

FEDERAL UNIVERSITY OF MINAS GERAIS
GRADUATE PROGRAM IN ELECTRICAL ENGINEERING
POLITECNICO DI TORINO
DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING

Variable Neighborhood Multiobjective Genetic Algorithm for the Optimization of Routes on IP Networks

Renata da Encarnação Onety

Prof. Ricardo Hiroshi Caldeira Takahashi
Prof. Roberto Tadei

A thesis presented to the Graduate Program in
Electrical Engineering of the Federal University of
Minas Gerais for the degree of Doctor in Electrical
Engineering and Department of Control and Com-
puter Engineering of Politecnico di Torino for the
degree of Ph.D in Computer Engineering

2013

To Leonardo, my love, my life, my son.

Abstract

The demand for different levels of Quality of Service (QoS) in IP networks is growing, mainly to attend multimedia applications. However, not only indicators of quality have conflicting features, but also the problem of determining routes covered by more than two QoS constraints is NP-complete (Nondeterministic Polynomial Time Complete). This work proposes an algorithm to optimize multiple Quality of Service indices of Multi Protocol Label Switching (MPLS) IP networks. Such an approach aims at minimizing the network cost and the amount of simultaneous requests rejection, as well as performing load balancing among routes. The proposed algorithm, the Variable Neighborhood Multiobjective Genetic Algorithm (VN-MGA), is a Genetic Algorithm based on the Elitist Non-Dominated Sorted Genetic Algorithm (NSGA-II), with a particular feature that different parts of a solution are encoded differently, at Level 1 and Level 2. In order to improve results, both representations are needed. At Level 1, the first part of the solution is encoded 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), whereas at Level 2, the second part of the solution is encoded by considering the sequence of requests as decision variables, and first part is kept constant. Pareto-fronts obtained by VN-MGA dominate fronts obtained by fixed-neighborhood encoding schemes. Besides potential benefits of the proposed approach application to packet routing optimization in MPLS networks, this work raises the theoretical issue of the systematic application of variable encodings, which allow variable neighborhood searches, as operators inside general evolutionary computation algorithms.

Keywords: Routing, Multiobjective Genetic Algorithm, Variable encoding.

List of Figures

2.1	Packet routing in an intra-domain network, represented by the cloud. In (a), the routing is done with no TE. Packets follow the shortest path, causing an overload of the link. In (b), the routing has an implementation of TE. It tries to distribute the flow over the available links, aiming at improving QoS (Maia, 2006).	8
2.2	The general model of QoS within the approaches provided by ETSI / ITU and IETF. It represents three QoS definitions for a general model: intrinsic, evaluated and perceived. The intrinsic QoS is obtained by a comparison of measured and expected performance characteristics. The perceived QoS reflects the user experience by using a specific service. The evaluated QoS begins to be analyzed when the customer decides whether to continue to use this service (Andrade, 2008; Maia, 2006).	14
2.3	The seven layers of OSI model.	16
2.4	The MPLS layer and its fields. The MPLS header is positioned after Layer 2 header and before Layer 3 header. The label is a short identifier that is assigned to each packet when it enters in the MPLS network. TC is the Traffic Class, that defines classes of services. S refers to stack, that supports queuing labels if the package receives more than one label. TTL refers to Time To Live, that counts the number of routers the packet has passed.	18

2.5	Routing packets in the MPLS domain. (Adapted from Maia (2006)). The cloud represents the MPLS domain. The ingress LER associates the IP packet to a FEC, explicitly stating the path to be followed - for instance, LSP1 or LSP2. Finally, the egress LER removes the label of the IP packet and it is forwarded on its way via standard IP routing. . .	19
4.1	Mapping from decision variable space to the objective space (Parreiras, 2006).	28
4.2	Non-dominated points. (i) There is no dominance relation between the points A, B and E. (ii) A dominates C and F. (iii) B dominates C, D and F. (iv) Between C, D and E there is no dominance relations. (v) C and D dominate F. (vi) E and F do not dominate any other point shown in Figure (Adapted from Takahashi (2007)).	29
4.3	Representation of two points of the Pareto-optimal set \mathcal{P} attained with ϵ -constraint method. A is the minimization of f_1 s.t. $f_2 \leq \epsilon_2$. B is the minimization of f_2 s.t. $f_1 \leq \epsilon_1$ (Takahashi, 2007).	30
4.4	The ϵ -constraint method guarantees weak Pareto optimality (Cerav- Erbas, 2004).	31
4.5	Diagram of Genetic Algorithm. It consists of an initial population that has its symbol representation decoded into variable values. With each variable, it is possible to assess the objective function of the problem. If it is according to a convergence criterion, the algorithm is concluded. Otherwise, it evolves again and again through genetic operators such as selection, crossover and mutation.	33
4.6	Genetic Encoding. The sequence of characters can be decoded into the problem variables (phenotype). Each character of the chromosome corresponds to a gene. The concatenation of all chromosomes belong- ing to an individual represents the genotype. The decoded genotype represents the phenotype (Parreiras, 2006).	33
4.7	Crossover with one cutoff point. A and B represent the parents indi- vidual. A' and B' represent the offspring individual generated from the crossover of A and B. K indicates the cutoff point.	34
4.8	Mutation of one bit. A represents the initial individual. K indicates the random chosen bit. A' represents the individual after mutation.	34
4.9	Diagram showing the steps of NSGA-II	36
4.10	NSGA-II procedure (Deb <i>et al.</i> , 2002).	37
5.1	Instances of networks (Santos, 2009).	44

