A Genetic Algorithm for Constructing a Reliable MCDS in Probabilistic Wireless Networks[*]

Jing (Selena) He, Zhipeng Cai, Shouling Ji, Raheem Beyah, and Yi Pan

Department of Computer Science,
Georgia State University, Atlanta, GA, USA
{jhe9,zcai,sji,rbeyah,pan}@cs.gsu.edu

Abstract. Minimum Connected Dominating Sets (MCDSs) are used as virtual backbones for efficient routing and broadcasting in wireless networks extensively. However, the MCDS problem is NP-Complete even in Unit Disk Graphs. Therefore, many heuristic-based approximation algorithms have been proposed recently. In these approaches, networks are deterministic where two nodes are assumed either connected or disconnected. In most real applications, however, there are many intermittently connected wireless links called *lossy links*, which only provide probabilistic connectivity. For wireless networks with lossy links, we propose a *Probabilistic Network Model* (PNM). Under this model, we measure the quality of Connected Dominating Sets (CDSs) using *CDS reliability* defined as the minimum upper limit of the *node-to-node delivery ratio* between any pair of dominators in a CDS. We attempt to construct a MCDS while its reliability is above a preset application-specified threshold, called *Reliable MCDS* (RMCDS). We claim that constructing a RMCDS is NP-Hard under the PNM model. We propose a novel Genetic Algorithm (GA) called RMCDS-GA to solve the RMCDS problem. To evaluate the performance of RMCDS-GA, we conduct comprehensive simulations. The simulation results show that compared with the traditional MCDS algorithms, RMCDS-GA can construct a more reliable CDS without increasing the size of a CDS.

1 Introduction

Different from wired networks, the topology of a wireless network may change from time to time, and the energy of nodes is very limited and irreplaceable. Therefore, designing an energy-efficient communication scheme for wireless networks is one of the most important issues that has a significant impact on the network performance. The effectiveness of many communication primitives for wireless networks, such as routing [1], multicast/broadcast [2], and service discovery [3], relies heavily on the availability of a *Virtual Backbone* (VB). A *Connected Dominating Set* (CDS) typically serves as a VB of a wireless network. A CDS is defined as a subset of nodes in a wireless network such that each node

[*] This work is partly supported by the NSF under grant No. CCF-0545667, CNS-0831634, and by the 111 project of China under the grant No. 111-2-14.

Y. Cheng et al. (Eds.): WASA 2011, LNCS 6843, pp. 96–107, 2011.

in the network is either in the set or adjacent to some node in the set, and the induced graph by the nodes in the set is connected. The nodes in a CDS are called *dominators*, otherwise, *dominatees*. In a wireless network with a CDS as its VB, dominatees only forward their data to their connected dominators. In addition to communication schemes, a CDS has many other applications, such as topology control [4], coverage [5], data collection [6], and data aggregation [7]. Clearly, the benefits of a CDS can be magnified by making its size (the number of the nodes in the CDS) smaller. In general, the smaller the CDS is, the less communication and storage overhead are incurred. Hence, it is desirable to build a Minimum-sized CDS (MCDS).

Ever since the idea of employing a CDS for wireless networks was introduced [8], a huge amount of effort has been made to find different CDSs for different applications, especially MCDSs. In the seminal work [9], Guha and Kuller first modeled the problem of constructing the smallest CDS as the MCDS problem in a general graph, which is a well-know NP-Hard problem [10]. Subsequently, many polynomial-time approximation algorithms for MCDS construction have been proposed in recent literature. The substraction-based CDS algorithms begin with a set of all the nodes in a network, then some nodes are removed by pre-defined rules to obtain a CDS. The work in [11] is such an algorithm. The addition-based CDS algorithms start from a subset of nodes (usually disconnected), then include additional nodes to form a CDS. One typical such algorithm is [12].

