Multiobjective Optimization of MPLS-IP Networks with a Variable Neighborhood Genetic Algorithm

Renata E. Onety[123], Roberto Tadei[2], Oriane M. Neto[1], and Ricardo H. C. Takahashi[4]

[1] Universidade Federal de Minas Gerais, Dep. Electrical Eng., Belo Horizonte, Brazil
[2] Politecnico di Torino, Dep. Control and Computer Engineering, Turin, Italy
[3] FUCAPI - Analysis, Research and Technological Innovation Center, Manaus, Brazil
[4] Universidade Federal de Minas Gerais, Dep. Mathematics, Belo Horizonte, Brazil

Abstract. This paper presents a Genetic Algorithm for the optimization of multiple indices of Quality of Service of Multi Protocol Label Switching (MPLS) IP networks. The proposed algorithm, the Variable Neighborhood Multiobjective Genetic Algorithm (VN-MGA), is a Genetic Algorithm based on the NSGA-II, with the particular feature that the solutions are encoded defining two different kinds of neighborhoods. The first neighborhood is defined by considering as decision variables the arrows that form the routes to be followed by each request, whilst the second part of the solution is kept constant. The second neighborhood is defined by considering as decision variables the sequence of requests, with the first part kept constant. Comparisons are performed with: (i) a VNS algorithm that performs a switch between the same two neighborhoods that are used in VN-MGA; and (iii) the results obtained with an integer linear programming solver, runing a scalarized version of the multiobjective problem. The results indicate that the proposed VN-MGA outperforms the pure VNS algorithm, and provides a good approximation of the exact Pareto-fronts obtained with the ILP approach, at a much smaller computational cost. Besides the potential benefits of the application of the proposed approach to the optimization of packet routing in MPLS networks, this work raises the theoretical issue of the systematic application of variable encodings, which allow variable neighborhood searches, as generic operators inside general evolutionary computation algorithms.

Keywords: Routing on IP Networks, Variable Neighborhood Search, Multiobjective Genetic Algorithm.

1 Introduction

The internet transmission of multimedia applications has become an achievable goal due to the emergence of new technologies. New applications such as videoconferences, Video on Demand (VoD) or Voice over IP (VoIP) brought the need of some guarantees of network characteristics with respect to the quality of the data flow, such as minimum bandwidth or maximum delay (De Giovanni et al.; 2004). However, in the conventional internet traffic, it is not possible to predict the path of the packets of transmission, i.e, there is no guarantee of the regularity of communication. For this reason, some mechanisms were developed for Quality of Service (QoS) that allow differentiation of the flows transmitted and the definition of conditions, in order to reach a level of quality from the prioritization of different flows according to their characteristics and objectives (Paul and Raghavan; 2002). Recently, several technologies have been proposed, based on the labeling of the information on IP networks that allow the processing of different packets according to specific policies in order to achieve QoS requirements. The MPLS (Multi Protocol Label Switching) is an example of such a technology that makes possible the explicit routing of packets, which facilitates the provisioning of QoS according to the requirements of multimedia applications.

Several studies have been proposed recently in order to develop an approach of Traffic Engineering for Routing with QoS. According to RFC-3272 (Request for Comments 3272), the Internet Traffic Engineering is defined as that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks (Awduche et al.; 2002). Many of those studies deal with routing on IP networks and MPLS, using single-objective GAs (Maia et al.; 2007; Andrade; 2008) or deterministic methods, like Lagrangian Relaxation (Dias; 2004). As the model in those studies is formulated with a single objective, the search can be biased to a specific goal, leading to solutions that are unsuitable under other objective viewpoint. For this reason, multi-objetive strategies have received some attention recently. However, the use of multi-objective methods applied to the problem of routing on IP networks is not so extensive. The works (Alvarado et al.; 2005; Erbas and Erbas; 2003; Santos; 2009) should be mentioned in such a context. This work employs the same objective functions employed by Santos (2009), which proposes a dynamic evaluation for routing in MPLS using multi-objective techniques.

The present study deals with the optimization of two or three merit functions which, according to Wang and Crowcroft (1996), render the problem NP-complete. Thus, the techniques based on non-deterministic heuristics are likely to be the most suitable ones. The stochastic optimization techniques such as Genetic Algorithms (GAs) and Variable Neighborhood Search (VNS) are examples of heuristic search strategies that can be used. GAs (Goldberg; 1989) are search techniques that consider sets of candidate solutions (each solution is an *individual*, and the set is the *population*), which are varied according to two kinds of probabilistic rules: the *mutations*, which introduce perturbations into current solutions, producing new ones, and the *crossover*, which combine the information from previous solutions, producing new ones. Finally, the current population passes a *selection* procedure, that probabilistically increases the frequency of the best solutions in a new population, reducing the frequency of the worst ones. In recent years, it has been recognized that a key factor that determines the performance of GAs is the encoding employed for the representation of the solutions in the population. This is attributed to the fact that different encodings induce different neighborhoods, which lead to different behaviors of the variation mechanisms of mutation and crossover (Carrano et al.; 2010). VNS techniques (Mladenovi and Hansen; 1997), on the other hand, usually evolve a single solution each time. This solution is subject to heuristic descent searches that find local minima in the attraction regions that are characterized by connected paths in a given neighborhood induced by an encoding. The heart of VNS techniques is the alternate usage of different encodings that induce different neighborhoods, which allows the algorithm to perform further descent steps after finding a local minimum in an encoding, by simply changing the encoding that is being used.

This paper deals with the problem of packet routing in MPLS systems. In the specific context of this problem, a new Multiobjective Genetic Algorithm, the VN-MGA (Variable Neighborhood Multiobjective Genetic Algorithm) is developed. The optimized routing tries to minimize the network cost and the amount of rejection of simultaneous requests, as well as to perform a load balancing among routes. The proposed algorithm allows to deal with these conflicting QoS indicators, described as independent objective functions. Moreover, the set of solutions provides flexibility for the decision maker to select one or other goal according to the current state of the network.

