STUDIA INFORMATICA
Nr 2(11) Systems and information technology 2008

Hyper-heuristics for power-aware routing protocols

Marek Pilski[1], **Franciszek Seredynski**[2]

[1] Institute of Computer Science, University of Podlasie,
Sienkiewicza Str. 51, 08-110 Siedlce, Poland

[2] University of Warmia and Mazury in Olsztyn,
Oczapowskiego Str. 2, 10-719 Olsztyn-Kortowo, Poland

Abstract. The idea underlying hyper-heuristics is to discover some combination of straightforward heuristics that performs very well across a whole range of problems. In this paper we describe genetic algorithm-based (GA) approach that learns such a heuristic combination for solving energy-efficient routing problem in mobile ad hoc networks (MANETs).

Keywords. Protocols, ad hoc networks, genetic algorithm, hyper-heuristic

1 Introduction

There are many situations where a communication network is required in places where there is no fixed networking infrastructure and no time to create such an infrastructure. Examples of such situations occur in military operations, law enforcement, rescue operations and personal area networking. Ad hoc networking, the ability to form a network dynamically from scratch using wireless connections, addresses this need. When the nodes of the network are mobile, as it is usually the case, the networks formed are called mobile ad hoc networks (MANETs). The dynamic nature of MANETs provides special challenges beyond those in standard data networks [2]. MANET requires special protocols, a review of a range of such protocols is given in [1]. Not only the network must be initiated, but it also needs to maintain itself as the nodes move around and as transmission properties change. With changing connectivity between nodes, including the entrance and exit of nodes, the network must have the ability to adapt its topology to reflect these changes. All nodes in such networks take two roles: producer and consumer in one of data packet streams, and routers for data packets destined for the other nodes. Equally important feature in mobile ad hoc networks is limited battery capacity of the nodes (for example, the nodes in MANETs may by devices as: laptops, palmtops, mobile-phones). The limited battery capacity poses yet another challenge for the routing algorithms: to distribute the packets on multiple paths in such a manner that the battery of different nodes deplete at an equal rate, as a result, the life time of the network could be increased. This feature of the nodes is the reason of creating a new group of protocols called Energy Efficient Routing Protocols or Power-Aware Routing Protocols for MANETs [3], [9]. The metrics for energy efficient routing are also introduced in [9], and it is evident that an energy aware routing algorithm is expected to degrade the traditional performance metrics of a routing algorithm i.e. throughput and packet delay. The real dilemma in MANETs is: how to design a routing algorithm which is not only energy efficient but also provides the same performance as that of the existing state-of-the-art algorithms.

In this paper we consider the continuous process of choosing paths between communicating nodes in MANETs and power management. The aim of the proposed routing algorithm is to maximize a number data packets sent between nodes in MANETs. A routing metric is data packets number from a moment when the network starts up, to the moment when the first battery in any nodes runs down. A problem solution is such a heuristics combination, which depends on an actual state of the network (energy of nodes) and will choose paths maximizing the routing metric. We propose genetic algorithm-based approach [4] to find a heuristic combination for solving power-aware routing problem.

The paper is organized in the following way. The next section presents an idea of a new energy efficient routing protocol. Section 3 describes representation of the network state and section 4 shows how information about the routes is collected. Section 5 presents five heuristics, which we used and section 6 shows genetic algorithm (GA) which uses these heuristics to establish connections between nodes in such a way to optimize accepted criterion of the network life. Section 7 shows results of experiment study. Last section contains conclusion.

2 The idea of a new energy efficient routing protocol

There exist a number of power-aware routing protocols, which perform well in specific network situations. However, a network state may change dynamically in time, and there is a need to use properly this protocols depending on the network state. Such a mechanism can be provided by a global mechanism which uses some meta-heuristics. The purpose of such a meta-heuristic is to use a combination of available protocols (heuristics) in order to solve a current problem in the network. In this paper we will use evolutionary-based hyper-heuristic [7] to discover a combination of protocols (heuristics) to solve power-aware routing problem. The problem solution procedure based on applying a meta-heuristic is the following:

- given a state *p* of the problem,
- find currently the best heuristic *h(p)* for a given state of the problem and apply it,
- this transforms the problem state to the state *p'*,
- repeat this until the problem has been solved.

