A Novel Ant Colony Optimization-Based Algorithm for the Optimal Communication Spanning Tree Problem

Nhu Gia Nguyen, Dac-Nhuong Le, and Nguyen Dang Le

Abstract—**The optimal communication spanning tree (OCST) problem finds a spanning tree that connects all node satisfies their communication requirements for a minimum total cost. In this paper, we present a new method of finding optimal solution for OCST problem based on Ant Colony Optimization (ACO) to reduce search space mentioned above but still converge to a global good solution. Our algorithm take account into node biased encoding (NBE) scheme to find nearly optimal solution. The new algorithm can achieve a result that is better than known heuristic algorithms do, as verified by a set of public benchmark problem instances.**

Index Terms—**Optimal communication spanning tree, node biased encoding, ant colony optimization.**

I. Introduction

The problem of finding optimal communication spanning tree was introduced by Hu in 1974 [1]. Its NP- hard property was shown by Johnson et al [2]. The formal definition of this problem is below.

Let $G=(V,E)$ is an undirected weighted graph, in which *V* is set of *n* vertices, *E* is set of *m* edges. Moreover, let $D=\left(d_{ij}\right)_{n\times n}$ and $R=\left(r_{ij}\right)_{n\times n}$ be the distant matrix and requirement matrix of the graph *G*, respectively. For any pair of nodes *i* and *j* in *V*, d_{ij} is the weight of the edge *(i, j)* and is infinite if not; r_{ij} is the communication requirement between them. Both *R* and *D* are all symmetric and their elements are nonnegative.

In this context, each spanning tree *T* of *G* has a cost $w(T)$ which is total of the weighted of paths over all pairs of vertices in *T*.

In general, $w(T)$ is computed by following equation:

$$w(T)=\sum\nolimits_{i<j;i,j\in T} f\left(r_{ij}\right) \tag{1}$$

where p_{ij}^{T} denotes the distant between node *i* and node *j* on the tree *T*.

Let $\{i=v_1,v_2,...,v_k=j\}$ be the path between node *i* and *j* in the tree *T*, then $p_{ij}{}^{T}$ is calculated by the following equation:

$$p_{ij}^{T}=\sum\nolimits_{1<l<k} d_{v_l} \tag{2}$$

The function *f* is defined as the product of two variables:

$$f\left(r_{ij},p_{ij}^{T}\right)=r_{ij}\times p_{ij}^{T} \tag{3}$$

The objective function is then:

$$minimize_{T\in\Im}w(T) \tag{4}$$

where $\Im$ is the set of *n* $^{n-2}$ label spanning tree of *G* [3]. We take the Palmer's *6*-nodes instance from [4] as an example to clear out the definition.

Example 1: The Palmer's *6*-nodes instance.

The distant matrix and requirement matrix are given below:

0	16661	18083	21561	21099	13461
16661	0	5658	9194	8797	10440
18083	5658	0	7230	6899	11340
21561	9194	7230	0	4300	13730
21099	8797	6899	4300	0	13130
13461	10440	11340	13730	13130	0

(a) The distant matrix

0	1	1	1	1	2
1	0	10	3	4	3
1	10	0	5	6	2
1	3	5	0	31	2
1	4	6	31	0	2
2	3	2	2	2	0

(b) The requirement matrix

Fig. 1. The palmer's 6-nodes instance.

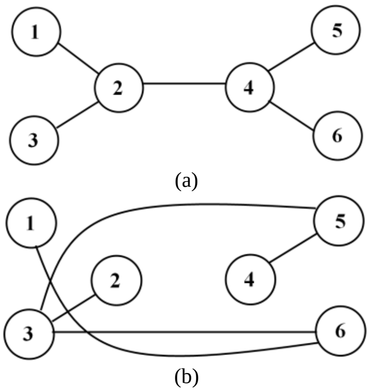

Fig. 2. Two communication spanning tree (a) of Palmer's 6-nodes

The cost of the tree *T* in Fig.2.a is compute as below:

$$\begin{aligned} w(T) &= r_{12}.d_{12} + r_{13}(d_{12}+d_{23}) + r_{14}(d_{12}+d_{24}) + \\ &\quad + r_{15}(d_{12}+d_{25}+d_{45}) + r_{16}(d_{12}+d_{24}+d_{46}) + \\ &\quad + ... + r_{56}(d_{54}+d_{46}) \\ &= 817626 \end{aligned} \tag{5}$$

Manuscript received October 9, 2012; revised December 26, 2012.

