A multi-objective approach for high quality tree-based backbones in mobile ad hoc networks

Apivadee Piyatumrong*

National Electronics and
Computer Technology Centre (NECTEC),
Klong Luang, Pathumthani 12120, Thailand
Fax: +66-(0)2564-6776
E-mail: apivadee.piyatumrong@nectec.or.th
*Corresponding author

Pascal Bouvry

Faculty of Science, Technology and
Communication (FSTC),
University of Luxembourg,
Campus Kirchberg 1359, Luxembourg
Fax: +352-46-66-44-5620
E-mail: pascal.bouvry@uni.lu

Frédéric Guinand

Laboratory for Computer Science,
Information Processing and Systems (LITIS),
University of Le Havre,
BP540 – 76058 Le Havre Cedex, France
Fax: +33-(0)2-327-44-314
E-mail: frederic.guinand@univ-lehavre.fr

Kittichai Lavangnananda

School of Information Technology (SIT),
King Mongkut's University of Technology Thonburi (KMUTT),
Thungkru 10140 Bangkok, Thailand
Fax: +66-(0)2872-7145
E-mail: kitt@sit.kmutt.ac.th

Abstract: Topology management for mobile ad hoc networks is usually based on the notion of virtual backbones. We address the notion of quality of such backbones as a multi-objective problem addressing quality of the mobiles composing the backbone and of the connection links. We validate the approach using extensive simulation based on different mobility models and using backbones composed of spanning forests. We provide ways to address the global behaviour of the algorithms by fine-tuning the greedy local decision policy and compare it to the solution quality that could be achieved using global knowledge and metaheuristics.

Keywords: topology management; mobile ad hoc network; MANET; multi-objective; spanning forest.

Reference to this paper should be made as follows: Piyatumrong, A., Bouvry, P., Guinand, F. and Lavangnananda, K. (2012) 'A multi-objective approach for high quality tree-based backbones in mobile ad hoc networks', *Int. J. Space-Based and Situated Computing*, Vol. 2, No. 2, pp.83–103.

Biographical notes: Apivadee Piyatumrong received her PhD in Computer Science in 2010 from Faculty of Science, Technology and Communication (FSTC), University of Luxembourg (UL), Luxembourg. Currently, she is working as a researcher in Large Scale Simulation Research Laboratory (LSR) at National Electronics and Computer Technology Centre (NECTEC), Thailand (http://www.nectec.or.th). Her research areas of interest are mobile ad hoc network, optimisation, parallel computing, information grid and dynamic system.

Pascal Bouvry received his PhD in Computer Science from the University of Grenoble, Grenoble, France in 1994. He is currently a Professor with the Faculty of Sciences, Technology and Communication, University of Luxembourg, Luxembourg City, Luxembourg, and the Head of the Computer Science and Communication Research Unit (http://csc.uni.lu). He specialises in parallel and evolutionary computing. His current research interests include the application of nature-inspired computing for solving reliability, security, and energy-efficiency problems in clouds, grids, and ad hoc networks.

Frédéric Guinand received his Bachelor and Master in Computer Science at Joseph Fourier University (Grenoble). He received his PhD in 1995 from Grenoble Institute of Technology (INPG) in France and did his Postdoc at Swiss Federal Institute of Technology (EPFL). Currently, he is a Professor at University of Le Havre. His research interests are in dynamic graph theory, distributed and mobile computing, optimisation under uncertainties and interactions between computer science and biology. He recently co-edited the book *Artificial Ants* of Wiley in 2010.

Kittichai Lavangnananda received his BSc in Computational Science from The University of Hull, UK in 1985 and MSc in Computing at The University College Cardiff, UK in 1986. He completed his PhD studies in Artificial Intelligence (AI) at Cardiff University, UK in 1996. At present, he is an Associate Professor and the Head of Software Technology Division at School of Information Technology (SIT), King Mongkuts University of Technology Thonburi (KMUTT), Bangkok, Thailand (http://www.sit.kmutt.ac.th). His current research interest is in the applications of AI, computational intelligence in particular, in data mining, knowledge discovery, spatial economics and IP-based communications systems.

1 Introduction

Mobile ad hoc networks (MANETs) gained the vast majority of interest in the past years. This class of network expects no infrastructure. The communication between nodes, typically are mobile nodes, is done through wireless technologies (Corson and Macker, 1999; Murthy and Manoj, 2004). The primitive concern of communication is quality of services. Due to lacking of central authority in MANET, provisioning and managing resources are difficult. Topology management has its aim to handling or control the network topology in order to provide better control over network resource and to increase the efficiency of communication (Santi, 2005). The constraints of managing topology in a MANET are defined by the characteristics of the network itself. Topology management algorithm must work in decentralised scheme, use asynchronous algorithm and use only local knowledge (Piyatumrong, 2010). This is because centralised algorithm and large amount of knowledge are not scalable and not realistic in MANET scheme.

Topology management for a network can be done either at physical or at logical level. Typically found in MANETs, topology management selects some mobile nodes and/or some communication links to be included in logical topologies, frequently called virtual backbones (Schleich et al., 2009). Depending on the desired properties of topology, there exists varieties of topology structures in literature. This work focuses on tree-based backbone due to multiple benefits found in literature such as low bandwidth usage for forwarding messages, each nodes retransmit message only once (Radhakrishnan et al., 2003; Jüttner and Magi, 2005). Furthermore, tree-based structure also has been used as an internal process for other network operations and protocols, for example, utilising existing tree-based structure for clustering technique (Srivastava and Ghosh, 2002) and for finding dominating set (Alzoubi et al., 2002). Thus, benefits found in this present work can be used to enhance such operations and protocols accordingly.

Not only selecting topology type according to the desired properties, the quality of elements to be included in the virtual backbone is also important. In a MANET, there exists two primitive elements which are communication node and communication link. This work addresses the notion of quality of virtual backbones as a multi-objective problem by addressing qualities of the mobile nodes and of the communication links composing the virtual backbone. Depending on the objective of virtual backbone at a particular time, the importance of each quality criterion may change which influent to the different topology selection and adjustment process accordingly. This work aims at constructing multi-objective spanning forest backbone in a *dynamic and disconnected environment of MANETs* [so called delay-tolerant MANETs (DT-MANETs)]. The difficulties are founded in the dynamicity of the underlying network and the lacking of central authority. These issues suggest the utilisation of an efficient decentralised algorithm that can construct multi-objectives tree-based backbone.

2 Characteristics of the targeting network – DT-MANETs

A MANET consists of mobile platforms, simply referred to nodes, which are free to move about arbitrarily. This ad hoc topology may change with time as the nodes move (Corson and Macker, 1999). We focus in particular at disconnected topologies of dynamic MANET, which can be called DT-MANETs. DT-MANETs employ the concept of delay tolerant and disruption tolerant network

(DTNs) (Delay Tolerant Networking Research Group, n.d.), but used in MANET environment instead of a deep space environment. Examples of related works for DT-MANETs are, such as Juang et al. (2002), Pathirana et al. (2005) and Abdulla and Simon (2006). The architecture includes the concepts of occasionally-connected networks that may suffer from frequent partitions. Actually, a DT-MANET is basically a combination of main key characteristics of DTNs and MANETs, which are the partitioning, mobile and ad hoc networks. The characteristic of DT-MANET as described above is similar to those networks called highly-partitioned wireless ad hoc network (Zhao and Ammer, 2003), sparse MANETs (Zhao et al., 2004) and disconnected MANETs (Pelusi et al., 2006). Regardless of the name, in summary, DTMANET is a challenged environment of MANETs where end-to-end connectivity cannot always be obtained. Such environment can be compared as a disconnected or partitioned MANETs. In this platform, usually the messages can be stored temporary on stations and are forwarded later when the situation allows to do so. Key problem of such network is the successful communication.

3 Related tree problems

Given a connected graph $G = \{V, E\}$, a spanning tree is a covering subgraph which includes every vertex of V and which is an acyclic graph. Spanning tree has been studied and used widely in many fields such as circuit design problems, communication network design problem, communication protocol problems, etc. In communication network, an important activity is to rule or to guide information through network, so that information arrive at the projected destination(s). The very first benefit of using spanning tree as a routing algorithm falls on the fact that tree has only $n - 1$ links over a connected network. Thus, in a fully connected network, spanning tree has a benefit as a small portion of communication links is used. Spanning trees are also used in broadcasting process which operates to sends a message from a source node to all other nodes within communication range. The acyclic structure of the broadcast tree ensures that the nodes in tree need to repeat the broadcast messages only once and thus spend little bandwidth usage (Jüttner and Magi, 2005).

From such benefits, we focus on studying topology management on a logical topology approach of tree-based topology. For the rest of this section, we discuss the problem of constructing spanning tree in different domains and provide accordingly state-of-the-art.

3.1 Minimum spanning tree problem – single-objective MST problem

Given a connected, undirected graph $G = \{V, E\}$, where V is the set of vertices, E is the set of edges. A weight $w(u, v)$ is associated to each edge $(u, v) \in E$. A minimum spanning tree (MST) $T = (V, E_T)$ is a spanning tree over G such that $w(T) = \sum_{(u,v)\in T} w(u,v)$ is minimised. Finding MST is a very well-known problem in which many contributions are proposed using different conditions and constraints. Single-objective minimum or maximum spanning tree problem has been proven to be solvable in polynomial time using greedy algorithms if global information of the system is known. The famous examples of such algorithms are Borůvka's (1926) algorithm, Kruskal's (1956) algorithm, Prim's (1957) algorithm. Furthermore, the distributed or parallel algorithm for MST problem has been studying since 1983 (Gallager et al., 1983; Faloutsos and Molle, 1995) and still is interesting in many research fields. Main example of such fields are of parallel computing and networking (e.g., overlay network, DT-MANETs). However, the algorithms proposed by both Gallager et al. and Faloutsos and Molle only consider a fixed network infrastructure, assume unlimited bandwidth and assume no collision of messages; all of these assumptions are not valid for DT-MANETs. Thus, they are not suit to the problem constraints assumed in this work. Meanwhile, some works focus at constructing MST within a bound communication rounds and number of messages such as Lotker et al. (2003). Unfortunately, it, again, considers no boundary level of local information, so every node will receive fragmented local information from all nodes in the network and then each node can compute global information. The constraints found in the previous works leads to the need of a new generation algorithm that solves single-objective MST using purely decentralised algorithm and a very local information as input. To date, there is no an algorithm for solving MST using such constraints found in literature.

3.2 Multi-objective MST problem

Considering a weighted graph $G = (V, E, w^k)$ where w_e^k denotes the weight of edge $e \in E$ at particular weight set k, where $k = \{1, 2, \ldots, m\}$ denotes different set of weight which may represent the distance, cost and so on. The multi-objective MST (or multi-criteria MST) problem can be formulated as follows:

Definition 3.1: Multi-objective MST problem

$$\min f_1(\gamma) = \sum_{e\in E\gamma} w_e^1;$$
$$\min f_2(\gamma) = \sum_{e\in E\gamma} w_e^2;$$
$$\ldots$$
$$\min f_m(\gamma) = \sum_{e\in E\gamma} w_e^m;$$

where $f_i(\gamma)$ is the i^{th} objective to be minimised for the problem.

Adding another dimension of objective to minimum or maximum problem has been proven to be an NP-Hard problem (Hamacher and Ruhe, 1994). Accordingly, approximation methods must be used for tackling such problem efficiently. An improved enumeration algorithm to give out all real Pareto optimal solutions for the multi-objective MST problem is presenting in Chen et al. (2007). However, Chen et al. assumed a connected and undirected graph which different from what we focus in this work. In literature, finding Pareto optimal solutions of a disconnected and undirected graph has not yet been addressed.

In this work, we interested in constructing and maintaining multi-objective tree-based backbone for a disconnected and dynamic network of DT-MANET. The goal of constructing and maintaining multi-objective tree-based backbone is to give the best tree-based topology according to multiple objectives in the given environment at a particular time. Since the network is dynamic and disconnected, distributed and decentralised method is needed for constructing and maintaining operations.