5.2	Flowchart of VN-MGA	47
5.3	Representation of solution encoding in two operation levels. In this figure, the individual I_1 processes a sequence of requests starting with R_1 followed by R_2 , and individual I_n processes a sequence of requests that starts with R_2 followed by R_m . In individual I_1 the request R_1 has origin in node 1 and destination in node 5, with a path that comprises nodes 2 and 3. In individual I_n the request R_1 has origin in node 1 and destination in node 5 (the origin and destination nodes are necessarily the same ones), with a path that comprises only node 4.	48
5.4	Representation of candidates paths in a given instance. In (a), the representation of the network instance. In (b), nodes 1 and 5 are source and destination respectively. In (c), 1 - 2 - 3 - 5 is a candidate path. In (d), 1 - 4 - 5 is another candidate path.	49
5.5	Representation of candidates paths in a given instance. In (a), the representation of the network instance. In (b), nodes 4 and 6 are source and destination respectively. In (c), 4 - 5 - 6 is a candidate path. In (d), 4 - 2 - 3 - 5 - 6 is another candidate path.	50
5.6	Level 1 Crossover: a common node n_c exists (Santos, 2009). (a) The route of parent I_1 for request r . (b) The route of parent I_2 for request r . (c) The resulting route in the offspring for request r	53
5.7	Level 1 Crossover: no common node (Santos, 2009). (a) The route of parent I_1 for request r . (b) The route of parent I_2 for request r . (c) The resulting route in the offspring for request r	53
5.8	Level 2 Crossover. (a) Request R_2 is randomly selected. (b) The routes of request R_2 in I_1 and I_2 are interchanged, generating the offspring. .	54
5.9	Level 1 Mutation. (a) The route for request r . (b) Random selection of two intermediate nodes which belong to the route for a new sub-route. (c) Mutated individual.	55
5.10	Level 2 Mutation. (a) Sequence of requests $R_1 \dots R_5$ from an individual I_1 . (b) Selected requests for mutation: R_1 and R_4 . (c) Mutated individual.	55
6.1	Comparison between VN-MGA (represented by asterisks) and ILP ϵ -constraint (represented by circles) results for 10, 20, 30 and 40 simultaneous requests for the network instance represented in Figure 5.1(e).	59

6.2	Comparison among ϵ -constraint, VNS and VN-MGA algorithm for R=30. 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. (Horizontal axis: F_1 . Vertical axis: F_3	62
6.3	Comparison among VN-MGA, single-level encoding GAs and the basic VNS for the network instance of Figure 5.1(d). Circles represent solutions achieved by Level 1 GA. Triangles represent solutions obtained by Level 2 GA. Crosses indicate solutions provided by VNS. Asterisks indicate solutions provided by VN-MGA.	63
6.4	Comparison among VN-MGA, single-level encoding GAs and the basic VNS for the network instance of Figure 5.1(e). Circles represent solutions achieved by Level 1 GA. Triangles represent solutions obtained by Level 2 GA. Asterisks indicate solutions provided by VN-MGA. . . .	64
6.5	Boxplot results of the hypervolume metric after 20 runs of algorithms with Level 1, Level 2, VNS and VN-MGA, for 40 simultaneous requests.	65

List of Algorithms

1	Iterated Local Search	39
2	Basic VNS	42
3	Solution Decoding	50
4	Level 1 Crossover	52
5	Level 2 Crossover	52
6	Level 1 Mutation	54
7	Level 2 Mutation	54
8	VN-MGA pseudocode	57

List of Acronyms and Abbreviations

AS	Autonomous Systems
Diffserv	Differentiated Services
ETSI	European Telecommunications Standards Institute
FEC	Forward Equivalence Class
GA	Genetic Algorithm
GEP	Gene Expression Programming
GRASP	Greedy Randomized Adaptive Search Procedure
IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
ILS	Iterated Local Search
Intserv	Integrated Services
IP	Internet Protocol
ITU	International Telecommunications Union
LER	Label Edge Router
LIB	Label Information Base
LSP	Label Switched Path
LSR	Label Switch Router
MLP	Multilayer Perceptron
MPLS	Multi Protocol Label Switching
NP-complete	Nondeterministic Polynomial Time Complete
NSGA	Non-Dominated Sorting Genetic Algorithm
NSGA-II	Elitist Non-Dominated Sorted Genetic Algorithm
OSI	Open Systems Interconnection
PAES	Pareto Archived Evolution Strategy
QoS	Quality of Service

RFC	Request for Comments
RSVP	ReSource Reservation Protocol
S	Stack
SPEA	Strength Pareto Evolutionary Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm Improved
TC	Traffic Class
TE	Traffic Engineering
TS	Tabu Search
TTL	Time To Live
VN-MGA	Variable Neighborhood Multiobjective Genetic Algorithm
VND	Variable Neighborhood Descent
VNS	Variable Neighborhood Search
VoD	Video on Demand
VoIP	Voice over IP

Contents

1	Introduction	1
1.1	Motivation	4
1.2	Objectives	4
1.3	Thesis Contributions	5
1.4	Document Outline	5
2	Traffic Engineering and Quality of Service on IP Networks	7
2.1	Traffic Engineering	7
2.1.1	Context of TE	10
2.2	IP protocol	12
2.2.1	Quality of Service on IP networks	13
2.2.2	Techniques for Obtaining Quality of Service	14
2.3	The MPLS Technology	16
3	Related works	21
3.1	Traffic Engineering and MPLS Technology	21
3.2	Routing applied to IP networks	22
3.3	Multiobjective optimization of networks	23
3.4	Hybridization of algorithms in routing problems	24
3.5	Considerations	25
4	Optimization and Algorithms	26
4.1	Optimization	26
4.1.1	Multiobjective optimization problems	27
4.2	Scalarization Methods	29
4.2.1	The ϵ -constraint method	30