Nhu Gia Nguyen is with the Duy Tan University, Danang, Vietnam (e-mail: Nguyengianhu@duytan.edu.vn).

Dac-Nhuong Le is with the Faculty of Information Technology, Haiphong University, Vietnam (e-mail: Nhuongld@ hus.edu.vn).

Nguyen Dang Le is with the Haiphong University, Vietnam (e-mail: Nguyenld@ hus.edu.vn).

DOI: 10.7763/IJCTE.2013.V5.739

For this instance, the optimal solution is given in Fig.2.b with the minimum cost is 693180.

It is important to note that the OCST problem is far more different from the minimum spanning tree problem, which is already solved by several polynomial algorithms (such as Prim [5] and Kruskal). Moreover, there are several variations of this problem when the form of the distant matrix D or the requirement matrix R changes. More information can be found in [6].

II. Related Works and Our Works

Due to its large application, there are many researched on OCST problem, from both exact and heuristic approaches. Authors in [1] defined the problem and discussed two simplest cases. The NP- hard property of the problem was showed [2] and Reshef showed that the problem is even a MAX SNP–hard one, which cannot be solved by a polynomial approximation algorithm unless *P=NP* [6].

Therefore, heuristic optimization methods have been chosen by researchers to find optimal or nearly optimal solutions for the OCST problems. One of the first heuristic approaches was presented by Ahuja and Murty, which is closes to local search methods. Palmer, in his PhD thesis [8], proposed a genetic algorithm (GA) using the node and link biased encoding that will be clearly descript in section III.A. From the same approach, Li and Bouchebaba proposed another GA which works on a tree chromosome without intermediate encoding and decoding [9]. This GA uses a Prim -based algorithm to randomly generate the initial population and performs better than the former one.

Some other GAs using variants of Prufer numbers were proposed, such as Picciotto [10], Julstrom [11], [12], etc. However, Jens Gottlieb stated that Prufer numbers based chromosome presentations are not good for constrained spanning tree optimization problems [13]. After all, from statistical view, Rothlauf argued that good solution for the OCST are biased towards minimum spanning trees [14].

In the latest paper have published in [15], we have proposed an algorithm to finding optimal solution for OCST problem based on Particle Swarm Optimization (PSO). In this paper, we present a new method of finding optimal solution for OCST problem based on Ant Colony Optimization (ACO) to reduce search space mentioned above but still converge to a global good solution. The NBE schema is used to represent candidate solutions. Our approach outperforms Palmer's and Li and Bouchebaba's GAs in not only final solution but also computation time.

The rest of the paper is organized as follows. The NBE schema is described in Section III.A. We present our method in Section III.B. Section IV presents some experimental results. We make conclusion and recommend some future works in Section V.

III. Ant Colony Optimization for Solving Graph Coloring Problem

A. Node Biased Encoding

The node biased encodings is among the class of weighted linear chromosome representation for spanning tree optimization problems. Its original version was developed by Palmer [8]. In this scheme, each spanning tree is encoded by assigning every nodes and link on graph to a corresponding weight. The details of encode and decode phases are given below.

1) *Encode phase:* Each spanning tree T of graph $G=(V,E)$ is represented by a n- components real vector $b=(b_1,b_2,...,b_n)$. Remind that n is the cardinant of V. b is now called the weighted vector.
2) *Decode phase:* Take two following steps to calculate spanning tree T correspond to the weighted vector b.

- *Step 1:* Construct the temple graph $G'=(V,E)$ with modified distant matrix $D'=(d_{ij})_{n\times n}$ computed as below equation:

$$d'_{ij}=d_{ij}+P_2\times(b_i+b_j)\times d_{\max} \quad (6)$$

where $d_{\max}=Max_{ij}(d_{ij})$ and P_2 is the node biased parameter which controls the node weights on the tree construction

- *Step 2:* Calculate T as the minimum spanning tree of temple graph G' by using Prim algorithm.