The purpose of GA is to define key-situations in the network states and associate with them combination of heuristics (communication protocols) which can be applied to provide efficient functioning of the network.

3 Representation of the network state

We assume that the network state is defined by four variables. The variables describe how many nodes of the network are into each of four categories at battery energy level:

- huge (*H*) stockpile of energy ($e_{best}/2 < n_i$),
- large (*L*) stockpile of energy ($e_{best}/3 < n_i \leq e_{best}/2$),
- medium (*M*) stockpile of energy ($e_{best}/4 < n_i \leq e_{best}/3$),
- small (*S*) stockpile of energy ($n_i \leq e_{best}/4$),

where: n_i - a node in mobile ad hoc networks and e_{best} - an energy of the best node (the node with the greatest stockpile of energy).

For example, the network state in time *t(1)* is described as follows: *H=100%*, *L=0%*, *M=0%* and *S=0%* (assumption: when the network starts up all nodes have the same energy in batteries). The example shows that all nodes are inside the *H* category (100% nodes), there are not nodes in *L*, *M* and *S* categories. For example, the network state in time *t(n)* may described as follows: *H=55%*, *L=40%*, *M=3%* and *S=2%*. It is worth noting that this representation throws away a lot of details.

4 Collecting information about the routes

When the node *S* (source) wants to sent data-packets to the node *D* (destination), it must know the route. To establish a connection between two hosts not directly connected, messages must be routed by intermediate node. The traditional broadcasting method used to discover routes relays on dissemination of a message called a *Route Request*. When the node *S* wants to find a route to the node *D*, it sends a route discovery message to its neighbourhood (on-demand). Each of its neighbours adds its name to the message and re-emits it. Every other node in the network does so and finally, the searched node *D* receives the message with the chain of nodes followed from the source node S. The final step of this process is the emission of a message called a *Route Reply* by the node *D* to inform *S* that a route has been found. This procedure is based on an IETF (Internet Engineering Task Force) as standardized protocol [5]. The broadcasting task is a fundamental mechanism in route discovery. Therefore, it is unceasingly the subject of the study [8]. The broadcasting method is illustrated in Fig. 4.1.

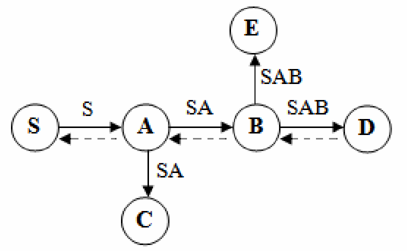


Figure 4.1. The broadcasting task the route request from node *S* to node *D* (continuous arrow) and route replay from node *D* to node *S* (dotted arrow).

The node *S* wants to find a route to *D*, so it sends a route request message to its neighbourhood with its *ID* as the initial chain of hosts. The node *A* receives this message, writes its name in the chain and re-emits it. *S* gets the packet, but ignores it (it already knows it and furthermore, it is the one that first emitted it). *B* does the same as *A* (as well as *C* and *E*), and finally *D* receives the request. Every node in the network retransmits once the message, upon receiving the first copy of it, and will consequently ignore further copies of the same message. It then just have to answer to *S* by emitting a *Route Reply*. This last message is addressed directly to *S*, because *D* knows the route to it. This one is simply written in the *route request message*, formed by the chain of hosts that relayed the message. In the example, the chain received by *D* was *SAB*, so it sends the route reply with *BAS* as the route to follow. Route discovery is performed by a broadcasting task, while the *Route Reply* is an unicast routing operation. Each answer (*Route Reply*) returned from *D* contains: the route (e.g. *SAB*) and the energy of each node on the route: energy of node *A* and energy of node *B*.

5 The set of heuristics used

We consider a simple ad hoc network with six nodes (Fig. 5.1). Let the energy values of batteries in nodes are the following: the energy in node *A=70%*, *B=80%*, *C=40%*, *E=90%*. We want to choose the path between the node *S* and the node *D*.

