

Analysis of Selection and Crossover Methods used by Genetic Algorithm-based Heuristic to solve the LSP Allocation Problem in MPLS Networks under Capacity Constraints

Alessandro Vivas Andrade^{1,2}, Luciano de Errico², André L. L. Aquino³, Luciana Pereira de Assis¹, Carlos H. N. R. Barbosa²

¹Departamento de Computação, Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM)
Rua da Glória, 187, Diamantina, MG, Brasil

²LabCOM, Departamento de Engenharia Eletrônica, Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627, Belo Horizonte, MG Brasil

³Departamento de Ciência da Computação, Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627, Belo Horizonte, MG Brasil

{vivas,l.errico}@cpdee.ufmg.br,{alla,lpassis}@dcc.ufmg.br,cbarbosa@gmail.com

1. Abstract

The Multiprotocol Label Switching (MPLS) is a popular routing technique for IP networks, where the core problem is to find a route (called LSP) that satisfy all the capacity constraints imposed by a specific traffic. Genetic algorithms come as a simple, appealing solution approach, but one that requires careful choices concerning initial population generation, crossover, mutation and selection. The present paper discusses the influence of different crossover and selection methods in achieving a fast and accurate convergence of the genetic algorithm, when solving the MPLS allocation problem. The experimental results, using different network topologies such as Carrier, Dora, and Mesh, have shown that uniform crossover and Stochastic Remainder Sampling selection are the most suitable combination to solve the problem.

2. Keywords: Computer Networks, Quality of Service and Genetic Algorithms.

3. Introduction

The ever-growing need for multimedia applications and the network increasing capacity have made feasible services like video on-demand, videoconference and IP telephony. Actually, those applications have become the main concern when dealing with traffic engineering and quality of service issues. In this context, the development of new transmission and optimization mechanisms is crucial, specially for networks that act as main traffic backbones

Nowadays, different classes of multimedia applications dispute network resources along with data applications. Every application has its own differentiated requirements and the network must satisfy each one according to the application individual needs. Traffic requirements may be fully identified by means of four parameters: reliability, time delay, jitter (delay variation) and bandwidth. These four parameters establish a level of Quality of Service (QoS) for each one of the traffic flows. Techniques based on QoS, like integrated and differentiated services, allow a significant improvement, although they are all based on the conventional routing algorithm still in use on IP networks.

To assure the correct packet flow through the network, one can use the Multiprotocol Label Switching (MPLS) framework. MPLS consists in routing packets based on a label, which is inserted in every packet between the corresponding link and network headers. The complexity reduction of the routing process of IP packets and the introduction of traffic engineering are two of the main motivations for using MPLS.

Traffic engineering is a management technique that works to guarantee that the network resources can satisfy all the traffic needs of the applications. However, the current resource allocation can be non-optimized and, as a consequence, the future requests may not succeed, even though there are available resources spread over the network. Thus, optimization applied to resource allocation is essential to achieve high performance and maximum utilization in a given network. Traffic engineering deals with different aims like maximization of the used bandwidth and re-routing caused by congestion or failure. MPLS can be used to implement traffic engineering, as explicit routes can be defined between sources and destinations. Since route assignment is a NP-hard problem, heuristics like genetic algorithms are generally used in order to find an acceptable solution.

Solving the route assignment problem through genetic algorithms has been gradually accepted by specialized literature and, in this sense, many methods and techniques have already been proposed. However, use of those methods implies on the preliminary definition of a number of parameters, which can cause deep effects on the results, leading effectively to the algorithm success or failure. This work aims to identify, as much as possible, the contribution of the selection and crossover methods for the success of the genetic algorithm, relating them to the overall efficiency in finding best route definition to a given set of requisitions in a MPLS network. This efficiency is considered in terms of processor time consumption and solution convergence. Based on experiments, one can demonstrate that genetic algorithm performance shows relevant improvements, when specific selection and crossover methods are combined.

In this context, previous research work uses orthogonal crossover as the solving method to the assignment problem [10, 2], while others combines genetic algorithms to local search, producing hybrid and memetic algorithms [3, 4]. Especially, a genetic algorithm to solve multi-objective optimization problems in MPLS networks is proposed in [5]. In this work, the one-point crossover method is adopted, the crossover and mutation rates are about 75% and 5%, respectively, and no elitism has been used. Pedro et al. [15] adopt different parameters: simple elitism, 60 individuals, crossover rate of 70%, and mutation rate of 7%.

