11-30-2004

Optimal Topologies for Wireless Sensor Networks

Jason Tillett
Rochester Institute of Technology

Shanchieh Jay Yang
Rochester Institute of Technology

Raghuveer Rao
Rochester Institute of Technology

Ferat Sahin
Rochester Institute of Technology

Follow this and additional works at: https://scholarworks.rit.edu/other

Optimal topologies for wireless sensor networks

Jason Tillett, Shanchieh Jay Yang, Raghuveer Rao and Ferat Sahin
Rochester Institute of Technology
Rochester NY 14623
jcteee@rit.edu, sjyeec@rit.edu, mrreee@rit.edu, feseee@rit.edu

ABSTRACT

Since untethered sensor nodes operate on battery, and because they must communicate through a multi-hop network, it is vital to optimally configure the transmit power of the nodes both to conserve power and optimize spatial reuse of a shared channel. Current topology control algorithms try to minimize radio power while ensuring connectivity of the network. We propose that another important metric for a sensor network topology will involve consideration of hidden nodes and asymmetric links. Minimizing the number of hidden nodes and asymmetric links at the expense of increasing the transmit power of a subset of the nodes may in fact increase the longevity of the sensor network. In this paper we explore a distributed evolutionary approach to optimizing this new metric. Inspiration from the Particle Swarm Optimization technique motivates a distributed version of the algorithm. We generate topologies with fewer hidden nodes and asymmetric links than a comparable algorithm and present some results that indicate that our topologies deliver more data and last longer.

Keywords: particle swarm optimization, topology control, wireless sensor networks

1. INTRODUCTION

A common theme in energy-aware sensor network operation is that the network lifetime is extended by reducing the power output of the nodes. However, reducing the transmit power of a wireless device may not translate into real energy savings. This is because computation of the nominal power required to transmit a packet of data must include the additional power required to retransmit it when the packet is not received correctly at the destination due to collisions. Therefore, finding an "optimum" transmit power for each node is a constrained optimization problem whose objective function must be evaluated in the context of the sensor network application. It has been shown that for a uniformly distributed wireless network, throughput and energy consumption can be optimized by reducing the common power level of the nodes[1, 2]. It has also been demonstrated[3] that the optimum transmit power of the nodes varies with the traffic load on the network, assuming a common transmit power. However, a common transmit power level may not be appropriate for inhomogeneous networks. Here, we allow each node to have a different transmit power, but we address the same problem of finding the optimal transmit power of the sensor nodes. By optimal, we mean that the topology generated by the algorithm will perform well in terms of throughput for a wide range of network loads using a contention-based communication protocol[4]. Our optimization specifically targets contention introduced by hidden nodes and asymmetric links. Symmetric links are needed for proper functioning of many protocols across different layers of the protocol stack. The MAC[4] layer relies on symmetric links for acknowledgements and many routing protocols assume symmetric links[5-7]. In Section 2, we formulate a statement of the constrained optimization problem. The Particle Swarm Optimization (PSO) algorithm is described briefly in Section 3. A distributed extension of the PSO algorithm (DPSO) is proposed and developed generally in Section 4. The subsections of Section 4 are used to cast the distributed algorithm in a form suitable for attacking the problem statement. Section 5 evaluates the topology generation tool. Section 6 presents simulation results via the Network Simulator[8] that compare our topologies to another topology generation scheme. A summary and future work section concludes the paper.

2. PROBLEM STATEMENT

Let the vector $\bar{r}$ represent the communication/transmit radii of the nodes. The problem is to determine $\bar{r}$ such that we minimize the average energy expended by the sensor network. If we assume that each node needs to transmit the same amount of data as every other node, then it is equivalent to minimize $\sum_i r_i$. For communication networks and most

routing protocols, it is important for nodes to determine to next unicast destination for each data packet. This imposes a constraint on the optimization problem. If the symmetric links between nodes are visualized as edges in a graph, then the constraint is equivalent to requiring that the graph be connected. Define $C(\bar{r})$ to be a function that returns 1 if all

the nodes are connected. The connectivity constraint is then $C(\bar{r}) = 1$. Most topology control algorithms stop here. We assert however that the cost of communication is not only impacted by transmit radius/power, but also by the presence of hidden nodes and asymmetric links. Specifically, hidden nodes and asymmetric links impact adversely on the energy expenditure of the sensor network and therefore should be minimized. The most restrictive form of this additional constraint arises if we require no hidden nodes and asymmetric links. To visualize the impact of this constraint, consider again a representation of the sensor network where all symmetric links are edges in a graph and assigned a cost equal to the Euclidean distance between the nodes that they join. Assume we begin with a graph where all nodes have direct links to every other node. Now assume we have removed some edges such that the remaining graph represents the solution to the constrained optimization problem being stated here. The zero hidden nodes and asymmetric links constraint means that if a retained edge emanating from a node has weight w, then all other possible edges emanating from the node that have weight less than or equal to w will also be included in the solution graph. This constrained optimization problem can be expressed as follows. Let $\Omega_i(\bar{r})$ be the set of node indexes that are able to correctly receive packets from node i. Given that d_{ij} is the Euclidean distance between nodes i and j, the problem can be expressed as,

$$\min(\sum_{i=1}^{N} r_i) \ subject \ to, \ \ C(\bar{r}) = 1, \ \bar{r}_{\min} < \bar{r} < \bar{r}_{\max}, \ \left\{ d_{ij} < r_j \mid j \in \Omega_i(\bar{r}) \right\}. \tag{1}$$

In (1), for clarity, we have explicitly included the dependence of Ω_i on the current communication radii of the nodes. Our goal is to find a way for a sensor network to approximate or approach the solution to (1).