4 Quality issues

In general, virtual backbone is used for some communication operations and management (e.g., routing, multicasting, broadcasting and disseminating information to a particular set of nodes). Quality of a backbone is described according to the objective of the backbone. In this work, we focus on construction of tree-based backbone due to its benefit found in literature. Furthermore, we aim at two different types of quality found at the primitive level of the network component, communication nodes and edges.

The qualities of communication nodes and edges can be found in variety aspects. In this work, we select only two aspects to be used as examples. They are cooperative enforcement aspect and capacity bandwidth (CB) aspect. The following sub-section describes both quality aspects which actually are the quality found in communication nodes and edges respectively. Later, qualities of tree-based backbone are defined by two different objectives based on nodes and edges' qualities accordingly. It is worth noting that an assumption of this work is there exists the quality value of node and edge and they are ready to be used.

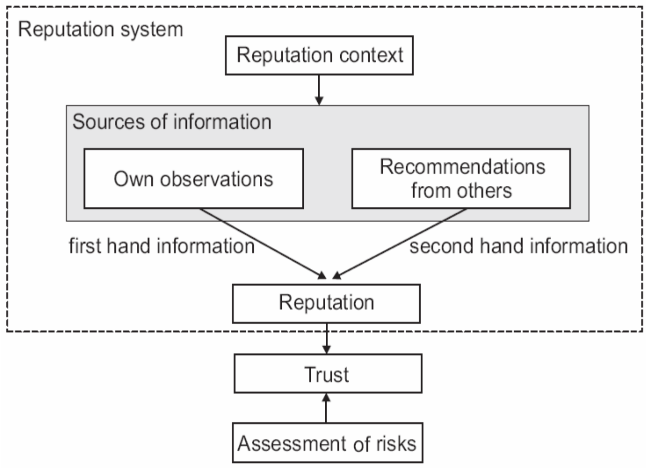
4.1 Nodes and edges' qualities

4.1.1 Cooperative enforcement aspect: trust level

Ad hoc networks rely on 'cooperation' of a set of nodes in order to emerge and operate the network. A single node can deteriorate the well-being of an ad hoc network simply by not forwarding messages. Cooperative enforcement approaches have been proposed by a number of researchers to enhance the robustness, the availability and/or the overall throughput (Marias et al., 2006) in pure MANETs. The main objective of those works is to cope with 'selfish nodes'. Such nodes can deteriorate the robustness of the network because they do not give collaborative efforts (e.g., by not forwarding a packet received from others). In this paradigm, the terms of 'trust' and 'reputation' are used as judgement values representing the cooperative level or trust level toward other nodes in the community. This work focuses on utilising this trust value for strengthening the quality of topology in a DT-MANETs.

Trust and reputation have a close relationship and often have been treated synonymously (Seredyński, 2009). However, they actually have different meaning and are depending on the context area. The relationship between trust and reputation has been summarised in Figure 1. To this work, we regard 'trust' as the resulting combination of value from reputation system and self-evaluation of risk according to the figure. The evaluation of trust in context of MANETs has been proposed in many works, good examples of such works can be found in Huynh et al. (2006), Marti et al. (2000), Wrona (2002), etc. A comprehensive survey on cooperation enforcement in MANETs can be found in Marias et al. (2006), while detailed discussion on peer-to-peer key and trust management approach in MANETs can be found in Merwe et al. (2007).

Figure 1 Relation between trust and reputation



Source: Seredyński (2009)

Definition 4.1: Trust

In a communication graph $G = (V, E)$, we denote $trust(v1)$ a trust value of node $v_1 \in V$.

4.1.2 CB aspect

Nodes in an ad hoc wireless network share a common broadcast radio channel. Since the radio spectrum is limited, the bandwidth available for communication in such networks is also restrained. Bandwidth represents the capacity of the connection. The greater the capacity, the more likely that greater performance will follow, though overall performance also depends on other factors, such as latency. The CB of each communication edge gives us an immediate idea of how well the edge prompts to transfer data. In summary, the higher the CB, the more data can be transfer through this edge. From an end-to-end communication perspective, the data cannot travel any faster than the smallest bandwidth link involved between two entities. In this work, such the smallest bandwidth link is also called the bottleneck bandwidth edge.

Definition 4.2: Capacity bandwidth

In a communication graph $G = (V, E)$, we denote $CB(v_1, v_2)$ a CB value of edge $(v_1, v_2) \in E$.

4.2 Tree-based backbone's qualities

Managing high quality tree-based backbone, a proper set of edges and nodes must be selected or arranged in order to achieve the desired quality. At a particular moment and a configuration of network, the desired topology may have different needs. Similar to other real world applications, high quality topology may comprises of many factors of network elements' qualities. For some qualities, they may have a strong correlation or no correlation. For the latter case, it may happen that they are conflicting qualities. This work studies on two conflicting criteria (non-correlation criteria) of nodes and path's quality.

The main idea of high quality tree-based backbone is to have high quality nodes acting as router (forwarder) and less quality nodes being at leaves position of tree. On the other hand, less quality edges should not be included in the tree structure if not necessary. These comprise of two different objectives of tree-based backbone and are presented here in the following two sub-sections.

4.2.1 High node's quality tree-based backbone

As discussed under Section 4.1.1, nodes with higher trust level are more likely to contribute and complete their tasks than lower one. Having high node's quality tree-based backbone is to have high quality nodes acting as router (forwarder) or to have less quality nodes being at leaves position of tree. This can be presented and measured using *nodeWeight*() function. The idea was first introduced in Piyatumrong et al. (2008b) to assess robust spanning trees with respect to the same objective. Having $V(\gamma)$ as the set of all nodes in a tree γ, the *nodeWeight*() function of a trusted spanning tree can be determined by the following equation:

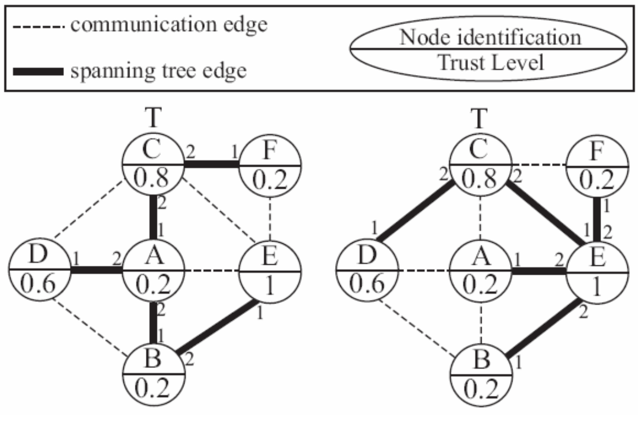
$$nodeWeight(\gamma) = \sum_{x \in V(\gamma)} trust(x) \times tree\_degree(x) \qquad (1)$$

The function *trust*(x) represents trust quality of node x, while *tree_degree*(x) represents the number of one-hop neighbours of node x in the tree (emphasise here that concerning tree neighbours and not physical neighbours).

Figures 2(a) and 2(b) are examples to illustrate how the *nodeWeight*() function can assess this quality where the *threshold* used in this particular example equals to 0.2. In Figure 2(a), the node with lowest trust level gets the highest *tree_degree*, while the node with highest level gets the lowest *tree_degree* (i.e., the node A has a trust level of 0.2 and *tree_degree* of 3, while the node E has a trust level of 1 and *tree_degree* of 1), hence the *nodeWeight*(γ_a) function for this trusted spanning tree is 4.4. Figure 2(b) depicts the opposite [i.e., the node with the highest trust level possesses the highest *tree_degree* (node E), while the node with the lowest level possesses the lowest *tree_degree* (node A)]. The *nodeWeight*(γ_b) function for this trusted spanning tree is 6.8. In order to measure this function on the whole graph (the spanning forest), equation (2) is introduced below

$$nodeWeight(G(t)) = \sum_{\gamma \in G(t)} nodeWeight(\gamma) \qquad (2)$$

Figure 2 An example scenario for Illustrating the details of *nodeWeight*() cost functions



Problem description. Each node is given a quality using trust level (*trust*()). We are looking for a set of spanning trees, (Γ^*), in a communication graph at moment t, $G(t)$ such that:

$$nodeWeight(\Gamma^*) = nodeWeight(G^*) = \max_{\forall \gamma \in G(t)} \left(\sum_{x \in V(\gamma)} trust(x) \times tree\_degree(x) \right)$$

4.2.2 High edge's quality tree-based backbone

As discussed under Section 4.1.2, edges with higher CB tends to give a higher quality of service than lower one. This can be presented and measured using *pathWeight*() function. Has its focus on an end-to-end communication, the quality measurement introduces by *pathWeight*() function measuring all possible paths in a constructed spanning tree. Since the low value of one edge can effect the quality of all the whole communication path as it is the bottle neck edge, *pathWeight*() function interests only the minimum value (the bottleneck value) of each path. The higher the sum value of all path, thus means the better the quality of the tree and the more robust the tree structure.

Each edge is given a quality. We denote *MinQuality* ($P(a, b)$) as the minimum quality value of any edge belonging to path $P(a, b)$, $\min_{e \in P(a,b)} quality(e)$. The robustness metric *pathWeight*() function is defined as follow:

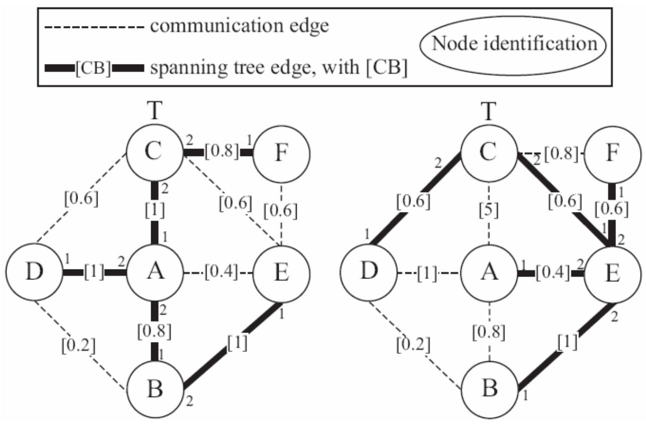
$$pathWeight(\gamma) = \sum_{\forall (a,b) \in E(\gamma)} MinQuality \;\; P(a,b) \qquad (3)$$

In order to measure this function on the whole graph (the spanning forest), equation (4) is introduced below.

$$pathWeight(G(t)) = \sum_{\gamma \in G(t)} pathWeight(\gamma) \tag{4}$$

Thus, the robustness metrics are reviewed here briefly using example in Figure 3. In the figures, value of quality is presented in the middle of each edge. The spanning tree in Figure 3(a) (γ_a) comprises of five edges where the minimum quality among these five edges is 0.8. This is the least quality value. Further investigating on this sub-figure, we found that the communication from node F using spanning tree to any other nodes in this connected subgraph will utilise maximum quality at 0.8. Meanwhile, the communication occurs between nodes A, C and D can use the maximum quality of edge at 1. Actually, the spanning tree in Figure 3(a) gives us the best spanning tree regarding to the given quality criteria. Since the minimum quality on this tree is 0.8, it is guaranteed that the communication can use the quality of edge at least 0.8 of quality level.

Figure 3 An example scenario for illustrating the details of robustness metric of *pathWeight*() function



On the contrary, Figure 3(b) gives us a spanning tree with the minimum quality at 0.4. This means many communications are limited their quality usage to this value. The *pathWeight*() function calculates all possible communication paths within a single tree. The results of the metric are 25.6 and 16.4 for Figures 3(a) and 3(b) respectively.

Problem description. The aim of this study is to construct robust topology of spanning forest by focusing on the CB of selected edges. Since the higher the CB regulate the higher quality of service, the spanning forest with a set of high CB edges are preferred. However, the end-to-end communication or communication path are limited its capacity by the edge with smallest CB value. Thus, this particular problem is to find a spanning tree of a connected component which maximise the minimum CB of all possible path. Each edge is given a weight using a CB. We denote *MinCB* (*P*(*a*, *b*)) as the minimum value of the CB of any edge belonging to *P*(*a*, *b*). We are looking for one path, *P**(*a*, *b*), such that:

$$MinCB(P^*(a,b)) = \max_{\forall (a,b) \in V(\gamma)} (MinCB(P(a,b))) \tag{5}$$

5 Optimum solution to each single-objective function

5.1 Reduction of weight(γ) objective

Given $G = (V, E)$ a communication graph

and γ a tree on $G(t)$

$\gamma = (V, E_\gamma)$

For each vertex v_i we note w_i its battery level

d_i its degree in γ

For a given γ we consider the following metric:

$$f_\gamma = \sum_{v_i \in V} w_i \times d_i^\gamma$$

Problem: We are looking for γ^* such that $f_{\gamma^*} = \max f_\gamma$.