The present paper is organized as follows. In Section 4, basic concepts related to genetic algorithms are presented. Mathematical modeling and implementation details are shown in Section 5. Section 6 contains the computational results and concluding remarks are given in Section 7.

4. Genetic Algorithm - an overview

Genetic algorithms are search and combinatory optimization methods based on the natural selection conceived by Charles Robert Darwin. The natural selection states that the most adapted generation remain while the less adapted disappear with time. Genetic algorithms are evolutionary algorithms, which initially consider an initial population and evolve through the genetic operators of selection, crossover, and mutation.

A genetic algorithm can be defined as a kind of biased random search technique, developed by Holland [6], able to get optimal global solution in a complex multidimensional space. One of the advantages of genetic algorithms is that they deal with a population of simultaneous points, selecting the best ones, making possible to create a subset from the original population not only near the global solution but also in other regions of the search space.

Evolution in a given population happens when selection, crossover and mutation operators are applied to several generations. These methods affect the success of a genetic algorithm and the associated effects can vary according to the kind of problem [7]. Factors such as crossover and mutation rates, population sizes and elitism techniques must be evaluated when a genetic algorithm is intended to a given problem.

Using the crossover, it is possible to explore new points on the search space. This operation is executed based on the crossover probability index (p_c). This probability range is defined as $0.5 \leq p_c \leq 1.0$ and it is always associated with two or more individuals. Crossover allows the exchange of genetic content between individuals, by combination among binaries values of their respective parents. There are many ways of doing the crossover, such as the one-point crossover, the two-point crossover, the uniform, and the orthogonal crossovers [8]. Some of them are explained below:

- In the one-point crossover, a random number defines the segment reference that splits off the chromosome into two parts. The children are generated through the region combination (generated by the segment point) of the two parents, selecting genetic material belonging to one and another parent in an alternating way. A variation of this method, known as the two-point crossover, consists in considering two (instead of one) segmentation of the chromosome.
- In the uniform crossover, a random vector of bits of the same size of the chromosome is used. In order to generate any child, one must simply choose the mask, bit by bit. If the chosen bit is zero, the current parent is picked up; if the chosen bit is one, one must take the other parent. When the first child has been generated, the value zero in the mask means that one bit is copied from parent A and the value one in that same mask indicates the copying of a bit from parent B. On the other hand, when the second child is generated, zero in the mask indicates that a bit of

the parent B must be copied and one in the same mask indicates that a bit must be copied from parent A. Another possibility is the adoption of one point for each variable included in the problem. So, genetic material exchanging will occur only between identical variables. This method allows the exchange of genetic material, involving only genes in the very same locus, between distinct individuals.

- In the orthogonal crossover [8] experimental design methods are used and, since it is impossible to test all possibilities in many cases, the Experimental Design Method [9] allows the selection of a small set of tests only, which is representative of the whole experiment. That is achieved through orthogonal arrays. This concept can be applied to the crossover in genetic algorithm and was used on some routing problems, as seen in [10, 2].

The selection algorithm defines which individuals will belong to the new generation. When we choose only the best individuals, a premature convergence could happen. Thus, there are many selection methods such as Roulette, Tournament, Deterministic Sampling and Stochastic Remainder Sampling, listed below:

- in the Roulette method [11], the first step is to calculate the cumulative fitness of the whole population through the sum of the fitness of all individuals. After that, the probability of selection is calculated for each individual as being $p_{sel_i} = f_i / \sum f_i$. Then, an array is built containing cumulative probabilities of the individuals. So, n random numbers are generated to the range 0 to $(0 - \sum f_i)$ and for each random number an array element which can have higher value is searched for. Therefore, individuals are selected according to their probabilities of selection.
- In the Deterministic Sampling, the average fitness of the population ap_{avg} is calculated and the fitness associated to each individual is divided by the average fitness, but only the integer part of this operation is stored. If the value is equal or higher than one, the individual is copied to the next generation. The remaining free places in the new population will be fulfilled with individuals with the greatest fraction.
- The Stochastic Remainder Sampling has identical concepts used in the Deterministic Sampling and the population must be formed with the integer part of the expression result ap_i / ap_{avg} . In this case, free places were filled based on the Roulette method.