3. PARTICLE SWARM OPTIMIZATION (PSO)

The PSO[9] approach utilizes a cooperative swarm of particles, where each particle represents a candidate (feasible) solution, to explore the space of possible solutions to the optimization problem of interest. Each particle is randomly or heuristically initialized and then allowed to 'fly'. At optimization step $t+1$, each particle adjusts its candidate solution (flies) according to,

$$v(t+1) = v(t) + \phi_1(x_p - x) + \phi_2(x_n - x)$$

$$x(t+1) = x(t) + v(t+1)$$

$$(2)$$

Subscripts for particle index and dimensionality have been left off of (2), which may be interpreted as the 'kinematic' equation of motion for one of the particles (test solution) of the swarm where the particle is one-dimensional. The variables in Eqn. 2 are summarized in Table 1.

Table 1 - List of variables used to evaluate the dynamical swarm response

v	The particle velocity.
x	The particle position (test solution).
t	Time
ϕ_1	A uniform random variable usually distributed over [0,2].
ϕ_2	A uniform random variable usually distributed over [0,2].
x_p	The particle's position (previous) that resulted in the best fitness so far.
x_n	The (current) neighborhood position that resulted in the best fitness.

Equation (2) can be interpreted as follows. Particles combine information from their previous best position and their neighborhood best position to maximize the probability that they are moving toward a region that will result in a better fitness.

Application of PSO to the optimization problem in (1) at a centralized location will require transmitting the solution back to the sensor nodes. This might be infeasible in an actual network. We therefore seek a distributed version of the PSO algorithm where each node acts like a swarm particle that seeks to find just its own communication radius.

4. DISTRIBUTED PSO (DPSO) [10]

In traditional PSO, a global fitness function is used by all the particles in the swarm. Particles in traditional PSO represent the candidate solutions to a single optimization problem. In contrast, in the distributed form developed here, particles have either no knowledge or only limited knowledge, of the global objective function. Particles do not represent a global solution to a single optimization problem. Rather, particles have individual objectives and their objective function is a function of their individual parameters. The function used by particle i can be written as,

$$f_i(p_{i1}, p_{i2}, \ldots p_{iM},) \tag{3}$$

where the M parameters, p_{ij}, in (3) can be: communication range, sensing range, carrier sense range, number of neighbors, or residual battery energy level. The system designer may have a global objective or optimization targeted, such as that given in (1), but the particles cannot evaluate the global objective function because they do not have access to all of the nodes' transmit radii.

Another difficulty arises in the interpretation and calculation of the neighborhood best solution. How does one calculate the neighborhood best (labeled x_n in Eqn. 2)? In traditional/centralized PSO, the neighborhood best is simply the solution represented by the most fit neighbor. In DPSO, the solution represented by the most fit neighbor evaluated using the particle's local objective function will not necessarily result in a better fitness for the particle. Therefore particles must be able to convert parameter values and fitnesses exchanged with neighbors into a possible "better fit" set of values for their own operational parameters. This mapping is expressed as $\tilde{Q}_i \Rightarrow \bar{p}_{i,nbest}$ where

$$\tilde{Q}_i = \begin{cases} (f_1, p_{11}, p_{12} \ldots p_{1M}), \\ (f_2, p_{21}, p_{22} \ldots p_{2M}), \\ \ldots \\ (f_{N_i}, p_{N_i 1}, p_{N_i 2} \ldots p_{N_i M}) \end{cases} \tag{4}$$

and $\bar{p}_{i,nbest}$ are the neighborhood best values for node i. $\tilde{Q}_i$ is a set of N_i tuples, for node i, of values of fitnesses, f, and parameters, p, that it receives from its N_i neighbors. In (4), each neighbor has a single fitness value, but may have up to M parameter values to report to particle i. A problem specific finite state machine, discussed in Section 4.2, is used to perform the mapping. Once each particle is able to evaluate its fitness function and is able to construct its $\tilde{Q}$ set with information from its neighbors, the computation can proceed as in traditional PSO. We can rewrite (2) as

$$\bar{v}_i(t+1) = \bar{v}_i(t) + \phi_1(\bar{p}_i - \bar{p}_{i,best}) + \phi_2(\bar{p}_i - \bar{p}_{i,nbest})$$
$$\bar{p}_i(t+1) = \bar{p}_i(t) + \bar{v}_i(t+1) \tag{5}$$

The subscript *best* denotes the previous best value for the particle, which determines the cognitive component of the particles' motions. The subscript *nbest* denotes the neighborhood best and determines the social component of the particles' motions.