In all the above mentioned methods, wireless networks are modeled using the *Deterministic Network Model* (DNM). Under this ideal model, any pair of nodes in a network is either fully connected or completely disconnected. In most real applications, however, the DNM model cannot fully characterize the behavior of wireless links. This is mainly due to the *transitional region phenomenon* which has been revealed by many empirical studies [13]. Beyond the "always connected" region, there is a *transitional region* where a pair of nodes are probabilistically connected. Such pairs of nodes are not fully connected but reachable via the so called *lossy links* [13]. As reported in [13,14], there are often much more lossy links than fully connected links. Therefore, their impact can hardly be neglected.

In order to well characterize a wireless network with lossy links, we propose a new network model called the *Probabilistic Network Model* (PNM). Under this model, in addition to transmission radius, there is a *Transmission Success Ratio* (TSR) associated with each link connecting a pair of nodes, which is used to indicate the probability that one node can successfully directly deliver a package to another. Obviously, the core issue under the PNM model is how to guarantee the node-to-node delivery ratio of all possible node pairs satisfying the user requirement, in other words, how to guarantee the Transmission Quality (TQ). For constructing a MCDS under the PNM model, we propose *CDS reliability* to measure its TQ. Given a PNM model, *CDS reliability* is defined as the minimum node-to-node delivery ratio between any pair of dominators. Thus, how to find a reliable MCDS under the PNM model is the major concern of this paper. The objective is to seek a MCDS whose reliability satisfies a certain application-dependent threshold denoted by σ (*e.g.*, $\sigma = 80\%$). If $\sigma = 100\%$, finding a reliable MCDS under the PNM model is the same as the traditional MCDS problem under the DNM model. However, a traditional MCDS algorithm may not find a reliable MCDS under the PNM mode. A counter-example is depicted in Fig. 1. By the latest algorithm proposed in [15], a spanning tree rooted at a specified initiator is first constructed, and then Maximal Independent Sets (MISs) are identified layer by layer. Finally a set of connectors to connect the MISs is ascertained to form a CDS. According to the topology shown in Fig. 1, the constructed CDS by [15] using s_4 as the initiator is $D = \{s_4, s_7, s_8\}$, whose reliability is 0.1. If the threshold $\sigma = 0.7$, the CDS D does not satisfy the constraint at all. The objective of our work is to find a MCDS whose reliability is greater than or equal to σ. One example of the satisfied reliable MCDS is $D' = \{s_3, s_6, s_7\}$ in Fig. 1.

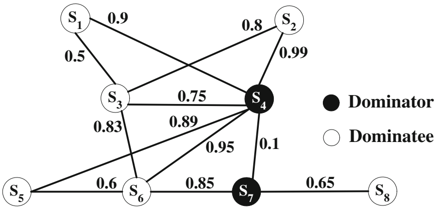


Fig. 1. A wireles network under the PNM model

To the best of our knowledge, this work is the first one attempting to construct a MCDS under the PNM model for wireless networks. Particularly, the main contributions of this paper are summarized as follows:

1. We identify and highlight the use of lossy links when constructing a CDS for wireless networks.
2. In order to measure the quality of a CDS under the PNM model, we define a new metric *CDS reliability*.
3. We propose a GA to build a reliable MCDS under the PNM model.
4. We also conduct extensive simulations to validate our proposed algorithms. The simulation results show that, compared with the traditional MCDS algorithms, our algorithm can obtain a more reliable CDS without increasing the size of a CDS.

2 Problem Statement

2.1 Network Model

We assume a static wireless network and all the nodes have the same transmission range. The Transmission Success Ratio (TSR) associated with each link connecting a pair of nodes is available and fixed. Under the *Probabilistic Network Model* (PNM), we model a wireless network as an undirected graph $G(V, E, P(E))$, where V is the set of n nodes, denoted by $s_1, s_2, \ldots, s_n$; E is the set of m lossy links, $\forall\ u, v \in V$, there exists an edge (u, v) in G if and only if: 1) u and v are in each other's transmission range, 2) $TSR(e = \{u, v\}) > 0$, for each link $e = \{u, v\} \in E$, where $TSR(e)$ indicates the probability that node u can successfully directly deliver a packet to node v; and $P(E) = \{< e, TSR(e) > | e \in E, 0 \leq TSR(e) \leq 1\}$. We assume edges are undirected (bidirectional), which means two linked nodes are able to transmit and receive information from each other with the same TSR value.