5. LSPs allocation in MPLS networks

The LSPs allocation problem can be expressed as a routing problem with some constraints. The network topology can be defined as a graph $G = (V, E, C)$, where V is a set of routing nodes and E links. C is the set of constraints of the graph. Our problem can be stated as follows: *Which set of LSPs (an LSP is a route) can be obtained in order to satisfy constraints imposed by the system and to minimize allocation costs?*

In the present work, it has been used the mathematical formulation proposed by Girish [12]. This formulation contains the following statements: routers are designated as LSRs; U is the set of source and destination LSRs; F is the LSP set; and D is a set of associated demands to F . However the mathematical model has the following parameters: u_l - LSR source; v_l - LSR destination; μ_l - bandwidth; e a_l - link cost.

The decision variable x_{ij} shows whether the LSP will be routed through the link:

$$x_{ij} = \begin{cases} 1, & \text{if the LSP } i \in F \text{ is routed to the link } l \in E \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The model is given by

$$\text{Minimize}(Z_R) = \sum_{l \in E} a_l \sum_{i \in F} \lambda_i x_{il}, \quad (2)$$

submitted to

$$\sum_{i \in F} \lambda_i x_{il} \leq \mu_l, \quad \forall l \in E \quad (3)$$

$$\sum_{\forall l|u_l=n} x_{il} = 1, \quad \forall n \in U \forall i | s_i = n \quad (4)$$

$$\sum_{\forall l|v_l=n} x_{il} = 1, \quad \forall n \in U \forall i | d_i = n \quad (5)$$

$$\sum_{\forall l|u_l=n} x_{il} - \sum_{\forall l|v_l=n} x_{il} = 0, \quad \forall n \in V \forall i | s_i \neq n, d_i \neq n \quad (6)$$

$$x_{il} \in \{0, 1\}, \quad \forall i \in F, l \in E \quad (7)$$

The constraint (2) assures that the link capacity will not be exceeded. The constraints (3) and (4) assure to all LSPs that start and terminate in a LSR are routed. The constraint (5) assures that every LSP is routed through intermediate nodes. The constraint (6) specifies that each decision variable will assume 0 or 1.