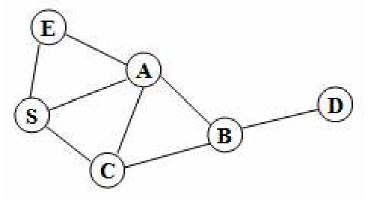


Figure 5.1. Example of ad hoc Network

Our GA-based meta-heuristic will use five basic heuristics:

5.1 Heuristic h1

Choose the path as follows:

- *Step 1.* For each path mark the weakest node (with the smallest energy in battery on the path).
- *Step 2.* Select a path from marked nodes which have the highest battery energy. If exists more than one path, select the shortest path.

```
S-E90%-A70%-B80%-D
S-A70%-B80%-D {selected path}
S-C40%-B80%-D
S-C40%-A70%-B80%-D
S-A70%-C40%-B80%-D
S-E90%-A70%-C40%-B80%-D
```

Figure 5.1.1. Example of choice a path by heuristic *h1*

5.2 Heuristic h2

Let *T* is a set of all paths. Choose the path as follows:

- *Step 1.* The set *T* of paths is sorted according to the weakest node on the path.
- *Step 2.* Select subset *T1* from *T*, such that it contains a node with the highest value of energy of the weakest node.
- *Step 3.* Randomly select a path from the set *T1*.

```
S-A70%-B80%-D                      }     }
S-E90%-A70%-B80%-D {selected path} } T1  }
S-C40%-B80%-D                      }     }
S-C40%-A70%-B80%-D                       } T
S-A70%-C40%-B80%-D                       }
S-E90%-A70%-C40%-B80%-D                  }
```

Figure 5.2.1. Example of choice a path by heuristic *h2*

5.3 Heuristic h3

Let *T* is a set of all paths. Choose the path as follows:

- *Step 1.* The set *T* of paths is sorted according to the weakest node on the path.
- *Step 2.* Select subset *T1* from *T*, such that it contains a node with the highest value of energy of the weakest node.
- *Step 3.* Randomly select a path from the set *T1*.

```
S-E90%-A70%-B80%-D                 }     }
S-A70%-B80%-D {selected path}      } T1  }
S-C40%-B80%-D                      }     }
S-C40%-A70%-B80%-D                 }     } T
S-A70%-C40%-B80%-D                       }
S-E90%-A70%-C40%-B80%-D                  }
```

Figure 5.3.1. Example of choice a path by heuristic *h3*

5.4 Heuristic h4

The heuristic *h4* behavior is similar to many another routing protocols which select the shortest path between two nodes in the network.

Let *T* is a set all paths. Choose the path as follows: select the shortest path from the set *T*.

$$\left.\begin{array}{l}\texttt{S-E90\%-A70\%-B80\%-D}\\ \texttt{S-A70\%-B80\%-D}\\ \texttt{S-C40\%-B80\%-D \{selected path\}}\\ \texttt{S-C40\%-A70\%-B80\%-D}\\ \texttt{S-A70\%-C40\%-B80\%-D}\\ \texttt{S-E90\%-A70\%-C40\%-B80\%-D}\end{array}\right\}T$$

Figure 5.4.1. Example of choice a path by heuristic *h4*

5.5 Heuristic h5

The heuristic *h5* with randomly selected path we use in order to check what is the influence of random choice of path. The strategy usually allows to leave the local optimum. In this case we will omit the most frequently used paths.

Let *T* is a set all paths. Choose the path as follows: randomly select path from the set *T*.

$$\left.\begin{array}{l}\texttt{S-E90\%-A70\%-B80\%-D}\\ \texttt{S-A70\%-B80\%-D}\\ \texttt{S-C40\%-B80\%-D}\\ \texttt{S-C40\%-A70\%-B80\%-D}\\ \texttt{S-A70\%-C40\%-B80\%-D \{selected path\}}\\ \texttt{S-E90\%-A70\%-C40\%-B80\%-D}\end{array}\right\}T$$

Figure 5.5.1. Example of choice a path by heuristic *h5*

6 Genetic Algorithms

GA are adaptive heuristic search algorithms premised on the evolutionary ideas of natural selection and genetic. We use GA-based approach to find a heuristic combination for solving routing problem and taking into account battery energy.