4.1 DPSO: the local fitness function
For applying DPSO to topology control for wireless sensors, we take the sensors/nodes to be the particles in the swarm, and a node's neighborhood consists of the set of nodes with which it has communication links. For approximating the constrained optimization problem presented here, we must identify the fitness function to be used by the sensor nodes. This involves identifying the form as well as the parameters of each sensor node's fitness function. Also to be determined is an appropriate choice for $\tilde{Q}_i \Rightarrow \bar{p}_{i,nbest}$. We assume each node knows its position so that communicating nodes can calculate distances. We also assume that the sensor node can adjust its power so it can vary its reach or transmit radius. [11] We assume that each node's carrier sense range, R_{sense}, may not be the same as its transmit radius. With these assumptions, a hidden node, whose number each node seeks to minimize, can be defined via the following inequalities, for which node k is hidden from node i.

$$d_{ij} \leq r_i , d_{ij} \leq r_j , i \neq j$$

$$d_{jk} \leq r_j , d_{jk} \leq r_k , d_{ik} > \frac{R_{sense}}{R_{receive}} r_k , i \neq k \qquad . \qquad (6)$$

Power consumption has two main components. The transmit power/radius of the node is the first component. The second component is the power consumed in re-transmitting frames that are lost due to collision at the MAC layer. Building minimization of power expended into the fitness function is straightforward; we simply make the sensor node fitness proportional to the transmit radius, r_i. (We are minimizing the fitness function.) Minimizing the power expended in re-transmissions can be achieved by minimizing the impact of medium contention. Either reducing the number of hidden nodes or minimizing the number of asymmetric links, N_i^A, can reduce the impact of contention. Therefore in a general expression for the fitness of a node, we will make the fitness proportional to the transmit radius, number of hidden nodes and number of asymmetric links. A node is better fit when it can send a packet successfully to any other node in the network. Therefore we include another term called cov_i in the node's fitness, which encourages it to form a connected network. [3] The above discussion leads to a general expression of the fitness function for node i as follows,

$$f_i = \frac{\left(N_{2,i} - N_{2,i}^A + 1\right)}{\left(N_{2,i} + 1\right)} + \frac{\left(N_i^A + 1\right)}{\left(N_{neigh,i}^A + 1\right)} + \frac{r_i}{r_{MAX}} + \frac{2\pi}{cov_i} . \qquad (7)$$

In (7) r_{MAX} is the maximum transmit radius of any given node, $N_{2,i}$ is the number of 2-hop neighbors of node i, $N_{2,i}^A$ is the number of 2-hop neighbors that will produce asymmetric links with node i and $N_{neigh,i}^A = \sum_{j \in \Omega_i} N_j^A$.

Each term in (7) is scaled by an appropriate factor so that it has about the same order of magnitude as the other terms. Since any given node cannot compute a count of hidden nodes, the number of hidden nodes is approximated by $N_{2,i} - N_{2,i}^A$, which approximates an upper bound on the number of hidden nodes for node i. The fitness expression of (3) encourages nodes to reduce their transmit radius, hidden nodes and asymmetric links while maximizing their coverage. The coverage maximization encourages spontaneous formation of connected graphs during the optimization. We may define the best topology in terms of any desired metric: hidden nodes, asymmetric links, sum-transmit-radius, $\sum_i r_i$, or max-transmit-radius, $\max\{r_i\}$. We can improve the fitness of (7). For a sensor network, each node should link minimally with its nearest neighbor. Therefore we can formulate a node's fitness function as composed of 2 functional forms. One form is used when a node is not minimally connected to its nearest neighbor, and (7) is used otherwise. When a node has no neighbors, it should increase its transmit radius. An expression for the fitness that encourages disconnected nodes to broadcast with greater and greater power in an attempt to gain connectivity to a supposed existing sensor network can be given as,

$$f_i(disconnected \ node \ i) = \frac{K}{r_i} , \qquad (8)$$

where K is a large constant. The form of (8) ensures that solitary nodes are more fit when they expand their transmit radius. Through the discussion above, we have identified the parameters, $\bar{p}$, in the fitness function. They are $N_{2,i}$, $N_{2,i}^A$, r_i, $N_{neigh,i}^A$ and cov_i. Note that more parameters may be necessary and will be introduced as needed.

$$p_i = \left(r_i, N_{2,i}, N_{2,i}^A, N_{neigh,i}^A, cov_i\right) \qquad (9)$$

In (9), $p_{i,1} = r_i$, $p_{i,2} = N_{2,i}$, $p_{i,3} = N_{2,i}^A$, $p_{i,4} = N_{neigh,i}^A$ and $p_{i,5} = cov_i$. We argue that the set $\{p_{ij} \mid i = 1, 2...N , j = 1, 2, 3, 4, 5\}$ that minimizes $\bar{f}$ component by component, will result in a locally optimal, probably non-pareto solution to the constrained optimization problem.

The only parameter in the node's fitness function over which it has independent control is its communication radius, r_i. Therefore (5) can be re-written as,

$$\vec{v}_i(t+1) = \vec{v}_i(t) + \phi_1(r_{i,best} - r_i) + \phi_2(r_{i,nbest} - r_i)$$
$$r_i(t+1) = r_i(t) + \vec{v}_i(t+1)$$
. (10)

When the nodes fix their transmit radii, then the topology is fixed and $N_{2,i}, N_{2,i}^A, N_{neigh,i}^A$ and cov_i are determined. The only quantity left to discuss in (10) is the neighborhood best. We hope to explain our interpretation of the neighborhood best in the DPSO approach and discuss explicitly how it is calculated. Again, this is one of the major challenges to applying this algorithm.

4.2 DPSO: the neighborhood best mapping

To start with, consider a node with no neighbors. It has a null neighborhood. That neighborhood is conveying information to the node to the effect, "you are solitary and should increase your transmit radius". We have already implicitly built in this neighborhood influence into the fitness function. When $\tilde{Q} = 0$ (null set), the node should adjust its neighborhood best $r_{i,nbest}$ to a larger value. We are free to experiment with different ways to adjust the neighborhood best $r_{i,nbest}$ value in this situation. We could select the maximum transmit radius of the node, for example. This illustrates our interpretation of how the neighborhood can allow a node to compute a value for $r_{i,nbest}$. Now we discuss in more detail the neighborhood best for non-null neighborhoods.