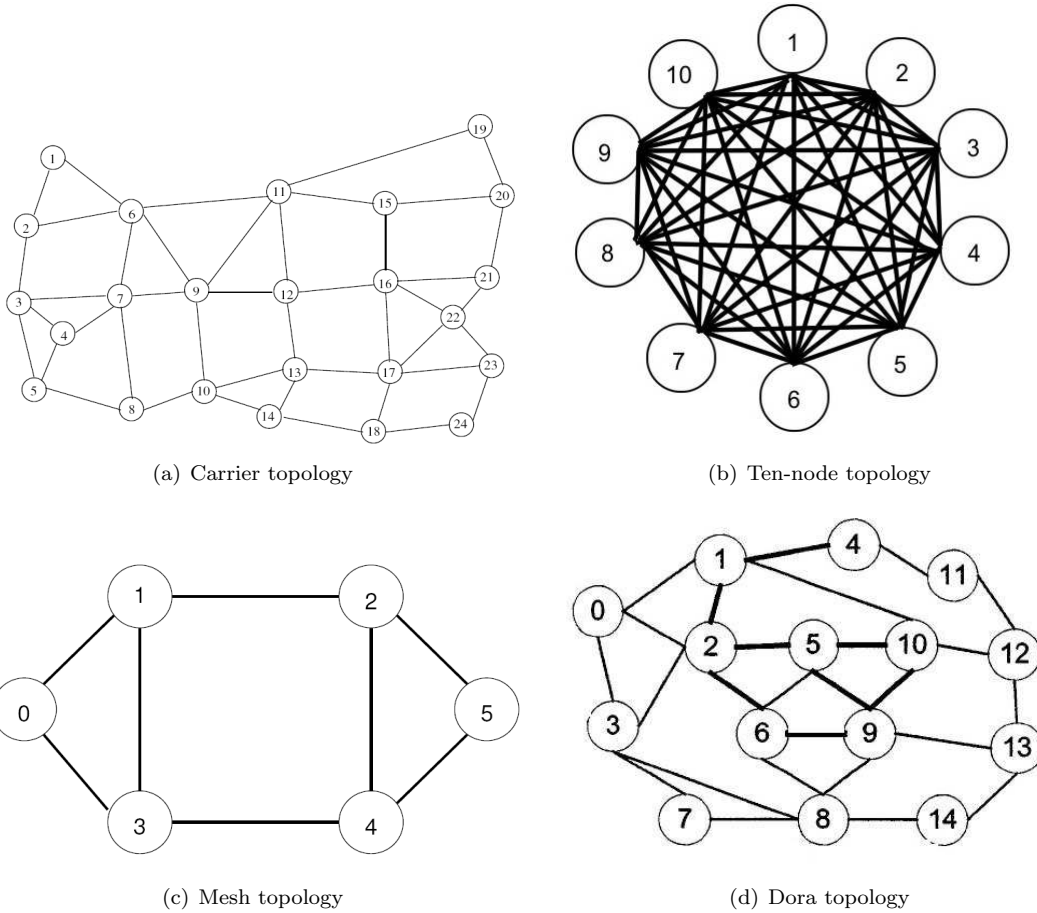


Figure 1: Topology Examples

For each sent requisition, k shortest paths are determined. These paths are obtained by cutting off arcs of the minimum cost in the shortest path. That procedure achieves solution without overloading the minimum cost arcs. The k -shortest paths are considered possible solutions to the problem, that is, a LSP. Binary encoding was adopted to the definition of the individuals, where each possible path was represented by a binary value.

In order to standardize tests, it was established two minimal paths for every LSP and all the topologies were tested with a number of requisitions varying from 50 to 500 LSPs. Considering 50 LSPs and two possible minimal paths for each LSP, we have a chromosome of 50 bit-long which implies in 2^{50} combinations in the search space. For 500 LSPs, the search space is 2^{500} combinations.

Tests were made in four topologies selected according to their nodes and arcs quantities. Carrier topology, presented in Figure 1(a), corresponds to a modified version of an IP backbone and is commonly used on simulations [13]. The Ten-node topology, presented in Figure 1(b), has 10 interconnected nodes and is a contribution of this work. The Mesh topology, shown in Figure 1(c), has six nodes and is interesting for simulations in order to get numerical results related to a more complex scenario [13]. The Dora topology, presented in Figure 1(d), was used in [14].

From the preliminary results obtained on testing the four topologies, we have chosen to work with populations no bigger than 30 individuals, crossover rate of 70%, mutation rate of 1%, bit by bit mutation, and simple elitism. About 100 tests were executed for every possible combination of selection algorithm (Roulette [ROL], Tournament[TOR], Determinist Sampling [DSA], and Stochastic Remainder Sampling [SRS]), crossover method (One-point [1PT], Two-point [2PT], Uniform [UNI], and One-Point by Variable [PTV]) and topology (Ten-node, Dora, Mesh, and Carrier). Another test using Orthogonal Crossover [OCX] was performed and this algorithm was considered being a crossover and selection method.

6. Computational Results

The system was developed using C language and tests were performed in a Pentium Core 2 Duo processor and 1GB RAM machine. Two scenarios were proposed in order to evaluate the algorithms. On the first scenario, we use 50 LSPs. On the second one, 500 LSPs were considered. For the 50 LSPs case, the optimal result were known and all the simulation achieved convergence.