6.1 The algorithm

The GA is can be described as follows. Firstly, we generate random population *P* of *n* chromosomes (suitable solutions of the problem). Next, we evaluate each chromosome *x* in the current population *P* according to the fitness function *f(x)*. In each iteration of the algorithm called a generation, we sequentially performs three basic genetic operators: selection, crossover and mutation. Then, the new children replace members of the current population *P*. Next, we again compute the fitness *f(x)* of each chromosome *x* in the population. The process is repeated a fixed number of times. If the algorithm stops, we return the best solution (chromosome) in the current population *P*.

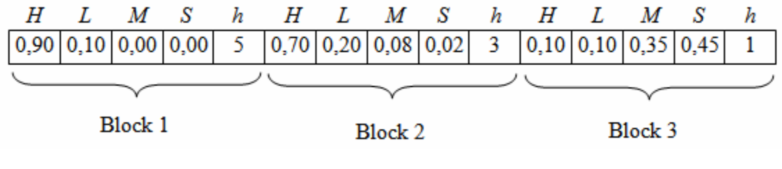
```
Chromosome: x=(b1, b2, ..., bn) {structure representing
a solution};
Optimization criterion: maximizing the routing metric;
Initialize population: P() {set of chromosomes};
t = 0;
Evaluate P(t);
WHILE termination condition NOT TRUE
       t = t + 1;
       Select P(t) from P(t-1) to reproduce;
       Crossover in P(t);
       Mutate in P(t);
       Evaluate P(t);
DO
Problem_solution = the best chromosome from P(t).
```

Figure 6.1.1. Pseudo-code of the genetic algorithm

6.2 Chromosome representation

A chromosome is composed of blocks. Each block j contains five numbers: H_j, L_j, M_j, S_j, h_j. The 1st four numbers represent the ad hoc network state, all in the range 0 to 1. The 5th number h_j, is an integer in the range 1 to 5 indicating which heuristic is associated with this block. An example of chromosome can be seen in Fig. 6.2.1.

The network state in time t is described by four numbers. The numbers describe how many nodes are into each of four categories: *L*, *H*, *M*, *S*. The current network state (*CNS*) is a point inside the 4-dimensional space. On the other hand, the blocks in the chromosome also represent a number of points inside the 4-dimensional space, and each point is associated with the heuristic *h*. Currently operating ad hoc network changes its states represented by *CNS* points. Each node in MANETs has the same chromosome which is the problem solution. If the node *S* wants to send data-packets to the node *D* it must choose a point (block) from the chromosome closest to point describing actual network state and use associated heuristic with the chosen block to routing. We do not know how many blocks are needed. Therefore we use a variable length chromosome.



H	*L*	*M*	*S*	*h*	*H*	*L*	*M*	*S*	*h*	*H*	*L*	*M*	*S*	*h*
0,90	0,10	0,00	0,00	5	0,70	0,20	0,08	0,02	3	0,10	0,10	0,35	0,45	1

Block 1 Block 2 Block 3

Figure 6.2.1. Example of chromosome consisting of three blocs

6.3 Fitness function

The fitness describes how well a chromosome solves a specified problem. In this case it is a number of data packets which were sent between network nodes computed from a moment when the network starts up, to the moment when the first battery in any nodes runs down.

6.4 Genetic operators

We have implemented a proportional selection operator and one-point crossover. We have variable length chromosomes, therefore the point to be selected on the shortest parent. We have three mutation:

- Add-block mutation. The mutation generates a new block and adds it to the chromosome. The first four numbers of a block are randomly generated in the range 0 to 1. The fifth number (an integer in 1 to 5) is generated randomly too.
- Remove-block mutation. The mutation removes a random block from the chromosome.
- Normal mutation. The mutation changes a locus (only one element from a block). In randomly chosen locus a suitable value is inserted. The value depends on the position in the block. For the first four cells in the block we inserted a value in range 0 to 1. For the fifth cell we inserted an integer in range1 to 5.

A child has a chance of being mutated with a certain probability. If it is to be mutated, then one type is chosen.