The proposed VN-MGA is based on the classical NSGA-II (Deb et al.; 2002) and has, as a distinctive feature, its crossover and mutation operators inspired in the concept of *variable neighborhood* of the VNS techniques. Two different encodings are employed: a low-level encoding, which encodes explicitly the routes that are followed by each request of service, and a high-level encoding, that encodes the permutations of the several requests of service, defining the order in which they will be included in the solution. The crossover and mutation operators, acting in these two levels, are able to explore and to exploit the decision variable space with enhanced efficiency, leading to solutions that dominate the ones that appear in algorithm versions using only one level. It should be noticed that the proposed operators are problem-specific. In problems of combinatorial nature, it has been established that algorithms employing specific crossover and mutation operators can be much more efficient than general-purpose GAs (Carrano et al.; 2006).

A group of routing problems has been solved using hybrid approaches (Perboli et al.; 2008). There are hybrid methods for the vehicle routing problem using Genetic Algorithms and Tabu Search (Perboli et al.; 2008) or combining VND (Variable Neighborhood Descent) and GRASP (Greedy Randomized Adaptive Search Procedure) (Freitas; 2008) and also problems of another characteristics, such as pipeline petroleum distribution using GA and VNS (de Souza Filho; 2007). However, those studies typically combine the different algorithms in a literal way, performing steps from one algorithm and from the other algorithm. The present authors have not identified any reference that performs an organic combination like the one proposed here. Some prelliminary results of the work described here have been published in the conference paper (Onety et al.; 2011).

This paper is organized as follows: Section II describes the problem and its modeling. Section III presents the VN-MGA. Section IV presents some results obtained with this approach and the section V concludes the paper.

2 Problem Description and Modeling

This study deals with the problem of choosing routes in a scenario of a corporative IP network with MPLS technology. The proposal is to minimize the network cost, to respond for the various user's requests ensuring the quality of service and to provide a load balancing between simultaneous streams. The network model is represented by the graph $G = (V, A)$, where V is the set of routers in the MPLS domain and $A = (i, j)$ is the set of links from node i to node j, or the links between the routers. The bandwidth of each link (i, j) is represented by B_{ij}. Each user request is represented by (o^k, d^k, b^k), where o^k and d^k indicate, respectively, routers of source and destination of traffic and b^k indicates the amount of bandwidth to be reserved for the request k. The set of requests is represented by R.

The objective functions are described by the equation (1), based on the work of Santos (2009).

$$\min \begin{cases} F_1 = \sum_{k \in R} \sum_{(i,j) \in A} x_{ij}^k \\ F_2 = \sum_{k \in R} (1 - a^k) \\ F_3 = \alpha \end{cases} \tag{1}$$

s.t.

$$\sum_{(i,j) \in \Gamma_i^+} x_{ij}^k - \sum_{(l,i) \in \Gamma_i^-} x_{li}^k = \begin{cases} a^k & (o^k); \\ -a^k & (d^k); \\ 0, & \forall i \in V, \forall k \in R \end{cases} \tag{2}$$

$$\sum_{k \in R} b^k x_{ij}^k \leq \alpha B_{ij}, \forall (i, j) \in A \tag{3}$$

$$\sum_{k \in R} a^k \geq C \tag{4}$$

where

$$x_{ij}^k \in \{0, 1\}, \forall (i, j) \in A, \forall k \in R \tag{5}$$

$$a^k \in \{0, 1\}, \forall k \in R \tag{6}$$

$$\alpha \in [0, 1] \tag{7}$$

The objective function F_1 represents the sum of links used to accept a request k. The fewer links are used, the smaller is the delay for the data travel from origin to destination. F_2 aims to reduce the number of rejections of requests. The amount of rejection of the requests is related to the admission control of new connections, which determines if a connection can be admitted or not, according to the load network condition and the amount of bandwidth requested. The minimum number of requests that must be responded is represented by C, shown in Equation 4. In F_3, α represents (in relative terms) the load of the most used edge, with values varying from 0 to 1. Minimizing the amount of data traffic on the links means that the load is evenly distributed. Consequently, the network is balanced. The constraint 2 represents the classical flow conservation. In 3, the requested bandwidth (b^k) for a link (i, j) must be less than or equal to the available bandwidth.

The problem stated in equation (1) has several functions to be minimized, and therefore it is a *multi-objective optimization* problem. A multi-objective optimization problem is defined as:

$$\begin{array}{l} \min \mathbf{f}(\mathbf{x}), \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_l(\mathbf{x})) \\ \text{subject to: } \; \mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathcal{X} \end{array} \tag{8}$$

in which $\mathbf{x} \in \mathcal{X}$ is the *decision variable vector*, $\mathcal{X}$ is the *optimization parameter domain*, $\mathbf{f} \in \mathcal{F}$ is the *objective vector*, and $\mathcal{F}$ is the *objective space*. In other words, $\mathcal{F} = \mathbf{f}(\mathcal{X})$.

The goal of some multi-objective optimization methods is to obtain estimates of the Pareto-optimal set (Ehrgott; 2000), which contains the set of non dominated solutions of the multi-objective problem. A point $\mathbf{x}'$ is said to be dominated by another point $\mathbf{x}$ if the following relation holds:

$$\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}') \text{ and } \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}')$$

in which the relation operators $\leq$ and $\neq$ are defined as:

$$\mathbf{f}(\mathbf{a}) \leq \mathbf{f}(\mathbf{b}) \Leftrightarrow f_i(\mathbf{a}) \leq f_i(\mathbf{b}), \quad \forall i = 1, 2, \cdots, l$$

and

$$\mathbf{f}(\mathbf{a}) \neq \mathbf{f}(\mathbf{b}) \Leftrightarrow \exists i \in \{1, 2, \cdots, l\} : f_i(\mathbf{a}) \neq f_i(\mathbf{b})$$

in which $\mathbf{a}$ and $\mathbf{b}$ represent two different decision vectors.

In this way, the Pareto set $\mathcal{P}$ is defined as the set of non dominated solutions:

$$\mathcal{P} = \{\mathbf{x}^* | \nexists \mathbf{x} : \mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}^*) \; \wedge \; \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}^*)\} \, . \tag{9}$$

All solutions that are not dominated by any other decision vector of a given set are called *non dominated* regarding this set. A Pareto-optimal solution is a non dominated vector $\mathbf{x} \in \mathcal{X}$. The Pareto-optimal set of the multi-objective optimization problem is the set of all Pareto-optimal solutions. The image of this set in the objective space is called the *Pareto front* $\mathbf{f}(\mathcal{P})$.