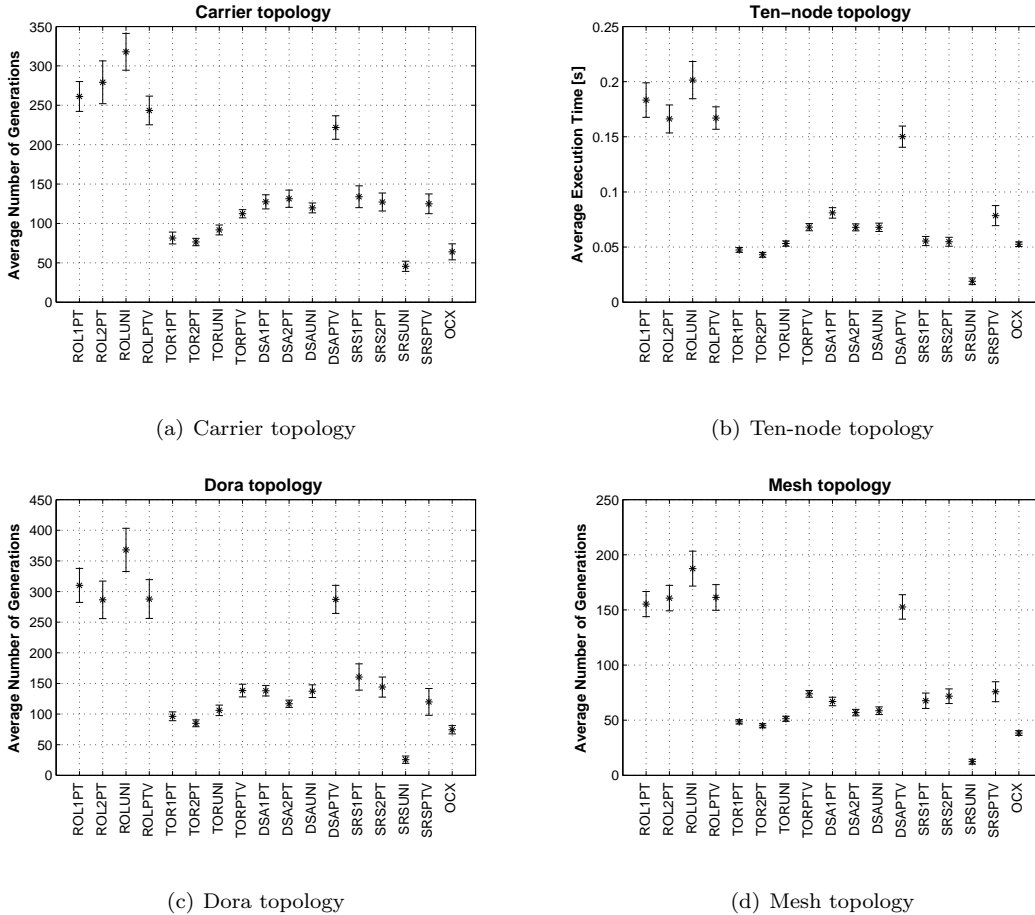


Figure 2: Generation number to distinct topologies

The presented results were obtained by 100 simulated tests for each combination. The maximum

number of generation was limited to 5000 and the trust interval associated to the calculation was 95%. The following stop criteria were adopted: maximum number of generation and non-significant improvement for 10 generation.

Figure 2 shows results associated to average generation number of the 50 LSPs scenario. Regardless to the verified topology, the patterns viewed on the plots were quite similar, better said, the fastest convergence method was achieved by combining SRS with uniform crossover. The slowest method used Roulette as the selection method.

Another relevant evaluation takes into consideration time spent in obtaining results and Figure 3 shows the related graphs. The convergence time depends on the generation number and the complexity of the algorithm being executed. Again, association between the uniform crossover method and SRS produced best results, no matter what topology was taken.