Proposition: We claim that maximising *f* for *G* is equivalent to find a maximum spanning tree for *G*.

Proof: For one particular spanning tree γ:

$$\begin{aligned} f_{\gamma^*} &= \sum_{v_i \in V} w_i \times d_i^\gamma \\ &= \sum_{v_j \in N(v_1)} \left(w_1 + w_j\right) \\ &\quad + \sum_{v_i \in V \setminus \{v_1\} \setminus \{N(v_1)\}} w_i \times d_i^\gamma \\ &\quad + \sum_{v_k \in N(v_1)} w_k \times \left(d_k^\gamma - 1\right) \end{aligned}$$

where $N(v_1)$ is the set of neighbours of v_1.

$$\begin{aligned} &= \sum_{v_j \in N(v_1)} \left(w_1 + w_j\right) \\ &\quad + \sum_{v_l \in N(v_2)} \left(w_2 + w_l\right) \\ &\quad + \sum_{v_i \in V \setminus \{v_1, v_2, N(v_1), N(v_2)\}} w_i \times d_i^\gamma \\ &\quad + \sum_{v_k \in N(v_1) \setminus N(v_2)} w_k \times \left(d_k^\gamma - 1\right) \\ &\quad + \sum_{v_m \in N(v_2) \setminus N(v_1)} w_m \times \left(d_m^\gamma - 1\right) \\ &\quad + \sum_{v_n \in \{N(v_1) \cap N(v_2)\}} w_n \times \left(d_n^\gamma - 2\right) \end{aligned}$$

We continue the same transformation and it comes that

$$\sum_{v_i \in V} w_i \times d_i^\gamma = \sum_{(v_i, v_j) \in E_\gamma} \left(w_i + w_j\right)$$

Thus, $\max \sum_{v_i \in V} w_i \times d_i^{\gamma} = \max \sum_{(v_i,v_j)\in E_\gamma} (w_i + w_j)$

and $\max \sum_{(v_i,v_j)\in E_\gamma} (w_i + w_j)$ is obtained for the maximum spanning tree for G where each edge is evaluated by the sum of the battery level of its adjacent nodes.

Example (illustration): The example show in Figure 4 illustrates the equivalent of our claim that maximising f for G is equivalent to find a maximum spanning tree for G. The resulting spanning tree is depicted using red line and the optimal value of that tree is 647.

Summary: The single objective of *weight()function* problem can be transformed into maximum spanning tree problem. This is done by giving value on edge using an addition of adjacent nodes' battery level. Thus, this single objective problem is able to find optimum value using either Kruskal or Prim's algorithm when the global information is allowed.

5.2 *Reduction of pathWeight(γ) objective*

Given $G = (V, E)$ a communication graph

and γ a tree on $G(t)$

$\gamma = (V, E_\gamma)$

For each edge $(v_i, v_j) \in E_\gamma$ we note

$CB(v_i, v_j)$ its capacity bandwidth

Foreach Path exists in γ, $P(v_s, v_d)$ we note

$MinCB(P(v_s, v_d))$ as the minimum value of the CB of any edge belonging to $P(v_s, v_d)$

For a given γ we consider the following metric:

$$f_{\gamma^*} = \max_{\forall (a,b)\in E(\gamma)} (MinCB(P(a,b)))$$

Problem: We are looking for γ^* such that $f_{\gamma^*} = \max f_\gamma$.

Proposition: We claim that maximising f for G is equivalent to find a maximum spanning tree for G.

Proof: Considering graph G and one particular spanning tree γ_1 shown in Figure 5.

Consider a path of (v_x, v_y), where $\exists\, P(v_x, v_y) \notin$ spanning tree γ_1 such that $MinCB$ is greater than in γ_1. If this is the case, then there is an edge $(v_i, v_j) \notin \gamma_1$. We denote a spanning tree γ_2 contains such path, $P(v_x, v_y)$, which contains an edge (v_i, v_j) such that

$$\sum_{\forall (v_x,v_y)\in E(\gamma_2)} MinCB(P(v_x, v_y)) > \sum_{\forall (a,b)\in E(\gamma_1)} MinCB(P(a,b))$$

and

$$\sum_{\forall (v_x,v_y)\in E(\gamma_2)} MinCB(v_i, v_j) > MinCB(w, z),$$

where $(w, z) = E_{\gamma_1} \setminus E_{\gamma_2} \setminus (v_i, v_j)$.

Thus, only the maximum $MinCB$ of $|V| - 1$ edges of graph G must contains in the spanning tree that give maximising f for G. Hence, $\max_{\forall (a,b)\in E(\gamma)} (MinCB(P(a,b)))$ is the same topology set of spanning tree, edge set, as the maximum spanning tree for G. Thus, v_i, v_j is edge (D, F) found in Figure 5(b). Figure 5(c) illustrates the maximum spanning tree to example graph G in Figure 5(a).

Figure 4 An example illustrates the equivalent of robust spanning forest topology problem to maximum spanning tree, (a) robust spanning forest topology problem (b) maximum spanning tree problem (see online version for colours)

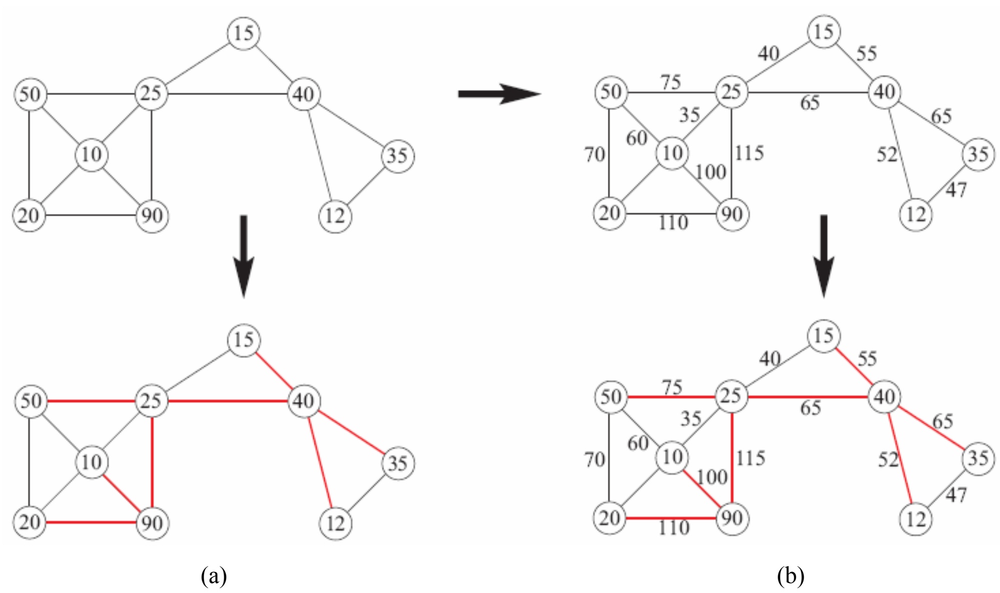
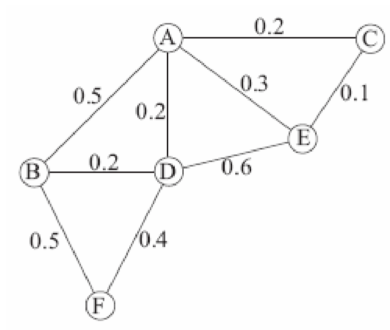
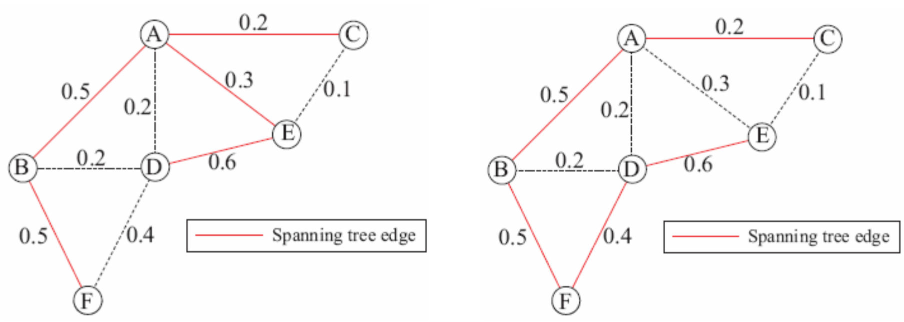


Figure 5 Example graph and spanning tree for reduction *pathWeight*() function proof, (a) graph G (b) one particular spanning tree γ_1 (c) maximum spanning tree of graph G (see online version for colours)

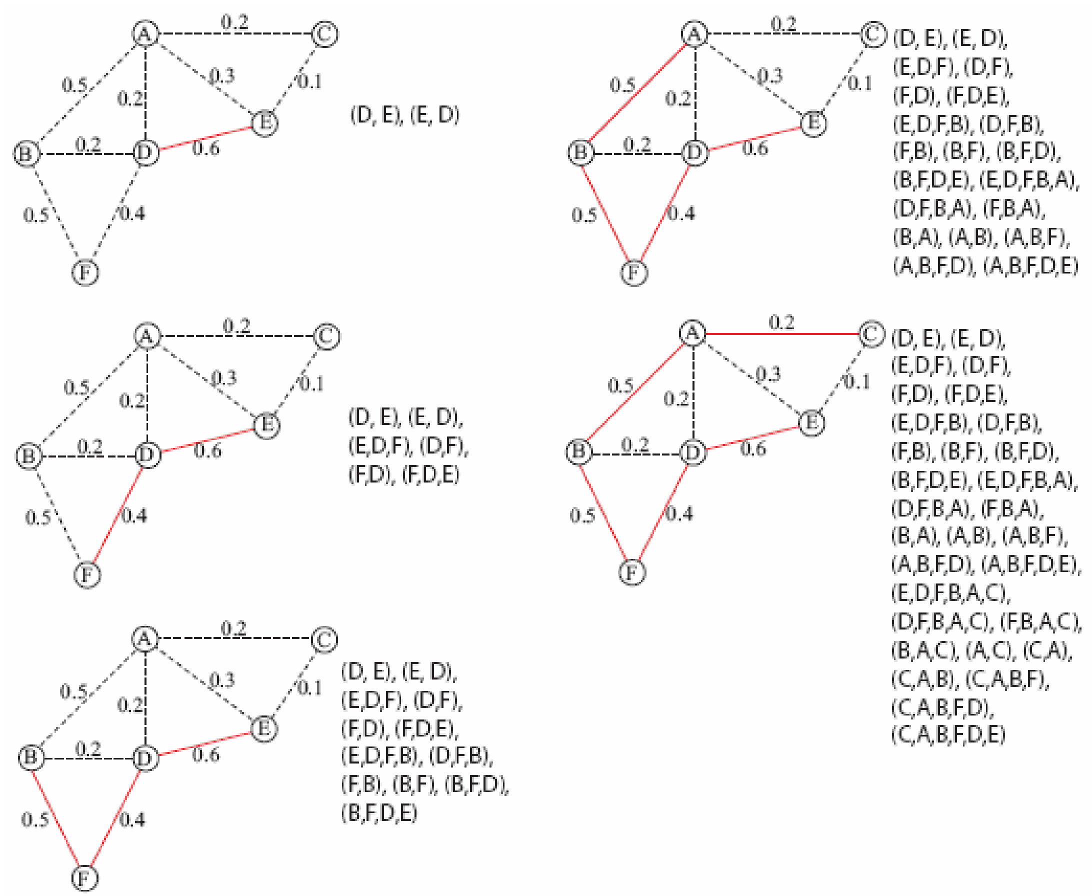


(a)



(b) (c)

Figure 6 Example of max *pathWeight*() spanning tree (see online version for colours)



Example (illustration): The example show in Figure 6 illustrates the equivalent of our claim that maximising f for G is equivalent to find a maximum spanning tree for G. The resulting spanning tree is depicted using red line. Although both figure show the same spanning tree, the optimal value obtained are different according to the *pathWeight*() function. However, it is guaranteed the maximisation of the solution.