3 Structure of Multiobjective Genetic Algorithm

The basic structure of the multiobjective genetic algorithm VN-MGA used here is the classical *Non-dominated Sorting GA* (NSGA-II), described in (Deb et al.; 2002). The following features of NSGA-II are used inside VN-MGA:

1. *Non-dominated sorting:* consists in sorting the solutions according to the non-dominance ranking. An individual belonging to rank 1 is not dominated by any of the solutions, while an individual belonging to rank k is dominated by at least one individual that belongs to rank $k-1$. This ensures that solutions belonging to lower dominance ranks are better than solutions situated at higher ranks.
2. *Crowding-distance:* The crowding distance is used as a measure of occupation in the neighborhood of a solution in the objective space. This indicator is defined as the sum of the lengths of the edges of a hypercube with vertices situated on the m nearest solutions (in which m means the dimension of the objective space). The crowding distance is used as the comparison criterion between solutions situated in the same rank, providing an advantage to the solutions which have the nearest neighbors at larger distances. This helps to avoid situations where the obtained solution set is too concentrated in a small (crowded) region, leading the algorithm to produce more uniform samplings of the Pareto-set.
3. *Binary tournament:* consists in choosing two individuals randomly and comparing them according to some fitness function. The one with best fitness evaluation is selected. In such a comparison, the rank is used as the first criterion and, in the case of solutions with same rank, the crowding distance is used in order to determine the result of the tournament.

We now depict the multiobjective optimization approach to the problem of optimizing the routing in IP networks, with the specific features that are necessary to deal with QoS parameters.

3.1 Variable Neighborhood Search

The Variable Neighborhood Search (VNS), proposed by Mladenovi and Hansen (1997), is a simple heuristic method which has the purpose of performing a global optimization using sequences of local searches. The main idea is to have a repertoire of operators which implicitly define two or more structures of neighborhood, and to switch between them. The search starts with a feasible solution and searches iteratively for a new one in the immediate neighborhood defined by the current search operators. By switching the operators, it is possible to change the neighborhood, which allows to perform descent searches in a new neighborhood. This allows to escape from points which represent local minima in some neighborhood, using the descent paths of other neighborhoods.

In literature, there are many variants of VNS which consider different sequences of neighborhoods, or different solution acceptance conditions. A basic version defines a set of neighborhoods $N = \left\{N^k, k = 1, ..., kmax\right\}$ and an initial solution x that will be used in the local search with the neighborhood N^1. This procedure is repeated for many iterations. The acceptance condition will choose between the previous local optimum and the new one. If this is the best solution, then the neighborhood for the next iteration will be in the first position. Otherwise, the neighborhood will follow the sequence. The algorithm proposed by Hansen and Mladenovi (2001) is described in Algorithm 1.

Algorithm 1 Basic VNS

$k \leftarrow 1$
while $k \leq kmax$ **do**
  a) Perturbation: generate randomly a point x' in $N^k(x)$;
  b) Local search: Local search from x'; Denote x'' as a local optimum obtained;
  c) Acceptance: if $f(x'') < f(x)$ then $x \leftarrow x''$ and continue the search on this neighborhood;
  else, $k \leftarrow k+1$
end while

Based on distinct neighborhoods, this work proposes the integration of VNS concepts within Genetic Algorithms. During the evolution process, genetic operators of crossover and mutation are developed for each such a neighborhood. In this way, the search in one neighborhood aids the search in the other one with alternated executions, exploring different search spaces.

3.2 The Proposed Multiobjective Genetic Algorithm

The figure 1 illustrates a schema of the proposed algorithm, the VN-MGA. Crossover and mutation operations are defined at two levels. The number of generations is the criterion used to determine when to switch encodings. Thus, after N generations searching new routes at Level 1, the search for new solutions is made at Level 2.

3.3 Genetic Representation

The encoding was designed in two levels of operations. The Level 1 represents the codification of routing, i.e. the genetic operations focus on the sequences of arcs that form routes. The Level 2 encodes the sequence of requests, i.e. the sequence in which the requests should be included in the solution. The requests indicate the demand of N flows, given as origins and destinations. The two levels are considered alternately. Figure 2 illustrates the population at each level.

At Level 1, the individual is represented by a group of routes on each request, denoted by $i_1, i_2, \ldots, i_n$. Each path is described by a source, intermediate and destination nodes, represented by the node numbers, where the first one is the source node and the latter one is the destination node. Each request has a specific need for bandwidth according to its application, called Bandwidth Requested. Each link has a total Bandwidth Capacity. Thus, in order to respond to requests, the bandwidth requested must be less than or equal to the available one.

Considering that it is a multiobjective problem, not only the shortest path should be examined. For this reason, aiming to generate a diversity of individuals, the initial routes are generated from the Dijkstra's algorithm with random costs distributed on links, given the origins and the destinations.

The bandwidth is then withdrawn from the available one, representing the allocation of routes with bandwidth reservation. If the request cannot be met,

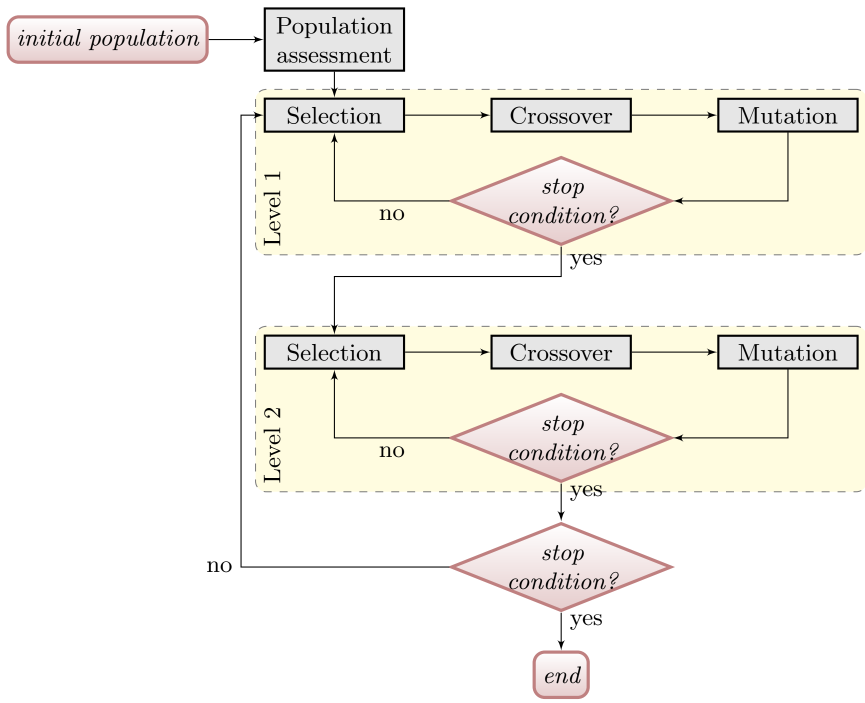


Fig. 1. Schema of VN-MGA

i.e. the requested bandwidth is greater than the available one, then that request is rejected.