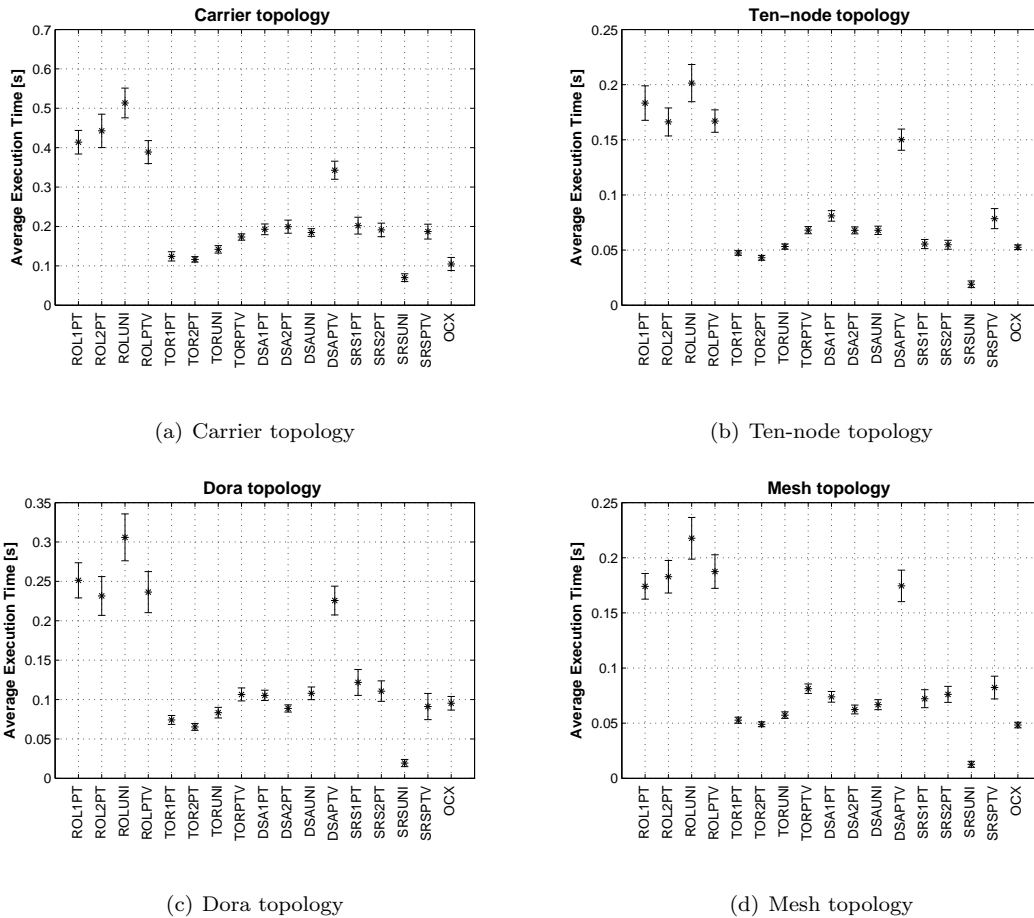


Figure 3: Average execution time for distinct topologies.

Figures 4 and 5 present the results obtained in 500 LSPs scenario. For this, the SRS selection method achieved better results, producing the least costs. The uniform crossover method association resulted in smaller computational times.

The orthogonal selection method presented the worst results in relation to execution time. This is mainly due to the algorithm complexity. The other selection methods like Roulette, Tournament, and DSA achieved times near 30 seconds, which is equivalent to an execution of 5000 generations. In those cases, the adopted stop criterion was the generation number.

Excepting SRS crossover, all the other methods needed about 5000 generations to find the final result. However, the least costs were obtained through association between SRS crossover method and one of the studied selection methods. Despite of the similarity among results, combination between SRS and

uniform crossover presented best average results.