The neighborhood best transmit radius of a node can be presented using a finite state machine representation. The state space is divided into the states that the node may control and the states that the environment determines. For each node, its internal states are its current choice of its neighborhood best transmit radius and its gateway status. Allowing nodes to become gateways[10, 12] enables nodes to cooperatively reach out to connect to an asymmetric link for example. The gateway status can take on 2 values, true and false. The neighborhood best transmit radius state could conceivably take on a continuous range of values. However, we discretize the variable and denote the discrete variable with a prime. The discrete values could be set to the distances to each neighbor. With this approach, the number of states would be dynamic as well as a function of which node is being considered. We simplify the model and allow $r'_{i,nbest}$ to take on 5 values. The node state variables and their allowed values are presented in Table 2.

Table 2 – node state variables relating to neighborhood best FSM (finite state machine)

$r'_{i,nbest} = 0$	set $r_{i,nbest} = r_{max}$
$r'_{i,nbest} = 1$	set $r_{i,nbest}$ = distance to nearest neighbor
$r'_{i,nbest} = 2$	set $r_{i,nbest}$ = distance to farthest neighbor
$r'_{i,nbest} = 3$	set $r_{i,nbest}$ = distance to nearest asymmetric link
$r'_{i,nbest} = 4$	set $r_{i,nbest}$ = distance to nearest gateway
$gw_i = 0$ (1)	node is (not) a gateway

The transition function that determines the next state of the node, and hence its $r_{i,nbest}$, is constructed to be dependent on the factors that should affect a node's $r_{i,nbest}$. Specifically, the presence of asymmetric links or the lack of neighbors should encourage a node to explore. If a node is directly linked to a gateway, then resolving asymmetric links should be relegated to the gateway. We therefore identify 5 environmental states that factor into the transition function of the node's finite state machine. They are the presence of neighbors (env_1), the presence of asymmetric links (env_2), the presence of 2-hop neighbors (env_3), the result of comparing a node's distance from its nearest asymmetric link and the distance between its nearest asymmetric links and its nearest neighbor (env_4), and whether the node is linked to a gateway (env_5). Each of these states we allow to take on values 0 and 1. The states and their meanings are summarized in Table 3. An examination of the list reveals that there are 2 more parameters that must be exchanged in

the neighborhood best determination, they are the node's gateway status, an the node's neighbor list. Also of note is that some parameters already enumerated in (9) need not be exchanged for the neighborhood best state machine presented here. They are $N_{2,i}, N_{neigh,i}^A$ and cov_i. So the revised a final parameter values that must be exchanged for the neighborhood best determination are

$$p_i = \left(r_i, \Omega_i, N_i^A, gw_i \right). \tag{11}$$

Table 3 – Environmental states in the finite state machine.

	state 0	state 1
env_1	node has no neighbors	node has neighbors
env_2	node has no asymmetric links	node has asymmetric links
env_3	node has no 2-hop neighbors	node has 2-hop neighbors
env_4	nearest asymmetric link node is closer than distance between the nearest gateway node and the nearest asymmetric link node	nearest gateway node is closer than distance between the nearest gateway node and the nearest asymmetric link node
env_5	node is not linked to a gateway	node is linked to a gateway

There are 2^5 combinations of the environmental states, but when a node has no neighbors, it can have no 2-hop neighbors and it cannot be linked to a gateway. Therefore 16 states collapse to 4 and the number of combinations is reduced to 20. These 20 environmental states are tabulated in Table 5. All 20 remaining combinations are listed but 2 of those listed are also forbidden. They are *U10* and *U9*.

Table 5 – Possible combinations of states of the environment.

	env_1	env_2	env_3	env_4	env_5
$U1$	0	0	0	0	0
$U2$	0	0	0	1	0
$U3$	0	1	0	0	0
$U4$	0	1	0	1	0
$U5$	1	0	0	0	0
$U6$	1	0	0	0	1
$U7$	1	0	0	1	0
$U8$	1	0	0	1	1
$U9$	1	0	1	0	0
$U10$	1	0	1	0	1
$U11$	1	0	1	1	0
$U12$	1	0	1	1	1
$U13$	1	1	0	0	0
$U14$	1	1	0	0	1
$U15$	1	1	0	1	0
$U16$	1	1	0	1	1
$U17$	1	1	1	0	0
$U18$	1	1	1	0	1
$U19$	1	1	1	1	0
$U20$	1	1	1	1	1