Because of the introduction of $TSR(e)$, the traditional definition of the node neighborhood has changed. Hence, we first give the definition of the 1-hop neighborhood and then extend it to the r-Hop neighborhood.

Definition 2.1 1-Hop Neighborhood. *$\forall\ u \in V$, the* 1*-Hop Neighborhood of node u is defined as:*

$$N_1(u) = \{v | v \in V, TSR(e = \{u, v\}) > 0\}$$

The physical meaning of 1-Hop Neighborhood is the set of the nodes that can be directly reached from node u.

Definition 2.2 Node-to-Node Delivery Ratio. *Given a source node u and a destination node v, one path between the node pair can be denoted by the edge permutation $\theta(u, v) = (e_1, e_2, \ldots, e_m)$, and the delivery ratio of the path is denoted by $DR_\theta = \prod_{i=1}^{m} e_i$. Furthermore, we use $\Theta(u, v)$ to denote the set of all the possible ways by which node v can be reached from node u. The Node-to-Node Delivery Ratio from node u to node v is then defined as:*

$$DR^*(u, v) = max\{DR_\theta, \forall\ \theta(u, v) \in \Theta(u, v)\}$$

Clearly, $DR^*(u, v)$ is equivalent to $DR^*(v, u)$.

Definition 2.3 CDS Reliability. *Given a wireless network represented by $G(V, E, P(E))$ under the PNM model, and its CDS denoted by D, the reliability of D R^*_D is the minimum Node-to-Node Delivery Ratio between any pair of the nodes in the CDS,* i.e.,

$$R^*_D = min\{DR^*(u, v), \forall\ u, v \in D, u \neq v\}$$

We use *CDS Reliability* to measure the quality of a CDS constructed under the PNM model. By this definition, when a CDS D has a reliability R^*_D satisfying a threshold σ (*i.e.* $R^*_D \geq \sigma$), we can state that for any pair of the nodes in the CDS, the probability that they are connected is no less than the threshold.

According to the reliability theory [16], we know that the computation of the Node-to-Node Delivery Ratio is NP-Hard. Therefore, the computation of the CDS reliability is also NP-Hard. In summary, we claim that, given a wireless network represented by $G(V, E, P(E))$ under the PNM model, a CDS for G denoted by D, and a pre-defined threshold $\sigma \in 0, 1]$, it is NP-Hard to verify whether $R^*_D \geq \sigma$.

2.2 Problem Definition

Definition 2.4 Reliable MCDS (RMCDS). *Given a wireless network represented by $G(V, E, P(E))$ under the PNM model, and a pre-defined threshold $\sigma \in (0, 1]$, the RMCDS problem is to find a minimum-sized node set $D \subseteq V$, such that*

1. *The induced graph $G[D] = (D, E')$, where $E' = \{e \mid e = (u, v), u \in D, v \in D, (u, v) \in E)\}$, is connected.*
2. *$\forall\ u \in V$ and $u \notin D$, $\exists\ v \in D$, such that $(u, v) \in E$.*
3. *$R^*_D \geq \sigma$.*

We claim that the problem to construct a RMCDS for a wireless network under the PNM model is NP-Hard. It is easy to see that the traditional MCDS problem under the DNM model is a special case of the RMCDS problem. By setting the TSR values on all edges to 1, we are able to convert the RMCDS problem to the traditional MCDS problem under the DNM model. Thus the RMCDS problem belongs to NP. The verification of the RMCDS problem needs to calculate the CDS Reliability. It is an NP-Hard problem, which is mentioned in Subsection 2.1. Therefore, the problem to construct a RMCDS for a wireless network under the PNM model is NP-Hard.