Note that, by running Prim algorithm, nodes i with low b_i will probably be interior nodes, and on the other hands, nodes j with high b_j will probably be leaf nodes. Therefore, the higher the weight b_i, the higher is the probability that node i will be a leaf node. We take the example below to illustrate the functionality of the NBE scheme.

Example 2: NBE scheme

Let G is a 5-nodes complete graph with distant matrix D given in Fig. 3.

0	2	1	3	4
2	0	5	6	3
1	5	0	4	3
3	6	4	0	10
4	3	3	10	0

Fig. 3. The distant matrix of graph in example 2

Then the modified distant D' correspond to vector $b=(0.7,\ 0.5,\ 0.8,\ 0.1)$ and parameter $P_2 = 1$ is given in Fig. 4 and by running the Prim algorithm we get the tree in Fig.5.

0	14	10	18	12
14	0	12	19	11
10	12	0	14	6
18	19	14	0	19
12	11	6	19	0

Fig. 4. The modified distant matrix of temple graph in example 2

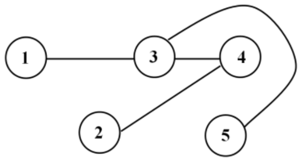

Fig. 5. Spanning tree computed in example 2

B. Solving the OCST Problem Base on ACO

The ACO algorithm is originated from ant behavior in the food searching. When an ant travels through paths, from nest food location, it drops pheromone. According to the pheromone concentration the other ants choose appropriate path. The paths with the greatest pheromone concentration are the shortest ways to the food. The optimization algorithm can be developed from such ant behavior.

The first ACO algorithm was the Ant System [16], and after then, other implementations of the algorithm have been developed [17-18].

In this section, we present application of ACO technique for the OCST problem. Without loss of generality, we can assume that *G* is undirected complete graph. Our new algorithm is described as follows:

1) *Expressing an element:* We use NBE scheme to represent ants. Each ant is a vector with n components; every component is a real number corresponds to one in *n* vertices. To decode an ant we do exactly steps described in previous section.
2) *Population Initialization:* As mentioned in section II, good solution for the OCST problem is biased towards minimum spanning trees. The ant population is randomly generated (50 ants) correspond to the minimum spanning tree one. This ant is the vector whose components are all zero. Others n specially initiated ants are of the form $\{1,1,...,1,0,1,1,...,1\}$. Those ants correspond to star- shaped trees. This is motivated by the fact that a star-shaped tree is the optimal solution for a special case [1].

All others *P-(n+1)* are vectors whose components are real numbers between 0 and 1 which are randomly generated (*P* is a designed parameter).

3) *Cost function:* The next node is selected according to the probability with which ant *k* will choose to go from current node *i* to next node *j* is calculated by the following formula:

$$p_{ij}^{k} = \frac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l\in N_i^k}\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}} \qquad (7)$$

In which, τ_{ij} is the pheromone content of the path from node *i* to node *j*, N_i^k is the neighborhood includes only nodes that have not been visited by ant k when it is at node *i*, η_{ij} is the desirability of node *j*, and it depends of optimization goal so it can be our cost function.

The influence of the pheromone concentration to the probability value is presented by the constant *α*, while constant *β* do the same for the desirability. These constants are determined empirically and our values are *α=1, β=10.*

The ants deposit pheromone on the locations they visited according to the relation.

$$\tau_j^{new} = \tau_j^{current} + \Delta\tau_j^k \qquad (8)$$

Here, $\Delta\tau_j^k$ is the amount of pheromone that ant k exudes to the node *j* when it is going from node *i* to node *j*.

This additional amount of pheromone is defined by:

$$\Delta\tau_j^k = \frac{1}{f_j} \qquad (9)$$

where f_j is the cost function of the k^{th} ant is computed by the formula (3).

The pheromone evaporates during time and diminishes if there are no new additions. The pheromone evaporation is applied to all locations as follows:

$$\tau_j^{new} = (1-\rho)\tau_j \quad (0 < \rho \le 1) \qquad (10)$$

The value of *ρ* is selected empirically, what is in our case *ρ= 0.1.*

C. Our Algorithms

The Fig. 6 presents process of our algorithm to solving OCST based on ACO.

