GENETIC ALGORITHMS FOR NETWORK CODING OPTIMIZATION

Zsuzsanna-Ilona KISS Zsolt Alfred POLGAR Mircea GIURGIU
Technical University of Cluj-Napoca, Communications Department
28 Memorandumului St., 400114 Cluj-Napoca, Romania, Tel: +40264401906
E-mail: Zsuzsanna.Kiss@com.utcluj.ro

Abstract: The optimization of networks which implement Network Coding techniques is necessary for efficient resource utilization. If the data is encoded only in a subset of nodes, selected in the appropriate way, the complexity of the coding operations could be decreased, while ensuring the imposed performances. Finding the minimum number of encoding nodes is a NP-hard problem, but genetic algorithms could offer a good solution to this problem. In this paper we investigate by computer simulations the performances of a genetic algorithm adapted to the network optimization problem and we try to find the combination of parameters which ensures low optimization time.
Keywords: *Network Coding, genetic algorithms, network optimization, node selection.*

I. INTRODUCTION

The Network Coding (NC) concept was firstly introduced for satellite communication networks in [1] at the turn of the millennium, and then fully developed in [2], where the term "Network Coding" first appeared. These coding techniques were proposed to increase the throughput in communication networks in compensation of the transfer rate limitations caused by congestions.

NC allows intermediary nodes to perform mathematical operations on the packets received on the incoming links, instead of simply store-and-forward these packets, being an alternative to classical routing. In [2] it was shown that in multicast networks the optimal transmission rate can be achieved using NC, and even unicast protocols perform better with NC than with classical routing.

A significant part of the theoretical studies regarding NC assume that coding operations take place at every intermediary node in the network. The disadvantage of this approach is that the necessary mathematical operations for coding at the intermediary nodes and decoding at the destinations nodes will increase the computational complexity of NC operations and will require a significant amount of additional resources. In many cases it is sufficient that only a set of intermediary nodes realize coding operations, without decreasing the throughput of the network, as it was shown in [3]. This will reduce the number of operations performed by the nodes, thus the complexity and the costs of NC operations. The problem is to find the minimum number of coding nodes, nodes that realize coding operations,, while ensuring a maximal throughput of the network. This problem is NP-hard and even close approximations of this problem are NP-hard. Considering the complexity of this problem there are only a few approaches for minimizing the necessary resources and the cost for NC.

Genetic algorithms may be a solution for the optimization problem of the networks where NC operations are performed [3], but these optimization algorithms are also computationally expensive and their theoretical performances are hard to be evaluated.

The paper proposes an evaluation of the genetic algorithms based network optimization process complexity by employing computer simulations. The relation between the number of iterations / optimization time and the different parameters of the genetic algorithms is analyzed. This study is important especially in the network optimization problem in discussions, because the number of iterations has a strong influence on the amount of necessary signaling overhead, while the optimization time has to be significantly smaller than the time periods during which the network topology remains unchanged.

This paper is organized as follows: Section II presents shortly the basic aspects related to NC techniques; Section III describes the optimization process of the network performing NC using genetic algorithms. Simulation results are presented in Section IV and Section V concludes the paper.

II. NETWORK CODING TECHNIQUES

In order to perform NC the communication network has to be modeled as a directed graph, in which every edge represents a communication link and every vertex a node from the network. In order to simplify the model we consider lossless links and every link is modeled by several elementary links with unit capacity, capable of transferring only one symbol in one time unit.

In Figure 1 the well-known butterfly network [2] is presented, as a basic example. The source nodes R1 and R2 transmit two bits, *b1* and *b2*, to destination nodes R3 and R4. If node R5 is allowed to mix the incoming bits from the source nodes by a simple bitwise XOR operation and forward the resulting bit, $b1 \oplus b2$, to R6, then every link in the network will be used exactly once, and both destinations will receive simultaneously the two bits.

Manuscript received October 29, 2010; revised December 9, 2010

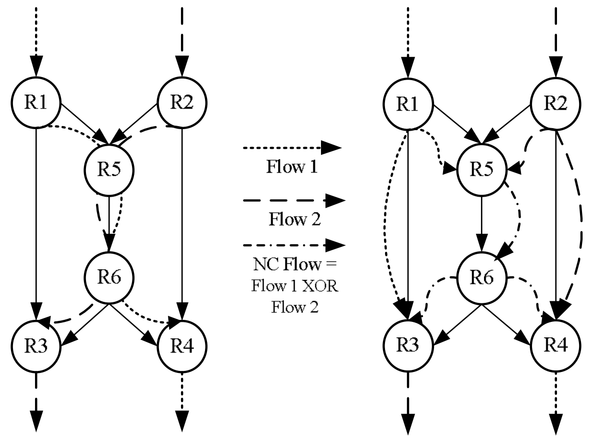

Figure 1. The butterfly network.