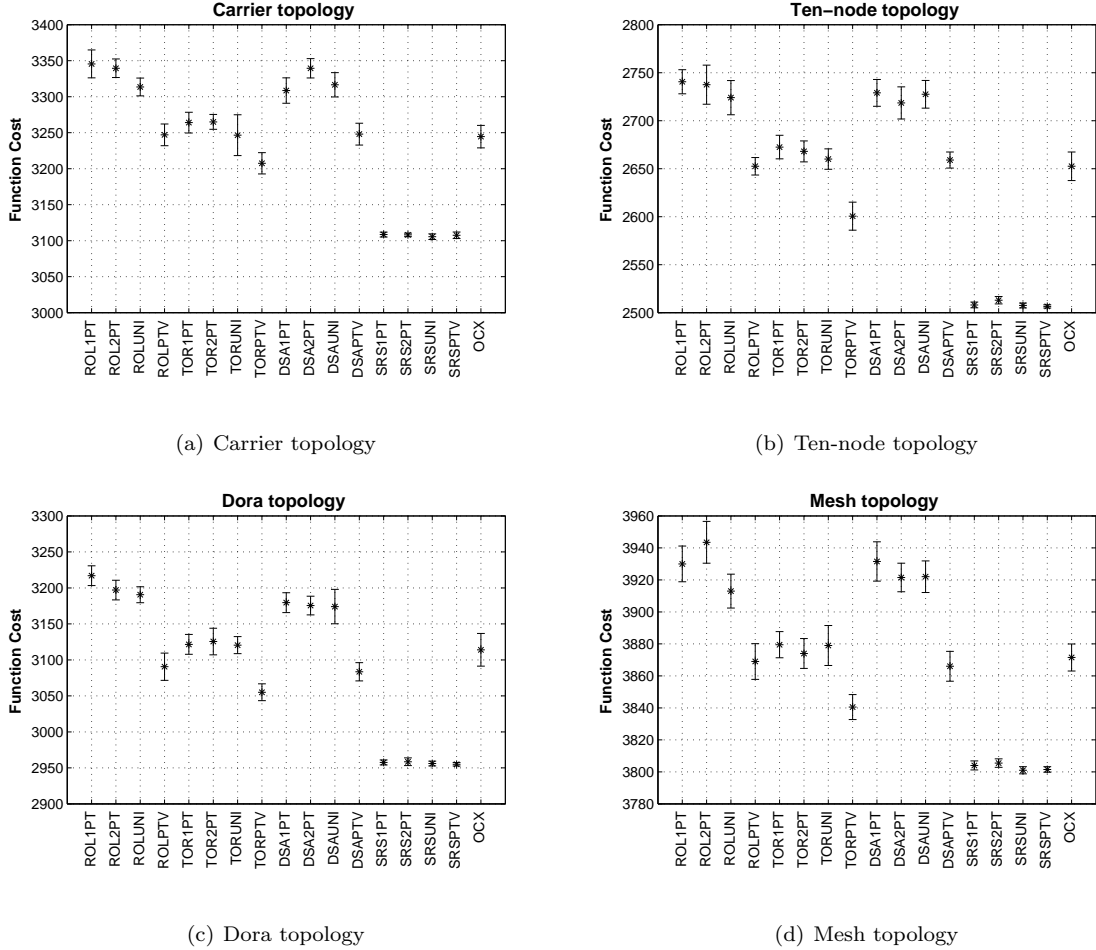


Figure 4: Costs to distinct topologies - 500 LSPs

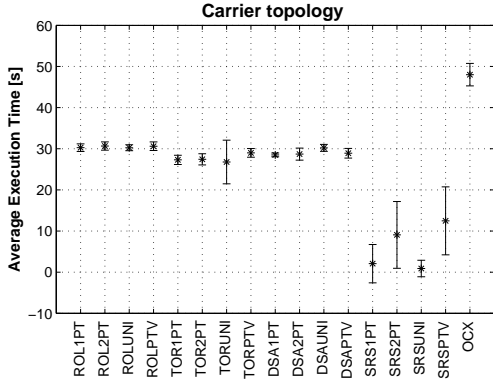
7. Concluding Remarks

For the 50 LSPs scenario, where all the algorithms achieved convergence in 100% of the cases, the ratio between the best (SRS selection and Uniform crossover) and the worst time (Roulette selection and Uniform crossover) was higher than 10 times. For the 500 LSPs scenario, the least cost was obtained through SRS selection and Uniform crossover combination. While the best method (SRS and Uniform) produced the optimal result within 0.68 seconds, the worst combination method reached its optimal result (better than the previous only by 3%) in 48.97 seconds.

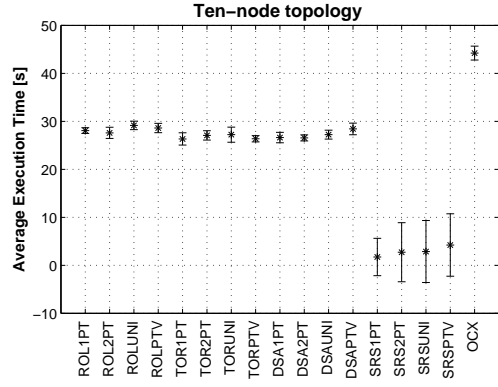
Several other methods and algorithms combinations, not tested in this work, can be used to improve the computational results, like adaptive mutation and crossover rates, space search reduction, and global elitism. Results suggested that research involving genetic algorithm must be concerned with exhaustive analysis of to the existing methods, in order to find better results.

