PWT 2007
Poznańskie Warsztaty Telekomunikacyjne
Poznań 6 - 7 grudnia 2007

POZNAN UNIVERSITY OF TECHNOLOGY ACADEMIC JOURNALS

THE IMPLEMENTATION OF GENETIC MULTICAST ALGORITHMS IN AD-HOC NETWORKS

Piotr Szymaniak
Maciej Piechowiak
Piotr Zwierzykowski
Poznań University of Technology

Abstract

Realtime multimedia applications in computer networks require techniques of providing data to a group of users at the same time. Building efficient multicast trees is a challenge for multicast routing protocols and algorithms. The paper focuses on the implementation of genetic multicast algorithms particularly in ad-hoc networks. It also proposes some approaches in ad-hoc networks modelling. The paper is the first stage of author's research work with ad-hoc networks.

I. Introduction

The multicast routing problem is known in literature as the Minimum Steiner Tree Problem, which has been shown to be $\mathcal{NP}$-complete for one and more link parameters (metrics) [1]. Literature presents many efficient heuristic solutions solving this problem in polynomial time [2]–[5].

In recent years, the genetic algorithms (GA) have become the field of interest for many researchers, because of robust and efficient search in complex space. They particurarly focus on adaptation of genetic operators in multicast routing algorithms [6]–[9].

The advantages of genetic algorithms are commonly known. GAs code solutions as bit strings, so difficult problems can be easily solve using long strings. Genetic operations, such as crossover and mutation, are easy to implement. The main advantage of GA is the parallelization of code – with a pool of chromosomes, GAs search the solution space at different corners in parallel. Randomized genetic operations, such as mutation, can keep the search from being trapped by local-optima [7].

The common goal of the abovementioned multicast algorithms (both heuristics and GAs) is their implementations in networks that represent real Internet topologies (on domain level and AS-level). This area is well studied and influence of network topology on the efficiency of QoS multicast heuristic algorithms has been examined [10], [11].

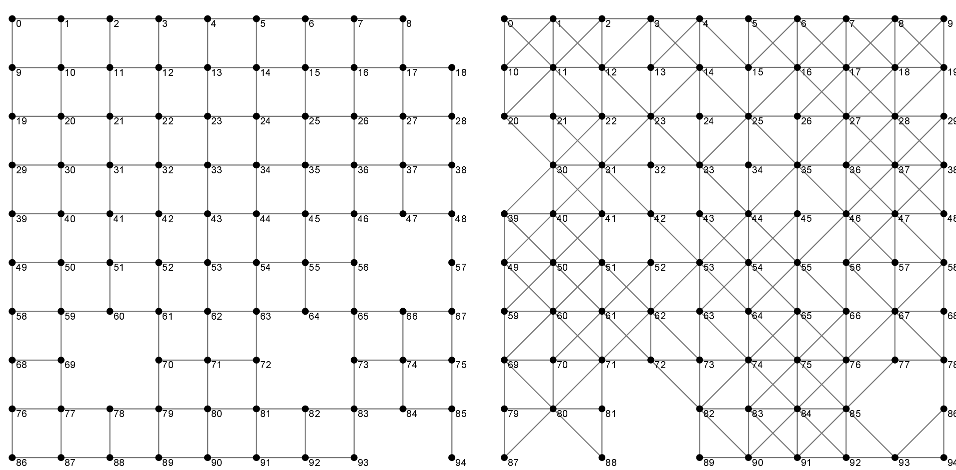
On account of increased interest in the transmission in wireless networks there is also a need to analyse the efficiency of routing algorithms dedicated mainly for ad-hoc networks. These are wireless networks of decentralized structure where mobile nodes can function both as a client (end terminal) and a router.

The models of network structures reflecting the real topologies of ad-hoc networks have not been studied in great detail. However, Li and Yu carried out some research on the properties of such networks [12]. Therefore, there is a need to create a tool which will allow to generate ad-hoc networks for algorithm analysis, which allow to compute them in similar topologies.

The article discusses the effectiveness of GA algorithms in ad-hoc networks. Chapter 2 presents a network model. Chapter 3 is an overview of implemented algorithm. Chapter 4 includes the results of the simulation of the implemented algorithms along with their interpretation. The final chapter sums up the discussion.