At Level 2, the individuals are represented by a set of requests, indicated by $R_1, R_2, \ldots, R_m$. As the algorithm does not differentiate the priority between simultaneous requests, the evaluation is made according to their position in the sequence. Thus, depending on the sequence, accepting or rejecting a request can impact the result of optimization.

3.4 Crossover Operators

In this study, crossover operators are proposed for the two encoding levels. The first one consists of the exchange of genetic material between individuals at Level 1, which represents the routes. Two individuals are selected randomly, responding for the same request r of source O_r and destination D_r. If there is a dominance relation between them, the dominant individual is selected. If there is no dominance, the crossover tries to join characteristics of both parents. If there is a node Nc in common between those individuals, the offspring is formed from node O_r up to Nc of the first parent i_1 and from Nc until Dr of the second parent i_2. Figure 3 illustrates the crossover process.

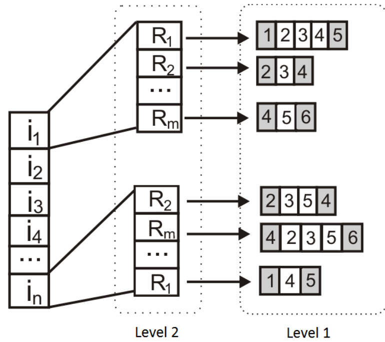


Fig. 2. Codification in two levels of operations

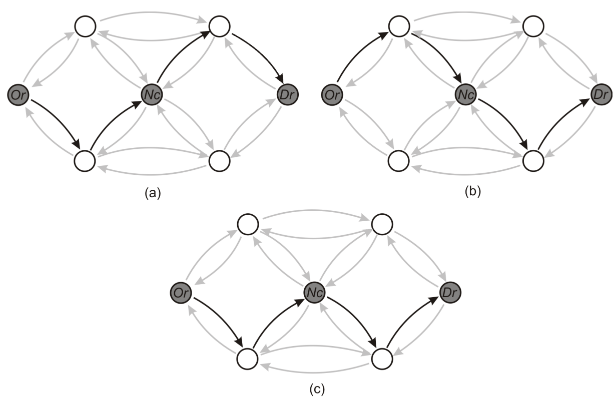


Fig. 3. Crossover with node in common (Santos; 2009). (a) represents the route of an individual i_1 to a request r. (b) represents the route of an individual i_2 to a request r. (c) represents the route of the offspring to a request r.

If there is not a common node between individuals, the crossover attempts to find edges to link the paths. Thus, the offspring inherits the nodes from O_r until the new node interconnection of the first parent i_1 and from the new node until D_r of the second parent i_2. Figure 4 illustrates the crossover without a vertex in common.

If there is no node to link the paths, the offspring inherits one of the parents' path, randomly selected in order to compose the set of population.

In the Level 2 crossover, the individual is analyzed from the perspective of request sequence. Two routes representing the same request are randomly selected and swapped, generating a new combination of them, and therefore, a new individual. Figure 5 illustrates this operation. The request R_2 is selected from the individual i_1. Then, looking for R_2 in i_2, the routes R_2 of i_1 and R_2 of i_2 are interchanged.

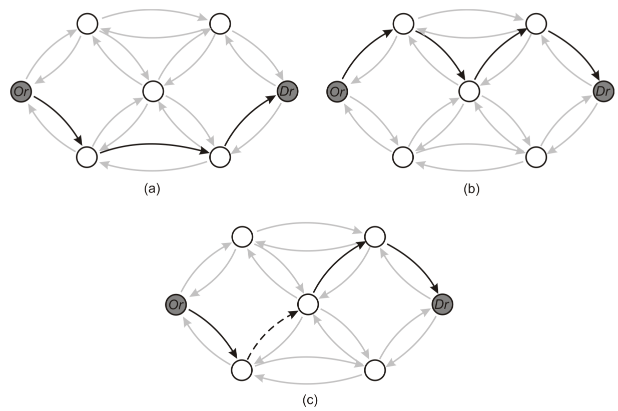


Fig. 4. Crossover without node in common (Santos; 2009). (a) represents the route of an individual i_1 to a request r. (b) represents the route of an individual i_2 to a request r. (c) represents the route of the offspring to a request r.

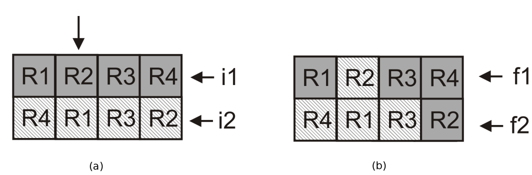


Fig. 5. Crossover at Level 2. (a) The request R_2 is selected in individual i_1. (b) The routes R_2 of i_1 and R_2 of i_2 are interchanged, generating the offspring.

Another crossover operator, still focusing on the Level 2, is implemented based on the Partially-Mapped Crossover (PMX). Initially two chromosomes are picked out. Then, two cutoff points are chosen randomly. Subsequently, the exchange of the material situated between the cutoff points is performed, producing two offspring, still intermediate. In order to conclude the crossover, an operation of mapping is performed, in order to repair of the chromosomes, making the genes unique. With this crossover, in addition to the exchange of routes between the parents, a change of the order of requests is also performed. Figure 6 illustrates this operation.

3.5 Mutation

The mutation operator is the responsible for the insertion of new genetic characteristics for individuals of the population. As in the case of crossover, the mutation is defined for the two encoding levels, with Level 1 concerning the sequences of arcs that form the routes and Level 2 dealing with the sequences of requests, given pre-defined routes that were found with Level 1 operations.