7 Experimental study

We have conducted an initial set of experiments to valid this approach. We simulate the routing algorithm in a stationary multihop wireless network where nodes are fixed (see Fig. 7.1). The ad hoc network have sixteen nodes called from *A* to *P*. The nodes are uniformly distributed over a square region of 300 meters on each side. The distance between each peer nodes equal 100 meters. A sender node may reach a node 105 meters away at its maximum transmission power.

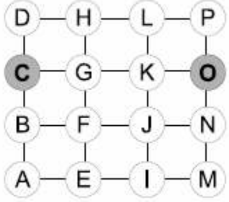


Figure 7.1. Topology of the tested network. Lines are symbolizing current connections between nodes in the network

Each nodes have the same battery capacity equal to 50 units. At the beginning the nodes have the same (full) energy. Exception are two nodes: *C* and *O*. We assume that they have an unlimited energy. Node *C* is a messages source. Node *O* is a messages destination. The node *C* sends messages only to node *O*. Each message has the same size. Sending one message or retransmission by intermediary node one message causes reducing energy of battery. One message reduces energy of battery in node in amount of one unit. Energy of battery in nodes *C* and *O* is constant, always full. The messages used to discover routes do not reduce energy of battery.

The GA described above was run five times with different seeds. The size of the population is 30 chromosomes, the GA runs 130 generations. The initial chromosomes are composed of 15 randomly created blocks. Crossover probability is equal to 0,7, mutation probability for each types of mutation is equal to 0,05. Problem solution - the best chromosome is evaluated 10 times. For each chromosome in each generation the ad hoc network starts from beginning. We record for each generation the best chromosome and its fitness function value. We record the average fitness function value of all chromosomes in each generation. Moreover, we record the network life - average remaining battery energy of the nodes. We evaluated separately each heuristic running the algorithm 10 times and record the average result. Similarly to the previous procedure, we evaluated the best heuristic combination returned by GA. The results are following: *h1* – 148 messages (the best results), *h2* – 69 messages, *h3* – 50 messages, *h4* – 50 messages, *h5* – 59 messages and heuristic combination – 148 messages. The results shows that the best results gives the *h1* heuristic, but the combination of heuristics have achieved the same performance.

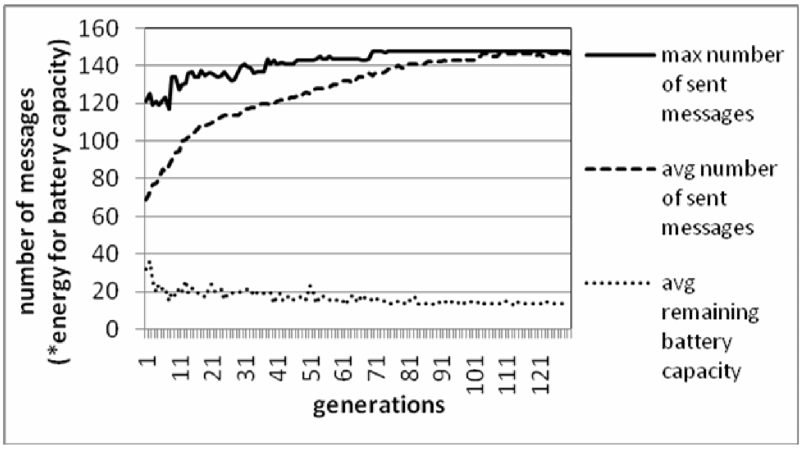


Figure 7.2. A number of sent and received messages as of function of a number of GA generations (averaged on 5 runs)

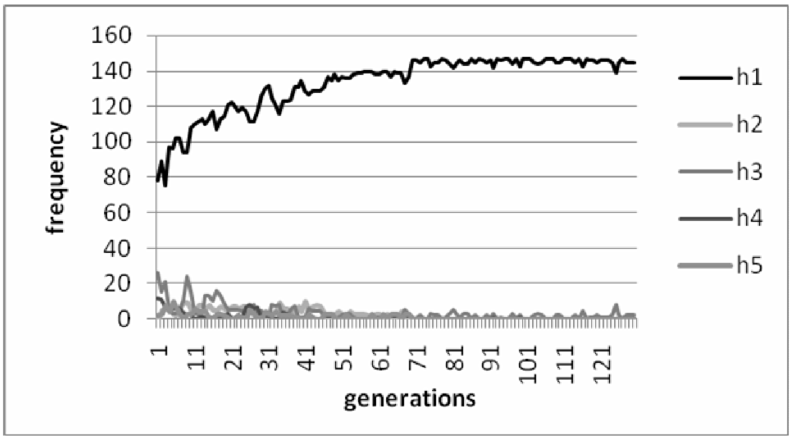


Figure 7.3. The frequency to use a single heuristic for the best chromosome from the population