II. Network model

It is assumed in the article that the network on which the algorithms will be tested is represented by a coherent directed graph $G = (V, E)$. V is a set of nodes and E is a set of directed edges connecting the nodes of the graph (it represents the links of the network). Each edge of the graph $e \in E$ is coupled with the cost metric $C(e)$ and the delay metric $D(e)$.



(a) (b)

Fig. 1. Example grid networks from ad-hoc generator: regular grid (the number of the nodes in the network $n = 95$, the number of edges $k = 164$, average node degree $D_{av} = 3.45$ [11]) (a) and grid where nodes have random range value ($n = 95$, $k = 273$, $D_{av} = 5.75$) (b)

The cost metric describes the general cost of setting up a connection with the execution of a given link. The cost function can be composed of: the flow capacity of the link, capacity of the buffers, error rate (link quality), etc. The delay metric represents the delay in the propagation of the signal carried by a given link.

The generator we have proposed constructs graphs modelling grid networks with a maximum coverage of the plane. This means that it is possible to arrange the nodes on the plane for the network with a low number of nodes as well as bigger networks (links are shorter than).

If $\sqrt{n}$ is not an integer value, a regular grid cannot be achieved. The algorithm implemented in the generator takes the closest integer value bigger than $\sqrt{n}$ to construct regular grid with $N > n$ nodes ($N = \lceil\sqrt{n}\rceil^2$). In the next step, $N - n$ random nodes are removed. Fig. 1 shows a regular grid network with 5 nodes removed.

The process of interconnecting nodes consist in a random assigning of range values for each node (corresponding to the transmitter power). For the range value equal 1 the closest neighbours are connected (left, right, up and bottom) – Fig. 1(a). For the range value equal 2 also diagonal nodes are linked. The range value can be randomly generated for each node (Fig. 1(b)).

III. Description of the implemented algorithm

The implemented algorithm is based on solutions described in [7] and [8]. However, the genres of genetic operations were chosen arbitrarily. The algorithm is constructed as follows:

Algorithm 1 Implementation of GA

```
for each multicast destination do
    generate a set of all possible routes fulfilling delay constraint from source to destination
end for
generate random set of individuals
for generations do
    perform crossovers with crossover probability
    perform mutations with mutation probability
    if any modified individual == any of the previous individuals then
        substitute modified individual with one randomly generated
    end if
    perform selection of individuals
end for
```

A. Generation of sets of routes

The process of generating a set of paths from a source node to each destination node is realized using an exhaustive in-depth-search method. It searches all possible routes that fulfill the delay limit. When a valid receiving node is hit, a path is copied into the corresponding set.

Searching for and storing all possible routes within a given delay limit is both time-consuming and memory-demanding. Second issue has been overcome. Each set, storing paths to one of the receiving nodes, is sorted in order to easily employ a RAM usage limiting mechanism. This mechanism limits the number of paths stored in each set, allowing only the best paths (in the unicast sense) to form organisms. The paths in each paths set are associated with a unique number.

B. Data representation and generation of members constituting first generation

Each individual is defined by the genetic code. Each gene contains a number of paths from the source to one of the destinations. The length of genetic codes is equal to the number of multicast receiving nodes. Each genetic code is created in a drawing process, which uses a linear congruential generator. Each gene is associated with a random number from 1 to the $number\ of\ paths\ to\ coresponding\ destination$. Multicast tree is formed by concatenating unicast paths. Since all the unicast paths are valid, the multicast graph is also valid. However, there is no explicit mechanism assuring that the multicast graph is a tree (loops are not avoided). It is hoped that the redundant loops will be eliminated in the evolution process by promotion of the cheaper solutions.

C. Crossover

Crossover is used to exchange fragments of genetic codes. The described implementation utilizes uniform crossover. First, with a given crossover probability, a decision is made whether to perform a crossover. Then, if the answer is true, two individuals are picked at random from a pool. The decision whether to exchange a gene or not is made for each corresponding pair of genes separately. Newly formed individuals are stored in another set in order to separate generations.

D. Mutation

Mutation allows to escape from local optimums. The implemented algorithm uses a typical mutation, where a gene is substituted with a random number in the valid range. First, a decision is made whether to perform the mutation. Then, if the answer is true, an individual is picked at random from a pool. Next, a mutating gene is decided and finally a new random value is substituted. Newly formed individuals are stored in another set in order to separate generations.