Destination node R3 receives bits *b1* and *b1*⊕*b2*, so it is able to decode bit *b2*. Similarly, node R4 is also able to decode bit *b1* from the received bits, *b2* and *b1*⊕*b2*. If node R5 simply forwards the incoming bits, without mixing them, at least one link has to be used twice, and the throughput of the two sources will decrease.

The coding operations performed in the considered example are representing deterministic linear NC operations, when for each node the coding coefficients are imposed. The drawback of such a solution consists in the increased signaling overhead required by the distribution of the coding coefficients among the nodes performing NC operations.

Random Network Coding, introduced in [4, 5], is an alternative solution which allows to each node to chose independently the coding coefficients, by generating randomly these coefficient. The coding operations are realized on symbols represented by vectors of length u, seen as elements of the finite field $F(2^u)$. The coded symbol $Y(j)$ transmitted on an outgoing link j of node v is a linear combination of the information symbols, X_i, generated at node v = tail(j) and of the data or coded symbols, $Y(l)$, received by node v on it's incoming links l, as it is depicted in Figure 2. For $Y(j)$ we can write the following equation:

$$Y(j) = \sum_{\{i:X_i \text{ generated at } v\}} a_{i,j} X_i + \sum_{\{l:head(l)=v\}} f_{l,j} Y(l) \quad (1)$$

where $a_{i,j}, f_{l,j}$ represent the coding coefficients used in node v for the locally generated respectively for the received data or coded symbols [4].

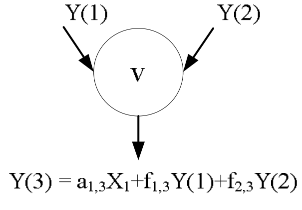

Figure 2. The coding process at node v.

The received data at the destination node, β, is also a linear combination of the data received on the input links:

$$Z(\beta,i) = \sum_{\{l:head(l)=\beta\}} b_{\beta_{i,l}} Y(l) \quad (2)$$

where $b_{\beta_{i,l}}$ represent the coding coefficients used in the destination node. The coefficients $\{a_{i,j}, f_{l,j}, b_{\beta_{i,l}} \in F_{2^u}\}$ form the matrices $A = (a_{i,j})$ and $B_\beta = (b_{\beta_{i,l}})$ of dimension $r \times w$, where r is the transmission rate in symbols/s and w is the total number of links in the network, and matrix $F = (f_{l,j})$ of dimension $w \times w$. The structure of these matrices depends on the topology of the network, as described in [4, 6]. The collection of matrices (A, F, B) represents a linear network code, including both the coding operations and the topology of the network where these operations are performed. The coding problem has solution if and only if the transfer matrix M of the network defined by (3) is nonsingular.

$$M = A \cdot G \cdot B_\beta^T \; ; G = (1 - F)^{-1} \quad (3)$$

The coding coefficients are selected randomly from the elements of the finite field $F(2^u)$. If the field size is sufficiently large, at the destination node, β, the coefficients $b_{\beta_{i,l}}$ can be selected such that the transfer matrix M is nonsingular. The increased size of the finite field, necessary to ensure coding solution, represents the main drawback of random NC operations.

III. GENETIC ALGORITHM BASED NETWORK OPTIMIZATION

In a multicast scenario it is not always necessary that coding operations to be realized at every node of the network. There are nodes which can simply forward the received packets without decreasing the throughput of the network. To minimize the cost of the NC operations and to optimize resources needed for NC, it is necessary to find the minimum number of coding nodes and the minimum number of links on which coded packets are transmitted, such that a given transmission rate is achievable. A simple example is presented in Figure 3, where the source node (node 1) transmits simultaneously two packets, *a* and *b*, to the destination nodes (node 9, 10 and 11). If NC is not used, the simultaneous reception of both packets by all the destination nodes is not possible. If NC is realized in its simplest form (bitwise XOR) at the intermediate nodes 5 and 6, without any optimization, the destination nodes 9, 10 and 11 receive the following packets: *a*, *a*⊕*b*; *a*⊕*b*, *a*⊕*b*; and *a*⊕*b*, *b* respectively. It can be seen that every destination node will be able to decode the received coded packets. The transmission rate in this case is 2 packets / time unit. If the network is optimized, the same transmission rate (2 packets/time unit) can be achieved, but it is sufficient that only one of the nodes 5 and 6 mixes the received packets.

