Energy-aware Topology Control in Sensor Networks
Using Modern Heuristics

Andreas Konstantinidis, Qingfu Zhang, Kun Yang, Ian Henning
University of Essex, Dept. of Electronic Systems Engineering and Dept. of Computer Science,
Wivenhoe Park, Colchester, CO4 3SQ, UK
{akonst | qzhang | kunyang | idhenn}@essex.ac.uk

Abstract— Cost-effective topology control is critical in wireless sensor networks. While much research has been carried out in this aspect using various methods, no attention has been made on utilizing modern heuristics for this purpose. This paper proposes a memetic algorithm-based solution for energy-aware topology control for wireless sensor networks. This algorithm (called ToCMA), using a combination of problem-specific light-weighted local search and genetic algorithm, is able to solve the minimum energy network connectivity (MENC) this NP-hard problem in an approximated manner that performs better than the classical minimum spanning tree (MST) solution. The outcomes of ToCMA can also be utilized for various network optimization and fault-tolerant purposes.

Keywords- wireless sensor networks, topology control, energy-awareness, memetic algorithms, genetic algorithms, local search

I. INTRODUCTION

Wireless sensor networks (WSNs) have attracted a phenomenon of research recently due to its ability of collecting data in hostile environment and reporting it back to a sink [1]. WSNs face several challenges with the most significant one being the energy consumption. Many protocols have been proposed to reduce the power consumption of sensors, and thereby the energy consumption of the network. An issue that is highly related to power consumption of a WSN is network topology. The randomness of sensor node distribution and the multi-hop nature of the wireless network routing, which renders the unevenness of the power distribution across sensor nodes, results in gradual topology change of the whole network. Connectivity is a basic requirement for sensor networks. Much research has been carried out on the energy-awareness of sensor networks, especially from the perspective of energy-efficient routing [2].

The problem of minimizing the transmission power of each node in the network, which results in minimizing the energy consumption of the network, while keeping at the same time its global connectivity is termed as the minimum energy network connectivity problem (MENC) by [3]. It has been proved that MENC is a NP-complete problem [3] and it is sometimes called topology control. Several heuristics have been developed to solve the MENC problem [3]. X. Cheng et al. [4] propose two heuristics based on a minimum spanning tree (MST) [5] and a broadcast incremental power (BIP) [5] method respectively. Based on the above work, M. Cheng et al. [3] present further improvement to the work in [4], e.g., a minimum incremental power (MIP) tree algorithm is designed. To the best of our knowledge, no research work has utilized

metaheuristic, for example Genetic algorithms (GA) [6] or Memetic Algorithms (MA) [7] which combines GA with Local Search (LS). This combination has been proved very successful in dealing with hard and complex problems [7]. This paper is to investigate how MA can be applied to the hard problems (in particular, the MENC problem) in wireless sensor networks and how MA is used to obtain performance gain. The ToCMA (Topology Control using Memetic Algorithm) algorithm proposed in this paper generates many different solutions and explores in an effective manner the solution space by using the genetic algorithm operators (i.e. crossover and mutation). Then ToCMA employs different problem specific algorithms to maintain the global connectivity of the network. It checks if the network is strongly connected and if it is not then ToCMA repairs it. Furthermore, ToCMA employs an improvement procedure to further minimize the overall energy consumption of the network as much as possible.

The rest of the paper is organized as follows. In Section 2 we describe the network assumptions, the MENC problem itself and a briefing on memetic algorithm. Section 3 details the proposed algorithm step by step. Section 4 presents the performance evaluation of ToCMA against MST. Finally the paper concludes in Section 5.

II. PRELIMINARIES

A. Network Assumptions

Since our work is to solve the MENC problem defined by [3], we follow similar assumptions as these in [3]:

- The sensors in the network are stationary and located in a two-dimensional plane. It is also assumed that the location of each sensor node can be obtained after the deployment, by using any positioning technology. The location information will be used for calculating distance between two sensor nodes.

- Omnidirectional antenna is used for each sensor. This means that a sensor radiates and receives equally in all directions. If a sensor transmits with a power level:

\[ p_t = \zeta \times d^\alpha \]  

then any sensor within the Euclidean distance \( d \) and a power threshold \( \zeta \) can receive the signal. The path loss exponent \( \alpha \) is between 2 and 4 [3].

- ToCMA uses transmission power in energy calculation without considering the transmission time. The same assumption is adopted by [3].
• Sensor nodes can operate in different initial power levels, within bounds. This consequently leads to asymmetric links and a directed graph. Note that only bidirectional links are considered in [4].

