Hierarchic Genetic Algorithm For Designated Multi-nodes Routing

Xiaoli Xiao
College of Computer and Communication Engineering
Changsha University of Science and Technology
Changsha , china
E-mail: ttxxli@163.com

Min Huang
College of Computer and Communication Engineering
Changsha University of Science and Technology
Changsha , china
huangmin_09@yahoo.com.cn

Abstract—**This paper proposed a hierarchic genetic algorithm used for solving the shortest path problem via designated multi-nodes routing of wireless networks. The hierarchy of solutions was plotted according to designated nodes, Variable-length chromosomes were employed in our genetic algorithm and the stochastic universal sampling was adopted as the selective operation. Crossover and mutation used the topological information from network to operate. A repair function was used to eliminate the unfeasible solution produced by genetic operations. The algorithm separated the solutions of feasible regions of objective function to reduce the size of populations and chromosome length, and each sub-population can run independently and simultaneously. Using this method can decompose complex routing issues into many sub-issues. By solving sub-issues the shortest path via designated multi-nodes can be obtained. The authors made simulation experiments on the algorithm by establishing a random network topology. The results indicated that the algorithm can effectively get the shortest path via designated multi-nodes , exhibits a much higher rate of convergence and is relatively less subject to network scale and the problem's complexity than traditional methods.**

Keywords-Wireless networks; hierarchic genetic algorithm; routing; random network topology

I. INTRODUCTION

In wireless networks, routing is the most important factor affecting network performance. In wireless networks environments, the packet from correspondent node to mobile node must transfer by his hometown agent and then reach the mobile node by foreign agent. During this process, its hometown agent and correspondent node is specified nodes, and the foreign agent of mobile node is also not changed in a certain period of time [1]. Therefore, this process is actually a routing one through designated multi-nodes.

Besides, in wireless networks environment security is also an important issue. If the address binding information has no powerful authentication mechanism, then in the network the security transferring data can not be protected. By using this routing scheme via designated multi-nodes the packets can get through the established security tunnel to avoid packets entering non-trusted zone[2].

The genetic algorithm applied to network routing is a hot issue of current genetic algorithm study. The reference[3] studied the genetic algorithm for a single specified node.

In this paper, the hierarchic genetic algorithm is presented for the routing problem of passing many designated nodes in the environment of wireless networks, it is designed to implement the routing passing many designed nodes. The proposed algorithm separated the feasible solutions of object function to reduce the complexity. The shortest routing of wireless networks via the designated multi-nodes is gained by executing independently genetic in multi sub-populations in parallel. The algorithm was simulated by emulation experiments to verify the effectiveness and feasibility of the algorithm.

II. MATHEMATICAL MODEL VIA DESIGNATED MULTI-NODES ROUTING EASE OF USE

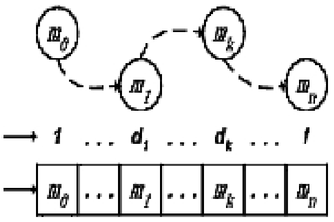
m_0
m_1
m_k
m_n
position
1 ... d_1 ... d_k ... l
chromosome
m_0 ... m_1 ... m_k ... m_n

Figure 1. The path via designed nodes

Network topology is defined as $G=(N,L)$, where N is a set of nodes, L is a set of edges, weight matrix is defined as matrix $C=[c_{ij}]$, c_{ij} is the weight value of link which connects node i with *node j* , the value represents cost of the link, if there is no link between nodes *i and j* , *cij* is 0. Matrix I_{ij} represents if there is a link between node i and j, S and D represent the source node and destination node of a packet respectively. $M=\{m_0,\ldots,m_n\}$ is a set which represents it should pass n + 1 specified nodes, $m_i \in N, n \geq 2$. The algorithm needs to find a path R, which should pass all the nodes in M orderly, while $R=(m_0,\ldots m_1,\ldots,m_k,\ldots m_n)$, where

The Project of Department of Education of Hunan Province of China under Grant No. 08C101

978-0-7695-3643-9/09 $25.00 © 2009 IEEE
DOI 10.1109/ISECS.2009.90

m_0=*S*，m_n=*D*，*R i*s a feasible region of the problem. That is shown in Figure 1. The authors considered the weight value of link and got following constraint optimization problems.