Summary: The single objective of *pathWeight*() function problem can be transformed into maximum spanning tree problem. This is done by finding the maximum spanning tree based on CB value on each edge. Thus, this single objective problem is able to find optimum spanning tree using either Kruskal's or Prim's algorithm when the global information is allowed. However, the optimal value to the function *pathWeight*() differ from the summation of minimum CB of all possible path from the maximum spanning tree. The optimum solution of *pathWeight*() function can be found by calculated from the resulting maximum spanning tree.

6 Multi-objectives approach for managing tree topology

So far, we separately addressed problems of high quality spanning forest topology based on criteria of node and edge. However, it is more suitable to address the problem considering multi-criteria at the same time. Thus, we propose the multi-objective approach for managing tree topology in DT-MANETs.

Problem description. Different qualities of nodes and their communication links are used as quality criteria to topology management in DT-MANETs problem. This work proposes to consider multi-criteria at the same time. Robustness objectives of tree topology is addressed as bi-objectives problem, considering one criteria of node's quality and one criteria of edge's quality, that are described in Sections 4.2.1 and 4.2.2 respectively. By having conflicting quality criteria, a bi-objectives problem according to our interest is described formally as follow:

Given $G(t) = (V(t), E(t))$ a communication graph at moment t and γ a tree on $G(t)$. The problem of robust spanning forest topology in DT-MANETs is:

$$maximize\ \ nodeWeight(G(t)) = \sum_{\forall_\gamma \in G(t)} nodeWeight(\gamma)$$

$$maximize\ \ pathWeight(G(t)) = \sum_{\forall_\gamma \in G(t)} pathWeight(\gamma)$$

where $nodeWeight(\gamma)$ is defined in equation (1) and $pathWeight(\gamma)$ is at equation (3). It must be emphasised that multi-objective problem comprises of conflicting objectives and there is usually no single optimal solution. Multi-objective optimisation problem (MOOP) presents a possibly uncountable set of solutions, which when evaluated, produce vectors whose components represent trade-offs in objective space. Therefore, the decision maker is required to select a solution (or solutions) from a finite set by selecting one or more of these vectors and making compromises. To the problem of managing topology in DT-MANETs like this one, the proposed algorithm must have ability to provide such a variety set of solution vectors: leave the decision on which compromising level is proper at the moment to the application level.

7 State-of-the-art – DA-GRS

To the best of our knowledge, DA-GRS (a graph relabelling system) has the best potential for constructing tree-based backbone in decentralised manner and using local information. We discusses about this potential and some drawbacks that needs improvements in details in this section.

Dynamicity aware-graph relabelling system (DA-GRS) (Casteigts, 2006) is an extension of graph relabelling system, GRS (Sopena and Metivier, 1997). It is a high level abstraction model that can improve the development of self-organised systems. All the mechanisms within DA-GRS are suitable for managing DT-MANETs efficiently. DA-GRS models topology changes and interaction between devices following such topology changes using label system. However, it does not create services or applications itself.

Definition 7.1: Spanning forest given a graph $G = (V, E)$, a spanning forest F is a set of trees $F = \{tree_1 = (V_1, E_1), \ldots, tree_m = (V_m, E_m)\}$ such that $V = \bigcup_{1 \ldots m} V_j$.

Following Definition 7.1, DA-GRS proposes some rules for constructing and maintaining a spanning forest in DT-MANETs which are presented in Figure 7. In this figure, the black circle represents a node. Letters on top of the nodes mean:

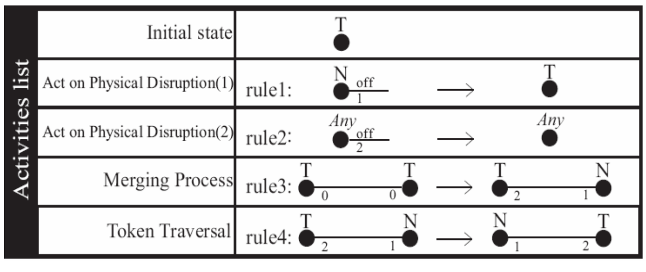
1 'T' if the node possesses the token

2 'N' if the node does not possess the token

3 'Any' when the node can possess or not the token.

The labels '0', '1' and '2' on the edge represent the route to the token. And finally, label 'off' describes a broken link.

Figure 7 DA-GRS rules for creating and maintaining spanning forest topologies with our mapping process name



Dynamic networks are characterised by mobility and possible connection disruptions, hence, devices need to handle with these changes when creating and maintaining the spanning tree. DA-GRS proposed four rules (as shown in Figure 7) to handle with four different situations. In the initial state every device has the token (what means it is a tree itself), and these four rules are:

- Rule 1: A tree link breaks, and the node belongs to the sub-tree which does not possess the token. In this case the node must regenerate the token, otherwise there will exist a tree without a token (which is an undesirable situation).
- Rule 2: A tree link breaks, and the broken link occurs at a node which currently belongs to the sub-tree which possesses the token. In this case, the node does nothing regarding the maintenance of the token.
- Rule 3: When a node with token meets another device possessing a token; both nodes will try to merge their trees in order to obtain a bigger tree from the two existing ones. The trees merging process starts. The result of this rule remains a bigger tree and only one token (the merging process discards one token automatically in order to remain one and only one token within a tree).
- Rule 4: Token traversal in general case: the token visits the nodes of the tree following a given strategy.

An important feature of this model is that in each tree one and only one token exists. Furthermore, only two nodes possessing token (thus there exists different trees) can start the trees merging process. As we are dealing with trees, cycles are not allowed. DA-GRS manages to avoid them since it is not possible to have two nodes belonging to the same tree and possessing a token at the same time.

Regarding the four rules of DA-GRS's spanning forest, we map all rules into different processes as shown in Figure 7 and give analysis as follow. First, it uses the token management to ensure the loop free of any tree. Moreover, the node possess the token can be regarded as a root of tree and is granted to do merge operation. This means the root of tree is always walking through tree which increases the chance in order to meet with other tokens and also reduce the communication messages for asking permission and granted or denied request to merge as found in Radhakrishnan et al. (2003). Furthermore, the algorithm described here concerns only the label changes (status changes) within one-hop neighbour. This follows the requirement of using only local information.

However, the synchronisation of the model is not compatible to asynchronous communication as we expected. DA-GRS's spanning forest uses rendez-vous assumption as synchronisation method at merging process. This rendez-vous assumption states that at one moment in time, only two nodes possessing token can meet and be merged. Furthermore, the token traversal process is based on randomness. This stochastic process makes it difficult to converge all trees into a spanning tree (i.e., slow to form a single spanning tree over a connected component). Pigné (2008) studied token traversal process of DA-GRS extensively and proposed TABU heuristics, a deterministic-based heuristic, which improves the convergence of trees above a connected component. Later, Piyatumrong et al. (2009) proposed *DFSmove* that can improved the convergence over Random process of DA-GRS and TABU heuristic of the previous work. Other important points are

1 DAGRS model is an abstract and high level model

2 it is incapable of communication realisation and neighbour discovering method.

In summary, DA-GRS proposed distributed algorithms that use only local information (e.g., one-hop information) to construct and maintain spanning forest over DT-MANETs. As drawbacks were found, this study improves DA-GRS to construct and maintain spanning forest in DT-MANETs. Hereinafter, we use the word 'DA-GRS reference algorithm' to represent these original four rules algorithm (shown in Figure 7) for constructing and maintaining spanning tree presented in this section, while 'DA-GRS model' is used to refer to its original meaning as a model.

8 Proposed algorithm for constructing multi-objective tree-based topology

Instead of tackling the multi-objective tree directly, we develop two different algorithms for constructing single-objective tree based on nodes and edges' quality which are G-NODE and G-PATH. Hence, we first introduce first G-NODE and G-PATH under this section. Then, our algorithm for multi-objective tree utilises these two primitive algorithms for constructing multi-objective tree-based backbone. We name this algorithm as G-Node-Path.

As mentioned in previous section, DA-GRS have some drawbacks to work in the environment and assumptions that we used for this work, such as the unrealistic synchronisation method and the lacking of communication realisation and neighbour discovering method. In our previous work (Piyatumrong et al., 2009), we proposed to relax rendez-vous assumption of DA-GRS as well as proposed details of communication operations for neighbour discovering, tree merging process and token traversal process. Furthermore, a new traversal strategy, named DFSmove, has been proposed to increase efficiency of tree construction. Our contributions in the current work are developed based on DA-GRS and our improvement in the previous work.

8.1 G-NODE

G-NODE has been introduced the idea in our previous work (Piyatumrong et al., 2008a). It is a decentralised algorithm that employs greedy algorithm for constructing spanning trees in a dynamic network like DT-MANETs. This algorithm attempts to create robust spanning forest by relying on quality of each neighbour (the quality of communication node). It is an extension of the DA-GRS reference algorithm but the rendez-vous assumption is relaxed. Thus, in G-NODE, several tokens can meet simultaneously. G-NODE relies on ‘greedy’ heuristics so it selects the highest quality node to merge when applicable. The merging operation in G-NODE is described in Algorithm 1.

Algorithm 1 G-NODE algorithm: looking for other tokens around token τ_i

```
τ^best is token owned by the most quality neighbourhood
if τ^best ≠ Ø then
    Merge_With(τ_i, τ^best) // merge the two tokens
else
    Move_Token(τ_i) // continue to move the token randomly
end if
```

Indeed, the advantage of G-NODE occurs when several tokens meet. It allows each node owning a token to choose the most quality neighbour to merge. Table 1 summarises the differences between DA-GRS algorithm and G-NODE.

Table 1 Summary heuristic used by DA-GRS and G-NODE at each activity concerning spanning forest construction and maintenance using token management strategy

	DA-GRS algorithm	*G-NODE*
Initial state	-	-
Act on physical disruption (1)	-	-
Act on physical disruption (2)	-	-
Merging process	(1) Rendez-vous synchronisation	(1) Relaxed rendez-vous
	(2) Randomness	(2) Greedy heuristic
Token traversal	Randomness	DFSmove

8.2 G-PATH

G-PATH follows the main idea of G-NODE, however it concerns the quality of edges. In G-NODE, when several tokens meet simultaneously, node has a chance to select the highest quality node (according to the specified criteria). Likewise, G-PATH relax the synchronisation method of the previous algorithm of DA-GRS. G-PATH relies on greedy heuristic to select the highest quality edge to merge when applicable. The merging operation in G-PATH is described in Algorithm 2.

8.3 G-Node-Path

‘G-Node-Path’ algorithm has its base on the composition of G-NODE and G-PATH described earlier. It takes into account the compromising issue and give high quality spanning forest solutions according to the compromise setting. Moreover, ‘G-Node-Path’ works in fully decentralised and asynchronous manner, which is suit to the characteristics of DT-MANETs. Indeed, the algorithm makes decision locally and in discrete manner. The preference of trade-off value is denoted as $[\rho, 1 - \rho]$ while ρ is preference value attached to one of the objective and the other objective will logically have preference value of $1 - \rho$. This preference value is used to influent the selection of used algorithm, G-NODE or G-PATH, at one moment of each node. By discretely using ρ to induce the selection of algorithms, the ultimate goal is to control the global behaviour of managed spanning forest, such that high quality topology follows the preference characteristics.

Algorithm 2 G-PATH algorithm: looking for other tokens around token τ_i

```
τ^best is token owned by the neighbour who possesses the
highest quality edge
if τ^best ≠ Ø then
    Merge_With(τ_i, τ^best) // merge the two tokens
else
    Move_Token(τ_i) // continue to move the token randomly
end if
```

9 Validation of multi-objective tree-based backbone

In order to validate the multi-objective tree-based backbone created by our proposal, G-Node-Path, we need to find the optimal tree for comparison. Finding an optimum solution for a single-objective tree problem, both maximising and minimising problem, can be done using either Kruskal or Prim’s algorithm when the global information is allowed. To our problems, finding a tree that gives the maximum *nodeWeight*() is done by giving value on edge using an addition of adjacent nodes’ trust level. Then, Kruskal or Prim’s algorithm is used for obtaining the maximum tree and the maximum value of *nodeWeight*() function directly. Similarly, finding a tree that gives the maximum *pathWeight*() can be done by transforming the problem into maximum spanning tree problem directly.