4.3	Evolutionary Computation	31
4.3.1	Genetic Algorithms	32
4.3.2	Multiobjective Genetic Algorithms	35
4.3.3	Decision Making	36
4.4	Local Search Algorithms	38
4.4.1	Iterated Local Search	38
4.4.2	Variable Neighborhood Search	40
5	The Variable Neighborhood Multiobjective Genetic Algorithm: VN-MGA	43
5.1	Problem definition	43
5.2	Model Formulation	44
5.3	The Implementation	46
5.3.1	Solution Encoding	46
5.3.2	Solution Decoding	49
5.3.3	Initial population	51
5.3.4	Crossover Operators	51
5.3.5	Mutation Operators	51
5.3.6	VN-MGA Structure	54
6	Results	58
6.1	Validating the implementation	58
6.2	Validating the proposed algorithm	59
6.2.1	Comparison with Integer Linear Programming (ILP) results . .	60
6.2.2	Comparison among VN-MGA, single-level encoding GAs and the basic VNS	62
6.2.3	Hypervolume metric	63
7	Conclusions and Future work	66
7.1	Conclusion	66
7.2	Limitations	67
7.3	Future works	68
7.4	Publications	68
	Bibliography	70

Introduction

With the emergence of new technologies, the transmission of multimedia applications has become an achievable goal. New applications such as videoconferences, Video on Demand (VoD) or Voice over IP (VoIP) brought the need of some guarantees of network characteristics regarding 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 packets, i.e, there is no guarantee of communication regularity. For this reason, some mechanisms were developed for Quality of Service (QoS). They allow differentiation of the transmitted flows. They also allow 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 & Raghavan, 2002).

Recently, several technologies have been proposed to develop approaches of Traffic Engineering for Routing with QoS. According to RFC-3272 (Request for Comments 3272), the Internet Traffic Engineering is defined as an aspect of Internet network engineering that deals with the issue of performance evaluation and optimization of operational IP networks (Awduche *et al.* , 2002). Many of these studies deal with routing on IP networks and MPLS (Multi Protocol Label Switching), using single-objective Genetic Algorithms (GAs) (Maia *et al.* , 2007; Andrade, 2008) or deterministic methods, such as Lagrangian Relaxation (Dias, 2004). As the model of these studies is formulated with a single objective, search can be biased to a specific goal, leading to solutions that are unsuitable under other objective viewpoint. For this reason, multiobjective strategies have been receiving some attention. However, the use

of multiobjective methods applied to the problem of routing on IP networks is not extensive. Nevertheless, most of works perform the optimization of two objectives (Alvarado *et al.* , 2005). Other studies use deterministic methods (Erbas & Erbas, 2003). The current study deals with the optimization of three objectives which, according to (Wang & Crowcroft, 1996), render the problem NP-complete (Nondeterministic Polynomial Time Complete). For this reason, techniques based on non-deterministic heuristics are likely to be the most suitable ones. Santos (2009) proposes a dynamic evaluation for routing in an ambient of MPLS using multiobjective techniques. That research represents an initial reference for the present work, employing the same scenarios and objective functions.

MPLS is an example of a technology that makes possible the explicit routing of packets. It facilitates the provisioning of QoS according to requirements of multimedia applications. This technology allows the addition of labels to packets in order to identify them.

A possible way to deal with various requirements of different applications is the use of search strategies for finding optimal or suboptimal solutions. 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 candidates solutions (each solution is an *individual*, and the set is the *population*), which vary according to two kinds of probabilistic rules: *mutation* and *crossover*. Mutations introduce perturbations into current solutions, producing new ones. Crossovers combine the information from previous solutions, producing new ones. The current population goes finally through 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 representation of solutions in the population. This is due 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 & Hansen, 1997), however, usually evolve a single solution each time. This solution is subjected to heuristic descent searches that find local minima in attraction regions. The regions are characterized by connected paths in a given neighborhood. It allows the algorithm to perform further descent steps after finding a local minimum in a neighborhood, by simply the neighborhood.

This work deals with the problem of packet routing in MPLS networks. 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 conflicting QoS indicators, described as independent objective functions. Moreover, the solution sets that are delivered as solutions for the multiobjective problem provide flexibility for the decision maker to select a specific solution that fits an operation 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 requirement of service, and a high-level encoding, that encodes the permutations of the several requirements of service, defining the order in which they will be included in the solution. Crossover and mutation operators, acting in 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).

The proposed algorithm is also compared with: (i) a pure VNS algorithm which employs the same neighborhoods employed in the proposed VN-MGA along with a typical VNS neighborhood change mechanism; and (ii) an Integer Linear Programming (ILP) solver. Both algorithms, which are scalar in their original formulation, are adapted to the multiobjective problem using an ϵ -constraint procedure (Ehrgott, 2000). The proposed VN-MGA reaches better solutions, when compared with the VNS, and requires a much smaller computational effort, compared with the ILP.

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) (de Freitas & Montané, 2008). Also problems with other characteristics, such as pipeline petroleum distribution using GA and VNS (de Souza Filho, 2007), should be mentioned. Those studies typically combine different algorithms in a literal way, performing steps from one and other algorithm.

Some preliminary results presented in this thesis have been published in a con-

ference proceedings (Onety *et al.* , 2011), and a more complete version have been published in a journal (Onety *et al.* , 2013).