$$Min \quad f(r) = \sum_{i=0}^{index(m_n)-1} c_{i,i+1} \qquad (1)$$

$$\text{subject to} \sum_{j=m_0, j\neq i}^{m_n} I_{ij} - \sum_{j=m_0, j\neq i}^{m_n} I_{ji} = \begin{cases} 1 & i = m_0 \\ -1 & i = m_n \\ 0 & otherwise \end{cases} \qquad (2)$$

$$\text{subject to} \sum_{j=s, j=i}^{D} I_{\text{ij}} \begin{cases} \leq 1 \text{ if } & \text{i} \neq \text{D} \\ = 0 \text{ if } & \text{i} = \text{D} \end{cases} I_{\text{ij}} \in \{0,1\} \qquad (3)$$

Where $r \in R$ and *r* is a non-circuit link, *index(m_i)* represents the order of m_i in link *r*. Constraints (2) and (3) ensure that the path is non-circuit[4] .

As seen in the feasible region of the problem, n+1 specified nodes will not changed. That is all the solutions have fixed pattern. By separating the feasible region we can reduce the populations size of genetic algorithm. By dividing the populations, the problem can be described as the following:

$$Min \quad f(r) = \sum_{j=0}^{n-1} \sum_{i=Index(m_j)}^{Index(m_{j+1})} c_{i,i+1} \qquad (4)$$

The minimum from (4) is $f(r^*) = \sum_{j=0}^{n-1} \min(\sum_{i=Index(m_j)}^{Index(m_{j+1})} c_{i,i+1})$, which $r^* \in R$ is the optimal solution. Therefore, the original question can be divided into n optimized issues, which means seeking shortest routing from the specified nodes M_j *to* M_{j+1} respectively.

$$Min \quad f_{m_j, m_{j+1}}(r) = \sum_{i=Index(m_j)}^{Index(m_{j+1})} c_{i,i+1} \qquad (5)$$

Then the entire shortest path $f(r^*)$ can be expressed as the sum of all shortest paths:

$$f(r^*) = \sum_{j=0}^{n-1} f_{m_j, m_{j+1}}(r^*) \qquad (6)$$

In (6) ,$m_j, m_{j+1} \in r, j \in [0, n-1]$. The routing problem between nodes M_j and M_{j+1} must satisfy the following constraint to avoid appearing circuit between two specified nodes.

$$\text{subject to} \sum_{k=m_j, j\neq k}^{m_{j+1}} I_{ik} - \sum_{k=m_j, j\neq i}^{m_{j+1}} I_{ki} = \begin{cases} 1 & i = m_j \\ -1 & i = m_{j+1} \\ 0 & otherwise \end{cases} \qquad (7)$$

Through the above method (1) is divided into multiple optimization problems, eventually, the shortest path will be obtained by merging several optimal solutions.

III. Hierarchic Genetic Algorithm for Calculating Designated Multi-Nodes Routing

There are many factors which is considered when we employ genetic Algorithm to solve real different problems such as encoding method, initial populations, selection, the selection of fitness function, crossover operation, mutation operation and the choice of parameters. These operators will be described in detail below.

A. Encoding

Encoding is the most fundamental operation in genetic algorithm. In the paper we use path to encode. a chromosome consists of ID number of nodes. The chromosome can be used to represent a single path. Gene location of the chromosome represents the order of nodes in the path, shown as in Figure 2. In the figure there are four nodes, then the corresponding chromosomes are *n1, n2, d*. the order of four nodes in the path can been fixed. You can find the first node and the last node will not be changed in the encoding method, the first node is the source node, and the last node is the destination node. Chromosomal length can changed, but the length is subject to total number of network nodes, chromosomes length *l* cannot exceed the number of nodes . The chromosomal codes comes from the network topology database, in this paper we use network adjacency matrix to represent them[5].