However, finding the optimal value of the multi-objective tree has been proven to be an NP-Hard problem by Hamacher and Ruhe (1994). We cannot find the optimal value in polynomial time and thus, researchers employ meta-heuristic for finding near optimal solutions in competitive time. In our case, we analysed the number of nodes and edges of communication graph represented DT-MANETs used in this work and found that we cannot enumerating all possible spanning tree to check for the optimal tree(s). This is because the size of node is big

(80–320 nodes). In order to avoid combinatorial explosion in finding optimal solutions for multi-objective spanning tree problems, the meta-heuristic approach is more suitable and is selected to be used.

We select evolutionary algorithm (EA) approach as the optimisation method because it tends to give a faster solution by finding and maintaining multiple solutions in one single simulation run. However, using EA for optimising a spanning tree problem, the solution (tree) is needed to be encoded so that evolutionary search operators like crossover or mutation can be applied. There are two different approaches for doing this: indirect representations and direct representations. The former approach usually encode a tree (phenotype) as a list of strings (genotypes) and apply standard search operators to the genotypes. The phenotype is constructed by an appropriate genotype-phenotype mapping. There are many indirect representations for trees such as the Prüfer numbers (Cayley, 1989), character vector (Davis et al., 1993), Blob code (Picciotto, 1999; Paulden and Smith, 2007), NetKeys (Rothlauf, 2006), etc.

In contrast, direct representations encode a tree as a set of edges and apply search operators directly to the set of edges. Therefore, mapping is not necessary. Instead, tree-specific search operators must be developed as standard search operators can no longer be used. Since there is no additional genotype-phenotype mapping, here tree-specific search operators are directly applied to the phenotypes. In this work, we use edge-set proposed by Raidl and Julstrom (2003) as representation of spanning trees. Not only the direct representation, Julstrom (Raidl and Julstrom, 2003) also provides some idea of initialisation, crossover, and mutation operators for working with edge-set too.

'jMetal' (Durillo et al., 2006) is an object-oriented Java-based framework aimed at the development, experimentation, and study of meta-heuristics for solving MOOPs. jMetal provides a rich set of classes which support the implementation of a new meta-heuristic and a new application problem. In this work, we focus to our own problem which is about finding optimal spanning forest or a set of spanning trees, according to our own metrics as shown in the previous Section 4.2.

10 Experimentation setup

10.1 Simulators and mobility models

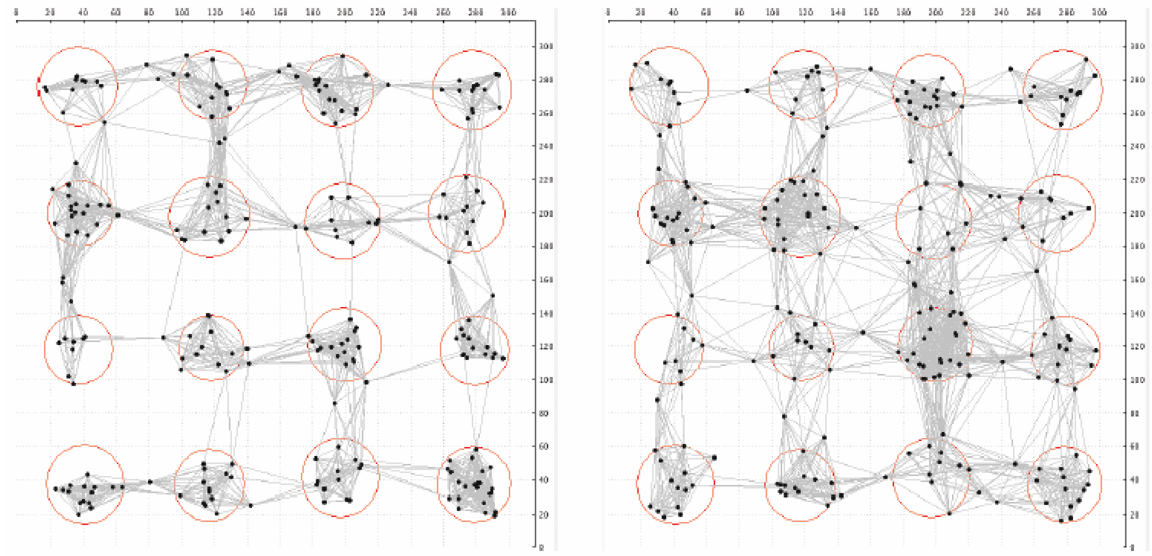
This study uses simulation as a mean for evaluation our proposed algorithms because it is practical for testing and changing the studying protocols in a controlled and reproducible manner. In order to evaluate performance of algorithms for ad hoc networks under simulation method, mobility models are used heavily. In this work, we are interested in human mobility model (HMM) provided by Hogie (2007). It is a generic mobility model that represents the intention-driven mobility of people in metropolitan areas and also provides different scenario modelling such as mall and highway mobility models. Hogie provides a network simulators named 'Madhoc' equipped with a set of HMM generator. Madhoc simulates MANET according to a set of pre-specified data such as time-interval between two iterations, types of communication technologies on mobile nodes, density of mobile nodes, simulation area, and mobility models.

The different mobility models provided by Madhoc allow realistic motion of citizens in variety of environment. All event of network simulation according to the set of pre-defined data can be captured in a trace file. These events are 'adding node', 'adding edge', 'removing node' and 'removing edge' at each time step t_i. According to a specified time interval given by the pre-defined data, list of event at each time step t_i is the result of event happened during the interval $t_i - 1$ and t_i.

This work uses GraphStream, which is a java library that manages dynamic graphs, as the main simulator. It gives a quick and easy way to add edges and nodes in a graph in order to evolve them. Utilising trace files from Madhoc in GraphStream is easy, since the syntax of trace file is compatible between two platforms. All of our simulations are done based on GraphStream simulator.

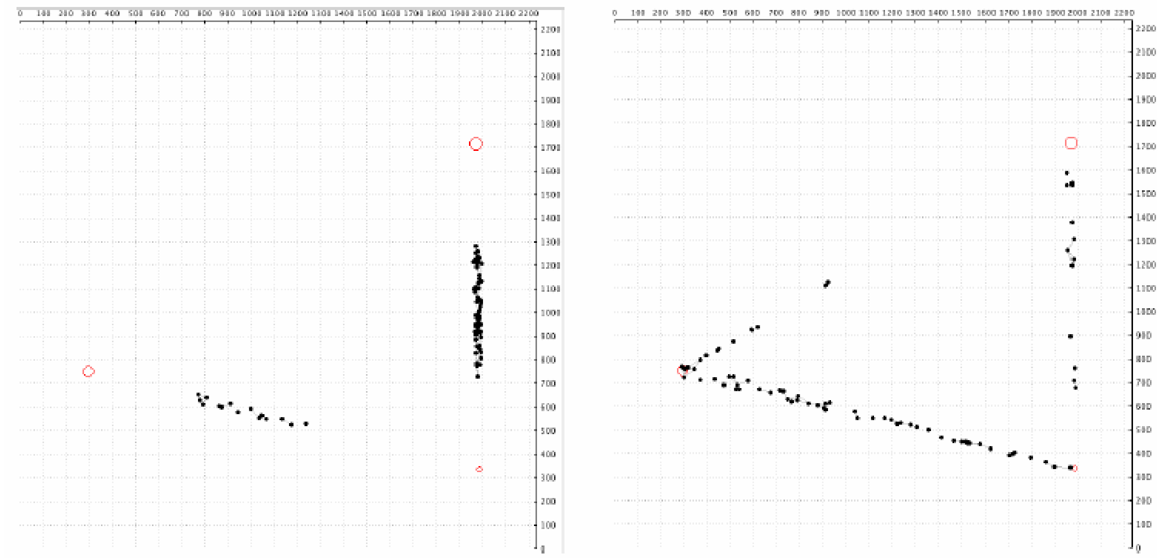
Two real-world mobility models, 'shopping mall' and 'highway', were selected for the simulations. They were chosen for this research as their characteristics are almost the opposite of each other, and thus, we could use them to ensure the validity and robustness of the verification. Shopping mall presents a mobility model that nodes are moving based on walking to running speed. Within a shopping mall area, there exists several spots representing different shops. Inside shops, the density of mobile nodes can be expected higher than in the corridor area. The direction of nodes in shopping mall model can have high fluctuation since there is no specific rule of walking. In contrary, the highway mobility model shows us the speedy movement of cars on highway but nodes move with guidance of road infrastructure. An important characteristic of highway model is the opposite movement of mobile nodes due to the two different directions of road infrastructure. Although, highway mobility model presents high speed of nodes, one common characteristic is also the group of node has same sped and move in the same direction. These nodes become neighbours to each other for a period of time before any node change their speed and direction. Some snapshots of the obtained mobility model network from Madhoc are shown in Figures 8 and 9 for shopping mall model and highway model respectively.

Figure 8 An example of shopping mall mobility model with 240 nodes, (a) a snapshot at G(1) (b) a snapshot at G(400) (see online version for colours)



(a) (b)

Figure 9 An example of highway mobility model with 80 nodes, (a) a snapshot at G(1) (b) a snapshot at G(400) (see online version for colours)



(a) (b)

We derived communication graphs from Madhoc which performs simulation in discrete-time. So the communication network corresponds to a series of static graphs: $G(t)$ for $t \in \{t_1, t_2, t_3, \ldots, t_{400}\}$. Between two consecutive times t_i and $t_i + 1$ the communication graph remains the same. A short timing-snapshot, 1/4 seconds between two consecutive times is considered sufficient to reflect the reality. As can be seen in Figures 8 and 9, differences in dynamicity of snapshots are shown in their sub-figures at different $G(t)$.

Four configurations were generated in this study. The parameters of these four different models are given in Tables 2 and 3 for shopping mall and highway mobility model networks respectively. Furthermore, the corresponding topology of each configurations can be found in Figures 10(a) and 10(b) for all shopping mall and highway configurations respectively.

The differences of these configurations are mainly the number of nodes and edges which are effected by the density, as shown in Tables 2 and 3. Furthermore, the number of connected components is also influenced by the density factor too. Figures 10(a) and 10(b) show the dynamic number of edges and connected components over the simulation time for both mobility models.