8. References

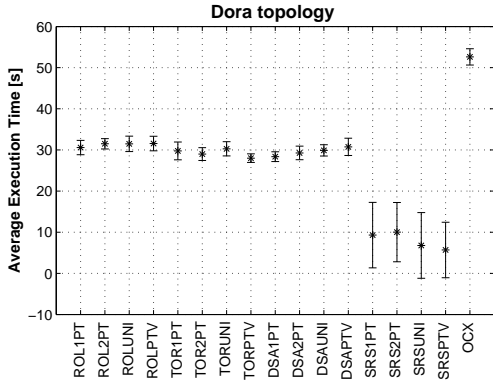
- [1] He Cuihong, Route Selection and Capacity Assignment in Computer Communication Networks based on Genetic Algorithms, *IEEE International Conference on Intelligent Processing Systems*, 1997, 1(1), 548-552.
- [2] Qingfu Zhang and Yiu-Wing Leung, An orthogonal genetic algorithm for multimedia multicast



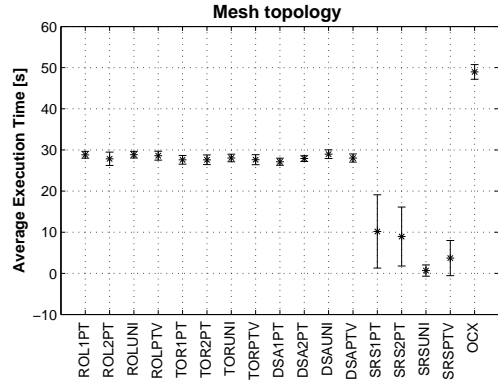
(a) Carrier topology



(b) Ten-node topology



(c) Dora topology



(d) Mesh topology

Figure 5: Computing time spent in distinct topologies - 500 LSPs scenario

routing, *IEEE Transactions on Evolutionary Computation*, 1999, 3(1), 53-62.

- [3] A. Riedl, A hybrid genetic algorithm for routing optimization in IP networks utilizing bandwidth and delay metrics, *IEEE Workshop on IP Operations and Management*, 2002, 166-170.
- [4] L.S. Buriol and M.G.C. Resende and C.C. Ribeiro and M. Thorup, A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing, *IEEE Networks*, 46(1), 2005, 36-56.
- [5] Huagang Shao and Xiao Chen and Weinong Wang, A Multiobjective Optimization Algorithm for LSP Setup in Diffserv and MPLS Networks, *First International Conference on Communications and Networking in China*, 2006, 1-5.
- [6] J. H. Holland. *Adaptation in Natural and Artificial Systems*. *Mit Press*, 1975
- [7] J. A. Vasconcelos and J. A. Ramírez and R. H. C. Takahashi and R. R. Saldanha, Improvements in Genetic Algorithms, *IEEE Transactions on Magnetics*, 37(5), 2001, 3414-3417.
- [8] Colin R. Reeves and Christine C. Wright, Genetic Algorithms and the Design of Experiments, *Evolutionary Algorithms*, 37(5), 1999, 207-226.
- [9] Douglas C. Montgomery. *Design and Analysis of Experiments*. *Wiley*, 1991

- [10] He Cuihong. Route Selection and Capacity Assignment in Computer Communication Networks based on Genetic Algorithms. *IEEE International Conference on Intelligent Processing System*, 1(1), 1991,548-552.
- [11] Marco Cesar Goldberg and Henrique Pacca L. Luna. *Otimização Combinatória e Programação Linear. 2nd ed. Campus*, 2000.
- [12] Muckai K. Girish and Bei Zhou and Jian-Qiang Hu, Formulation of the Traffic Engineering Problems in MPLS Based IP Networks, *ISCC Proceedings of the Fifth IEEE Symposium on Computer and Communications*, 2000, 214-219.
- [13] Adriana de Andrade Oliveira and Martín Gómez Ravetti and Geraldo Robson Mateus, Traffic Engineering in MPLS Networks: ARS - An Adapative Routing Scheme Based on Control Charts, *Simpósio Brasileiro de Redes de Computadores*, 2005;
- [14] R. Boutaba and W. Szeto and Y. Iraqi, DORA: Efficient Routing for MPLS Traffic Engineering, *Journal of Network and Systems Management*, 10(3), 2002,309-325.
- [15] M. Pedro and E. Monteiro and F. Boavida, A two-phase algorithm for off-line inter-domain traffic optimization, *Proceedings of Advanced Industrial Conference on Telecommunications*, 2005,396-401.