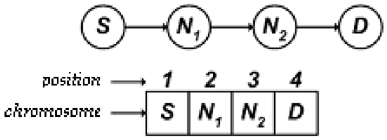

Figure 2. Encoding

Routing problems via n designated nodes in the second section can be divided into n optimization problem, therefore, we can employ genetic algorithm with n sub-populations, the whole population for solving the problem is defined as follows:

$$POP = \bigcup_{i \in [0, n-1]} pop_j \qquad (8)$$

That is $POP = [pop_0, pop_1, \cdots pop_{n-1}]$, sub-population pop_j is composed of all paths that link node *Mj* and *Mj*$_{+1}$, node *Mj* is the source node and node *Mj*$_{+1}$ is the destination node, $pop_j = \begin{bmatrix} m_j^1 & \cdots & m_{j+1}^1 \\ \vdots & \ddots & \vdots \\ m_j^{pSize} & \cdots & m_{j+1}^{pSize} \end{bmatrix}$, *pSize* is the size of population. Length of the path from node *Mj to* node *Mj*$_{+1}$ can be changed, therefore, the population pop_j is composed of variable - length chromosomes. Of course, the length of non-loop path can not exceed the number of entire network nodes.

B. Initial populations

In this paper all solutions of the problem are composed of the paths that are represented by the node's ID. Therefore, initial populations are randomly formed, The initial populations must satisfy the constraints. Random operation must be carried out in accordance with the given network topology information. The initializing process is described as follows:

Step1: Select the source node *S* and the destination node *D* from any non - empty network topology *G* and the specified nodes' set *M*
Step2: Initialize an empty chromosome *chr,* push (*chr, S*)
Step3: Get last element *T* from *chr*, pop (*chr, T)*
Step4: If $T \in M$, delete *T* from *M*
Step5: Choose a subsequent node *LT* of the node *T* from *G* randomly, push (*chr, LT*), remove *LT* from *G*;

Step6: If $G = \Phi \cap LT \neq D$, go step 3
Step7: If $LT = D \cap M = \Phi$ then you get a chro-mosome *ch*;
Step8: Calculate the fitness values of chromosome *ch*;
Step9: Insert *chr* into the initial population
Step10:If the size of initial population satisfy constrains then go 11, otherwise go 3

Step11: stop

By Removing choose nodes from the network topology, you can avoid duplication so as to avoid loops formation.

C. Fitness function

Fitness function is defined as follows:

$$f_i = \frac{1}{\sum_{k=0}^{n-1} \sum_{j=Loc(m_k)}^{Loc(m_{k+1})} c_{chr_i(j),chr_i(j+1)} \times I_{chr_i(j),chr_i(j+1)}} \quad (9)$$

Where fi represents the fitness function value of ith chromosomal, Loc(m_i) represents the position of node mi in the chromosome, $Chr_i(j)$ represents the jth gene of ith chromosomes, C is a weight of links. You can find the fitness function increases when the weight of link changes small. there are several calculating method of the weight , in the paper Euclid distance between two nodes is used as the weight , The weight value of entire link is the sum of weight of links between nodes.

D. Selection

In the paper rank- based fitness assignment is used, in the method populations are sorted according to objective value. Individual fitness values only depends on its position of the population, and do not an actual objective value. Blickle T's research [6] indicates that rank- based fitness values assignment overcomes the measure problem that proportional fitness assignment causes x individuals in a population is sorted by fitness function value in ascending order, here is probability distribution:

$$p_i = \frac{2i}{x(x+1)}, \quad 1 \leq i \leq x \qquad (10)$$

Therefore, the optimal population is copied to next generation with the maximum probability 2/(x+1).

In the paper stochastic universal sampling is used, This method provides zero deviation and minimum individual extensions. If nSelect is the number of selected individuals, in accordance with equidistance to choose individuals, the distance between chosen pointers is 1 / nSelect, The position of first pointer is determined by the uniform random number in [0,1 / nSelect], shown as in Figure 3.

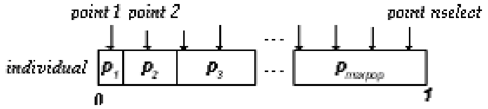

Figure 3. Stochastic universal sampling

E. Crossover

Crossover checks the current solutions in order to find better ones. Crossover in the routing problems plays the role of exchanging each partial route of two chosen chromosomes in such a manner that the offspring produced by the crossover will only be one route. In the paper a single node's crossover is employed, we select a same intermediate node in two chromosomes to implement intercross, the information of the specified nodes do not be changed in the way. The process is described as follows:

Step1: Select two chromosomes *chr1* and *chr2* from population *pop*
Step2: Select a same node *C* from chr1 and chr2 $C \neq D$
Step3 : If $C \notin M$ go step 6
Step4: Select a successor node *LC* of the node *C* from *M*;
Step5: Swap two chromosomal substrings from chromosomes *C* and *LC*, then generate a new couple of chromosomes, go 8

Step6: Select $m \in M \cap m \in chr1 \cap m \in chr2$ and *C* is in front of *m*
Step7: Swap two chromosomal substrings from chromosomes *C* to *m* ,
Step8:Calculating the fitness values of the couple of new chromosome, if it do not satisfy cross probability then go 1
Step9: stop

F. Mutation

For Solving shortest paths problem, when choosing mutant nodes, we can not employ random algorithm to avoid generating infeasible solution. In the paper the algorithm carries out complete mutation on the partial path, i.e. the algorithm chooses a mutant node from two designated nodes, carries out complete mutation on all path from mutant node to designated node, it can guarantee individuals is still available after mutating. The mutant process is described as follows:

Step1: Select a chromosome *chr* according to the given mutant probability from populations
Step2: Select a mutant node *C* from *Chr* randomly
Step3 : if $C \notin M$ go step 5

Step4: Select m, $m \in M$, and *m* is a successor node of the node *C* in *M*
Step5: Delete nodes that locates in the front of node *C* and behind node m from *G*
Step6: Select a successor node *LC* of node *C* from *G,* push(*chr*, LC)*, and, delete *LC* from *G*
Step7: If *LC*=*m* then use *chr** to replace chromosomal segment of chr from node *C* to node *LC*.
Step8: If mutant probability can not been satisfied, then go 1, otherwise algorithm stop.

G. Repair function

Genetic operation may cause infeasible chromosomes because of loop path. In the paper authors use a repair function that avoids the formation of loop and remove infeasible chromosomes. The repair function is shown below:

Step1: Select a chromosome *chr* from new popula-tions
Step2: Select a node *C* of chromosome *Chr* in order
Step3: When searching from the rear to front of a chromosome, If node *C** exists and $C^* \in chr \cap C^* = C$, the loop exists.
Step4: Select m, if $m \in M$ and *m* is located between chromosomes *C* and *C**, it is infeasible, otherwise go 1
Step5: Delete all nodes from node *C* to node *C** in order to remove loop.
Step6: Recalculate the fitness value of chromosome chr
Step7: If all chromosomes are repaired, the algorithm stops, otherwise go s*etp1*.

H. Procedure of the algorithm

Base on the analysis above, the procedure for the hierarchic genetic algorithm can be summarized as the following:

Step1: Initialize information database of network topology and designated multi-nodes information
Step2: Generate the initial populations according to the topology information database
Step3: Calculate the fitness of the initial population
Step4: Divided the initial populations up according to the designated multi-nodes
Step5: Select individual by rank- based fitness assignment
Step6: Perform crossover according to the method above
Step7: Analyse the individual after crossover operation , remove infeasible chromosomes by use of a repair function
Step8: Perform mutation according to the method above
Step9: Generate new populations
Step10: If stop criteria met，output result ,and the algorithm would be finished；otherwise，go to step 3

IV. SIMULATING EXPERIMENT AND PERFORMANCE ANALYSIS

In order to analyze the performances and effects of the hierarchic genetic algorithm, the authors have randomly selected a network topology to perform the simulation. The random network was generated by the method that Waxman proposed [8], and $\alpha=0.15, \beta=0.2$,adjacency matrix is used to represent the network topology. We randomly generated 100 node's network environment by MATLAB7.1, we assign link cost adopts Euclid distance, the population size is equal to 33, crossover probability is 0.8, mutation probability is 0.05, the number of sub-populations are related to the given nodes. For a connected map, a link must exists between any two nodes, therefore, firstly we give two specified nodes that are the source node 1 and the destination node 100 to verify the algorithm convergence. First, we calculate the shortest path by Dijkstra algorithm, the optimal solutions is 913.76, the shortest path is <1,4,14,30,100>. We can obtain the optimal solution at 58 iteration by the hierarchic GA, shown in figure 4.