1.1 Motivation

Currently, several technologies have been proposed in order to enable the recognition of the type of information that is transmitted over IP networks and, therefore, to offer support for the QoS requirements. The technology called MPLS, for instance, is an effective alternative, since it allows the determination of explicit routes. With the assignment of routes, it helps the QoS provisioning with the view to meet the requirement of multimedia applications. This technology allows the addition of labels to packages, identifying them. Thereafter, it makes possible to perform routing, considering several indicators of QoS.

However, different QoS indicators that should be enhanced have conflicting nature. The minimization of delay, for example, may generate an unbalanced load distribution due to the concentration of packages on the shortest path links. Moreover, the problem of determining optimal routes under two or more QoS constraints can become NP-complete (Wang & Crowcroft, 1996). In this way, the use of deterministic methods is not an efficient approach for routing. In the specific case of multimedia applications, it is desirable the assignment of reliable and delay-free routes for the transmission of information, in order to enable those applications. These routes should consider multiple QoS requirements, many of them conflicting, such as reducing delays and performing network load balancing.

A viable alternative to deal with the various conflicting requirements of different applications is the implementation of search strategies for finding the optimal or suboptimal solutions based on heuristics. Through heuristics, it is possible to find solutions for which the feasibility is guaranteed with a reasonable computational effort. Multiobjective Genetic Algorithms and Variable Neighborhood Search are some examples of heuristic search strategies.

1.2 Objectives

This work proposes an approach for the optimization of routes on IP networks in an MPLS domain, based on a Traffic Engineering methodology that is capable of controlling some Quality of Service (QoS) parameters. In particular, it proposes specific genetic operators for the problem of ensuring QoS in IP networks, aiming to increase the probability of generation of feasible individuals during the evolution process.

The methodology is based on a Multiobjective Genetic Algorithm. As an innovative feature, it has specific genetic operators that consider different neighborhood structures, such as suggested by the technique of Variable Neighborhood Search. Thus, using this change of neighborhood structure, it can explore new areas, not glimpsed before.

This thesis assumes an MPLS-based environment due to its explicit routing with labels on its packages. This environment consequently favors the QoS. The use of multiobjective genetic algorithms, in turn, can deal with different QoS indicators of conflicting nature. These indicators are described as independent objective functions, i.e., there is no need to adopt a single goal to weight the different metrics. Moreover, the fact that multiobjective algorithms deliver sets of solutions at the same time, can offer flexibility to the decision maker, which can analyse the current state of the network and decide in each moment which objective should be privileged, by choosing different solutions.

The definition of a new method for route generation is a contribution of this work. The proposed method exploits the fact that, depending on the request order, the route generation can privilege a particular request that could be impaired if it were in a different priority. The variation of request order, therefore, increases the diversity of solutions, allowing the development of new possibilities of routes.

1.3 Thesis Contributions

As result of this work, we expect to provide a viable methodology for package routing in MPLS IP networks.

The particular feature of the proposed approach is that solutions are encoded such that different neighborhoods become defined, allowing the application of principles of VNS algorithms. In this way, specific VNS-based operators for the GA are able to perform global search with acceptable computational cost.

In addition to the technological contribution, we expect that the proposed concept of variable-neighborhood operators for genetic algorithms might become a relevant contribution to the theory of evolutionary computation algorithms.

1.4 Document Outline

This thesis is organized as follows. In Chapter 2, we introduce the basic concepts necessary for understanding the purpose of this research, such as Traffic Engineering, Quality of Service and MPLS technology. We describe, in the context of Traffic

Engineering, some techniques for obtaining Quality of Service.

In Chapter 3, we present basic concepts about multiobjective optimization. We also describe a scalarization method, called ϵ -constraint, which may be solved exactly by ILP solvers in the case of mixed-integer linear programming problems, which will be used for extracting benchmarks for the comparison of the proposed method. Afterward, we present some discussion about evolutionary computation methods, including genetic algorithms in single-objective and multiobjective versions. In addition, we present some discussion about meta-heuristics, focusing on Variable Neighbourhood Search, which is the main inspiration of the proposed method. Finally, we present a brief discussion about the issue of Decision Making, in order to contextualize the proposed method.

In Chapter 4, we mention some related works that have been done, considering Evolutionary Computation, MPLS, Routing and hybrid methods.

In Chapter 5, we describe the problem tackled in this thesis and its modeling. We present the VN-MGA, its operators, and the corresponding algorithms.

In Chapter 6, we present some results obtained with this approach. We compare them with other algorithms.

Finally, conclusions and future developments of the research activity are reported in Chapter 7.

Traffic Engineering and Quality of Service on IP Networks

This chapter introduces basic concepts from the field of Telecommunication Networks. They are necessary for understanding the purpose of this research, such as Traffic Engineering, Quality of Service and MPLS technology.

2.1 *Traffic Engineering*

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. Its mission is to make the Internet work better by producing high quality, relevant technical documents that influence the way people design, use, and manage the Internet (IETF, 2013). Recommendations of IETF are usually published in documents called RFCs (Request for Comments). According to the scope of this work, we are interested in principles of Traffic Engineering.

Traffic Engineering (TE) comprises a set of techniques for the management of telecommunication networks. By arranging the traffic distributions, it aims at reducing congestion, instability or impaired Quality of Service (Maia, 2006). According to definitions of RFC-3272 (Awduche *et al.* , 2002), TE is an aspect of Network Engineering that deals with issues of evaluation and optimization of performances of IP

networks. It is capable of applying technologies and scientific principles in order to measure, to characterize, to model and to control the network traffic. Performance measures include delay, delay variation (jitter), packet loss and throughput.