Some environmental states may be grouped together and relabeled as they have identical impact on the node. Let *U1'={U1,U2,U3,U4}*, *U2'={U5,U6,U7,U8,U13,U14,U15,U16}* and *U3'={U9,U10,U11,U12}*. Note that some environmental conditions may be impossible, e.g., *U10*. With these definitions, we can provide visualization for the full finite state machine in Figure 1. The figure is split into 3 diagrams to improve readability. The rectangular boxes

denote possible nodes internal state configurations. A node makes a transition along indicated directions of lines provided the environment matches the line label. Inaccessible node states are "bricked" in the figures. Possible final states in each diagram are colored solid gray.
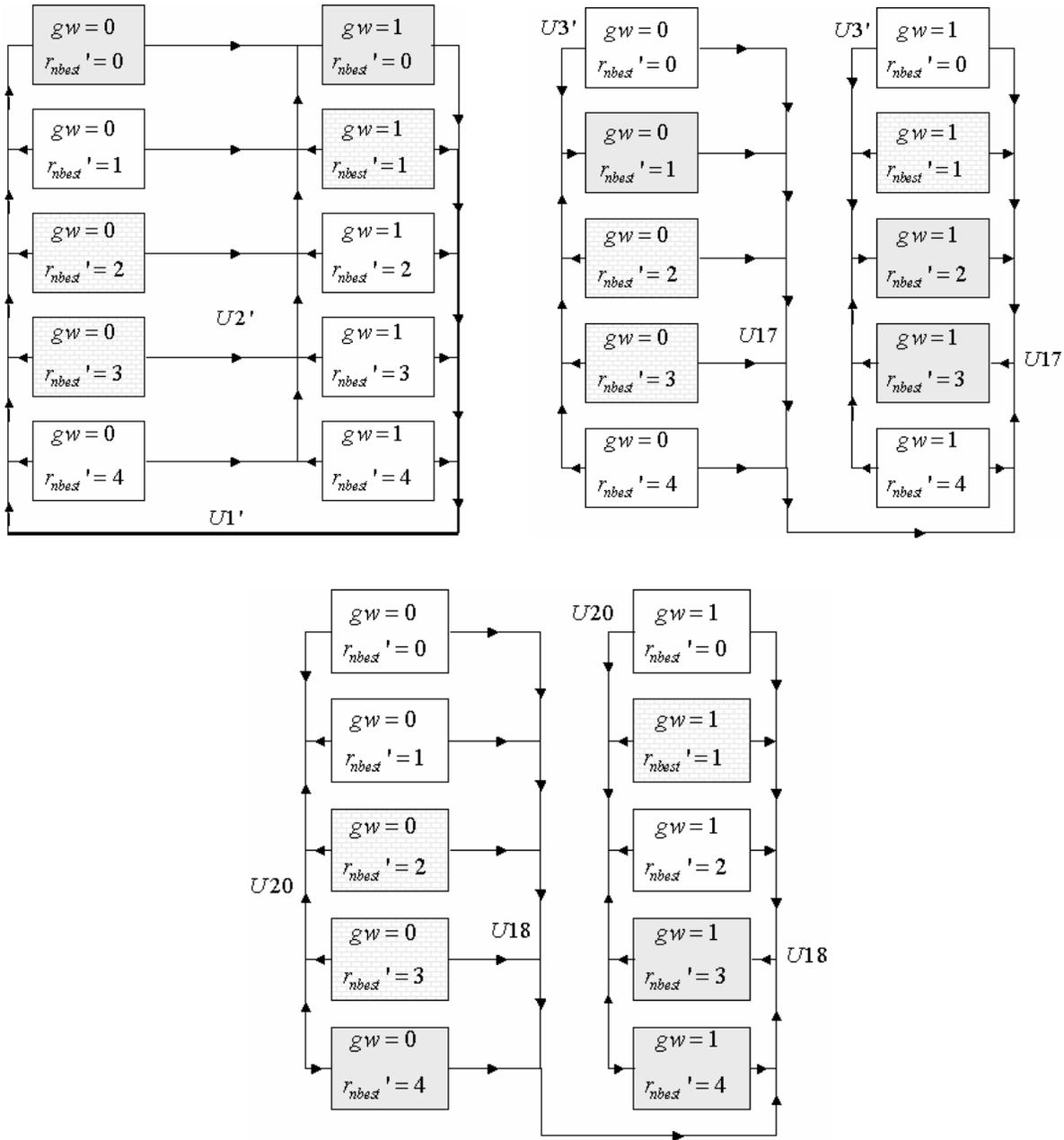

Figure 1 -- State machine to determine neighborhood best

5. EVALUATION

A simulation environment, using C++ was constructed to allow us to place nodes either constructively or randomly, assign initial transmit radii and execute the algorithm. Nodes are created in a 50 unit radius circle. The maximum transmit radius is set at 66 and the minimum transmit radius, r_{min} is set to 0.34. Note that the maximum transmit radius is chosen, so as to have a high probability of generating connected graphs by using the maximum value for all the

nodes, even for the case where the numbers of nodes is small, e.g., 5 nodes[13]. The initial transmit radius of each node could be set to any value between the minimum and the maximum values. The choice of these initial values can affect the simulation results. For investigations below, we choose to initialize the transmit radii of the nodes using the minimum value of 0.34.

The PSO algorithm requires a choice for the weighting of the cognitive and social components of the particle's motion. The "off-the-shelf"[14] PSO indicates that 2.8 and 1.3 are reasonable choices for the weighting of the cognitive and social components, respectively. We, however, adopt values of 1.75 and 1.35 based on our experiments. This is due to that the form of the fitness is such that larger radii are desirable for disconnected nodes while smaller radii are desirable for connected nodes, and oscillations may occur. We use a random variable to weigh the node's positional motion in (10) to quench the oscillations. We implemented a "fitness timer" to effectively restart each node's search for a best transmit radius. Allowing a node to forget about a previous best transmit radius helps a node to maintain its gateway status in the event that it was previously a part of a connected network and had a smaller transmit radius. Table 6 summarizes the base parameters used for our experiments of DPSO presented in the sequel.