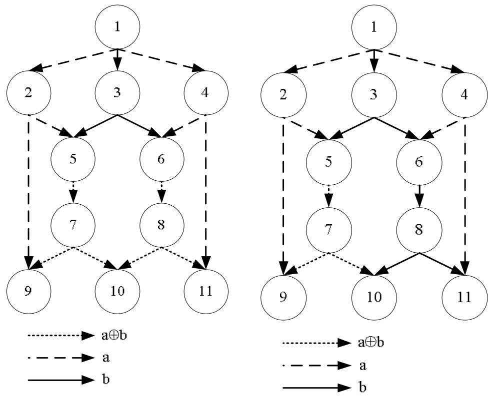

Figure 3. Network Coding cost optimization

Supposing that node 5 performs the coding operations on the received packets, the destination nodes 9, 10 and 11 receive the following packets: *a*, *a*⊕*b*; *a*⊕*b*, *b*; and *a*, *b* respectively. Both data packets, *a* and *b*, can be recovered in all destination nodes.

Finding the minimum number of coding nodes in a more complex network topology is not a simple task (like in the previous example) and a possible solution could be to adapt genetic algorithms to this problem. A genetic algorithm operates on a set of possible solutions, called population, which improve sequentially using different mechanisms inspired from biological evolution.

A particular variant of genetic algorithms which can be easily used to solve the optimization problem of networks employing NC is presented in [3]. In this algorithm the chromosomes of the initial population are defined as follows: for each pair of incoming link $i\in\{1, \ldots, d_{in}\}$, $d_{in}\geq 2$, and outgoing link $j\in\{1, \ldots, d_{out}\}$, $d_{out}\geq 1$, of an intermediate node, a binary value $a_{i,j}$ is associated randomly. This value is 1 if the information received on the incoming link *i* contributes to the information sent on the outgoing link *j*, and is 0 otherwise. For the outgoing link *j* ($j\in\{1, \ldots, d_{out}\}$), the set of associated binary values $a_j=(a_{i,j})_{i\in\{1, \ldots, din\}}$ forms the coding vector and every chromosome is defined as the collection of these coding vectors. Thus, a given chromosome indicates which inputs will contribute to which outputs at every intermediate node, as presented in Figure 4.

If $d_{in}(v)$ is the number of incoming links of node *v*, and $d_{out}(v)$ is the number of outgoing links of the same node, then the length of a chromosome can be defined as:

$$m = \sum_{v\in V} d_{in}(v) d_{out}(v)\,, \qquad (4)$$

where *V* is the set of intermediate nodes. Every gene in the chromosome represents an incoming-outgoing link pair from the network.

The size of a population is limited usually to a few hundreds of chromosomes and it is possible that none of these chromosomes is feasible, due to their random generation. To avoid such a situation, if the size of the population is *N*, only *N-1* chromosomes are generated randomly, and the chromosome containing value 1 at every position will be added to the population. This chromosome is feasible, but it represents a non-optimal solution.

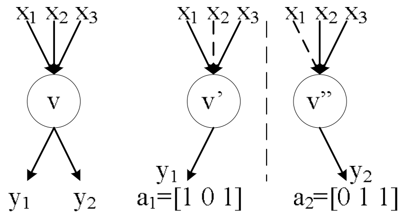

Figure 4. Coding vectors

For the evaluation of the chromosomes a fitness function is defined. In every chromosome, the number of links which carry coded data is equal to the number of coding vectors which contain at least two values of 1. The number of coding nodes is equal to the number of nodes which have at least one output link carrying coded symbols. Using Random Network Coding it has to be checked if the transmission rate can be achieved after the data transmission is restricted in the way that a given chromosome specifies. The input links which have a 0 value assigned in a coding vector composing the chromosome are not used for data transmission and coding operations on the output link related to that coding vector. A chromosome is feasible if the transmission rate can be achieved according to the restriction given by the chromosome, otherwise it is not feasible. In other words a chromosome is feasible if the transfer matrix at every destination node is nonsingular. The fitness function according to [3] can be defined as:

$$F(z) = \begin{cases} number\ of\ coding\ nodes, & \text{if z is feasible} \\ \infty, & \text{if z is not feasible} \end{cases} \qquad (5)$$

From the current population a new population is created using three operations: selection, crossover and mutation.