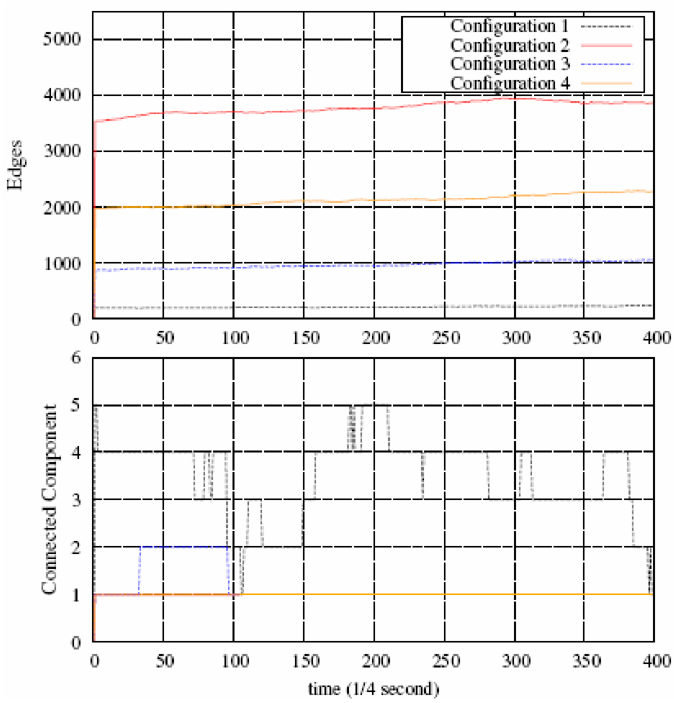
Table 2 Parameters used in the experiments for shopping mall mobility model

	Configuration			
	1	*2*	*3*	*4*
Surface (km^2)	0.32			
Node density (per km^2)	250	500	750	1,000
No. of nodes	80	160	240	320
Avg. no. of partitions	3.48	1.16	1.00	1.00
No. of connections	219.79	970.27	2,121.83	3,772.72
Average degrees	5.51	12.16	17.72	23.63
Velocity of nodes (m/s)	0.3–3			
Radio transmission range	40–80 m			
Network technology	IEEE802.11b			

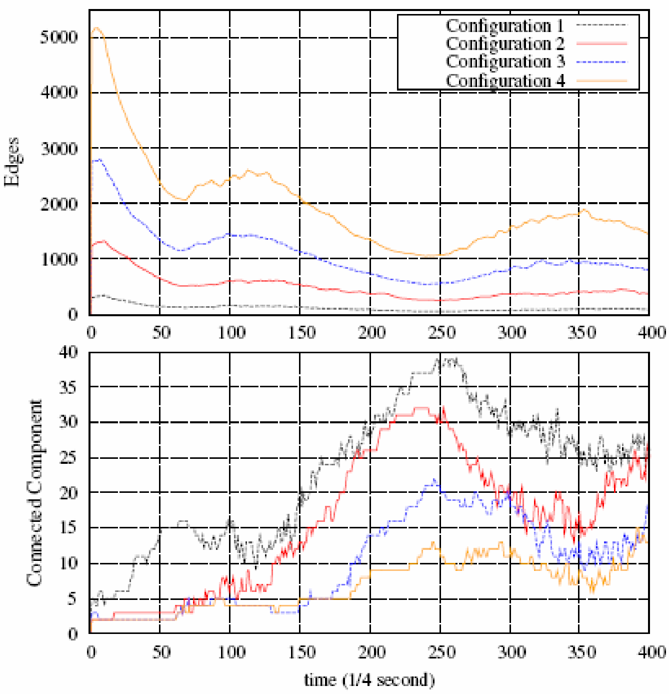
Table 3 Parameters used in the experiments for highway mobility model

	Configuration			
	1	*2*	*3*	*4*
Surface (km^2)	2.24			
Node density (per km^2)	35.72	71.43	107.14	142.86
No. of nodes	80	160	240	320
Avg. no. of partitions	22.78	15.90	9.89	6.91
No. of connections	117.59	498.00	1,078.00	1,983.04
Average degrees	2.96	6.23	9.04	12.47
Velocity of nodes (m/s)	20–40			
Radio transmission range	40–80 m			
Network technology	IEEE802.11b			

Figure 10 Number of edges and connected component of each configurations from both mobility model according to Tables 2 and 3, (a) shopping mall topology (b) highway topology (see online version for colours)



(a)



(b)

10.2 Communication assumptions

GraphStream is selected to replay the realistic trace files from Madhoc in a deterministic way. However, the shortcoming of GraphStream is the absence of the communication renderer. In order to overcome this shortcoming, we limit the amount of communication by using a boundary for each nodes and for the whole simulation graph per simulation step. This section describes the detail of those communication boundaries used in this study.

Gupta and Kumar (2000) summarises that when n identical randomly located nodes, each capable of transmitting at W bits per second and using a fixed range, form a wireless network, the throughput $\lambda(n)$ obtainable by each node for a randomly chosen destination is:

$$\Theta\left(\frac{W}{\sqrt{n \log n}}\right) \text{bits/sec} \tag{6}$$

This is under a non-interference assumption. Fundamentally, it is the sharing channel/resources between every node in all over the domain. Sharing resources with nodes in its local neighbourhood is the reason for the constriction in capacity. According to Gupta's finding, we render our simulation by limiting throughput of each node as well as of each communication graph. According to the wireless network capacity in the equation (6) above, W equals to the maximum raw data rate of used communication technology, which in this study is IEEE802.11b. Thus, W is 11 Mbit/sec. The number of n is different according to the level of limitation, at neighbourhood level and at communication graph level. The number of n at the communication graph level equals to the number of all nodes in V of $G = (V, E)$. The number of n at neighbourhood level of a particular node equals to the number of its neighbours. Hence, we can calculate the boundary of transmitting data of a node as:

$$\left(\frac{11 \text{ Mbits/s}}{\sqrt{n \log n}}\right) \text{bits/sec},$$

where n is the number of neighbour of this node.

This calculation gives us the upper bound throughput. In our simulation, we test using 50%, 75% and 100% of this maximum throughput. We found that the results given by these different setting have similar characteristics. That is the DA-GRS reference algorithm gives the lower bound both in terms of robustness and efficiency. However, the lower the throughput limitation, the lower the value of robustness metrics for both the DA-GRS reference algorithm and our proposal algorithms. We select to use the limitation at 75% of the maximum throughput because it gives a reasonable trade-off between the maximum throughput and the half-theoretical throughput. Table 4 shows the upper bound throughput per network with different network sizes of this study.

Table 4 Calculation of capacity of wireless network following Gupta's finding

Network size	*Max. throughput per network (bits/sec)*	*Max. no. of packet (184 bits/packet) (packet/sec)*	*Throughput limitation at 75% (packet/sec)*
80	934,798	5,080	3,810
160	614,207	3,338	2,503
240	482,591	2,622	1,966
320	407,380	2,214	1,660

Not only the throughput that has been limited, based on Beaconing Rate of IEEE802.11 IEEE Standard 802.11: Wireless LAN medium Access Control and Physical Layer Specifications (1999), the time interval used for periodically sending the beacon is 100 millisecond. This means the beacon will be delivered ten times per second. For our used networks, timing-snapshot is 1/4 seconds between two consecutive times and, thus, 2.5 beacons should be sent during this 1/4 seconds for one node. The limitation of beaconing also limits the number of packets that can be sent using our algorithms and heuristics.

Algorithm 3 Driving mechanism from a graph $G(t)$ to a consecutive graph $G(t + 1)$ in GraphStream

t is time step in a dynamic graph $G(t)$, $t \in (t_1, \dots, t_i\}$
n_j is a node in $G(t)$, $j \in (1, \dots, j\}$
α is the maximum beaconing packet per a snapshot network $G(t_i)$
B_{n_j} is the allowed number of packet of node nj
while $G(t) \leq G(t_i)$ **do**
  reset α and B_{n_j} for this snapshot graph
  for all n_j **do**
    if $\alpha > 0$ and $B_{n_j} > 0$ **then**
      do DA-GRS or G-NODE or G-PATH or G-Node-Path
      if n_j sends beacon **then**
        $\alpha \leftarrow \alpha - 1$
        $B_{n_j} \leftarrow \beta - 1$
      else
        if n_j responses to some requests and $B_{n_j} \geq 0$ **then**
          $B_{n_j} \leftarrow B_{n_j} - k$, k is number of packet sent
        end if
      end if
    end if
  end for
end while

11 Methodology

In this study, the length of simulation is 100 seconds or 400 time steps (t). Each time step graph $G(t)$ is the snapshot of the dynamic graph G at every 0.25 seconds. Each snapshot is a graph G at moment t. At any $G(t)$, it may exist one or many connected component m. Figure 11 depicts a snapshot of graph at a time t emphasising different connected components in red circles.

We separate each connected component of a graph $G(t)$ and work one by one using meta-heuristic, NSGAII in jMetal. The product of this process is the Pareto front of each connected component. In order to have the Pareto front of the graph $G(t)$, all Pareto front from m connected component of $G(t)$ are processed by addition of all Cartesian products of m connected components.

Figure 11 Examples of communication graph at different time step t with emphasising of connected components in red circles (see online version for colours)

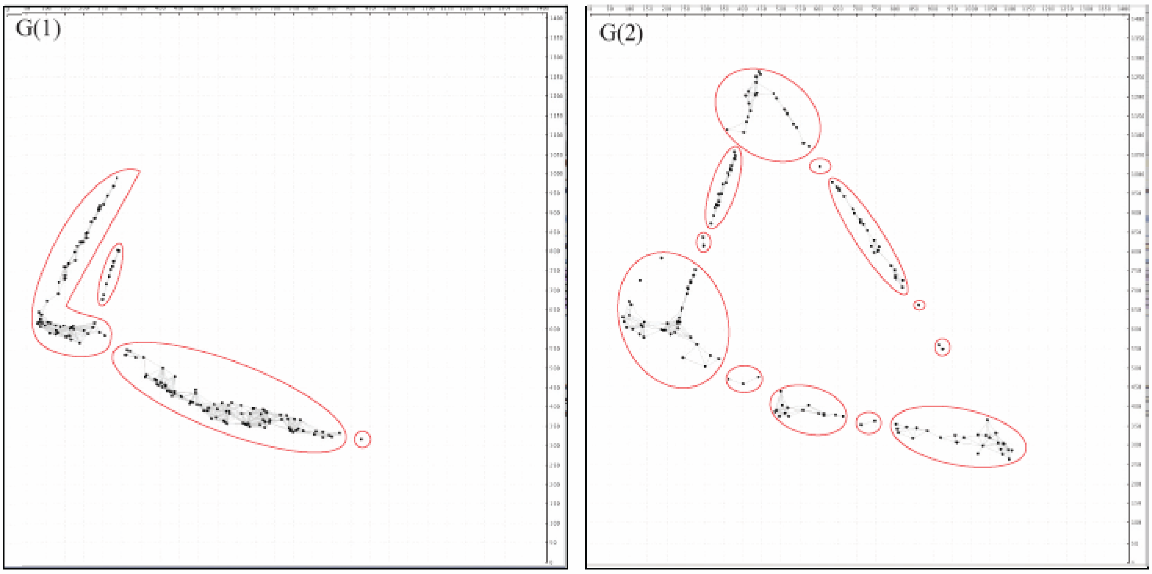
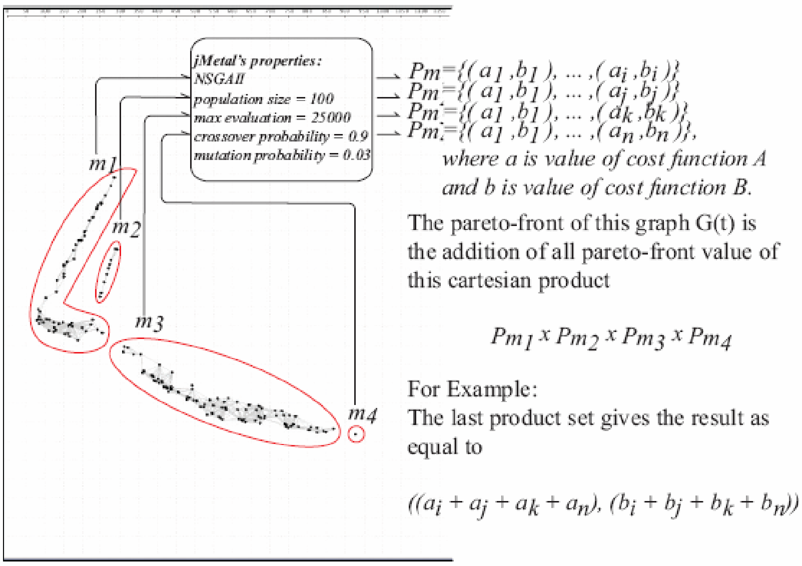


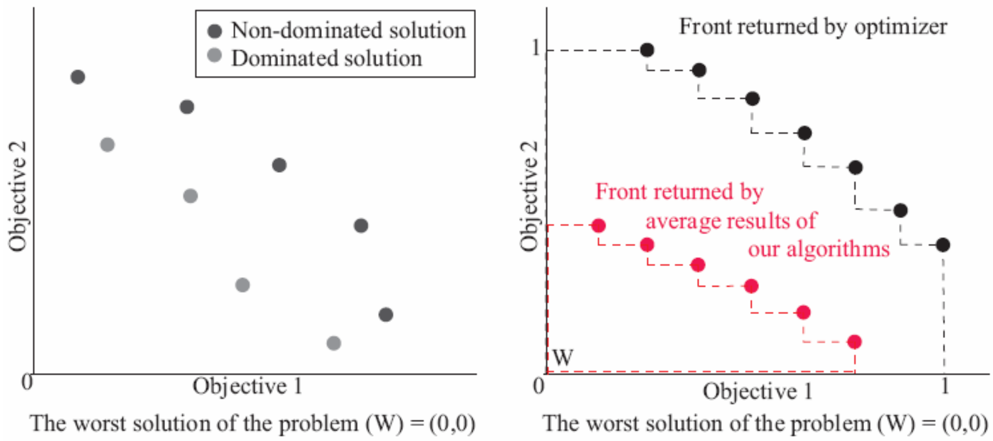
Figure 12 Finding Pareto front of a graph $G(t)$ (see online version for colours)



Let M is a set of all connected component of graph $G(t)$: $M = \{m_1, m_2, \ldots, m_m\}$. Figure 11 show two different graphs, $G(1)$ and $G(2)$, where $G(1)$ has four connected components (M_4) and $G(2)$ has 12 connected components (M_{12}). jMetal finds the Pareto front of a connected component i, namely set P_i. In order to find the Pareto front of $G(1)$, the Cartesian product of $P_{m_1} \times P_{m_2} \times P_{m_3} \times P_{m_4}$ is calculated. The value in each product set is the summation within each set. The result is used to select the Pareto front of all Cartesian product again. These processes are summarised in Figure 12. Furthermore, parameters used for EA (NSGA-II) in this work are also presented in the same figure.