Experiments shows that tested GA found the best solved already after 73 generations (see Fig. 7.2.). This chromosome have 14 blocks and it is using 8 times block with *h1* heuristic. Examples of found chromosomes (solutions): 10,4,45,38,(1); 5,26,29,38,(2); 40,27,2,29,(1); 18,22,28,88,(1); 14,34,43,8,(2); 8,14,45,30,(1); 43,7,29,19,(1); 91,49,17,19,(1); 38,36,10,14,(2); 15,9,30,44,(1); 18,16,26,39,(3); 37,38,20,3,(2); 19,36,14,29,(1); 17,26,21,33,(5). Next generations shows that GA found the next optimum for the energy efficient routing problem. Moreover, in the next generations the *h1* heuristic is favorite and often used (see Fig 7.3). To sum up, GA in the evolutionary process favours single the best *h1* heuristic.

8 Conclusion

This work presents an approach to developing the algorithm of an energy efficient routing protocols in MANETs. The problem is solved by continuous process of choosing paths between communicating nodes in MANETs and power management. The purpose of the proposed algorithm is to maximize a number data packets sent between nodes in MANETs. We proposed GA-based approach to find a heuristic combination for solving power-aware routing problem. A problem solution is such a heuristics combination, which depends on an actual state of the network and will choose optimal paths. We have already started our research in implementing the simulator program and energy efficient routing algorithm for MANETs. In the future we want to develop engineering solutions for largest MANETs (at least 100 nodes).

References

1. Abolhasan M., Wysocki T.A., Dutkiewicz E., (2004). *A review of routing protocols for mobile ad hoc networks*, Elsevier, Ad Hoc Networks Vol. 2, No. 1, 1-22.
2. Basagni S., Conti M., Giordano S., Stojmenovic I., (2004) *Mobile Ad Hoc Networking*, Published Wiley-IEEE Press.
3. Cui Y., Xue Y., Nahrstedt K. (2006). *A utility-based distributed maximum lifetime routing algorithm for wireless networks*, Vehicular Technology IEEE Transactions on Vehicular technology, Volume 55, Issue 3, 797-805.
4. Goldberg D.E., (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley.
5. Johnson D.B., Maltz D.A., Hu Y.C., (2004, visited: January 2008). *The dynamic source routing protocol for mobile ad hoc networks (DSR)* http://tools.ietf.org/id/draft-ietf-manet-dsr-10.txt
6. Pilski M., Seredyński F., (2008). *Hyper-heuristics for Power-aware Routing Protocols*, Proceedings of Artificial Intelligence Studies, Vol. 5 (28)/2008 Publishing House of University of Podlasie, Siedlce 2008, 93-100.
7. Ross P., Marin-Blazquez J. G., Schulenburg S., Hart E., (2003). *Learning a Procedure That Can Solve Hard Bin-Packing Problems: A New GA-Based Approach to Hyper-heuristics*, Genetic and Evolutionary Computation - GECCO 2003, Springer-Verlag Berlin Heidelberg, LNCS 2724, 1295-1306.
8. Tseng Y.C., Ni S.Y., Chen Y.S., Sheu J.P., (2002). *The broadcast storm problem in a mobile ad hoc network*, Wireless Networks, vol. 8, Short version in MOBICOM 99, 153-167.
9. Wedde H.F., Farooq M., Pannenbaecker T., Vogel B., Mueller C., Meth J., Jeruschkat R., (2005). *BeeAdHoc: An Energy Efficient Routing Algorithm for Mo-bile Ad Hoc Networks Inspired by Bee Behavior*, Proc. of Genetic and Evolutionary Computation Conference, 153-160.