B. Problem Definition

If for each node in a wireless network, there is a route to reach any other node in the same network, then such a network is regarded strongly connected [3]. Let \( V \) denote the set of wireless sensor nodes and \( G(V,E) \) denote the super-graph on \( V \) that contains all possible edges \( E \) if each node transmits at its maximum transmission power. Graph \( G \) sets an upper bound on the maximum connectivity that a wireless network can have. The topology control algorithm returns a topology \( T \) constructed from \( G \), i.e., \( T \) is a subgraph of \( G \) on \( V \). A wireless network should fulfill the following connectivity requirement: for any pair of nodes \( u \) and \( v \), if there is a path from \( u \) to \( v \) in \( G \) then there is also a path from \( u \) to \( v \) in \( T \).

The formal definition of the MENC problem is given as follows [3]:

Given a set of wireless nodes \( V = \{n_1,n_2,...,n_n\} \) and the cost function \( f : (V,V) \rightarrow Z \), MENC is to determine a power assignment of nodes \( P:V \rightarrow Z \) such that:

1. The induced directed graph \( T \) is strongly connected.
2. The total energy consumption of the network \( \sum_{i=1}^{n} p_i \) is minimized, where \( p_i \) denotes the power assigned to node \( n_i \) and is calculated via Formula (1).

C. Memetic Algorithm

A memetic algorithm (MA) is a combination of a genetic algorithm and a local search [6]. It is based on the principle of evolution operations such as crossover and mutation and the concept of fitness. It also utilizes various problem specific heuristics to improve and/or repair the solutions generated by the evolution. In MA each solution is usually encoded as an integer string, with each integer representing different physical parameter that is specific to the problem to be solved. A solution is termed as an individual in a population. A population is associated with certain generation \( t \) in the whole evolution of individuals. The initial populations are usually generated in a random manner and then the evolution of these populations is carried out by the genetic operators such as crossover and mutation. Local search is utilized to check the feasibility of each population and to divide the population into feasible solutions and infeasible solutions. Then a repair procedure is invoked on infeasible solutions trying to “repair” them. Even the feasible solutions can be further improved to obtain a better fitness. Here fitness is used to express how good a solution is, i.e., how close it is to the optimal solution. Note that the three terms, individual, chromosome, and solution, represent the same meaning but from different points of view. For instance, a solution to a problem is represented as an individual, and from genetics’ perspective an individual is a chromosome which again is composed of multiple genes. The MA algorithm proposed in this paper for the MENC, ToCMA, follows the above procedure, which is illustrated in Figure 1.

III. TOCMA ALGORITHM

This section starts with a presentation as to how the MENC problem is represented by ToCMA, the search space and the fitness function of a solution. Then it gives a detailed presentation as to how each step is designed and implemented for the MENC problem following the flow in Figure 1.

A. Genetic Representation

A ToCMA solution to the MENC problem is represented by a positive integer string where 1) the integer numbers of the string are the power levels assigned to each node respectively and 2) the position of an integer number in the string represents the node id. All sensor nodes in the sensor network concerned are numbered from 1 to \( n \), where \( n \) is the total number of nodes in the sensor network. In GA terminology [6], a solution is represented as a chromosome: \( c = (p_{1,1}, p_{2,1},..., p_{n,1}) \), where \( c_i \) denotes the \( i \)-th solution in the population and \( p_{j,i} \in \{1,2,...,n\} \) denotes the power assigned to the \( j \)-th sensor as far as the \( i \)-th solution is concerned. In general, the composing entities of a chromosome are called genes. In ToCMA, genes are the power of sensors. The quality of each solution is measured by a fitness function. In ToCMA, the fitness function is defined as the sum of the power assigned to each sensor, namely:

\[
  f(c_i) = \sum_{j=1}^{n} p_{j,i}
\]  

(2)

To the MENC problem, the smaller a solution’s fitness value is the better the solution is. An optimal solution is defined as a solution that has minimal \( f(c_i) \). The fitness function is to be used as a criterion when ranking and selecting a chromosome.
B. Population Initialization

ToCMA adopts a random process to generate its initial populations due to the speediness and less complexity of this method. However, ToCMA can equally take benefit of more intelligent initialization method but at a cost of more computational complexity. A random number generator is used by ToCMA to generate \( p_i, 0 \leq p_i \leq \Delta \) for each sensor node \( i \) where \( \Delta \) is the maximum power a node can be assigned to.