Table 6 – Base simulation parameters.

ϕ_1	distributed on [0,1.75]
ϕ_2	distributed on [0,1.35]
r_{min} , r_{max}	0.34, 66
initial swarm particle velocities	randomly and uniformly distributed on [-5,5]
v_{max} (heuristically constructed)	$5 - 4*(i/I)$, where i is the current iteration and I=1000 is the stopping iteration
initial transmit radius	set to r_{min}

To evaluate DPSO for its capability of generating fitness-based topologies, we create 10 different random "deployments" of 30 nodes uniformly in a circle of radius 50. The simulator we build will compute and track a global metric to represent the fitness of the entire network. This global metric will not be used as part of DPSO in searching for the solutions, but will be used to determine how good the overall topology is. We emphasize that the primary purpose of this experiment is to generate topologies that exhibit specific target properties. The global metrics considered in this paper are the number of hidden nodes, the number of asymmetric links and the sum of the transmit radii of all the nodes. The algorithm is executed 5 times for each deployment and for each global metric. The number of hidden nodes, asymmetric links and the sum-transmit-radius are averaged over the 5 replicates of each deployment and normalized by their maximum (ceiling) value recorded over all scenarios simulated, regardless of the deployments and the global metrics. The results are displayed in Figures 2, 3 and 4.
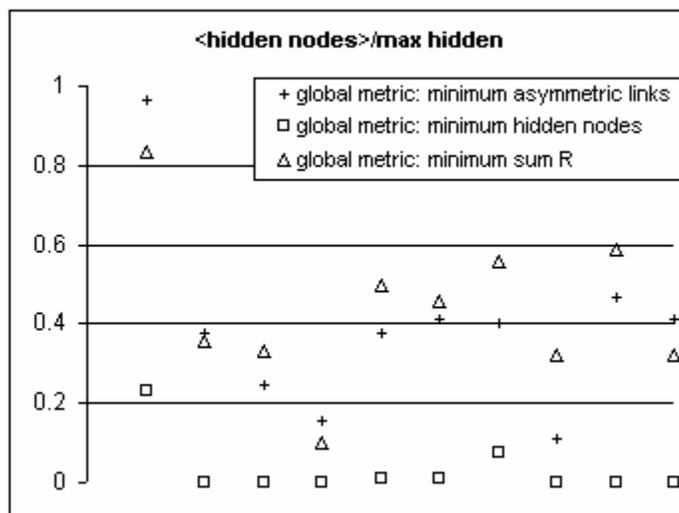

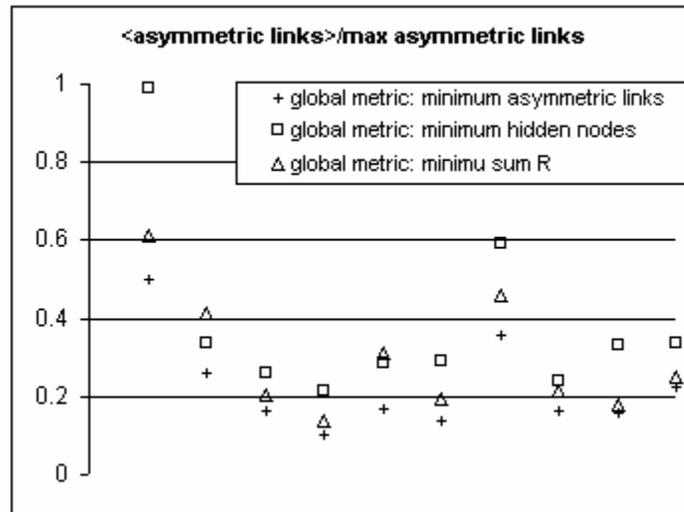Figure 2 – Normalized average hidden nodes for the 10 different deployments (x-axis).

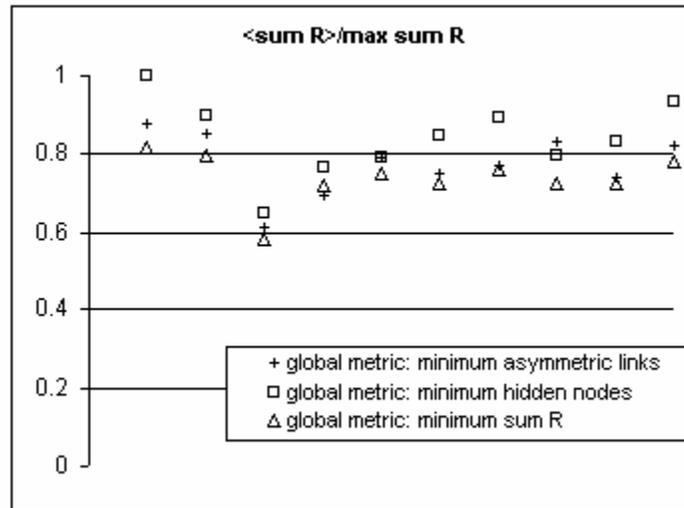Figure 3 – Normalized average asymmetric links for the 10 different deployments (x-axis).

Figure 4 – Normalized average sum-transmit-radius for the 10 different deployments (x-axis).

Figures 2, 3 and 4 show that, for a given global metric, the generated topologies on average exhibit smaller value for the selected metric. For example, in Figure 2, the 5 topologies generated based on the global fitness of minimizing hidden nodes contains fewer hidden nodes than topologies resulted from using the other 2 global metrics. Among all executions, we observe only a single situation in which using the global asymmetric link metric for one of the deployments results in a lower sum-transmit-radius than that achieved using the global sum-transmit-radius metric.