Figure 2.1 illustrates the packet forwarding with and without TE. In a domain that does not implement TE, all packets must be forwarded along the shortest path, in this case, the middle way, which often results in congestion. The existence of congestion may cause QoS degradation (Maia, 2006).

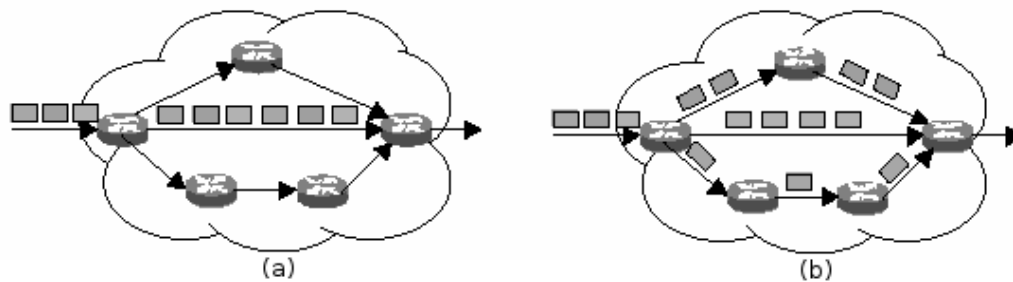


Figure 2.1: Packet routing in an intra-domain network, represented by the cloud. In (a), the routing is done with no TE. Packets follow the shortest path, causing an overload of the link. In (b), the routing has an implementation of TE. It tries to distribute the flow over the available links, aiming at improving QoS (Maia, 2006).

Among many objectives of TE, it aims at facilitating reliable operations, reducing the vulnerability to errors, faults and failures that can occur in the infrastructure. Another important goal of TE is to perform the control and optimization of the routing function when it sends the traffic through the most effective way of network. Considering the optimization, it refers to the management of capacity and traffic. Capacity management includes capacity planning, routing control and resource management such as bandwidth, buffer size and computational resources. Traffic management refers to the analysis of traffic conditions, queue management, scheduling and other functions that regulate traffic flow through the network (Awduche *et al.*, 2002).

Summarizing the goals, TE includes four basic problems, that is, admission control of new connections, constraint-based routing, re-routing of established connections, and network resources planning (Maia, 2006).

The first problem, admission control, determines if a request can be admitted or not. If so, it selects a route for this connection over the network.

The second problem, constraint-based routing, is responsible for the selection of optimal paths that satisfy a given set of constraints and requirements. Metrics used for constraint-based routing include cost in money, number of intermediate nodes, transmission rate, reliability, delay and jitter. The first step of this process aims at deleting links with not enough bandwidth for new connections or that somehow do

not meet the requirements of incoming demand. The second step consists of choosing the “best path” to be used on the residual network. The best path is not necessarily the shortest path calculated by an algorithm such as Dijkstra’s, but a path that meets all the requirements (Awduche *et al.* , 1999).

The third problem, re-routing of established connections, is the process in which some traffic flows are re-routed in order to improve network efficiency. The network administrator may periodically modify the routing of traffic flows due to new network conditions. For example, a traffic flow can be re-routed into a secondary path in case of failure in the primary path, or in the case of a higher priority traffic using the main path resources. When a fault occurs, if there were no re-optimization, the network could become increasingly distant from the optimal point (Andrade, 2008).

The last problem, network resource planning, includes planning of link capacities and sizes of buffers. It aims at providing the network ability to comply with new traffic demands. The planning of network resources should consider the possibility of future demands, which can be estimated from the historical data of network traffic (Girish *et al.* , 2000).

The RFC-3272 states that TE has not a goal that is reached only once. Instead, its achievements are continuous, obtained in an iterative process. The reason for this is that the optimization objectives may change over time whenever new requirements are imposed. Due to the complexity of those tasks, the TE requires a continuous development of new technologies and methodologies for enhancing network performance.

TE can also be seen from a control perspective. This aspect of control can be classified as pro-active and reactive. Pro-active control consists of a preventive action that avoids unfavorable network states. Reactive control responds correctively and adaptively, dealing with events that have already occurred in the network.

One of the major challenges of TE is the capability of automated control and of performing adjustments for significant changes in the current network state, while keeping its stability. Therefore, it is important to assess its performance in order to determine the effectiveness of the chosen methods. Results of assessments can help to identify problems and to predict potential problems. Moreover, they can guide the network in a re-optimization procedure. This evaluation can be achieved in many different ways. The main techniques include analytical methods, simulation and empirical methods based on measurements. In the case of analytical methods or simulations, the network nodes and links are modeled in order to capture relevant operational features such as topology, bandwidth, buffer space, and nodal service policies (link scheduling, packet prioritization, buffer management, etc.). Analytical

traffic models can be used to describe dynamic characteristics and behavior of traffic, such as burstiness, statistical distributions and dependence.

Another important objective of TE is to combat congestion at a reasonable cost. A network resource is considered congested if the arrival packet rate exceeds the output capacity of the resource over an interval of time. Congestion may result in some of the arrival packets being delayed or even dropped. Congestion increases transit delays, jitter, packet loss, and reduces the predictability of network services (Awduche *et al.* , 2002).

Considering that performance evaluation is not a trivial issue, several techniques are used to simplify the analysis, such as abstraction, decomposition and approximation. A simplification of concepts such as available bandwidth and effective buffer, for example, can approximate nodal behaviors at packet level and simplify the analysis at connection level. For instance, techniques of network analysis using queuing models and approximation schemes based on asymptotic and decomposition techniques can render the analysis more tractable.