C. Local Search: Checking, Repairing and Improvement

1) Checking function

In ToCMA, check process is to check if the graph generated based on a solution gives a directed strongly connected network. Based on the analysis of sensor network topology, the following four cases, as depicted in Figure 2, might cause an infeasible network topology.

In Case #1, as shown in Figure 2 (a), there is one or more totally isolated node in the network represented by graph \( g \), for example, the node circled. Such a solution is put into the infeasible set of the current generation \( t \), i.e.,

\[
POP^{IF} \leftarrow c = \{g(V,E) \mid \exists n_i \in V, \text{inDegree}(n_i) = 0 \land \text{outDegree}(n_i) = 0\}
\]

where \( V \) denotes the set of the nodes in graph \( g \) and \( \text{inDegree}(n_i) \) and \( \text{outDegree}(n_i) \) stands for the in-degree and out-degree of node \( n_i \) respectively.

In Case #2, as shown in Figure 2 (b), there is one or more one-way isolated node in the network represented by graph \( g \), for example, the two nodes circled. The notations are the same as these in Case #1. For example, in Figure 2(b), \( n_5 \) can reach \( n_7 \) but not reachable by \( n_7 \). Any solution falling into this case is also put into the infeasible set of the current generation \( t \), i.e.,

\[
POP^{IF} \leftarrow c = \{g(V,E) \mid \exists n_i \in V, \text{inDegree}(n_i) + \text{outDegree}(n_i) = 1\}
\]

In Case #3, as shown in Figure 2 (c), there is one or more loop in the network represented by graph \( g \). For example, there is a loop between node \( n_1 \) and node \( n_2 \). For directed graph, this means \( n_1 \) and \( n_2 \) are each other’s only next-hop neighbouring nodes. Any solution falling into this case is also put into the infeasible set of the current generation \( t \), i.e.,

\[
POP^{IF} \leftarrow c = \{g(V,E) \mid \exists n_i, n_j \in V, (n_j \in \text{neighbour}(n_i) \land \text{outDegree}(n_j) = 1) \land (n_i \in \text{neighbour}(n_j) \land \text{outDegree}(n_i) = 1)\}
\]

where \( \text{neighbour}(n_i) \) means the one-hop neighbours of \( n_i \).

In Case #4, as shown in Figure 2 (d), a partition occurs in the network and it is represented by graph \( g \). For example, there are two sub-networks in the graph, one composed of nodes \( \{n_6, n_7, n_8, n_9, n_1\} \) and the other composed of node \( \{n_5, n_6, n_7, n_8, n_9\} \) and there is no edge linking these two sub-networks. This case is formally expressed as: there are at least two sub-graph \( g_1, g_2 \), where the following conditions hold: (C1) these two graphs do not share any common node and (C2) there is no link connecting these two sub-graphs. Any solution falling into this case is also put into the infeasible set of the current generation \( t \), i.e.,

\[
POP^{IF} \leftarrow c = \{g(V,E) \mid \exists g_1, g_2, C1 \land C2\}
\]

Any other solutions, which are feasible solutions, are put into the feasible population set: \( POP^{F} \).

2) Repair Function

As discussed in the previous sub-section, four cases cause an infeasibility of a solution. Repair function is provided for each of them.

Case #1 & #2: In these two cases, the repairing heuristic firstly discovers the origin of infeasibility, e.g., node \( n_i \), and secondly it finds the node’s nearest neighbour. Thirdly, it measures the distance to that neighbour and calculates the necessary power needed by \( n_i \) to be able to communicate with that neighbour. And finally the repairing heuristic assigns this power value to \( n_i \).

Case #3: the repairing heuristic firstly discovers the involved nodes. For each involved node \( n_i \), it carries out the following steps: 1) it tries to find other next-hop neighbours and selects the one \( n_j \) which is nearest to it; 2) it calculates the power needed for \( n_i \) to reach \( n_j \); 3) then it assigns this power value to \( n_i \).

Case #4: the heuristic firstly discovers the partitioned groups, e.g., \( g_1, g_2 \). Then it tries to find a node \( n_j \) in group \( g_1 \) which is the nearest neighbour of a node \( n_i \) in group \( g_2 \). Finally, it calculates the power needed for \( n_i \) to communicate with \( n_j \) and assigns this power value to \( n_i \). Then communication from \( g_2 \) to \( g_1 \) is set up. Repeat this procedure to all partitioned groups until network connectivity is resumed.