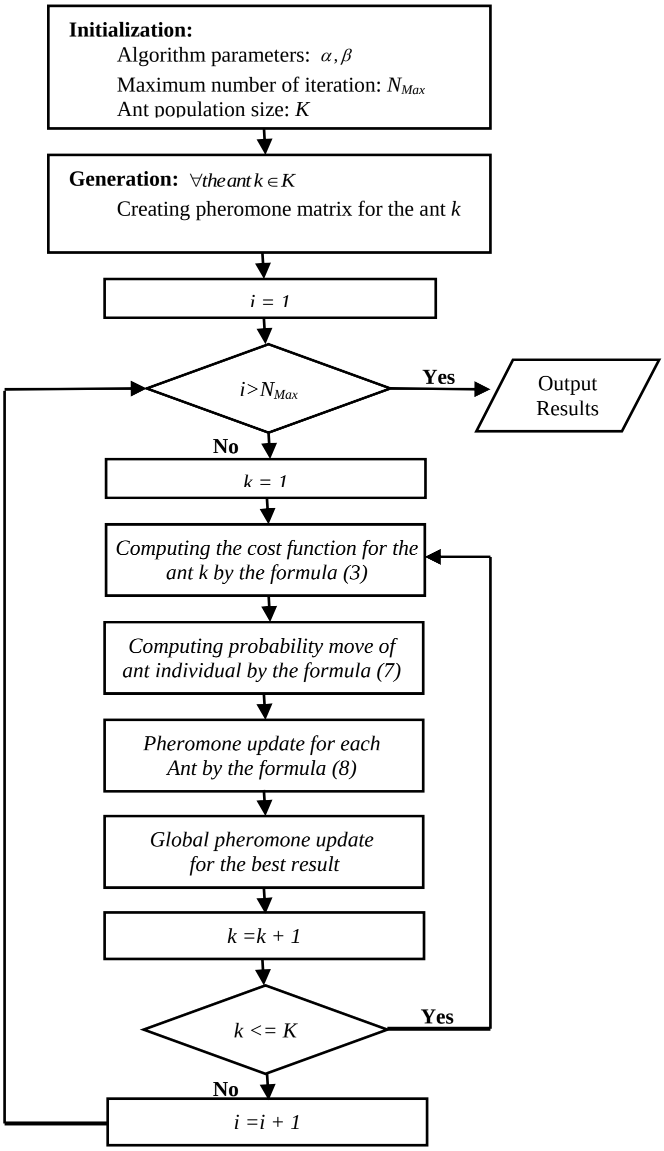
Initialization:
Algorithm parameters: α,β
Maximum number of iteration: N_{Max}
Ant population size: K
Generation: $\forall$ the ant $k \in K$
Creating pheromone matrix for the ant k
i = 1
i>N_{Max}
Yes
Output Results
No
k = 1
Computing the cost function for the ant k by the formula (3)
Computing probability move of ant individual by the formula (7)
Pheromone update for each Ant by the formula (8)
Global pheromone update for the best result
k =k + 1
k <= K
Yes
No
i =i + 1

Fig. 6. The ant colony algorithm's flow chart

IV. Experiment Results

In order to measure the performance, we use a set of standard benchmark instances:

- 6 nodes (*Palmer6*), 12nodes (*Palmer12*) and 24 nodes (*Palmer24*) networks from Palmer;
- 10 nodes (*Raidl10*), 20 nodes (*Raidl20*), 50 nodes (*Raidl50, Raidl50gen*) networks from Raidl.

- 6 nodes (*Berry6*) networks from Berrry.

These instances can be found in [4].

We run our ACO algorithm on those above instances and compare to the results produced by Palmer's GA, Li and Bouchebaba's GA and our PSO algorithm [15] in Table I.

Comparision cost function of the optimization methods:

- For Palmer series, we have the two optimal solutions (*Palmer6 and Palmer12*) with ACO and one optimal solution (*Palmer12*) with PSO.
- For Raild series, we have the same two optimal solutions (*Raild10 and Raild20*) with Li & Bouchebaba's.
- In other instances (*Raidl50 and Raidl50gen*), both PSO and ACO algorithm have better solution than Palmer's and Li & Bouchebaba's.
- The last instance (*Berry6*), all four algorithms have the same an optimal solution. However, our algorithms are still the fastest.