In multi-objective approach, the solution space to be explored becomes n-dimensional, where n is the number of objectives to optimise. Consequently, the fitness of an individual changes from one value to set of values, each corresponding to one objective in the solution space. In such a situation, a comparison of fitness or cost function value is not sufficient as it is in single objective approach. For a multi-objective problem, a solution is rather non-dominated front or Pareto front. Therefore, the assessment of quality must be done for a whole group of solutions at the same time. Hypervolume (HV) is used here to assess the quality of a whole group of solutions at the same time. The motivation for using HV is for comparing between solutions given by jMetal (Pareto front given by meta-heuristic, NSGA-II, in this case), DA-GRS reference algorithm and our proposed heuristic. HV has been discussed in details in Nebro et al. (2008). We present here briefly the methodology to obtain HV using maximisation problem as an example.

Figure 13 Multi-objective search space, (a) Pareto front and dominated solutions (b) HV of a Pareto front (see online version for colours)



(a) (b)

Figure 13(a) shows the relation between a set of solutions in grey colour that is dominated by another set of solutions, in dark grey colour. In another figure, the Pareto front given by optimiser is in black colour. All solutions in this particular figure has been normalised by the value of Pareto front so that the maximum value of each objective reach one. The HV of Pareto front can be obtained by calculating the area under the black dotted line. Please note that this example is for maximisation problem which has the worst solution point at (0; 0). In the next section, the comparison has been made between the HV of Pareto front given by NSGA-II and the average result of our proposed algorithms at different preference of trade off value, ρ. In Figure 13(b), the set of front in red colour represent a set of results to compare with Pareto front. The robustness metrics are also presented to measure how well G-Node-Path handle the construction of spanning forest at different preferences, ρ.

For ease of reference and comparison of result, the preference of trade off value, denoted as [ρ, $1 - \rho$], is set for the experimentation from [0, 100], [10, 90], …, [90, 10], [100, 0]. So, there are eleven different experimentation for each configuration network. Each experimentation in each configuration network uses the preference trade off value for heuristics [G-NODE, G-PATH] accordingly.

12 Results

12.1 HV results

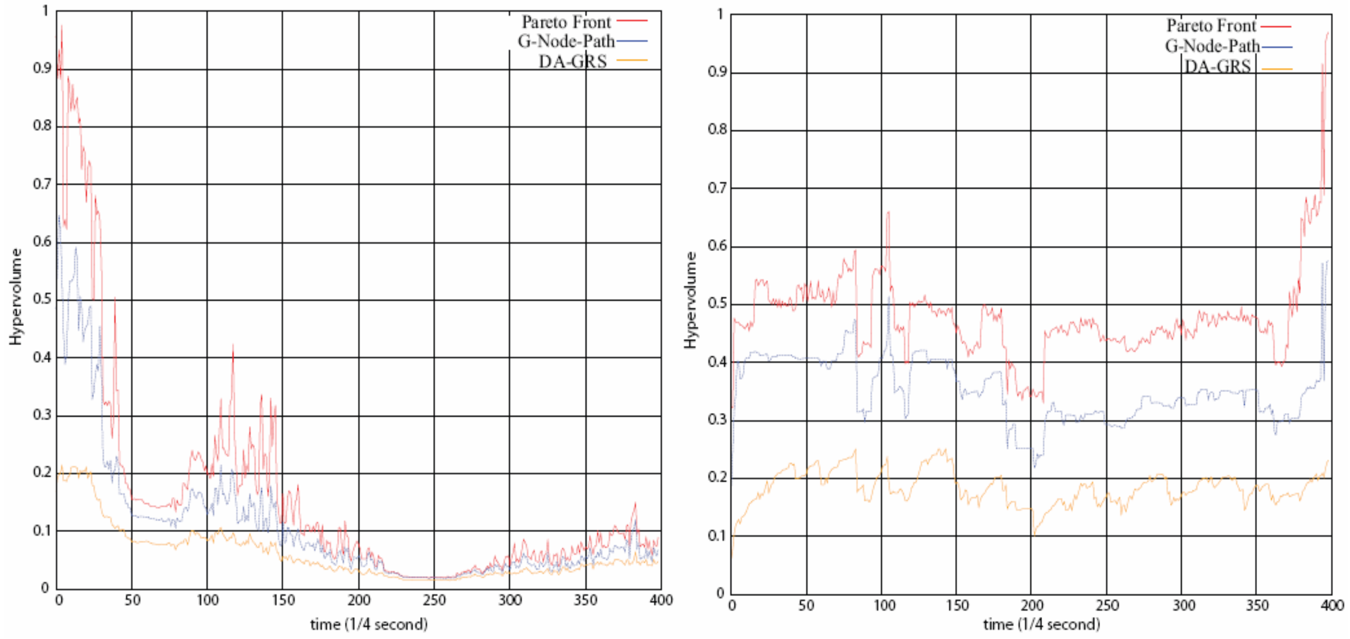
The comparison between the HV of Pareto front and other heuristic results give us the quality of local decision making heuristic against the best solution when global information is provided. In this section, different graphs show the comparison of HV between Pareto front given by NSGA-II, DA-GRS and our G-Node-Path heuristic. The experimentation has been done using 100 runs in each configuration of network.

The result graphs 15 reveals and advantage of G-Node-Path. It is clearly see that G-Node-Path algorithm achieves more than a half of distance between Pareto front and DA-GRS on both mobility models. Both sub-figures also depict the strong differences of mobility characteristics over simulation time. Focusing at Figure 14(a), between time step 300–350 where the topology is highly fluctuant, G-Node-Path still provide a better solution. In summary, these results confirm the benefit of G-Node-Path for handling multi-objective topology of spanning forest in DT-MANETs. Next subsection we shall discuss the results based on different metrics that form problem description at different preference value, ρ.

12.2 Metric results

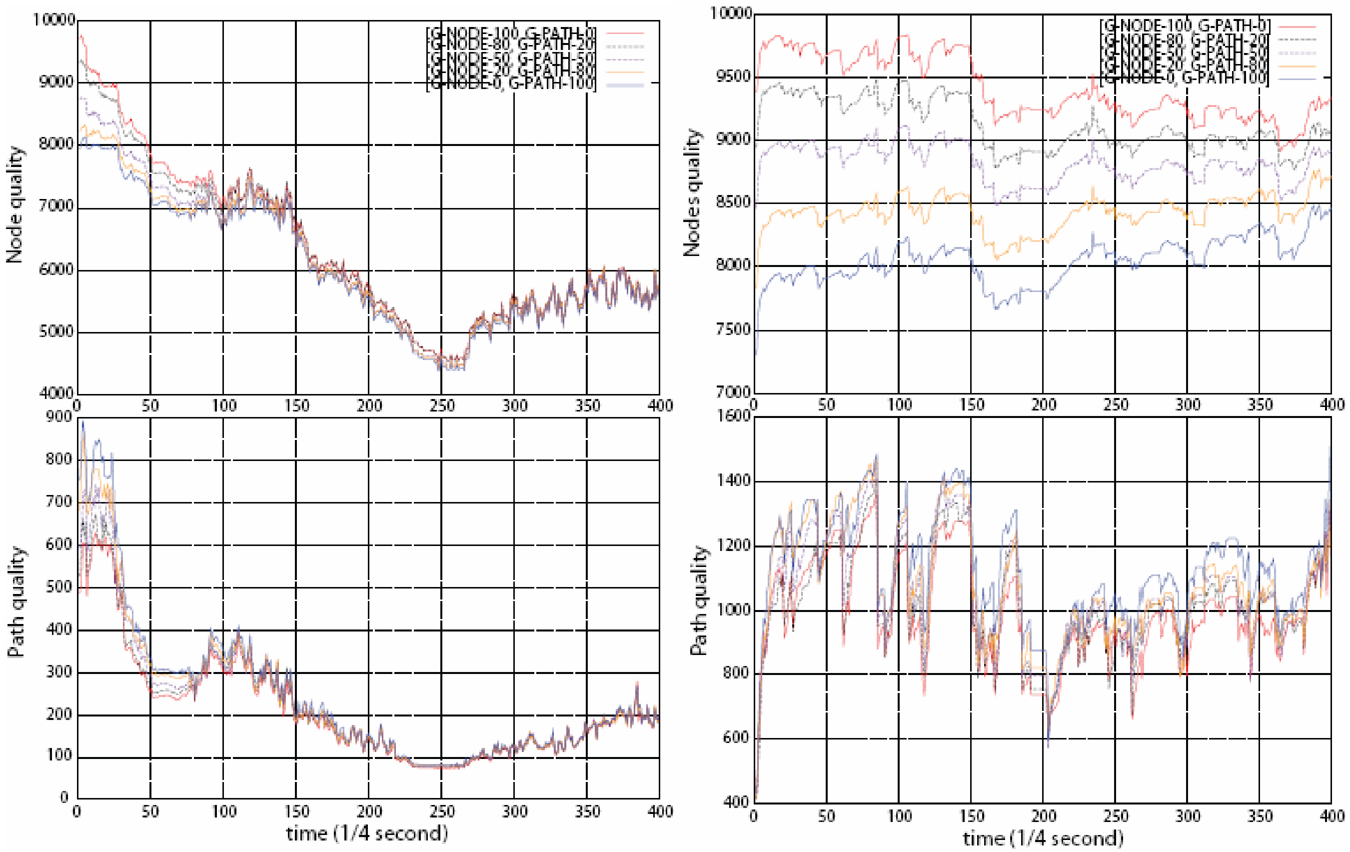
The comparison between different preference value is used to confirm the usability of G-Node-Path with preference value. At Figure 15, the overall output topology is influenced by the preference value, and thus, each metric value presents this fact accordingly. Figure 15 has two sub-figures for presenting results from Highway and Shopping mall mobility model. In each sub-figure, there are two different graphs which illustrate node and path quality value over simulation time and are given as a result of different preference value (ρ) of G-Node-Path heuristic. Each information line can be distinguish by colour and corresponding key such as [GT-100, GP-0] which means this vector of information are generated by G-Node-Path at 100% of G-NODE and 0% of G-PATH. The path quality used here is the CB and the nodes' quality is the battery level. The experimentation has been done using 25 runs for each configuration of network.

Figure 14 Comparison between HV of NSGA-II, DA-GRS and G-Node-Path over simulation time, (a) highway mobility model (b) shopping mall mobility model (see online version for colours)



(a) (b)

Figure 15 Comparison between path quality and node quality of five different preference value (ρ) over simulation time, (a) highway mobility model (b) shopping mall mobility model (see online version for colours)



(a) (b)

In Figure 15(b), it can be seen clearly that each set of experimentation are distinguishable quite well, especially for nodes' quality. Although, the differences between each preference set is quite slim for path quality and also for highway mobility model, the pattern of differences can be seen.