The chromosomes intended to form the new population are selected based on their fitness values, chromosomes with better fitness value being selected with a higher probability. From the current population *M* chromosomes are selected randomly, and the one with the best fitness value is kept, where *M* is a parameter of the genetic algorithm called selection pressure. This selection method is repeated *N* times, where *N* is the size of the population.

The chromosomes selected for the new population undergo uniform crossover. The chromosome pairs are selected randomly, and they interchange part of their genes according to a given crossover probability. The chromosomes generated by the crossover process undergo a binary mutation, the bits of the chromosomes being inverted according to the mutation probability.

The new population obtained after crossover and mutation is evaluated using the fitness function. The best chromosome of the population (the chromosome with the smallest fitness value in our situation) is kept unchanged for the next generation, while the worst chromosome, having the largest fitness value, is eliminated. New populations are generated until the termination criterion is reached, meaning that the number of generations reaches the imposed limit. Another termination criterion can be adopted, which stops the iterations when in a given number of generations the fitness value of the best chromosome does not change significantly.

IV. EXPERIMENTAL RESULTS

If the optimization of the network is successful, the genetic algorithm will return the coding coefficients which will be used for the NC operations involved by the data transmission. Considering that the network topology can change relatively fast (it is the case of wireless networks), it is a subject of interest how fast the genetic algorithm finds the best solution. The influence of the parameters of the genetic algorithm – population size, selection pressure, crossover probability, mutation probability, maximum number of generations, finite field size – on the optimization time and on the necessary number of generations is evaluated in this section. For the evaluation we use a simulator implemented in Matlab and the Random Network Coding operations are realized in $GF(2^5)$ in all simulations.

The first test topology is the same as presented in Figure 3. If every link in the network has a capacity of 1 data unit / time unit, then without NC it is not possible to transmit simultaneously two packets to every destination node. If the intermediate nodes can mix the received packets – NC is used in the network – the maximum multicast transmission rate of 2 data units / unit time, can be achieved. Without optimization both intermediate nodes, 5 and 6, perform coding operations, although for achieving the maximum transfer rate it is sufficient that only one of the nodes 5 and 6 mixes the received information. For this topology the simulations were performed considering the following parameters of the genetic algorithms: selection pressure = 5, crossover probability = 20%, mutation probability = 2% and maximum number of generations = 150.

First we evaluate the performances depending on the population size. The simulations are done for different population sizes: {20, 30, 50, 70, 100, 150} and the results are presented in Figure 5. The optimization is successfully done in 7 generations and it does not depend on the size of the population, if the size is large enough. The time needed for successful optimization increases proportionally with the size of the population, for 50 chromosomes in the population the duration is 10.24 s, for 100 chromosomes it is 20.45 s and for 150 chromosomes it is 30.68 s, as presented in Figure 5; the optimization time intervals are valid only for the simulations performed. The operations of the genetic algorithm (selection, crossover, mutation and evaluation) take place for every chromosome from the population, so larger populations need higher number of operations, and the algorithm needs more time for optimization.