3) Improvement

The feasible solutions are further improved by ToCMA. The purposes are mainly twofold. Firstly it is to provide remedies to the fact that the random power assignment used for population initialization might assign unnecessary high power to certain nodes as such leading to unnecessary power consumption. Secondly it is to avoid the situation where direct connections are two frequently used whereas there is an indirect route between two nodes.

Improvement #1 occurs when a node’s \( n_i \) current power \( p_i \) is greater than the power needed to reach its farthest neighbour \( n_f \). Thus the power of this node is decreased to the power level that is just enough to reach its farthest neighbour(s). Furthermore, if
there is a route \( r \) between a pair of indirectly connected nodes: \( n_i \) and \( n_j \) then there is a chance to take the benefit of Improvement #2.

Improvement #2, occurs when the power needed for a node \( n_i \) to reach directly its farthest neighbour \( n_j \) is more than the power needed if \( n_i \) follows a route to \( n_j \). Thus the power of node \( n_i \) is decreased to the level that is just enough to reach the intermediate neighbour of the less power consuming route.

4) Population Update and ToCMA termination

After local search, all the solutions are feasible and are kept in \( POP_{\text{IMP}} \) which keeps only feasible solutions for one particular generation. ToCMA maintains another set of feasible solutions (also of length \( \mu \) ) that collects the best solutions across all generations. We denote this set as \( BestInd \). The solutions in \( BestInd \) will be used by GA operators to generate next generation populations. Using best parents across previous generations to create the next generation rather than only the current generation increases the chance of creating better solutions in the next generation.

ToCMA uses a popular population replacement strategy, the elitism strategy [8]. A solution from the current generation is used to update \( BestInd \) only when it is at least better than the worst one in \( BestInd \). And it is always the worst solution in \( BestInd \) that is replaced. At the end of each generation, this strategy ensures that only \( \mu \) chromosomes with best fitness value will be kept in \( BestInd \) and survive to get involved in the creation of next generation via genetic operators.

The algorithm terminates after \( \eta \) generations. If the termination criterion is not met, the GA’s operators manipulate on \( BestInd \) to generate the new population.

D. Genetic Operators: Selection, Crossover and Mutation

Three steps are needed in order to generate a new generation while keeping its high quality and diversity:

1) Selection

The selection operator is to improve the average quality of the population by giving the high-quality chromosomes a better chance to get copied into the next generation. ToCMA uses the tournament selection [6] as its selection operator. The reason is that it is very simple and it only needs a preference ordering between strings. The preference ordering in ToCMA is already carried out when generating \( BestInd \). In ToCMA the tournament size \( m \) is equal to \( (\mu / 2) + 1 \) (note that if \( \mu \) is an odd number then the fraction part is ignored) and they are the first \( m \) best individuals stored in \( BestInd \). The purpose of the selection operator is to get a pair chromosomes which are forwarder for further crossover operation.

2) Crossover

In this phase ToCMA utilizes a fast and simple technique, the single-point crossover operator, as illustrated in Figure 3. This operator involves two steps: 1) the selection of the crossover site \( IX \), and 2) the generation of the two new chromosomes (also called offspring). The crossover site is selected randomly in the interval \([1, n]\). Offspring are generated by swapping the characters between positions \( IX+1 \) and \( n \) of the parents (the pair selected earlier by the selection operator).

The problem with crossover is that all the population generated tends to gather together as such covering only a limited area in the search space. In order to increase the diversity of the population and thereby increasing the chance of finding better solutions, mutation is further utilized after crossover.

3) Mutation

The mutation operator is applied to all offspring of the new population \( POP_{\text{CRS}} \). This operator simply selects randomly \( g = \lfloor n / 2 \rfloor \) (round the fraction up if \( n \) is an odd integer) genes in each offspring and randomly changes their value in the power interval \([0, \Delta]\). This number is chosen to be big enough to increase the diversity of the population as much as possible in order to compensate the diversity loss in crossover operator and the local search process. After the mutation, the new mutated population, \( POP_{\text{MUT}} \), will then be forwarded to the local search heuristic discussed above to go through checking/repairing/improvement again until the termination criterion is met.

IV. EVALUATION RESULTS AND ANALYSIS

In this section ToCMA is compared against MST (minimum spanning tree) in terms of total energy consumption while maintaining network connectivity. MST is selected because it is a simple and neat solution to the MENC problem and it is also popularly selected as a benchmark in topology control problem solving [3].