In summary, we found that results confirm the benefit of greedy heuristic used by G-NODE and G-PATH, both of them are fully decentralised algorithm and use only local information. By using only one-hop information, G-NODE and G-PATH achieve a better results than DA-GRS algorithm but not yet reach the optimality. The multi-objective problem presented here is the combination of two objectives problem concerning the quality of node and path. This study implemented the G-Node-Path as a distributed and localised algorithm for constructing spanning forest topology based on bi-criteria. Results show that the set of solutions provided by G-Node-Path can achieve different level of desired solutions. Furthermore, the HV of the solutions set provided by G-Node-Path achieve a better solution than DA-GRS reference algorithm.

13 Conclusions and future works

This work address tree-based topology management problem in DT-MANETs which presents a dynamic and disrupted topology of MANETs. Thus, we focus on disconnected and arbitrarily graphs. At anytime, obtaining global information is impossible and impractical. Managing topology in such environment requires a new generation algorithm that is decentralised algorithm, using local information and working in asynchronous manner.

Topology management problem is more suitable to tackle considering multi-criteria at the same time. This work propose to manage a high quality tree-based topology based on different criteria of communication node and edge. Examples of such quality used in this work are trust level of nodes and CB of edges. Problem definitions and quality metrics are proposed accordingly. The multi-objectives tree-based topology of this work wants to maximise *nodeWeight*() function and maximise *pathWeight*() function at the same time. Thus, the solution space becomes 2-dimensional and the assessment of quality should be done for a whole group of solutions at the same time. HV is used here to assess the quality of a whole group of solutions. The validation is done by comparing the solutions given by metaheuristic as the upper bound and the solutions given by DA-GRS reference algorithm as the lower bound.

In our previous work, G-NODE has been proposed to solve a single-objective tree topology based on trust level quality (G-NODE is an extension algorithm from DA-GRS reference algorithm). This present work applies the idea for quality of tree-based topology based on edge and accordingly path quality. G-PATH is a new generation of algorithm for obtaining a single-objective of high path quality tree topology. Our aim is to provide an algorithm that can generate solutions to multi-objective tree-based topology. Accordingly, G-Node-Path is proposed in this work for such purpose. By utilising preference value preinstalled in each node, all the node can mutually construct tree topology according to the preference value in asynchronous and decentralised way. Also G-Node-Path only uses one-hop information. Thus, the overhead in communication and memory used are very low.

For future work, the adaptation of preference value to the situation such as the mobility model and the actual environment is very interesting. Node should have a way to assess the situation and give feed back to the topology managing unit so that the decision is made in real time on what quality is more important. Thus, the definition of high quality tree-based topology is suitable to the current situation. Accordingly, the study of multiple mobility models is needed. In this current work, we select two extremely different models which are Highway and Shopping Mall models. For future work, we are also interested in city mobility model that combines both highway and shopping mall together. The results given by G-Node-Path in this work still not reach the optimal solutions, thus the improvement is still needed. From the analysis, we found that G-Node-Path still lack of adaptation after tree-construction. One of the main future work should be carried out to include the tree-adaptation idea into G-Node-Path.

References

Abdulla, M. and Simon, R. (2006) 'A simulation analysis of multicasting in delay tolerant networks', in *WSC '06: Proceedings of the 38th Conference on Winter Simulation*, pp.2234–2241.

Alzoubi, K.M., Wan, P-J. and Frieder, O. (2002) 'Distributed heuristics for connected dominating sets in wireless ad hoc networks', *Journal of Communications and Networks*, Vol. 4, No. 1, pp.22–29.

Borůvka, O. (1926) 'O jistém problému minimálním', *Práce Mor. Přírodověd. v Brně (Acta Societ. Scient. Natur. Moravicae)*, Vol. 3, pp.37–58.

Casteigts, A. (2006) 'Model driven capabilities of the DA-GRS model', in *ICAS '06: Proceedings of the International Conference on Autonomic and Autonomous Systems*, p.24.

Cayley, A. (1989) 'A theorem on trees', *Quarterly Journal of Mathematics*, Vol. 23, pp.376–378.

Chen, G., Chen, S., Guo, W. and Chen, H. (2007) 'The multi-criteria minimum spanning tree problem based genetic algorithm', *Inf. Sci.*, Vol. 77, No. 22, pp.5050–5063.

Corson, S. and Macker, J. (1999) 'Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations', IETF.org, available at http://www.ietf.org/rfc/rfc2501.txt (accessed on 24 July 2011).

Davis, L., Orvosh, D., Cox, A. and Qiu, Y. (1993) 'A genetic algorithm for survivable network design', in *Proceedings of the 5th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp.408–415.

Delay Tolerant Networking Research Group (n.d.) 'Delay tolerant networking research group', available at http://www.dtnrg.org/wiki (accessed on 7 June 2010).

Durillo, J.J., Nebro, A.J., Luna, F., Dorronsoro, B. and Alba, E. (2006) 'jMetal: a java framework for developing multi-objective optimization metaheuristics', Technical Report ITI-2006-10, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos.

Faloutsos, M. and Molle, M. (1995) 'Optimal distributed algorithm for minimum spanning trees revisited', in *PODC '95: Proceedings of the Fourteenth Annual ACM Symposium on Principles of Distributed Computing*, ACM, New York, NY, USA, pp.231–237.

Gallager, R.G., Humblet, P.A. and Spira, P.M. (1983) 'A distributed algorithm for minimum-weight spanning trees', *ACM Trans. Program. Lang. Syst.*, Vol. 5, No. 1, pp.66–77.

Gupta, P. and Kumar, P.R. (2000) 'The capacity of wireless networks', *IEEE Transactions on Information Theory*, Vol. 46, No. 2, pp.388–404, available at http://dx.doi.org/10.1109/18.825799.

Hamacher, H. and Ruhe, G. (1994) 'On spanning tree problems with multiple objectives', *Annals of Operations Research*, Vol. 52, pp.209–230.

Hogie, L. (2007) 'Mobile ad hoc networks: modelling, simulation and broadcast-based applications', PhD thesis, University of Le Havre (France) University of Luxembourg (Luxembourg).

Huynh, T.D., Jennings, N.R. and Shadbolt, N.R. (2006) 'An integrated trust and reputation model for open multi-agent systems', *Autonomous Agents and Multi-Agent Systems*, Vol. 13, No. 2, pp.119–154.

IEEE Standard 802.11: Wireless LAN medium Access Control and Physical Layer Specifications (1999) IEEE Computer Society.

Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L-S. and Rubenstein, D. (2002) 'Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with Zebranet', in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, Vol. 37, ACM, pp.96–107.

Jüttner, A. and Magi, Á. (2005) 'Tree based broadcast in ad hoc networks', *Mobile Networking Applications*, Vol. 10, No. 5, pp.753–762.

Kruskal, J.B. (1956) 'On the shortest spanning subtree of a graph and the traveling salesman problem', in *Proceedings of the American Mathematical Society*, Vol. 7, pp.48–50.

Lotker, Z., Pavlov, E., Patt-Shamir, B. and Peleg, D. (2003) 'Mst construction in o(log log n) communication rounds', in *SPAA '03: Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, ACM, New York, NY, USA, pp.94–100.

Marias, G.F., Georgiadis, P., Flitzanis, D. and Mandalas, K. (2006) 'Cooperation enforcement schemes for manets: a survey', *Wirel. Commun. Mob. Comput.*, Vol. 6, No. 3, pp.319–332.

Marti, S., Giuli, T.J., Lai, K. and Baker, M. (2000) 'Mitigating routing misbehavior in mobile ad hoc networks', in *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ACM, New York, NY, USA, pp.255–265.

Merwe, J.V.D., Dawoud, D. and McDonald, S. (2007) 'A survey on peer-to-peer key management for mobile ad hoc networks', *ACM Comput. Surv.*, Vol. 39, No. 1, p.1.

Murthy, C.S.R. and Manoj, B. (2004) *Ad Hoc Wireless Networks: Architectures and Protocols*, Pearson Education, Inc., Prentice Hall PTR, eBook ISBN: 9780131470231, Print ISBN: 013147023X.

Nebro, A.J., Luna, F., Alba, E., Dorronsorro, B., Druillo, J.J. and Beham, A. (2008) 'Abyss: adapting scatter search to multiobjective optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 4, pp.439–457.

Pathirana, P.N., Bulusu, N., Savkin, A.V. and Jha, S. (2005) 'Node localization using mobile robots in delay-tolerant sensor networks', *IEEE Transactions on Mobile Computing*, Vol. 4, No. 3, pp.285–296.

Paulden, T. and Smith, D.K. (2007) 'Some novel locality results for the blob code spanning tree representation', in *GECCO '07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ACM, New York, NY, USA, pp.1320–1327.

Pelusi, L., Passarella, A. and Conti, M. (2006) 'Opportunistic networking: data forwarding in disconnected mobile ad hoc networks', *IEEE Communications Magazine*, Vol. 44, No. 11, pp.134–141.

Picciotto, S. (1999) 'How to encode a tree', PhD thesis, University of California.

Pigné, Y. (2008) 'Modélisation et traitement décentralisé des graphes dynamiques – application aux réseaux mobiles ad hoc', Phd thesis, L'Université du Harve.

Piyatumrong, A. (2010) 'Robust topology management in DT-MANETs: an efficient tree-based decentralized and multi-objective approach', PhD thesis, University of Luxembourg, Luxembourg.

Piyatumrong, A., Bouvry, P., Guinand, F. and Lavangnananda, K. (2008a) 'Trusted spanning tree for delay tolerant MANETs', in C-Z. Xu and M. Guo (Eds.): *IEEE/IPIP International Conference on Embedded and Ubiquitous Computing (EUC 2008)*, Vol. 2, pp.293–299, IEEE Computer Society.

Piyatumrong, A., Bouvry, P., Guinand, F. and Lavangnananda, K. (2008b) 'Trusted spanning tree for delay tolerant mobile ad hoc networks', in *SMCia '08: IEEE Conference on Soft Computing in Industrial Applications*, pp.131–136.

Piyatumrong, A., Ruiz, P., Bouvry, P., Guinand, F. and Lavangnananda, K. (2009) 'Token traversal strategies of a distributed spanning forest algorithm in mobile ad hoc – delay tolerant networks', in *Advances in Information Technology*, Vol. 55 of Communications in Computer and Information Science, Springer-Verlag Berlin Heidelberg, pp.96–109.

Prim, R.C. (1957) 'Shortest connection networks and some generalizations', *Bell System Technology Journal*, Vol. 36, pp.1389–1401.

Radhakrishnan, S., Racherla, G., Sekharan, C.N., Rao, N.S.V. and Batsell, S.G. (2003) 'Protocol for dynamic adhoc networks using distributed spanning trees', *Wireless Networks*, Vol. 9, No. 6, pp.673–686.

Raidl, G. and Julstrom, B. (2003) 'Edge sets: an effective evolutionary coding of spanning trees', *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 3, pp.225–239.

Rothlauf, F. (2006) *Representations for Genetic and Evolutionary Algorithms*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Santi, P. (2005) T*opology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley and Sons, Ltd, West Sussex, England.

Schleich, J., Bouvry, P. and An, L.T.H. (2009) 'Decentralized fault-tolerant connected dominating set algorithm for mobile ad hoc networks', in H.R. Arabnia and V.A. Clincy (Eds.): *ICWN*, pp.354–360, CSREA Press.

Seredyński, M. (2009) 'An evolutionary approach towards cooperation enforcement in ad hoc networks', PhD thesis, University of Luxembourg (Luxembourg).

Sopena, E. and Metivier, Y. (1997) 'Graph relabeling systems: a general overview', *Computers and Artificial Intelligence*, Vol. 16, No. 2, pp.167–185.

Srivastava, S. and Ghosh, R.K. (2002) 'Cluster based routing using a k-tree core backbone for mobile ad hoc networks', in *DIALM '02: Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ACM, New York, NY, USA, pp.14–23.

Wrona, K. (2002) 'Distributed security: ad hoc networks and beyond', presented at the *Ad Hoc Network Security Pampas Workshop*, RHUL, London, 16–17 September.

Zhao, W. and Ammer, M.H. (2003) 'Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks', in *Proceedings of the Ninth IEEE Workshop on Future Trends of Distributed Computing Systems*', IEEE Computer Society, p.308.

Zhao, W., Ammar, M. and Zegura, E. (2004) 'A message ferrying approach for data delivery in sparse mobile ad hoc networks', in *MobiHoc '04: Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ACM, pp.187–198.