E. Elimination of identical individuals

In order to avoid degeneration of individuals to a set of identical solutions, which has been observed in test runs, a mechanism of elimination of identical individuals has been implemented. After crossover and mutation, if any individual in the offsprings set is equal to any individual in the parents set, the offspring individual is substituted with a randomly generated one. This allows the algorithm to search the solutions space more comprehensively and further assures that the algorithm will not fail in the local optimum.

F. Selection

After applying genetic operations each newly created individual is being evaluated. After that a selection process takes place. In this algorithm a tournament selections has been applied. As long as the quantity of population is bigger than expected, two random individuals are drawn from the population and costs of multicast graphs they resemble are compared. The cheaper multicast graph is preserved, while the more expensive one is discarded.

IV. Performance evaluation

The proposed genetic algorithm has been compared to two classical heuristic algorithms: KPP [3] and CSPT [13]. In the trials, the crossover probability and mutation probability was set to 1 (number of crossovers and mutations equaled number of individuals). The number of paths to each multicast node, generated by the in-depth-search procedure, was limited to 1000.

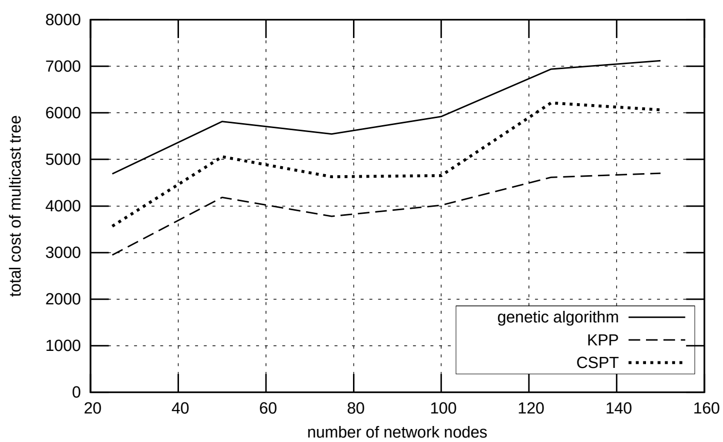


Fig. 2. Total cost of multicast tree versus the number of network nodes n (number of receiving nodes $m = 10$, maximum initial path delay $\Delta = 14$, number of individuals 100, number of generation 300)

KPP proved to be unbeatable throughout all trials. However, a prolonged evolution process with more individuals taking part in the evolution process results in generating better solutions than the ones obtained with CSPT. However, the execution time of studied genetic algorithm was always incomparably longer than the execution time of KPP.

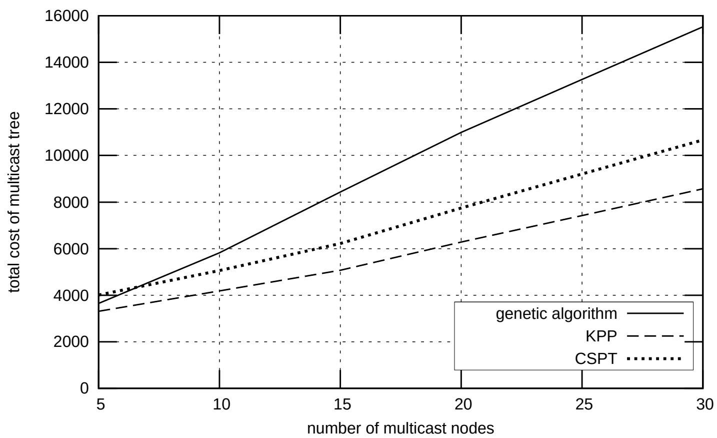


Fig. 3. Total cost of multicast tree versus number of receiving nodes m (number of network nodes $n = 50$, maximum initial path delay $\Delta = 14$, number of individuals 100, number of generation 300)