2.3 Remarks

As we already know, computing the Node-to-Node Delivery Ratio and the CDS reliability are NP-Hard problems. Therefore, instead of computing the accurate Node-to-Node Delivery Ratio, we design a greedy based algorithm to approximate the ratio denoted by $DR(u, v)$. Based on the approximate Node-to-Node Delivery Ratio, we then calculate the approximate *CDS Reliability* denoted by R_D. When there is no confusion, $DR^*(u, v)$ and $DR(u, v)$, R^*_D and R_D are interchangeable in the paper.

Based on Definition 2.4, the key issue of the RMCDS problem is to seek a tradeoff between the minimum-sized CDS and the CDS reliability. GAs are population-based search algorithms, which simulate biological evolution processes and have successfully solved a wide range of NP-Hard optimization problems [17]. In the following, algorithm RMCDS-GA is proposed to solve the RMCDS problem to search the feasible domain more effectively and reduce the computation time.

3 RMCDS-GA Algorithm

3.1 GA Overview

GAs, first formalized as an optimization method by Holland [18], are search tools modeled after the genetic evolution of natural species. In nature, over many generations, natural populations evolve according to the principles of natural selection and *survival of the fittest*. By mimicking this process, GAs work with a population of *chromosomes*, each representing a possible solution to a given problem. Each chromosome is assigned a *fitness score* according to how good a solution to the problem it is. The highly fittest chromosomes are given opportunities to *reproduce*, by *crossover* with other chromosomes in the population. This produces new chromosomes as *offsprings*, which share some features taken from each *parent*. A whole new population of possible solutions is thus produced by selecting the best chromosomes from the current *generation*, and mating them to produce a new set of chromosomes. In the following part of this section, we will explain RMCDS-GA step by step.

3.2 Representation of Chromosomes

In the proposed RMCDS-GA, each node is mapped to a gene in the chromosome. A gene value indicates whether the node represented by this gene is a dominator or not. Hence, a chromosome is denoted as: $C_i = (g_1, g_2, \cdots, g_j, \cdots, g_n)$, where $1 \leq i \leq k$ and k is the number of the chromosomes in the population; $1 \leq j \leq n$ and n is the total number of the nodes in a wireless network.

$$\begin{cases} g_j = 1, & node\ s_j\ \ is\ \ a\ \ dominator \\ g_j = 0, & node\ s_j\ \ is\ \ a\ \ dominatee \end{cases}$$

All the nodes with $g_j = 1$ form a CDS denoted by $D = \{s_j | g_j = 1, 1 \leq j \leq n\}$.

An example wireless network under the PNM model is shown in Fig.1 to illustrate the encoding scheme. There are 8 nodes and the CDS is $D = \{s_4, s_7\}$. Thus, the 8 nodes can be encoded using 8 genes in a chromosome, *e.g.* $C_1 = (g_1, g_2, \cdots, g_8)$, and then set the values of genes representing the dominators to 1. Finally, the encoded chromosome is $C_1 = (0, 0, 0, 1, 0, 0, 1, 0)$.

3.3 Population Initialization

According to the proposed RMCDS-GA algorithm, after we decide the encoding scheme of the RMCDS problem, the first generation (a population with k chromosomes) should be created. This step is called population initialization. A general method to initialize the population is to explore the genetic diversity. That is, for each chromosome, all dominators are randomly generated. However, the dominators must form a CDS. Therefore we start to create the first chromosome by running an existing MCDS method, *e.g.*, Wan's work [15], and then generate the population with k chromosomes by modifying the first chromosome. We call the procedure, generating the whole population by modifying one specific chromosome, Inheritance Population Initialization (IPI).

An example is shown in Fig.1 to illustrate the IPI process. In Fig.1, the network and its CDS $D_1 = \{s_4, s_7\}$ are given. The values on the edges are TSR values and black nodes are dominators. Furthermore, we assume the CDS is constructed by a traditional MCDS method. According to the encoding scheme mentioned in subsection 3.2, $C_1 = (0, 0, 0, 1, 0, 0, 1, 0)$ represents the CDS generated by Wan's work [15] shown in Fig.1. Subsequently, we need to generate more chromosomes based on the first chromosome. The IPI algorithm is summarized as follows:

1. Start from the node with the smallest ID, reduce one dominator each time from the original CDS D_1 represented by C_1. If the new obtained node set is still a CDS D_i, then encode it as a chromosome C_i and add it into the initial population. Otherwise, remove the node with the second smallest ID from the original CDS D_1 and make the same checking process as for the node with the smallest ID. Repeating the process till no more new chromosome can be created. The CDS shown in Fig.1 is a minimum-sized CDS, *i.e.*, we cannot further reduce its size. Thus we go to step 2.
2. If the size of the original CDS D_1 cannot be reduced, and the number of the generated chromosomes is less than k, then for all the existing chromosomes $C_1, C_2, \cdots, C_i$ doing the following steps till k non-duplicated chromosomes are generated.
   (a) Let $t = 1$.
   (b) In the CDS D_t represented by the chromosome C_t, start from node u with the smallest ID, and add one dominatee node in its 1-hop neighborhood $N_1(u)$ by the order of its ID into the CDS each time. If the new obtained node sets form CDSs, then encode them as chromosomes, and add them into the initial population. The node with the smallest ID is s_4 in D. Therefore, the chromosomes from C_2 to C_6 are generated by adding one node from set $N_1(s_4) = \{s_1, s_2, s_3, s_5, s_6\}$ each time.
   (c) Move to the node with the second smallest ID in CDS D_t till every node in D_t are checked. The 1-Hop neighborhood of the node with the second smallest ID s_7 is $N_1(s_7) = \{s_6, s_8\}$. Since s_6 has already been marked as a dominator, we cannot add it to create a new CDS. By eliminating the duplicates, the chromosome C_7 is created.
   (d) If all the dominators in the current D_t are checked, move to the next CDS by setting $t = t + 1$, repeat the step from 2b) to 2d).

3.4 Fitness Function

Given a solution, its quality should be accurately evaluated by the fitness value, which is determined by the fitness function. In our algorithm, we aim to find a minimum-sized CDS D whose reliability R_D should be greater than or equal to a preset threshold σ. Therefore, the fitness function of a chromosome C_i in the population is defined as:

$$f(C_i) = \frac{R_D^2}{|D|^2} \tag{1}$$

The purpose of raising $|D|$ and R_D to the power of 2 in Equation 1 is to enlarge the weight of the size of the CDS D. The denominator in Equation 1 needs to be minimized while the numerator needs to be maximized. As a result, the fitness function value will be maximized.

As mentioned in the previous section, precisely calculating the CDS reliability is an NP-Hard problem. According to Definition 2.3, we can easily compute the CDS reliability based on the Node-to-Node Delivery Ratio of all possible dominator pairs in the CDS. Therefore, we propose a greedy based approximate algorithm to calculate the Node-to-Node Delivery Ratio. We adopt a greedy based routing protocol, Greedy Perimeter Stateless Routing (GPSR) [19], to find the pathes between all dominator pairs. In this work, we modified the greedy criterion to be the largest TSR values in 1-Hop Neighborhood based on GPSR.

3.5 Selection (Reproduction) Scheme

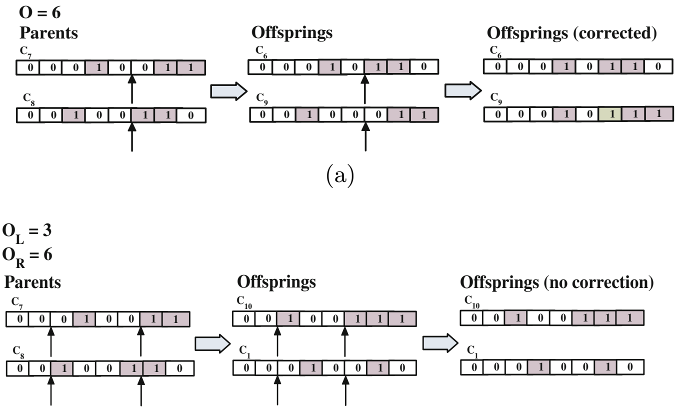
During the evolutionary process, election plays an important role in improving the average quality of the population by passing the high quality chromosomes to the next generation. We adopt *Roulette Wheel Selection* (RWS) since it is simple and effective. RWS stochastically selects individuals based on their fitness values $f(C_i)$. A real-valued interval, S, is determined as the sum of the individuals' expected selection probabilities, *i.e.* $S = \sum_{i=1}^{k} P_i$, where $P_i = \frac{f(C_i)}{\sum_{j=1}^{k} f(C_j)}$. Individuals are then mapped one-to-one into contiguous intervals in the range [0, S]. The size of each individual interval corresponds to the fitness value of the associated individual. The circumference of the roulette wheel is the sum of all fitness values of the individuals. The fittest chromosome occupies the largest interval, whereas the least fit has correspondingly smaller interval within the roulette wheel. To select an individual, a random number is generated in the interval $[0, S]$ and the individual whose segment spans the random number is selected. This process is repeated until a desired number of individuals have been selected.