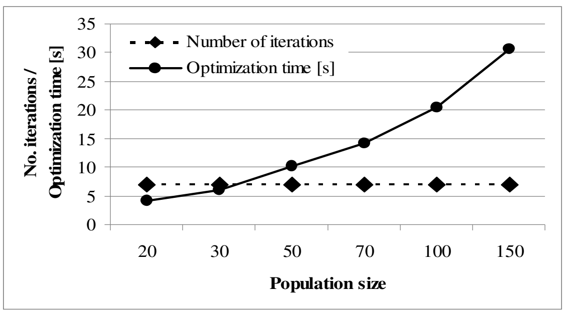

Figure 5. Number of iterations and optimization time vs. population size

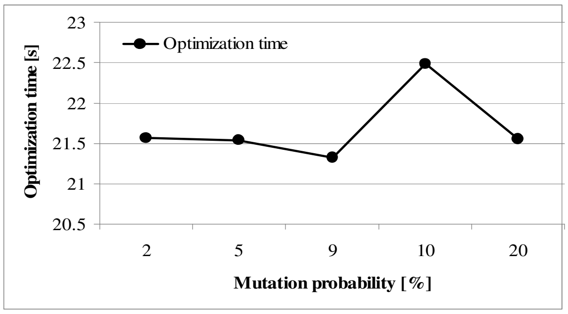

Figure 6.Optimization time vs. mutation probability

The population size depends on the test topology. If the population is too small, it is possible that the optimization will not be successful, because none of the randomly generated chromosomes offers a solution for the optimization problem. If the population size is too big, then the algorithm will take too long to find the solution.

Next we assign to the mutation probability the following values: {2%, 5%, 9%, 10%, 20%}; the population is composed of 100 chromosomes. The results presented in Figure 6 show us that for small values of mutation probability (<10%) the optimization time decreases with the increase of mutation probability. If the mutation probability is higher than 10%, the optimization time increases, because more inversions will take place in the chromosomes. The value of mutation probability depends on the application for which the genetic algorithm is used; for optimization of networks performing NC a mutation probability of 2% is recommended.

The second test topology is a modified butterfly topology, in which every link has a different non-elementary capacity. The capacities of the links are indicated in Figure 7.a. The maximum transmission rate which can be achieved in this network topology with NC is 3 data units / time unit. In Figure 7.b is presented the same network topology after optimization. Before optimization each link was used for data transmission and 6 links were used for the transmission of coded packets. After optimization 3 links will become unused and coded packets will be transmitted only on 2 links. The optimization process reduces the complexity of the coding and decoding operations through the simplification of the transfer matrix of the network and it frees some links, which can be used for other transmissions.

The termination criterion is that the fitness value of the best chromosome does not change in 5 generations.

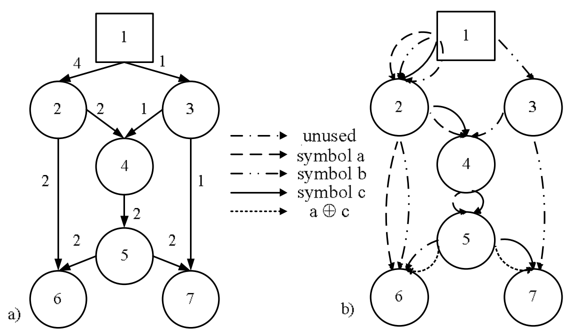

Figure 7. Modified butterfly test topology

For this topology the simulations were performed considering the following parameters of the genetic algorithm, if not specified otherwise: population size = 100, selection pressure = 5, crossover probability = 20%, mutation probability = 2% and maximum number of generations = 150.

Figure 8 presents the variation of the optimization time as a function of the population size. In the simulations performed the population size takes the following values: {20, 30, 50, 70, 100, 150, 200}. The optimization process is successful usually after 7 generations, regardless of the size of the population. The optimization time increases proportionally with the size of the population. For 50 chromosomes in the population the optimization is achieved in 13.8 s, for 100 chromosomes in 26.24 s, and for 150 chromosomes in 39.68 s; the time periods are valid only for the simulations performed. The optimization time of this topology is higher than that of the first test topology, because this topology has a higher number of links, which increases the dimension of matrix F (which contains the coding coefficients). Also the transmission rate is higher, which increases the number of operations required to compute the transfer matrix.

Next we evaluate the effect of the selection pressure on the optimization time and on the number of generations needed for successful optimization. The simulations were performed considering the following values for the selection pressure: {4, 5, 6, 7, 8, 9, 10, 15} and the results are presented in Figure 9. Neither the number of iterations nor the optimization time changes uniformly with the selection pressure. The optimization time is proportional to the number of iterations needed for successful optimization. The duration of one iteration step is approximately constant with the selection pressure.

Figure 10 shows the effect of the crossover probability on the optimization time. The crossover probability has the following values in the simulations performed: {5%, 10%, 15%, 20%, 25%, 30%, 40%, 50%, 100%}. The number of generations needed for successful optimization is constant with the crossover probability; 7 generations are enough for the optimization of the network. If the crossover probability is small (<20%) the optimization time increases with the increase of the crossover probability. For crossover probability higher than 20% the optimization time is constant and reaches its minimum.