One puzzle brought forth during our experiment is the relative impact of the detrimental factors of hidden nodes and asymmetric links on wireless sensor networks. Therefore, we take an alternative view of the data presented in Figures 2, 3 and 4, by combining the average number of hidden nodes and average number of asymmetric links. The combined results are shown in Figure 5. Note that the global minimum asymmetric link metric always produces fewer h+a than the other 2 metrics. The data also seem to indicate that the global sum-transmit-radius metric produces fewer h+a on average than the global hidden nodes metric. To determine whether the minimum hidden nodes or minimum sum-transmit-radius global metric produces on average fewer h+a, we average the collected data over the 10 deployments, and re-plot them in Figure 6.

Figure 6 suggests that one should choose the "minimum asymmetric link" as the global target metric (from the three experimented) to generate topologies, if the desired property of the topology is to have a small value of h+a and a small value of sum-transmit-radius. Our results shows that, by choosing the minimum asymmetric link as the target metric, the topology generated on average contains the fewest number of h+a and results in only marginally higher value in terms of the sum-transmit-radius. Our experiments suggest that the minimum hidden nodes global metric is the worst performer on average.
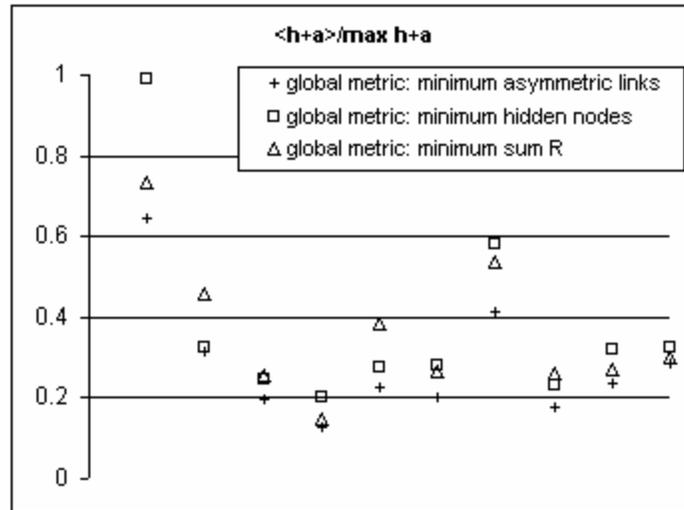

Figure 5 – Normalized average of sum of hidden nodes and asymmetric links (h+a) for the 10 different deployments (x-axis).
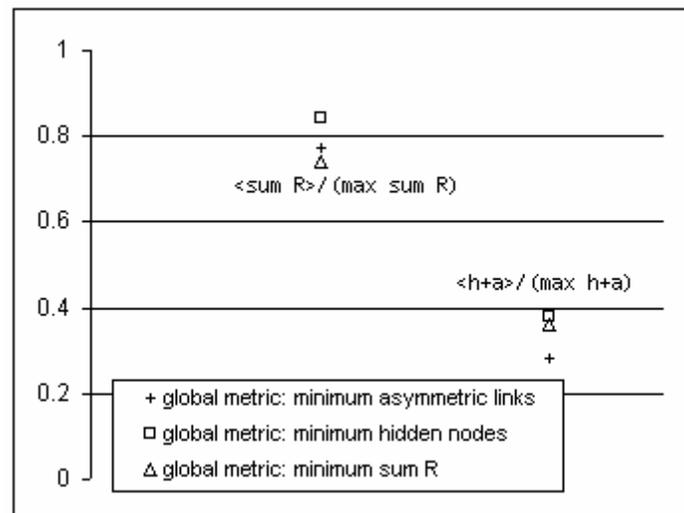

Figure 6 – The normalized sum-transmit-radius and normalized sum h+a averaged over all 10 deployments.

6. OPERATIONAL PERFORMANCE OF FITNESS-BASED TOPOLOGY

A Cone-Based[15] approach to topology generation has been proposed and is a distributed heuristic for minimizing node power under the constraint of connectivity. The algorithm can form provably connected networks, as long as the network will be connected when all nodes are transmitting at maximum power). The Cone-Based approach yields heterogeneous power topologies. Heterogeneous power may or may not result in topologies with non-zero counts of hidden nodes and asymmetric links. The algorithm makes no explicit attempt to minimize hidden nodes or asymmetric