Comparision time processing of the optimization methods is presents in the Fig. 7 and Fig. 8.

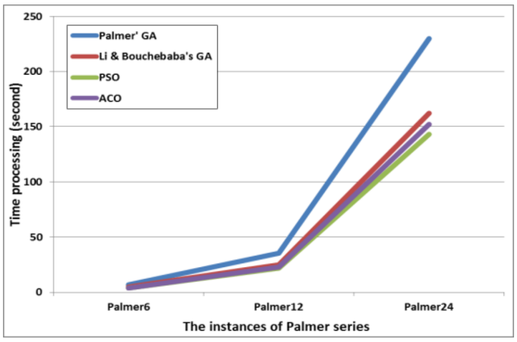

Fig. 7. Comparision time processing of algorithms with the instances of Palmer series

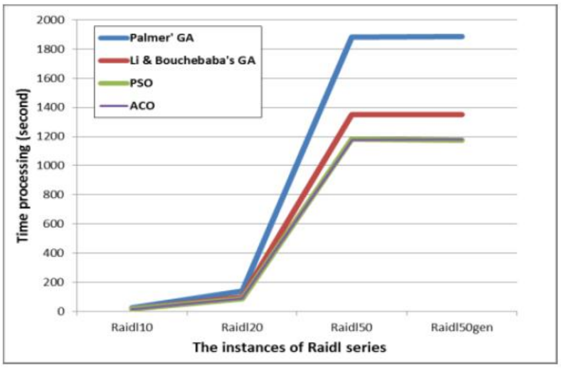

Fig. 8. Comparision time processing of four algorithms with the instances of Raidl series

Comparision results of cost function of the optimization methods is presents in the Fig. 9.

V. Conclusion and Future Work

In this paper, we present a novel Ant Conoly Algorithm (ACO)-based algorithm to solve the optimal communication spanning tree (OCST) problem in which the each ant is encode using NBE scheme. The experiments results show that the proposed algorithm is effective to get optimal and near optimal solution in term of final solution cost and running time.

In the future, we will test our algorithm on more problem instances, both from literature and real world. We will also try find out a better encoding scheme and adapt our method to other problems.

TABLE I: Comparision of the Results for Standard Benchmark Instance
(An Entry Marked by a Star (*) Denotes a Optimal Solution)

Instance	Palmer' GA		Li & Bouchebaba's GA		PSO		ACO	
	Cost	*Time*	*Cost*	*Time*	*Cost*	*Time*	*Cost*	*Time*
Palmer6	709770	6.88	708090	5	693180	4.1	693162*	3.92
Palmer12	3876488	35.53	3457952	24.7	3428509*	22	3428509*	23
Palmer24	1959790	229.67	1086656*	162	1138360	143	1138364	152
Raidl10	58352	22.21	53674*	15.93	53674*	14	53674*	14.71
Raidl20	165788	138.50	157570*	97.3	157570*	87.3	157570*	86.9
Raidl50	911987	1883.09	134931	1350	107746*	1178	107746*	1176
Raidl50gen	123013	1883.86	964140	1351	826499*	1177	826499*	1179
Berry6	534*	6.80	534*	5.74	534*	4.6	534*	4.5

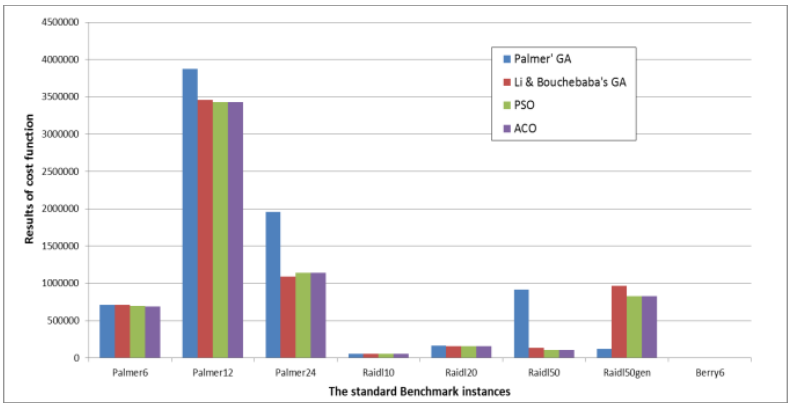

Fig. 9. Comparision results of cost function of four algorithms with the standard Benchmark instances

References

[1] T. C. Hu, *Optimum Communication Spanning Trees*, SIAM Journal of Computing, vol. 3, no. 3, pp. 188-195, 1974.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979

[3] A. Cayley, "A theorem on trees," *Quart. J. Math*. vol. 23, pp. 376-378, 1889.