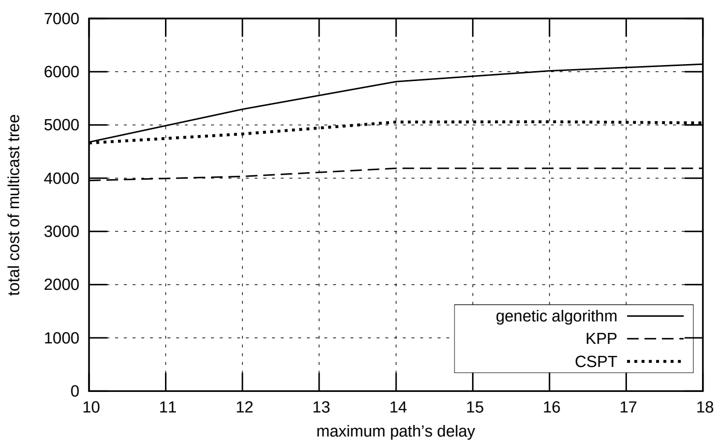


Fig. 4. Total cost of multicast tree versus maximum paths' delay constrain Δ (number of network nodes $n = 50$, number of receiving nodes $m = 10$, number of individuals 100, number of generation 300)

The total cost of multicast tree versus Δ presented in Fig. 4 requires a comment. The method used to generate networks and choose receiving nodes was random and there was no mechanism assuring that the network is routable within a specified maximum paths' delay. Because of that, the actual number of networks taking part in each simulation varied. Since weakening of the constrain entailed the ability to find feasible paths to each receiving node in more networks, more networks with higher average multicast tree cost supplied the set of solutions.

Significant increase of total cost of multicast tree occurring with the increase

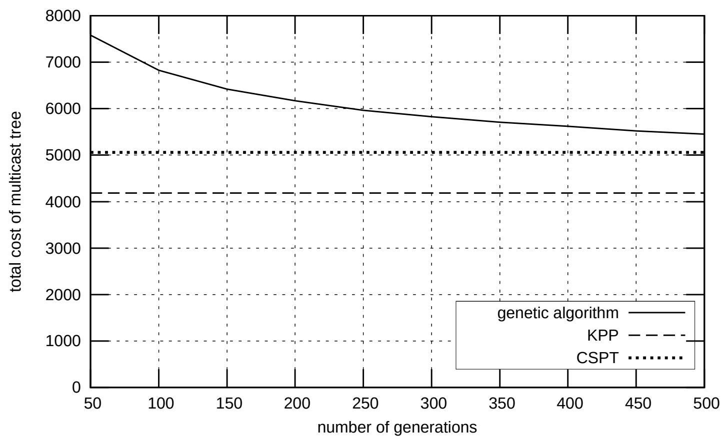


Fig. 5. Total cost of multicast tree versus length of evolution process (number of network nodes $n = 50$, number of receiving nodes $m = 10$, maximum initial path delay $\Delta = 14$, number of individuals 100)

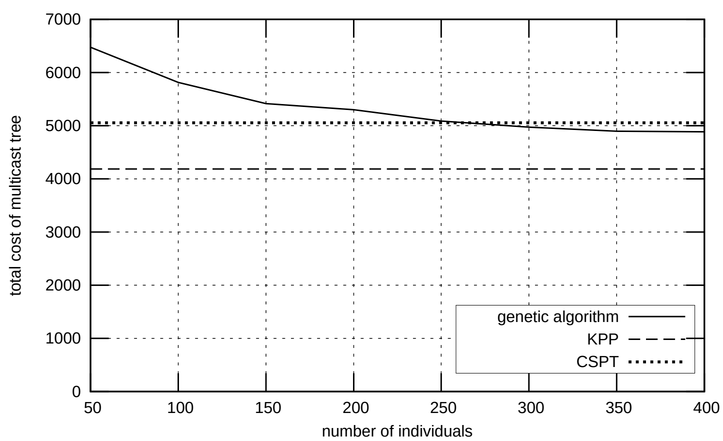


Fig. 6. Total cost of multicast tree versus number of individuals (number of network nodes $n = 50$, number of receiving nodes $m = 10$, maximum initial path delay $\Delta = 14$, number of generation 300)

of the number of receiving nodes and with the increase of the number of maximum paths' delay proves that the length of the evolution process has to be strongly correlated with the number of receiving nodes and number of paths to each receiving node the algorithm has to choose from to allow the algorithm to work out mature solutions.