links. We reproduced the Cone-Based algorithm using $\alpha = 5\pi/6$ and generated topologies by executing the "cone" phase followed by both the "back-off" and "redundant-edge removal" phases. Interested readers may find a detailed description of the algorithm in [15]. Our experiments suggest that the Cone-based scheme generates topologies, in general, with more hidden nodes and more asymmetric links than our DPSO does. We further examine the impact of hidden nodes and asymmetric links on how wireless sensor networks perform, using NS-2, in terms of the network capacity (i.e., total data collected by the network) and the energy efficiency (i.e., joules spent per unit data collected). Figures 7 and 8 below show the normalized network capacity and the energy efficiency achieved with the networks generated by the Cone-based and the DPSO algorithms. It can be seen that the topologies with fewer hidden nodes and asymmetric links, which happen to be those produced by the DPSO technique, delivered more data and use energy more efficiently than those produced by the Cone-based topology generator.
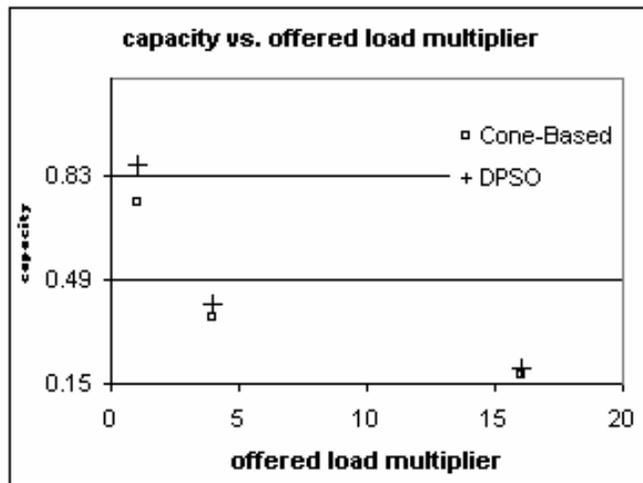

Figure 7 – The normalized capacity achieved by topologies generated by DPSO and Cone-based scheme, as the offered load increases.
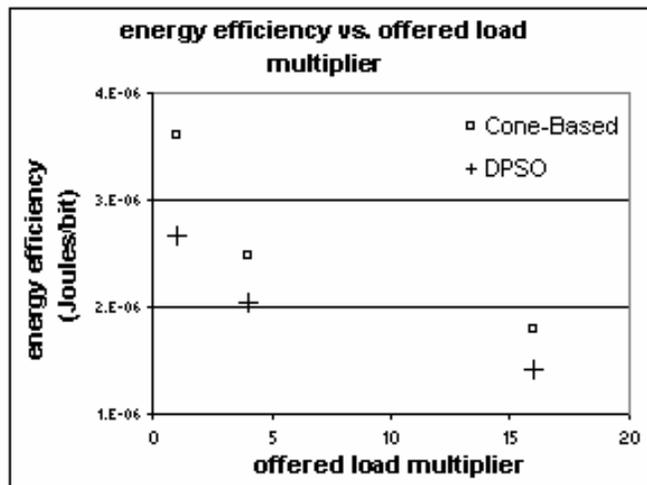

Figure 8 – The energy efficiency achieved by topologies generated by DPSO and Cone-based scheme, as the offered load increases.

7. CONCLUSION

A DPSO algorithm is proposed in this paper to generate fitness-based topologies with specific properties. We have shown that through modification of the global target metric, topologies can be produced with preference to minimizing either the number of hidden nodes, the number of asymmetric links or sum-transmit radius. We believe incorporating other global metrics should be a straight forward extension of the current simulator. Our experiments suggested that, by choosing the asymmetric links as the target global metric, the topologies produced provide both a low number of hidden nodes and asymmetric links and a low sum-transmit-radius. Given these topologies, we simulated using NS-2 and demonstrated the superior performance of our fitness-based optimal topologies to those produced using the Cone-based[3] topology generation scheme.

REFERENCES

[1]   P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. IT-46, pp. 388-404, 2000.

[2]   S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," presented at European Wireless Conference, 2002.

[3]   S.-J. Park and R. Sivakumar, "Adaptive Topology Control for Wireless Ad-hoc Networks," presented at MobiHoc, Annapolis MD, 2003.

[4]   V. Bharghavan, A. Demers, S. Shneker, and L. Zhang, "Macaw: A media access protocol for wireless lans," presented at Proc. ACM SIGCOMM '94, 1994.

[5]   C. E. Perkins, E. M. Royer, and S. Das, "Ad hoc on demand distance vector routing," presented at 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999.

[6]   C. E. Perkins and P. R. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," presented at ACM SIGCOMM, 1994.

[7]   "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Standard 802.11," IEEE 802 LAN/MAN Standards Committee 1999.

[8]   UCB/LBNL/VINT, "The Network Simulator (NS-2)," 2000.

[9]   J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*: Morgan Kaufmann Publishers, 2001.

[10]   J. Tillett, R. Rao, F. Sahin, and T. Rao, "A distributed evolutionary algorithmic approach to the least-cost connected constrained sub-graph and power control problem," presented at SPIE Security and Defense Symposium, Orlando, FL, 2004.

[11]   S. Agarwal, S. Krishnamurthy, R. H. Katz, and S. K. Dao, "Distributed power control in ad hoc wireless networks," presented at PIMRC01 -- IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2001.

[12]   J. Tillett, R. Rao, F. Sahin, and T. M. Rao, "A distributed evolutionary algorithmic approach to the least-cost connected constrained sub-graph and power control problem," presented at 2nd IEEE Upstate New York Workshop on Sensor Networks, University Sheraton Hotel, Syracuse, NY, 2003.

[13]   B. Krishnamachari, S. Wicker, R. Bejar, and M. Pearlman, "Critical Density Thresholds in Distributed Wireless Networks," in *Communications, Information and Network Security*: Kluwer Publishers, 2002.

[14]   A. Carlisle and G. Dozier, "An off-the-shelf PSO," presented at Workshop on Particle Swarm Optimization, Indianapolis, IN: Purdue School of Engineering and Technology, 2001.

[15]   L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a Cone-Based Distributed Topology Control Algorithm for Wireless Multi-hop Networks," presented at ACM Principles of Distributed Computing, 2001.