[4] F. Rothlauf, *Representations for Genetic and Evolutionary Algorithms*, Springer, Heidelberg New York, 2nd Edition, 2006.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, Second Edition. MIT Press and Mc Graw-Hill, 2001.

[6] B. Y. Wu and K. M. Chao, *Spanning Trees and Optimization Problems*, Chapman & Hall/CRC, 2004

[7] E. Reshef, "Approximating minimum communication cost spanning, trees and related problems," Master's thesis, Feinberg Graduate School of the Weizmann Institute of Science, Rehovot 76100, Israel. April 1999.

[8] C. C Palmer, "An approach to problem in network design using genetic algorithms," PhD Thesis, Polytechnic University, Computer Science Department, Brooklyn, New York, 1994.

[9] Y. Li and Y. Bouchebaba, "A new genetic algorithm for the optimal communication spanning tree problem," in *Proceedings of Artificial Evolution: Fifth European Conference*, Berlin, Springer, pp.162-173, 1999.

[10] S. Picciotto, "How to encode a tree," Ph. D. thesis, University of California, San Diego, USA, 1999.

[11] B. A. Julstrom, "The blob code: A better string coding of spanning trees for evolutionary search," A. S. Wu (Ed.), in *Proceedings of the 2001 Genetic and Evolutionary Computaton Conference Workshop Program*, San Francisco, California, USA, pp. 256-261, 2001.

[12] B. A. Julstrom, "The blob code is competitive with edge-sets in genetic algorithms for the minimum routing cost spanning tree problem," in Beyer, Hans-Georg et al. (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference 2005*, New York, ACM Press, pp. 585-590,2005.

[13] J. Gottlieb, B. A. Julstrom, G. R. Raidl, and F. Rothlauf, "Prufer numbers: A poor representation of spanning trees for evolutionary search," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 343-350. 2001

[14] F. Rothlauf, J. Gerstacker, and A. Heinzl, "On the optimal communication spanning tree problem," Technical Report 15/2003, Department of Information Systems, University of Mannheim. 2003.

[15] A. T. Hoang, V. T. Le, and N. G. Nguyen, "A novel practicle swarm optimization- based algorithm for the optimal communication spanning tree problem," in *Proceedings of The 2010 International Conference on Communication Software and Network (ICCSN 2010)*. pp.232-236, 2010.

[16] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. on System, MAN, and Cybernetics-Part B*, vol. 26, pp. 29-41, February 1996.

[17] E. R. Iglesias and O. Q. Teruel, "Linear array synthesis using an ant colony optimization based algorithm," *IEEE Trans. on Antennas and Propagation*, 2005.

[18] M. Dorigo, M. Birattari, and T. Stitzle, "Ant colony optimization: arificial ants as a computational intelligence technique," *IEEE Computational Intelligence Magazine*, November, 2006.

Nhu Gia Nguyen received the BSc degree in computer science and the MSc degree in information technology from Dannang University, Vietnam, in 1998 and 2006, respectively. He currently works in Duy Tan University, Danang, Vietnam. His research interests include algorithm theory, network and wireless security.

Dac-Nhuong Le received the BSc degree in computer science and the MSc degree in information technology from College of Technology, Vietnam National University, Vietnam, in 2005 and 2009, respectively. He is currently a lecturer at the Faculty of information technology in Haiphong University, Vietnam. His research interests include algorithm theory, computer network and networks security

Nguyen Dang Le received the BSc degree in computer science and the MSc degree in information technology from College of technology, Vietnam National University in Hanoi, Vietnam, in 1996 and 2005, respectively. He currently works in Haiphong University, Vietnam. His research interests include algorithm theory, network and wireless security.