V. Summary and direction of further research

Tests have proven the genetic algorithm to perform worse than traditional non-genetic solutions. Furthermore, algorithm exhibits huge computation complexity

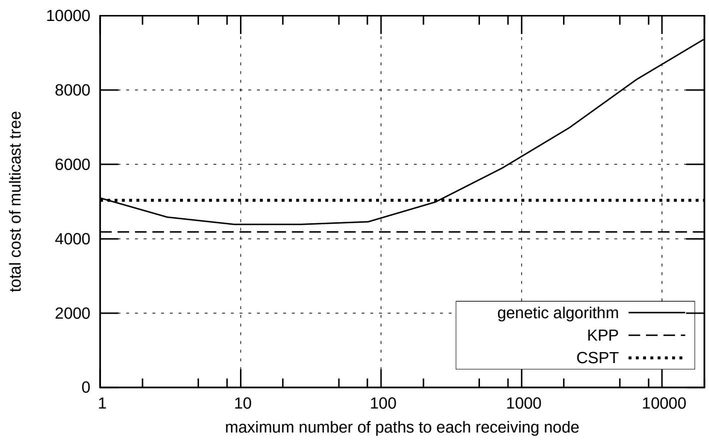


Fig. 7. Total cost of multicast tree versus paths to each receiving node the genetic algorithm has to choose from (number of network nodes $n = 50$, number of receiving nodes $m = 10$, maximum initial path delay $\Delta = 20$, number of individuals 100, number of generation 300)

resulting in long execution times, incomparably longer than CSPT and even KPP. This inspires the author to further improve the efficiency of the proposed solution. One of the possible improvements is related to implemented paths' search mechanism: time consuming exhaustive in-depth-search. Great improvement is expected by implementation of modified Dijkstra algorithm instead, which would enable practical exploration of thousand-nodes' networks. Furthermore, as shown in the Fig. 7, a limitation on the number of paths to each multicast node the algorithm can choose from gives significant positive impact on the cost of the final multicast tree.

REFERENCES

[1] R. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations*, pp. 85–104, 1972.

[2] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," *Acta Informatica*, no. 15, pp. 141–145, 1981.

[3] V. P. Kompella, J. Pasquale, and G. C. Polyzos, "Multicasting for Multimedia Applications," in *INFOCOM*, 1992, pp. 2078–2085.

[4] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, "A source-based algorithm for delay-constrained minimum-cost multicasting," in *INFOCOM '95: Proceedings of the Fourteenth Annual Joint Conference of the IEEE Computer and Communication Societies (Vol. 1)-Volume*. IEEE Computer Society, 1995, p. 377.

[5] Q. Sun and H. Langendoerfer, "Efficient Multicast Routing for Delay-Sensitive Applications," in *Proceedings of the 2-nd Workshop on Protocols for Multimedia Systems (PROMS'95)*, October 1995, pp. 452–458.

[6] R. Hwang, W. Do, and S. Yang, "Multicast routing based on genetic algorithms," *Journal of Information Science and Engineering*, vol. 16, pp. 885–901, 2000.

[7] R. Bhattacharya, P. Venkateswaran, S. Sanyal, and R. Nandi, "Genetic Algorithm Based Efficient Routing Scheme For Multicast Networks," *IEEE International Conference on Personal Wireless Communications 2005*, pp. 500–504, 2005.

[8] M. Hamdan and M. El-Hawary, "Multicast Routing with Delay and Delay Variation Constraints Using Genetic Algorithm," *Canadian Conference on Electrical and Computer Engineering*, pp. 2363–2366, 2004.

[9] J. Crichigno and B. Baran, "Multiobjective Multicast Routing Algorithm," *LNCS*, no. 3124, pp. 1029–1034, 2004.

[10] P. Zwierzykowski and M. Piechowiak, "Performance of Fast Multicast Algorithms in Real Networks," in *Proceedings of EUROCON 2007 The International Conference on: Computer as a tool*, Warsaw, Poland, September 2007, pp. 956–961.

[11] ——, "Efficiency Analysis of Multicast Routing Algorithms in Large Networks," in *Proceedings of The Third International Conference on Networking and Services ICNS 2007*, Athens, Greece, June 2007, pp. 101–106.

[12] H. Li and D. Yu, "A Statistical Study of Neighbor Node Properties in Ad Hoc Network," in *International Conference on Parallel Processing Workshops*, 2002, pp. 103–108.

[13] J. S. Crawford and A. G. Waters, "Heuristics for ATM Multicast Routing," *Proceedings of 6th IFIP Workshop on Performance Modeling and Evaluation of ATM Networks*, pp. 5/1–5/18, July 1998.