3.6 Genetic Operations

The performance of a GA relies heavily on two basic genetic operators, *crossover* and *mutation*. Crossover exchanges parts of the current solutions (the parent chromosomes selected by the RWS scheme) in order to find better ones. Mutation flips the values of genes, which helps a GA keep away from local optimum. For crossover, we can adopt all classical operations, however, the new obtained solutions may not be **valid** (the dominator set represented by the chromosome is not a CDS) after implementing the crossover operations. Therefore, a correction mechanism needs to be performed to guarantee validity of all the new generated solutions. Similarly, all traditional mutation operations can be adopted to the RMCDS problem, followed by a correction mechanism.

Crossover. In our algorithm, since a chromosome is expressed by binary codes, we adopt two crossover operators called single-point crossover, two-point crossover respectively. With a crossover probability P_c, each time we use the RWS scheme to select two chromosomes C_i and C_j as parents to perform one of the three crossover operators randomly. We use Fig.2 to illustrate the three crossover operations.

Suppose that two parent chromosomes $C_7 = (00010011)$ and $C_8 = (00100110)$ are selected from the population. By the single-point crossover (shown in Fig.2(a)), the genes from the crossover point to the end of the two chromosomes exchange with each other to get $C_6 = (00010110)$ and $C_9 = (00010111)$. The crossover point denoted by $O = 6$ is generated randomly. After crossing, the first offspring $C_6 = (00010110)$ is a valid solution. However, the other one $C_9 = (00100011)$ is not valid, thus we need to perform the correction mechanism. The correction starts from the gene in the position of the crossover point O, *i.e.* g_6. Since g_6 is 1 in the parent chromosome C_8, it changes to 0 after crossing. We correct it by setting $g_6 = 1$. Then $C_9 = (00010111)$ is now a valid solution. In general, we can keep correcting the genes till the end of the chromosome. By the two-point crossover (shown in Fig.2(b)), the two crossover points are randomly generated which are $O_L = 3$ and $O_R = 6$; and then the genes between O_L and O_R of the two parent chromosomes are exchanged with each other. The two offsprings are $C_{10} = (00100111)$ and $C_1 = (00010010)$ respectively. Since both of the offspring chromosomes are valid, we do not need to do any correction. As we already know, C_1 is the fittest in the population. This is a good illustration, we can obtain a fitter solution during the evolutionary process through genetic operations.



(a)

(b)

Fig. 2. Illustration of Crossover Operations: (a) single-point crossover; (b) two-point crossover; (c) uniform crossover

Mutation. The population will undergo the mutation operation after the crossover operation is performed. With a mutation probability P_m, we scan each gene g_i on the parent chromosomes. If the mutation operation needs to be implemented, the value of the gene flips, *i.e.* 0 becomes to 1 and 1 becomes to 0.

Replacement Policy. The last step of RMCDS-GA is to create a new population using an appropriate replacement policy. Usually, two chromosomes from the evolution process are utilized to replace the two worst chromosomes in the original population for generating a new population. However, when creating new population by crossover and mutation, we have a big chance to lose the fittest chromosome. Therefore, an elitism strategy, in which the best chromosome (or a few best chromosomes) is retained in the next generation's population, is used to avoid losing the best candidates.