Simulations can be used to evaluate the network performance or to verify and validate analytical approximations. However, they can be computationally expensive and not very clear. Thus, the analysis of network performance should involve a hybrid combination of analytical techniques, simulations and empirical methods.

As a general rule, according to the RFC 3272, concepts and mechanisms of TE must be sufficiently specific and well-defined for known requirements and, at the same time, flexible and extensible to future demands. For this purpose, it is necessary to clearly define the context in which the TE is applied, specifying the appropriate rules for each question. Hereafter, we present the contexts that are applied in Chapter 4 to the problem addressed in this study.

2.1.1 Context of TE

The use of a TE methodology is important to establish scenarios in which the rules of Traffic Engineering are applied. Awduche *et al.* (2002) proposed in RFC-3272 the use of contexts, which delimit issues to be addressed and how they should be evaluated.

Network Context

According to Awduche *et al.* (2002), *Network Context* defines the universe of discourse and, in particular, the situation which the TE problem occurs. This context includes the structure, network policies, constraints and characteristics, quality at-

tributes and optimization criteria.

Network elements and resources may have specific characteristics that restrict the manner that demands are handled. Network resources may be equipped with traffic control mechanisms, which should be used to control, for example, various packet processing activities within a given resource, to arbitrate contention for access to the resource by different packets, and to regulate traffic behavior through the resource. Details about packet transport are specified in the policies of network administrators. Types of services offered by the network generally depend on the technology and characteristics of the network elements and protocols.

Problem Context

The *Problem Context* defines general and specific issues that the TE deals, such as the identification, abstraction of relevant issues, representation, formulation, specification of requirements on the solution space and specification of the desired features of acceptable solutions.

Important questions to be taken into account are how to formulate explicitly the problems that the TE may solve, how to identify the requirements on the solution space, how to specify the desirable characteristics of good solutions, and how to measure and to characterize the effectiveness of solutions. Another class of problems is how to measure and to estimate relevant network parameters and how to characterize the state of the network in order to evaluate its performance in a variety of scenarios. There is still another class of problems that concerns on how to effectively optimize network performance. Performance optimization may entail translating solutions to specific traffic engineering problems into network configurations. Optimization may also entail some degree of resource management control, routing control, and capacity augmentation.

Solution Context

The *Solution Context* indicates how to address the issues identified by the Problem Context, including analysis, evaluation of alternatives, prescriptions and resolutions. It demands reasonable estimates of traffic load, characterization of network state, in this way deriving solutions for the problems of TE that can be implicitly or explicitly formulated.

Implementation and Operation Context

The *Implementation and Operation Context* indicates where the solutions are methodologically instantiated. It includes planning, organization and execution. This context is characterized by constant changes that occur at multiple levels of abstraction. Aspects of planning should involve a set of priority actions to achieve certain goals. The organization involves assigning responsibility to the various system components of TE and coordinating their activities. Execution involves measuring and applying corrective actions when they are necessary.

2.2 IP protocol

The Internet can be described as a collection of Autonomous Systems (AS) interconnected by backbones. Each AS is an independent node or a sub-network that defines its own rules such as routing policy, internal structures and access technologies. A protocol that enables the communication between each AS is called Internet Protocol (IP). IP is a protocol that has the task of delivering packets from the source to the destination host based on their addresses. For this purpose, IP defines datagram structures that encapsulate the data to be delivered. It provides the service of best effort, i.e., it does the best service as possible, but with no guarantee of transport or sequence, with a variable bit rate and with delay in delivery time (Andrade, 2008; Maia, 2006).

According to Awduche *et al.* (2002), in a very basic abstraction level, an IP network may be represented as a distributed dynamic system, consisting of:

1. A set of interconnected resources that provides transportation services of IP traffic subject to certain constraints;
2. A demand system representing the offered load to be transported through the network;
3. A response system consisting of network processes, protocols, and related mechanisms that facilitate the movement of traffic through the network.

To provide a successful multimedia network, many studies have focused on the issue of how to ensure the quality of services on IP networks. We discuss such an issue in next section.

2.2.1 *Quality of Service on IP networks*

With the increment of data-intensive technologies such as multimedia applications and of the capacity of networks, new applications have appeared requiring a better network performance. However, the transportation of multimedia data is not trivial. Apart from bandwidth, there are constraints related to maximum delay, maximum jitter and packet loss. For the success of a multimedia network, major innovations in the Quality of Service are essential.

Many multimedia applications are known as constant rate media applications. Nevertheless, there is another class of applications which deal with variable rate transmission. Several classes of multimedia applications share resources of a network among themselves and with data applications. Some examples are applications of streaming audio and video such as Internet radio, recorded lectures, interactive applications such as IP telephony and video conferencing. Thus, classes of multimedia applications have different requirements and should be treated in a particular way, according to their needs. Each source sends a stream of packets to a given destination. The needs of each flow can be characterized by four parameters: reliability, delay, jitter and bandwidth. These parameters define the Quality of Service for each flow requirement (Andrade, 2008).

Reliability can be described as the acceptable percentage of lost packets for an application. Delay is defined as the time that a packet takes to reach its destination. It is a result of queuing in routers, propagation time and processing time along the path from source to destination. Jitter delay or fluctuation is defined as the difference between arrival times of successive packets. The bandwidth or throughput is the amount of information sent or received per second (Andrade, 2008).