The mutation probability takes two values: 2% and 3%, the population size having also two values: 100 and 150, for the next simulations. The obtained results are presented in Figure 11 and show us that for the same size of the population, the duration of an iteration step is smaller if the mutation probability is higher. For the same mutation probability the duration of an iteration step is higher when the population is larger.

Finding the maximum number of generations after which the algorithm stops is also important and in the simulations performed the following values were considered: {3, 5, 7, 8, 10, 15, 20, 25, 30, 40, 50, 75, 100, 150, 200}. As it is shown in Figure 12, if the number of generations is too small (<20), the optimization is not successful in several cases, because in a small number of generations it is not always possible to improve the fitness value of the best chromosome, such that it can offer a solution to the optimization problem. If the maximum number of generations is above a threshold value (~ 25 generations in this case), the optimization is always successful. This parameter depends on the topology of the network which has to be optimized, increasing with the size of the network.

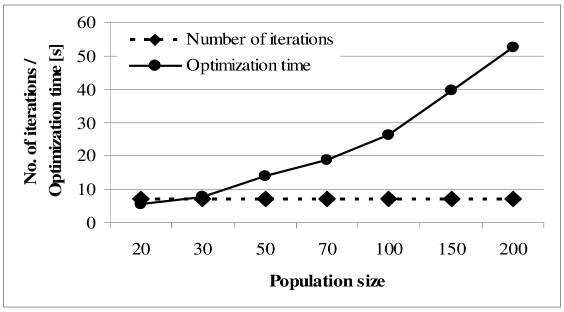

Figure 8. Optimization time vs. population size

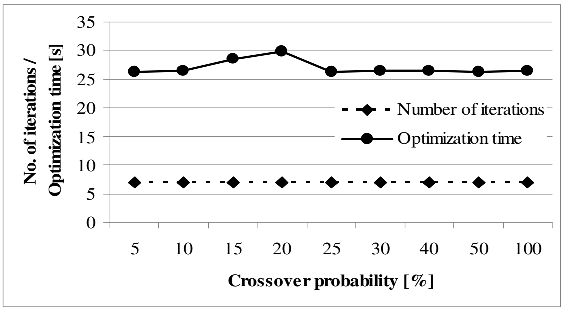

Figure 10. Number of iterations and optimization time vs. crossover probability

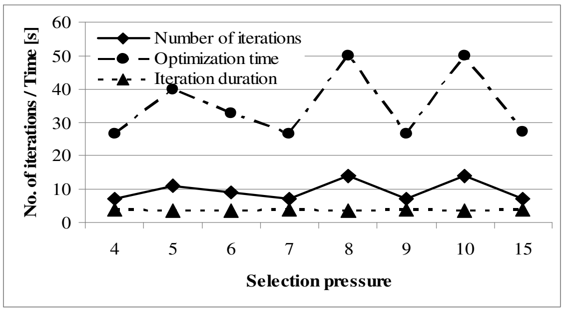

Figure 9. Optimization time vs. selection pressure

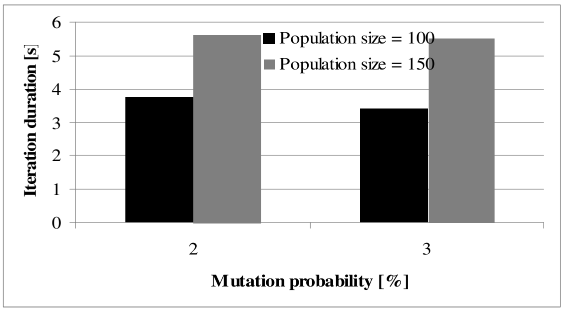

Figure 11.Iteration duration vs. mutation probability

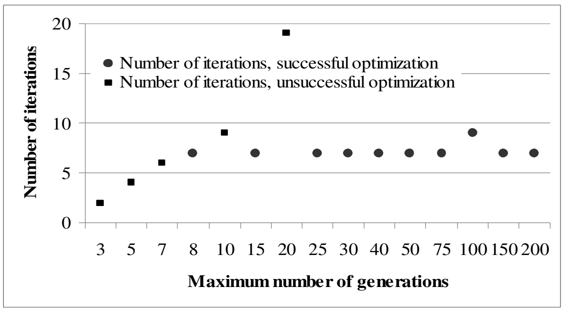

Figure 12. Number of iterations vs. maximum number of generations

The size of the finite field is a significant factor in the evaluation of the performances of the genetic algorithms used for network optimization, because Random Network Coding is employed for the evaluation of populations. The total number of iterations and the optimization time needed for successful optimization are evaluated for two different finite field sizes: $GF(2^4)$ and $GF(2^5)$. We consider the same parameters of the genetic algorithm as in the previous simulation and the maximum number of generations is 100. For each field size 7 simulations were performed and the results are presented in Figure 13. The obtained results show that the optimization time is higher if the size of the finite field is smaller. Even if the algebraic operations (addition and multiplication) are performed faster in a smaller field, the optimization time is higher, because by decreasing the size of the field the probability of generating linearly independent coding coefficients decreases as well.

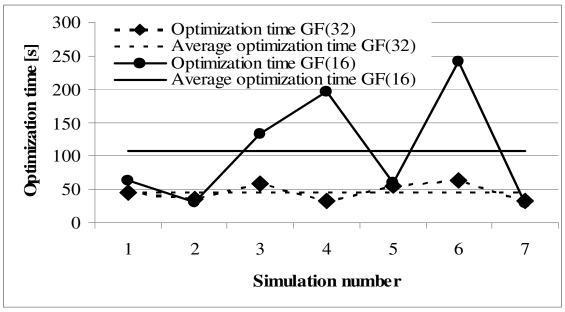

Figure 13.Optimization time vs. field size

V. CONCLUSIONS

The paper considers the employment of genetic algorithms in the optimization of communications networks implementing Network Coding techniques. The main goal was to evaluate the performances of a particular genetic algorithm adapted to the mentioned optimization problem. The evaluation was realized using computer simulations.