The RMCDS-GA stops and returns the current fittest solution until the number of total generations G is achieved or the best fitness value does not change for continuous 10 generations. In the RMCDS-GA algorithm, we use G to stop the algorithm.

4 Performance Evaluation

In the simulations, we implement the RMCDS-GA to solve the RMCDS problem. These algorithms are compared with Wan's work [15] denoted by MIS, which is the latest and best MIS-based CDS construction algorithm.

4.1 Simulation Environment

We build our own simulator where all nodes have the same transmission range ($10m$) and all nodes are deployed uniformly in a square area. Moreover, a random value between $[0.9, 0.98]$ is assigned to the TSR value associated to a pair of nodes inside the transmission range, otherwise, a random value between $(0, 0.8]$ is assigned to the TSR value associated to a pair of nodes beyond the transmission range. For a certain n, 100 instances are generated. The results are averaged among 100 instances. Additionally, the particular GA rules and control parameters are listed in Table 1.

Table 1. GA Parameters and Rules

Population size (k)	20
Number of total generations (G)	100
Selection scheme	Roulette Wheel Selection
Replacement policy	Elitism
Crossover probability (P_c)	1
Mutation probability (P_m)	0.001

4.2 Simulation Results

In Table 2, we show that traditional MCDS construction algorithms cannot solve the RMCDS problem under the PNM model, especially for large scale wireless networks. In Table 2, we list the number of times that MIS and RMCDS-GA can find a CDS with a reliability greater than or equal to σ by running 100 simulations separately. σ is decreased from 0.6 to 0.4 by 0.1. From Table 2, we find that, with increasing n, the number of the times of satisfied CDSs for MIS and RMCDS-GA both decrease. This is because the sizes of CDSs increase which leads to a lower Node-to-Node Delivery Ratio. Moreover, RMCDS-GA can guarantee more satisfied CDSs than MIS, especially when $n \geq 200$. In other words, for large scale wireless networks, it is hard to construct a satisfied CDS for MIS since the MIS algorithm does not consider reliability. Additionally, both MIS and RMCDS-GA can find more satisfied CDSs when σ decreases. In

Table 2. MIS-based CDSs and RMCDS-GA generated CDSs

n	$\sigma = 0.6$		$\sigma = 0.5$		$\sigma = 0.4$	
	MIS	GA	MIS	GA	MIS	GA
50	100	100	100	100	100	100
80	94	100	100	100	100	100
120	57	100	98	100	100	100
160	21	100	90	100	100	100
200	5	96	44	100	88	100
250	2	91	12	93	56	100
400	1	90	4	17	10	100

conclusion, traditional MCDS construction algorithms do not take reliability into consideration, while RMCDS-GA can find a satisfied reliable MCDS which is more practical in real environments.

In Table 3, R_{MIS} and R_{GA} represent the reliability of a CDS generated by MIS and RMCDS-GA, respectively. $|D_{MIS}|$ and $|D_{GA}|$ represent the size of the CDS constructed by MIS and RMCDS-GA, respectively. In Table 3, the reliability of CDSs decreases when the area size increases, since the number of the dominators increases. RMCDS-GA can guarantee to find a more reliable CDS than MIS, *i.e.*, $R_{GA} > R_{MIS}$. More importantly, the sizes of the CDSs obtained by MIS and RMCDS-GA are almost the same. On average, RMCDS-GA can find a CDS with 10% more reliability without increasing the size of a CDS than MIS. In summary, RMCDS-GA does not trade CDS size for CDS reliability.