For Level 1, the chromosome to be mutated and two cutoff points on that are chosen randomly. A new path, from the initial to the second cutoff point, is

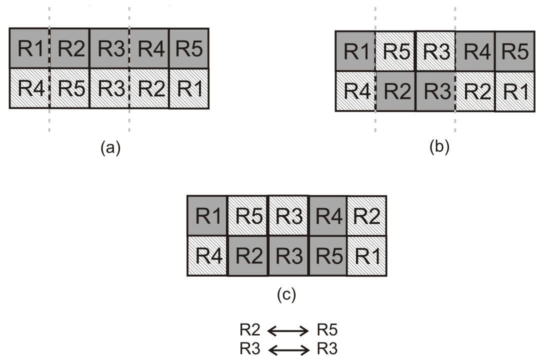


Fig. 6. PMX crossover at Level 2.

searched in the adjacency matrix. Therefore, a new section of the route is created. The search for a new sub-route is performed both in forward and backward directions (from the first to the second cutoff point, and in the opposite direction, from the second to the first cutoff point), alternately, avoiding any bias in this search. Figure 7 illustrates the process of mutation.

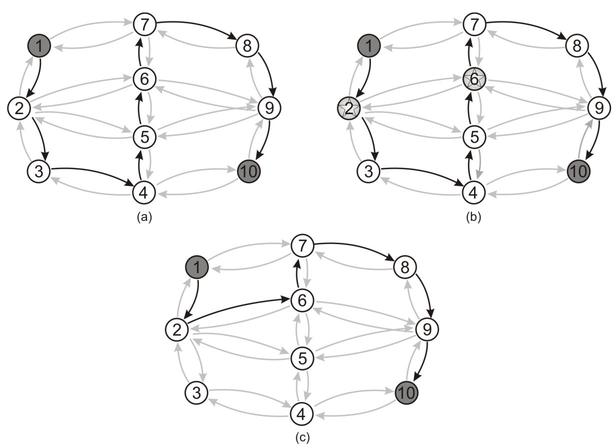


Fig. 7. Mutation at Level 1. (a) represents the route of an individual to a request r. (b) selection of the points 2 e 6 for a new sub-route. (c) Mutated individual.

The mutation in Level 2 performs the permutation of requests. Similarly to the Level 1 mutation, at this stage two cutoff points are selected randomly, but now, considering the requests. Then, a swap between these two requests changes the sequence of individuals. Figure 8 represents the mutation at Level 2.

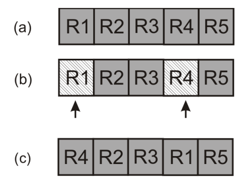


Fig. 8. Mutation at Level 2. (a) Represents the sequence of requests from individual. (b) The points selected for mutation. (c) Mutated individual.

4 Effects of Two-Level Encoding

With the purpose of determining the specific effect introduced by the two-level encoding, which was introduced in the VN-MGA, some tests have been performed for comparing the performance of VN-MGA with two reduced versions of the same algorithm, which employ operators either only on Level 1 or on Level 2. Those tests have been performed with some network instances, commonly used for telecommunications problems, such as that presented on Fig. 9(a) and Fig. 9(b), which will be named here as Instance #1 and #2, respectively.

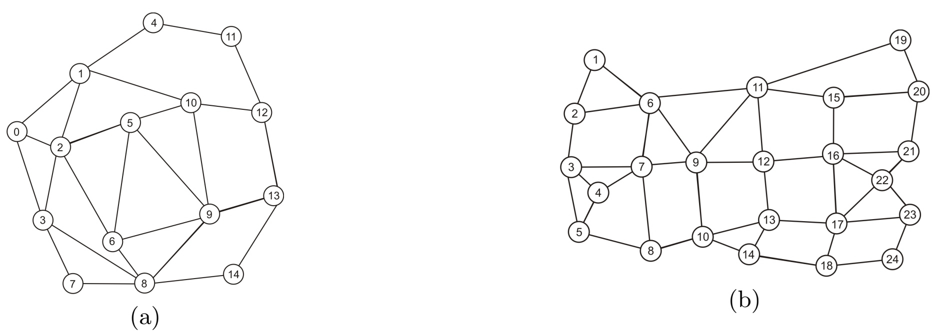


Fig. 9. (a) Instance #1. (b) Instance #2.

The parameters used in the experiments described in this section are displayed in Table 1.

As can be observed in Figures 10 and 11, the variable encoding described by two levels, improves the final quality of the routing. It can be observed that the solutions delivered by VN-MGA dominate the solutions obtained by single-level searches.

5 Results

In order to evaluate the VN-MGA proposed here, some numerical tests were performed. The first comparison was performed with the results delivered by an

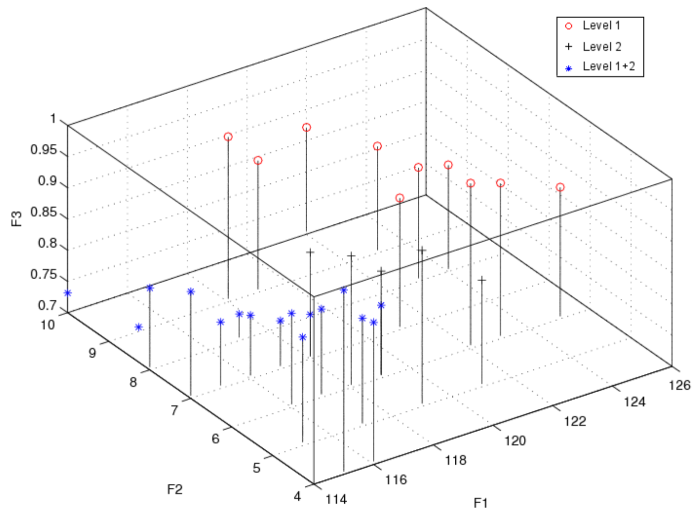


Fig. 10. Results for Instance #1. Red circles represent the solutions achieved by Level 1 algorithm (encoding of paths), black crosses represent solutions obtained by Level 2 algorithm (encoding of requests), and blue asterisks indicate the solutions provided by the combination of both levels 1 and 2 (the VN-MGA).

integer linear programming (ILP) solver, which provides the exact solutions for the problem instances. The ILP formulation delivers the solutions of the multi-objective problem by using an scalarization approach, the ϵ-constraint method (Ehrgott; 2000). The multi-objective optimization problem is converted into a series of single-objective ones, each one with only one objective function being optimized and the other ones treated as constraints of the problem. For each objective function considered as a constraint, a constraint value ϵ_i is assigned. Each solution of such a problem is at least a weakly non-dominated solution (Ehrgott; 2000), and by varying the values of ϵ_i it is possible to generate all the solutions belonging to the Pareto-optimal set of the problem. The main drawback of this approach, apart the fact that the ILP solver should be run once