The definition of QoS through parameters of delay, jitter, throughput and packet loss is not the only possible one. Although this definition is well-established, there are other approaches provided by International Telecommunications Union (ITU), European Telecommunications Standards Institute (ETSI) and the Internet Engineering Task Force (IETF). According to ITU/ETSI, there are three QoS definitions for a general model: intrinsic, evaluated and perceived. Figure 2.2 presents the general model of QoS.

The intrinsic QoS is determined by the network project of transmission mechanisms and provisioning of access networks, terminals and connections. It is obtained by a comparison of measured and expected performance characteristics. Another aspect that should be emphasized is that the intrinsic QoS is not affected by user perception (Andrade, 2008; Maia, 2006).

The perceived QoS reflects the user experience by using a specific service. This

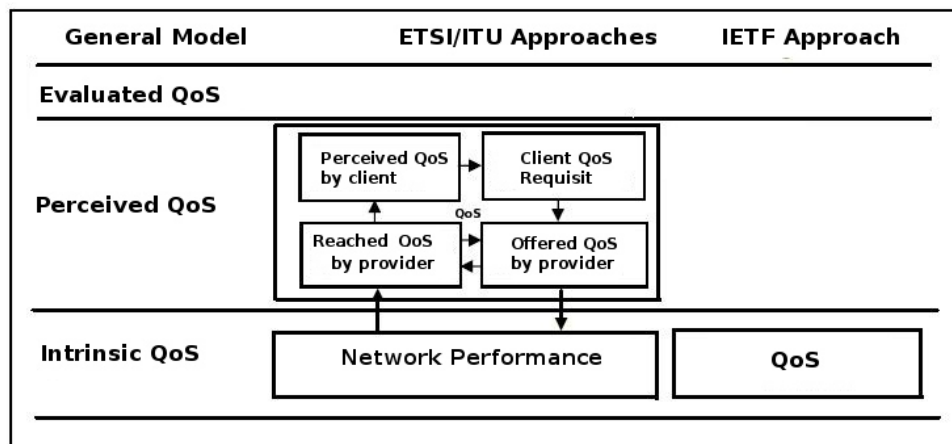


Figure 2.2: The general model of QoS within the approaches provided by ETSI / ITU and IETF. It represents three QoS definitions for a general model: intrinsic, evaluated and perceived. The intrinsic QoS is obtained by a comparison of measured and expected performance characteristics. The perceived QoS reflects the user experience by using a specific service. The evaluated QoS begins to be analyzed when the customer decides whether to continue to use this service (Andrade, 2008; Maia, 2006).

measure is subjective and will reflect the relationship between user's expectations and experience.

The evaluated QoS begins to be analyzed when the customer decides whether to continue to use this service. This decision is influenced by factors such as price of service, perceived quality and how well the customer is served.

The level of overall QoS should be evaluated in all dimensions (intrinsic, evaluated and perceived) separately. Each entity should separate responsibilities in the process of obtaining QoS.

For the purpose of this research, some techniques for obtaining intrinsic QoS will be discussed.

2.2.2 Techniques for Obtaining Quality of Service

In order to support multimedia applications, the network can use different techniques. Some techniques may involve some structural aspects, while others may involve traffic shaping and differential data treatment.

A basic solution is to oversize all network elements (routers, links and network elements). In this way, problems as delay, jitter and loss packet will be reduced and the bandwidth will be increased. Although simple, and sometimes impossible, this solution is expensive and does not provide a rational usage of the available resources.

Another simple solution is to increase the buffers in the routers. This solves the

problem of jitter, but also increases the delay. Both oversizing the network components and increasing buffers do not solve the problem of QoS in the long term.

There are more elaborate mechanisms, such as Integrated Services and Differentiated Services, which allow greater control by administrators.

Integrated Services

The Integrated Services (Intserv) is a structure developed by the IETF Integrated Services working group. This structure aims at providing QoS for individualized applications. It requires resources, such as bandwidth and buffers, to be reserved a priori for a given traffic flow. It includes resources such as packet classifiers, packet schedulers, and admission control, which are beyond those used in the best-effort model. A packet classifier identifies flows to receive a certain level of service. A packet scheduler handles the scheduling of services to different packet flows to ensure that QoS commitments are met. Admission control is used to determine whether a router has the necessary resources to accept a new flow (Andrade, 2008).

The main issue with the Integrated Services model has been scalability [RFC-2998], especially in large public IP networks which may potentially have millions of active micro-flows in transit concurrently. A noticeable feature of the Integrated Services model is that it requires explicit signaling of QoS requirements from end systems to routers [RFC-2753] (Awduche *et al.* , 2002).

The main IETF protocol for Integrated Services is the ReSource Reservation Protocol (RSVP) (Braden *et al.* , 1997). This protocol is used to reserve resources. It does a multicast routing with spanning tree, where each group receives a group address. A sender adds the address of a group in their packets in order to transmit data to this group. Then the algorithm builds a spanning tree that covers all members (Tanenbaum, 1989). A drawback of this approach is the requirement of resource reservation before sending packets. This leads to problems of scalability and excessive complexity of routing elements. To remedy this problem, another approach, called Differentiated Services, was developed.

Differentiated Services

In the differentiated services architecture it is not necessary to reserve resources in advance. It operates on the principle of traffic categorization, creating multiple classes of services. This way, each router is configured to differentiate traffic based on its class. The Service Level Agreements specify which class of traffic would be served, the guarantee and the amount of data that should be granted to each class.

2.3 The MPLS Technology

The formerly discussed techniques for obtaining QoS allow a significant performance improvement. However, as they still depend on the traditional routing algorithms from IP networks, and as IP routing algorithms provide no guarantees, it is not possible to ensure the quality of multimedia applications.