Table 3. R & $|D|$ results of MIS and RMCDS-GA algorithms

Area (m^2)	n	R_{MIS}	R_{GA}	$\|D_{MIS}\|$	$\|D_{GA}\|$
40×40	50	0.65	0.77	17	18
50×50	80	0.59	0.72	24	26
60×60	120	0.51	0.68	33	33
70×70	160	0.46	0.62	40	44
80×80	200	0.44	0.58	51	51
90×90	250	0.39	0.53	63	62
100×100	400	0.32	0.49	78	78

5 Conclusions

In this paper, we have investigated the RMCDS problem using a new network model called PNM. The PNM model is based on empirical studies show that most wireless links are lossy links which only probabilistically connect pairs of nodes. Different from the traditional DNM model which assumes that links are either connected or disconnected, the PNM model enable the employment of lossy links by introducing the TSR value on each lossy link. In this paper we focus on constructing a minimum-sized CDS while its reliability satisfies a preset application-dependent threshold. We claim that RMCDS is an NP-Hard problem and propose a GA to address the problem. The simulation results show that compared to the traditional MCDS algorithm, RMCDS-GA can find a more reliable CDS without increasing the size of a CDS.

References

1. Wassim, E.H., Ala, A.F., Mohsen, G., Chen, H.H.: On Efficient Network Planning and Routing in Large-Scale MANETs. IEEE Transactions on Vehicular Technology 58(7) (2009)
2. Polat, B.K., Sachdeva, P., Ammar, M.H., Zegura, E.W.: Message Ferries as Generalized Dominating Sets in Intermittently Connected Mobile Networks. In: MobiOpp 2010, Pisa, Italy (2010)
3. Helmy, A., Garg, S., Pamu, P., Nahata, N.: CARD: A Contact-based Architecture for Resource Discovery in Ad Hoc Networks. MONET 10(1), 99–113 (2004)
4. Deb, B., Bhatnagar, S., Nath, B.: Multi-resolution state retrieval in sensor networks. In: IWSNPA (2003)
5. Ammari, H.M., Giudici, J.: On the Connected k-Coverage Problem in Heterogeneous Sensor Nets: The Curse of Randomness and Heterogeneity. In: ICDCS (2009)
6. Ji, S., Li, Y., Jia, X.: Capacity of Dual-Radio Multi-Channel Wireless Sensor Networks for Continuous Data Collection. In: Infocom (2011)
7. Wan, P.J., Huang, S.C.-H., Wang, L., Wan, Z., Jia, X.: Minimumlatency aggregation scheduling in multihop wireless networks. In: MobiHoc 2009 (2009)
8. Ephremides, A., Wieselthier, J., Baker, D.: A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling. Proceeding of the IEEE 75(1), 56–73 (1987)
9. Guha, S., Khuller, S.: Approximation Algorithms for Connected Domimating Sets. Algorithmica 20, 374–387 (1998)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. WH Freeman & Co., New York (1979)
11. Wu, J., Li, H.: On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks. In: DIALM (1999)
12. Li, Y., Thai, M., Wang, F., Yi, C., Wang, P., Du, D.: On Greedy Construction of Connected Dominating Sets in Wireless Networks. Wireless Comm. and Mobile Computing 5(8), 927–932 (2005)
13. Cerpa, A., Wong, J., Kuang, L., Potkonjak, M., Estrin, D.: Statistical model of lossy links in wireless sensor networks. In: IPSN 2005, Los Angeles, CA (2005)
14. Liu, Y., Zhang, Q., Ni, L.-M.: Opportunity-Based Topology Control in Wireless Sensor Networks. In: TPDS 2010, vol. 21(9) (2010)
15. Wan, P.J., Huang, S.C.-H., Wang, L., Wan, Z., Jia, X.: Minimum-latency aggregation scheduling in multihop wireless networks. In: MobiHoc (2009)
16. Agrawal, A., Barlow, R.: A Survey of Network Reliability and Domination Theory. Operations Research 32, 298–323 (1984)
17. Hu, X., Zhang, J., Yu, Y., Chung, H., Lim, Y., Shi, Y., Luo, X.: Hybrid Genetic Algorithm Using a Forward Encoding Scheme for Lifetime Maximization of Wireless Sensor Networks. ITEC 14(5) (2010)
18. Holland, J.H.: Adaptation in Natural and Artificial System. University of Michigan Press, Ann Arbor (1975)
19. Karp, B., Kung, H., GPSR: greedy perimeter stateless routing for wireless networks. In: MobiCom 2000, New York, 2000.