A. Experimental Design and results

A spanning tree, \((V, E')\), is a subgraph induced from a supergraph \( G(V, E) \), where \( V \) is a set of nodes common for both graphs, \( E \) is a set of links in the supergraph and \( E' \in E \) is a set of links in the sub-graph. The number of links in \( T \) is equal to \(|E'|=|V|-1\). A graph may have many spanning trees. From all these spanning trees, there is one that has minimum cost, based on the sum of the weight on links. This is called the MST \( T \) of graph \( G \) [5]. The same method as that in [3] is utilized to calculate the energy consumption of MST.

The population size is \( \mu = 30 \) and the number of nodes \( n \) varies from 10 to 100. ToCMA employs a tournament selection with tournament size \( m = 16 \). A single point crossover is then performed to each pair of chromosomes. Each node is randomly assigned a power within the interval \([0, 500]\), where the path loss exponent \( \alpha = 2 \). Each experiment is terminated at a maximum number of generations \( \eta = 15 \).

For the comparison of MST and ToCMA simulation is used and draws the results in Figure 4. Figure 4 gives the total energy consumption of the best solutions found by each algorithm. In most cases the ToCMA algorithm outperforms MST algorithm.
This result is also depicted by the following example as illustrated in Figure 5. In this example a random topology is created and the links between the nodes are calculated based on the location of the nodes and their power assignment.

\[ e_g(MST) = \sum_{i=1}^{n} p_i = 4^2 + 4^2 + 4^2 + 5^2 + 1^2 + 5^2 + 5^2 + 5^2 = 149 \]

Similarly, for ToCMA, there is:

\[ e_g(ToCMA) = \sum_{i=1}^{n} p_i = 1^2 + 1^2 + 1^2 + 5^2 + 5^2 + 5^2 + 5^2 + 5^2 = 128 \]

Apparently, \[ e(ToCMA) < e(MST) \].

B. Other Benefits

Figure 4 indicates a narrow win of ToCMA over MST. In addition to the better quality of solution that ToCMA can offer, ToCMA can also provide some other benefits.

ToCMA can take advantage of the information stored in BestInd for a variety of purposes. Apart from being effectively used for genetic operations, BestInd, when stored in the sink of the sensor network, can be utilized for fault tolerance purpose. For example, if for any reason a sensor node has to operate with less power than the one assigned by the best solution i.e. \( c_1 \), then the sink can search in BestInd to try to find another best solution that satisfies this new power constraint requirement. And then the sink broadcasts this solution to all the sensors in the network to easily accommodate a sudden change in the network. For instance, in Figure 5, after the best solution \( c_f = 1^2 + 1^2 + 1^2 + 5^2 + 5^2 + 5^2 + 5^2 + 5^2 = 128 \) has been deployed, suddenly node \( n_2 \) has to operate with a power \( p_{2s} \leq 3 \). ToCMA can easily locate the best solution, e.g. \( c_f = 2^2 + 2^2 + 2^2 + 2^2 + 2^2 + 2^2 + 2^2 + 2^2 = 64 \), for this situation and potentially reduce the chance of network partitioning.

In order to check that a solution provides a strongly connected network, the checking function actually needs to guarantee a route, either directly connected or via intermediate nodes, between any pair of nodes in the network. This routing information can be well stored alongside the power information (e.g. in an extended BestInd table). With this routing information, route discovery process can be much simplified.

Though being more computationally complex than MST, the outputs from ToCMA can be utilized for many other purposes than energy consumption efficiency to the overall benefit of the wireless network as a whole.

V. Conclusions and Future Work

In this paper, we have proposed an alternative approach to tackle the MENC problem in wireless sensor networks, which utilizes modern heuristics and more precisely memetic algorithm. The proposed ToCMA explores in an effective manner the solution space by using a combination of the GA operators and the local search technique. It employs repair and improvement methods to refine solutions. The concrete MA solutions are guided by problem-specific features such as network connectivity, avoiding loop etc. Simulation results have shown that better solutions can be obtained by ToCMA than MST. ToCMA also demonstrated its strength in generating initial routing information and fault tolerance and robustness.

Based on the encouraging results of this paper, we will further investigate how an intelligent initialization, guided mutation and smarter local search mechanism will make impact on quality solutions and the overall performance of the algorithm. To further reduce the computational complexity and test different performance metrics is also the next-step target when accommodating the above future research plans.

REFERENCES