Table 1. Parameters for the algorithm

Mutation Probability	0,4
Crossover Probability	0,9
Available Bandwidth for each link	1024Kbps
Requested Bandwidth	200Kbps e 400Kbps
Number of generations	50
Number of individuals	50
Number of requests	50

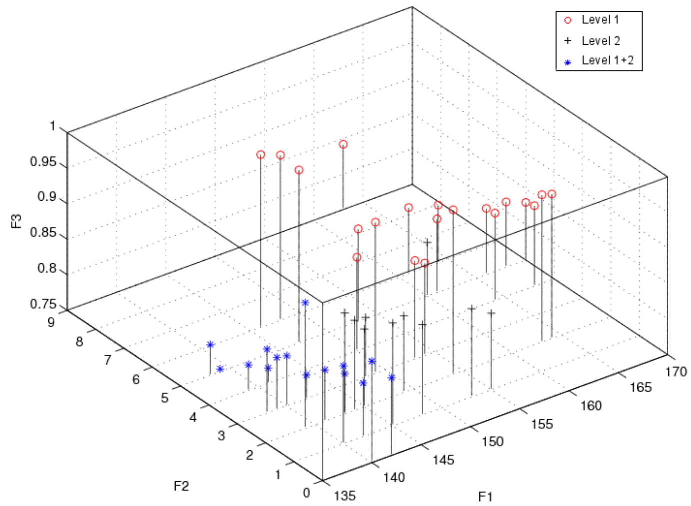


Fig. 11. Results for Instance #2. Red circles represent the solutions achieved by Level 1 algorithm (encoding of paths), black crosses represent solutions obtained by Level 2 algorithm (encoding of requests), and blue asterisks indicate the solutions provided by the combination of both levels 1 and 2 (the VN-MGA).

for generating each solution of the Pareto-optimal set, is that the computational complexity of the approach is exponential in the number of decision variables. Therefore, this kind of approach is not suitable for large problem instances.

A problem with the objective function vector represented by $f(.)$ and the constraint function vector represented by $g(.)$ can be modeled in terms of m problems:

$$x^* = \arg\min f_i(x), \tag{10}$$

subject to:

$$\begin{cases} g(x) \leq 0 \\ f_j(x) \leq \epsilon_j \ ; \ j = 1, \ldots, m \ ; \ j \neq i \end{cases} \tag{11}$$

where

$$x \in \mathbb{R}^n, f(.) : \mathbb{R}^n \longmapsto \mathbb{R}^m \text{ and } g(.) : \mathbb{R}^n \longmapsto \mathbb{R}^p \tag{12}$$

As ILP tools, both the FICO$^{\text{TM}}$XPress Optimization and the CPLEX solver have been used here. In order to simplify the search, it was considered the minimization of two objective functions: F_1 that represents the cost of time delay in the network, and F_3 that represents the load of the most used link. The second objective function, that represents the number of package rejections, was set with a fixed value, initially zero. Firstly, the ILP solver searched the minimum value for the objective function F_1. Afterwards, starting from the optimal value of F_1, it minimized the objective function F_3, by relaxing F_1 with an increasing ϵ_1.

The solutions delivered by the ILP solver, using the ϵ-constraint scalarization approach, are used here in order to evaluate the ability of the proposed algorithm for reaching the Pareto-optimal set of the problems.

Some parameters of the VN-MGA which are kept fixed in the experiments are described in Table 2.

Table 2. Parameters for the VN-MGA algorithm

Mutation Probability	0,4
Crossover Probability	0,9
Available Bandwidth for each link	1024Kbps
Requested Bandwidth	200Kbps e 400Kbps

The same instances that were represented in Figure 9, were used in the tests described in this section.

Instance #1

The Instance #1 is a simple one, and it is used for the purpose of evaluating the proposed algorithm. It is composed of 15 vertices and 52 links. For 10 and 20 simultaneous requests, 40 generations and 40 individuals are used. Twice these values are employed for 30 and 40 simultaneous requests. The results are shown in Figure 12.

Instance #2

The Instance #2 is composed of 24 vertices and 43 links. For 10 simultaneous requests, the values obtained by VN-MGA are equal to the best values reached by the ϵ-constraint method. For 20, 30 and 40 simultaneous requests, most of the values obtained by VN-MGA are equal to the best values attained by the ϵ-constraint method, but in some cases the results by VN-MGA presented a small gap in relation to the exact solution. Although finding most of the optimal points, the VN-MGA was not able to cover the whole Pareto-optimal front. The VN-MGA clearly outperforms the basic VNS algorithm, in the cases of 30 and 40 requests, both in terms of the number of Pareto-optimal estimates that were generated and in terms of dominance. Figure 13 shows these results.

5.1 Hypervolume metrics

The hypervolume metrics was employed in order to provide some numerical figures for the comparisons between the VN-MGA and the basic VNS algorithm. In the comparisons, the reduced-version algorithms involving only Level 1 and