The particular genetic algorithm considered, also shortly described in the paper, was evaluated from the point of view of complexity and processing time. These parameters are important for optimization of real, resource constrained and dynamic topology networks.

The performance evaluation was realized considering the main parameters of genetic algorithms: population size, selection pressure, crossover probability, mutation probability, maximum number of generations and the size of the finite field used for network coding operations.

Based on the simulation results obtained for several network topologies, we can state, as a rule of thumb, that the "optimal" parameters of the implemented algorithm are close to the following values: population formed from 100 chromosomes, selection pressure 5, crossover probability 20%, mutation probability 2% and maximum number of iterations 100.

An important conclusion is that the number of generations needed for successful optimization does not change significantly with the above mentioned parameters, which proves that the implemented genetic algorithm is very robust. However the parameters of the genetic algorithm have a big impact on the optimization time. A large optimization time could be a serious disadvantage of this algorithm when it is applied in rapidly changing environments, such as wireless networks. This algorithm in particular and genetic algorithm in general could be used only for the optimization of networks which change their topology relatively slow in time.

ACKNOWLEDGMENT

This paper was supported by the project "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC" contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

REFERENCES

[1] R. W. Yeung and Z. Zhang, "Distributed Source Coding for Satellite Communications", *IEEE Transactions on Information Theory*, Vol. 45, No. 4, pp.1111-1120, 1999.
[2] R. Ahlswede, N. Cai, S.-Y. Robert Li and R. W. Yeung, "Network Information Flow", *IEEE Transactions on Information Theory*, Vol. 46, No. 4, pp. 1204–1216, 2000.
[3] M. Kim, M. Médard, V. Aggarwal, U.-M. O'Reilly, W. Kim, C. W. Ahn, M. Effros, "Evolutionary Approaches to Minimizing Network Coding Resources", *in Proceedings of the 26th IEEE International Conference on Computer Communications INFOCOM 2007*, pp. 1991-1999, 2007.
[4] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, Jun Shi and B. Leong, "A Random Linear Network Coding Approach to Multicast", *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, 2006.
[5] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction", *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
[6] M. Medard and R. Koetter: "Beyond Routing: An Algebraic Approach to Network Coding", *in Proceedings of the 21st IEEE International Conference on Computer Communications INFOCOM 2002*, vol.1., pp. 122-130, 2002.
[7] C. R. Reeves and J. E. Rowe, *Genetic Algorithms - Principles and Practice. A Guide to GA Theory*, Operations Research Computer Science Interfaces Series, Vol. 20, Kluwer Academic Publisher, Massachusetts, USA, 2002.