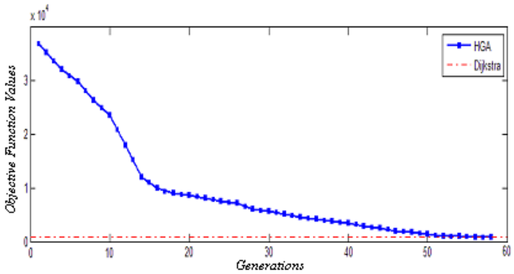

Figure 4. Comparison of Dijkstra and hierarchic GA algorithm

We add specified intermediate node 50, nodes that must pass are nodes 1, 50, 100. All network nodes can be divided into two parts, and the population can be divided into two sub-populations. We can obtain the optimal solution at 52 iteration by hierarchic GA, the link cost is 2085.8, and the shortest path that passes nodes 1,50,100 is <**1**,4,14,**50**,82,19,30,**100**>. We can calculate the shortest path by Dijkstra algorithm, that is <**1**,4,14,**50**>, and remove all nodes of the path except the destination node from the topological graph, then calculate the shortest path that passes nodes 50, 100. It is <**50**,82,19,30,**100**>, therefore, the shortest path that passes nodes 1,50,100 is <**1**,4,14,**50**,82,19,30,**100**>, and two algorithms obtain the same results.

We simulated experiments many times and the results are tabulated in Table 1, table 1 has illustrated a failed probability. Failed probability illustrates genetic algorithm does not converge to the optimal solution, but the algorithm will give a suboptimal solution, it can also satisfy the constraint conditions. The results depicted in table I show when the number of designated nodes is gradually increasing, the running time is increasing fast in Dijkstra's algorithm, while the running time is increasing smoothly in hierarchic GA. Simulations show that the hierarchic algorithm exhibits a optimal solution and a much higher rate of convergence than Dijkstra's algorithm. The hierarchic GA is relatively independent of network sizes and topologies.

TABLE I. RUNNING RESULTS OF 100 NODES

Number of designed nodes	***Optimal solution***		***Running time comparison（ms）Dijkstra / Hierarchic GA***	***Failed probability (GA)***
	Dijkstra	*Hierarchic GA (iteration)*		
2	913.76	913.76（58）	28.2/94.3	0.24
3	2085.8	2085.8（52）	79.4/104.2	0.24
4	3222.5	3222.5（52）	166.4/93.1	0.20
5	5956.6	5956.6（41）	266.9/110.6	0.18

V. CONCLUSION

The paper proposed a hierarchic genetic algorithm used for calculating the shortest path problem via designated multi-nodes routing of wireless networks, which dijkstra needs more time for solving the problem. It is relatively independent of network sizes and topologies. Experimental results show that the algorithm can solve effectively the routing problem passing many designed nodes which is used to separate the feasible region to reduce the complexity of problem and heighten the efficiency of genetic algorithm, it is favorable to distributing execution. By using this routing method via designated multi-nodes the packets can get through the established security channel and avoid entering non-trusted areas.

REFERENCES

[1] Perkins, "IP Mobility Support for IPv4," RFC 3220, January 2002.

[2] Chang Wook Ahn, and R. S. Ramakishna, "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations", IEEE Transactions on Evolutionary Computation, vol. 6 : 566-579 2002.

[3] Jianjun Yang , "Optimum Routing Based on Genetic Glgorithm in Mobile IP", Journal of Zhejiang University, Vol.38 No.11 1426-1430, 2004.

[4] G. Bao, Z. Yuan, Q. Zheng and X. Chen, "A Novel Genetic Algorithm to Optimize QoS Multicast Routing," Lecture Notes in Control and Information Sciences, Vol. 344, pp. 150-157, 2006.

[5] Z. Xiawei, C. Changjia, and Z. Gang, "A Genetic Algorithm for Multicasting Routing Problem",. International Conference Communication Technology Proceedings, WCC-ICCT 2000, pp. 1248-1253, 2000.

[6] T. Blickle, and Thiele L, " A Comparison of Selection Schemes Used in the Genetic Algorithms", TIK- Report, Swiss Federal Institute of Technology (ETH), 1995

[7] Wenyi Wang, Guangjun Qin, and Ruoyu Wang, "Research on Genetic Algorithm Based on Particle Swarm Algrithm",Computer Science,China,vol.24,no.10,2007,pp.34-38.

[8] D. Waxman, " Routing of Multipoint Connections", IEEE Journal on Selected Area in Communica tions, 1988, 6(6): 1617-1622.