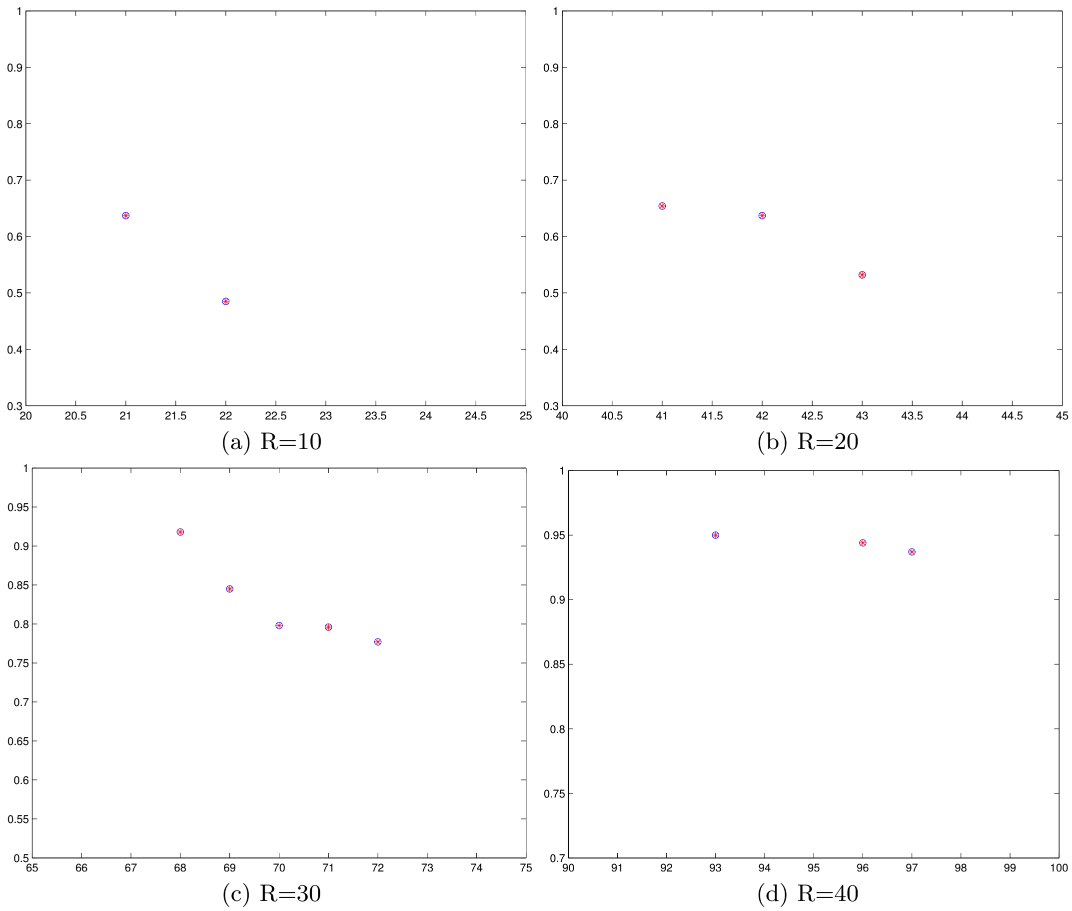
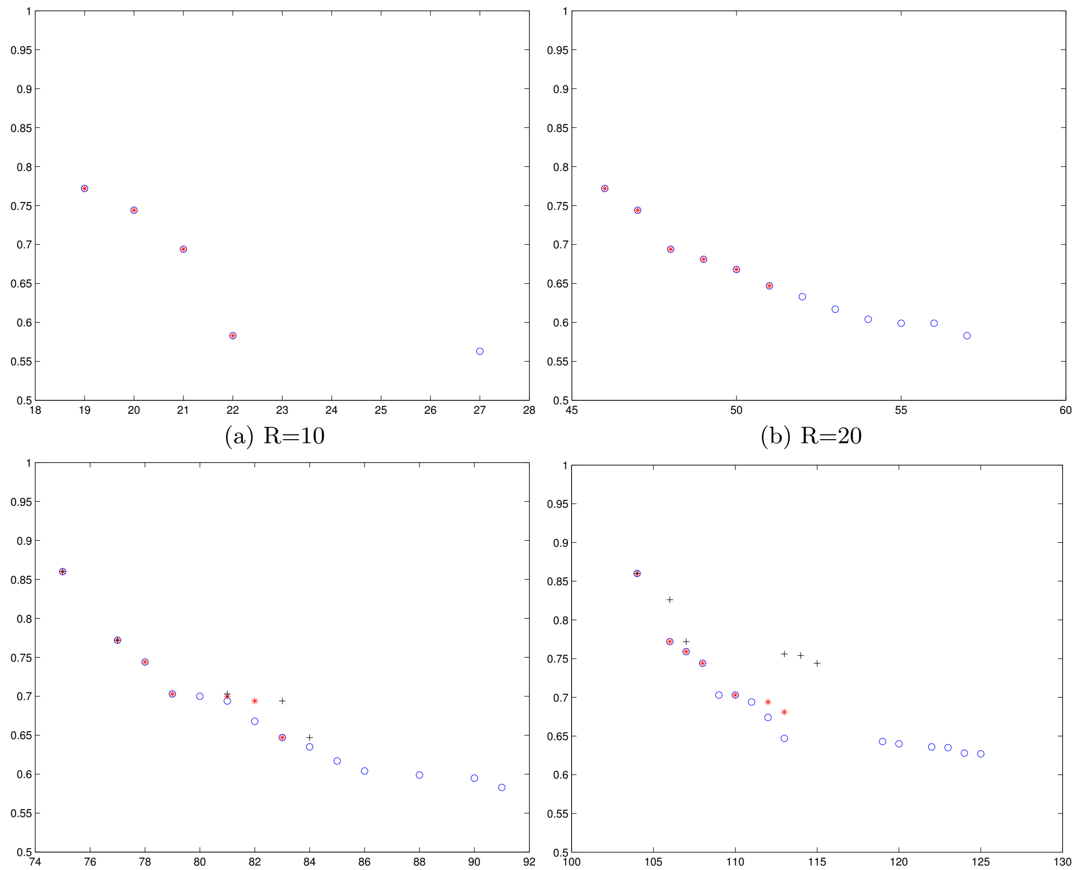


Fig. 12. Comparison between e-constraint and the VN-MGA algorithm. Circles represent the solutions achieved by the ϵ-constraint ILP technique, and the solutions delivered by the proposed VN-MGA are represented by red asterisks. Instance #1, R=10, R=20, R=30, R=40

Level 2 operations are also included, with the purpose of quantifying the performance gain of the employment of both levels. The package described by Fonseca et al. (2006) was employed in order to compute those figures. The table 3 shows the values of hypervolume metric calculated for the Pareto-Optimal set of the Instance #2, produced by each algorithm, after 10 tests, with the same number of function evaluations assigned to each algorithm. These results show that even the worst results produced by the systematic change of neighborhood, such as performed within VNS and VN-MGA, are better than the best values of any single-level algorithm.

Comparing the variable-neighborhood algorithms, one observes that the VN-MGA attains better hypervolume values than the VNS. This suggests that the



(a) R=10 (b) R=20

(c) R=30 (d) R=40

Fig. 13. Comparison between e-constraint and the VN-MGA algorithm. Circles represent the solutions achieved by the ϵ-constraint ILP technique. The solutions delivered by VNS are represented by black crosses and the solutions of the proposed VN-MGA are represented by red asterisks. Instance #2, R=10, R=20, R=30, R=40

crossover operators employed by the VN-MGA play an important role in the search for good solutions in the problems examined in this study.

Table 3. Hypervolume metric for different algorithms.

Hypervolume	Level 1	Level 2	VNS	VN-MGA
Best value	21.7	23.95	34.62	37.67
Worst value	17.55	20.68	24.81	26.04
Average value	20.62	22.41	28.57	30.37
Standard Deviation	1.42	1.14	3.6	5.15

6 Conclusions and Future Work

This paper proposed an algorithm to optimize multiple objectives that represent Quality of Service indices on IP networks. The proposed algorithm, VN-MGA, is a Genetic Algorithm based on the NSGA-II, with the particular feature that each solution has two different encodings, at Level 1 and Level 2. At Level 1, the solution is encoded considering as decision variables the edges that form the routes to be followed by each request. At Level 2, the solution is encoded with the routes considered as fixed, and the sequence of requests considered as the decision variable. The results suggest that local minima can be indeed avoided using this approach.

The good behavior presented by the proposed VN-MGA, outperforming both some algorithm versions that do not employ variable neighborhoods and a basic VNS algorithm, raises an interesting issue to be studied: should the evolutionary algorithms specialized in combinatorial optimization problems employ variable neighborhood operators as a standard methodology? The authors intend to investigate this issue in the near future.

Concerning the MPLS problem, a challenging area of future work concerns a quantitative analysis, covering sensitivity and scalability. The sensitivity deals with fault tolerance in paths or routers and the capacity of re-routing of the proposed method. Using new scenarios, it is possible to assess the scalability in order to quantify the gain that is expected with the application of the proposed algorithm. Within this perspective, it is also possible to suggest new models for telecommunication networks. In any case, the proposed approach delivers a reasonable diversity of solutions belonging to the Pareto Front. So, it offers a larger range of options for the decision maker in different situations, such as in: (i) network congestion that occur in rush moments, or (ii) using applications that require a small delay, or (iii) responding to concurrent requests that do not present stringent requirements of delay, but require large bandwidths, among others.

Bibliography

Alvarado, C., Herazo, I., Ardila, C. and Donoso, Y. (2005). Aplicación de NSGA-II y SPEA-II para la optimización multiobjetivo de redes multicast, *Ingeniería y Desarrollo* **17**: 28–53.

Andrade, A. V. (2008). *Provisionamento de Qualidade de ServiÃ§o em Redes MPLS utilizando Algoritmos Bio-inspirados em um Ambiente de TrÃ¡fego Auto-Similar*, PhD thesis, UFMG.

Awduche, D., Chiu, A., Networks, C. and Elwalid, A. (2002). Overview and Principles of Internet Traffic Engineering, *Request for Comments* **3272**.

Carrano, E. G., Soares, L. A. E., Takahashi, R. H. C., Saldanha, R. R. and Neto, O. M. (2006). Electric distribution network multiobjective design using a problem-specific genetic algorithm, *IEEE Trans. Power Delivery* **21**(2): 995–1005.

Carrano, E. G., Takahashi, R. H. C., Fonseca, C. M. and Neto, O. M. (2010). Non-linear network topology optimization - An embedding vector space approach, *IEEE Transactions on Evolutionary Computation* **14**: 206–226.

De Giovanni, L., Della Croce, F. and Tadei, R. (2004). On the impact of the solution representation for the Internet Protocol Network Design Problem with max-hop constraints, *Networks* **44**(2): 73–83.

de Souza Filho, E. M. (2007). *Variable Neighborhood Search (VNS) aplicado ao problema de distribuição dutoviária*, Master's thesis, UFRJ - Universidade Federal do Rio de Janeiro.

Deb, K., Pratap, A., Agrawal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii, *IEEE Trans Evol Comput* **2**(6): 182–197.

Dias, R. A. (2004). *Engenharia de Trafego em Redes IP sobre Tecnologia MPLS: Otimizaçao Baseada em Heurısticas.*, PhD thesis, UFSC.

Ehrgott, M. (2000). *Multicriteria Optimization*, Spring-Verlag.

Erbas, S. and Erbas, C. (2003). A multiobjective off-line routing model for MPLS networks, *Providing Quality of Service in Heterogeneous Environments-Proc. of the 18th International Teletraffic Congress (ITC-18)*, Vol. 16, Citeseer, pp. 471–480.

Fonseca, C. M., Paquete, L. and López-Ibáñez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator., *IEEE Congress on Evolutionary Computation*, pp. 1157–1163.

Freitas, Lia Mara Borges de ; Montané, F. A. T. (2008). Metaheurísticas vns-vnd e grasp-vnd para problemas de roteamento de veículos com coleta e entrega simultânea., *XI Simpósio de Pesquisa Operacional e Logística da Marinha*.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Hansen, P. and Mladenovi, N. (2001). Variable Neighborhood Search: Principles and applications, *European Journal of Operational Research* **130**(3): 449–467.

Maia, N. A., de Errico, L. and Caminhas, W. M. (2007). Combining MPLS, Computational Intelligence, and Autonomic Computing into a Self-Managing Traffic Engineering System., *Proceddings of the Second Latin American Autonomic Computing Symposium LAACS 2007* pp. 1–6.

Mladenovi, N. and Hansen, P. (1997). Variable neighborhood search, *Computers & Operations Research* **24**(11): 1097–1100.

Onety, R. E., Moreira, G. J. P., Neto, O. M. and Takahashi, R. H. C. (2011). Variable neighborhood multiobjective genetic algorithm for the optimization of routes on IP networks, *Proc. 6th International Conference on Evolutionary Multicriterion Optimization (EMO 2011)*, Vol. LNCS 6576, Ouro Preto, Brazil, pp. 433–447.

Paul, P. and Raghavan, S. (2002). Survey of QoS routing, *Proceedings of the international conference on Computer communication*, Vol. 15, Citeseer, p. 50.

Perboli, G., Pezzella, F. and Tadei, R. (2008). EVE-OPT: a hybrid algorithm for the capacitated vehicle routing problem, *Mathematical Methods of Operations Research* **68**(2): 361–382.

Santos, F. A. (2009). *Otimização Multiobjetivo aplicada a alocação dinâmica de rotas em redes de telecomunicações*, Master's thesis, UFMG.

Wang, Z. and Crowcroft, J. (1996). Quality-of-service routing for supporting multimedia applications, *IEEE Journal on Selected areas in communications* **14**(7): 1228–1234.