In order to guarantee QoS in IP networks it is necessary to use mechanisms for a greater control over network resources. With this goal, the Multiprotocol Label Switching (MPLS) was developed. Concerning that it is the selected scenario of this research, the MPLS will be described in some detail.

With the purpose of locating the MPLS in the layers model, the Open Systems Interconnection (OSI) model is briefly introduced as follows. The OSI model standardizes internal functions of a communications system by partitioning it into abstraction layers. It classifies computer networks in seven layers, as shown in Figure 2.3 (Tanenbaum, 1984).

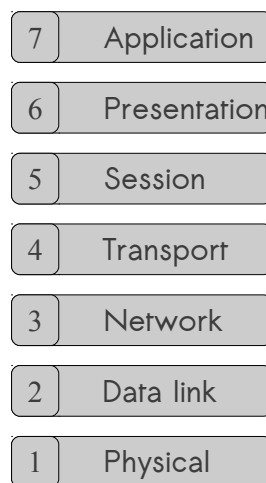


Figure 2.3: The seven layers of OSI model.

According to Tanenbaum (1984), basically, the main rules to define those seven layers are:

1. A layer should be created if there is the necessity of another abstraction level.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen based on the definition of international standardized protocols.
4. The boundaries of layers should be chosen to minimize information flow through interfaces.

5. The number of layers should be large enough such that distinct functions do not need to be placed at the same layer, and small enough such that the architecture does not become hard to control.

A brief description of each one was specified by ITU-T Rec.X.200 (1994), as follows.

Layer 1: The Physical Layer provides mechanical, electrical, functional and procedural means to activate, maintain, and de-activate physical connections for bit transmission between data-link entities. Its entities are interconnected by a physical medium. This layer defines the relationship between a device and a transmission medium such as a copper cable or an optical fiber cable.

Layer 2: The Data Link Layer detects and possibly corrects errors which may occur in Physical Layer. It also enables the Network Layer to control the interconnection of data-circuits within the Physical Layer.

Layer 3: The Network Layer controls the operation in the subnet, deciding which physical path the data should take based on network conditions, priority of service and other factors. This layer is also responsible for forwarding packets between networks, to control traffic on the subnet and to maintain the quality of service requested by the transportation layer.

Layer 4: The Transportation Layer provides transparent data transfer between end-users, ensuring reliable services to the upper layers. It optimizes the use of available network services in order to provide the required performance by each session entity at minimum cost.

Layer 5: The Session Layer controls dialogues between computers, managing their data exchange. It establishes a session connection between processes running on different stations with the aim at supporting orderly data exchange interactions and releasing the connection in an orderly manner.

Layer 6: The Presentation Layer establishes context between application-layer entities. It transforms data in a form that the application accepts, formatting and encrypting data to be sent across a network.

Layer 7: The Application Layer is the closest one to the end-user. It interacts directly with software application implementing a communicating component. In this way, the end user may access the network services.

The MPLS technology operates at a layer that is generally considered to lie between traditional definitions of Layer 2 (Data Link Layer) and Layer 3 (Network Layer), and thus is often referred to as a "Layer 2.5" protocol.

MPLS (*Multi Protocol Label Switching*) is one of the technologies proposed by the IETF, which enables sophisticated routing schemes based on the ability of a prior

establishment of routes to be followed by packets. The explicit routing of packets is possible due to the addition of short path labels. Thus, it avoids complex lookups in a routing table. This is an important feature for the maintenance of the network QoS indicators.

Figure 2.4 shows the MPLS layer and its fields. The MPLS header is positioned after Layer 2 header and before Layer 3 header. The Label is a short identifier that is assigned to each packet when it enters in the MPLS network. Thus routers just analyze the labels in order to direct the packet. The TC field means Traffic Class. It defines classes of services, in order to indicate priority levels to support DiffServ in MPLS network, among others. The field S refers to Stack. It supports queuing labels if the package receives more than one label. The field TTL means Time to Live. It counts the number of routers the packet has passed by, in a total of 255. If the packet exceeds 255 routers, it is discarded to avoid possible loops.

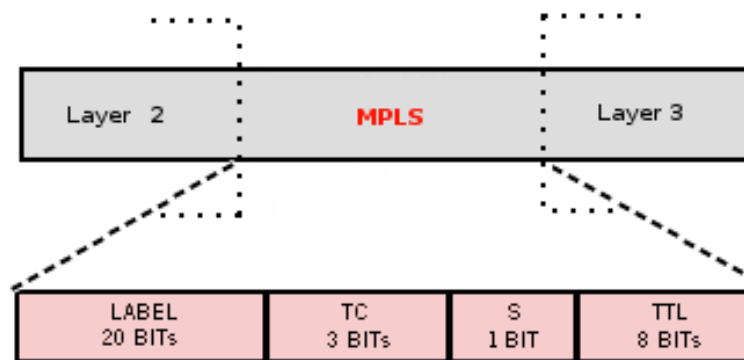


Figure 2.4: The MPLS layer and its fields. The MPLS header is positioned after Layer 2 header and before Layer 3 header. The label is a short identifier that is assigned to each packet when it enters in the MPLS network. TC is the Traffic Class, that defines classes of services. S refers to stack, that supports queuing labels if the package receives more than one label. TTL refers to Time To Live, that counts the number of routers the packet has passed.

The term *Multi Protocol* from the acronym MPLS represents the possibility of integration of this technology with different network protocols. It consists of specific software and hardware. In software it focus on communication protocols. In hardware, it needs routers that can handle labeled packets within this domain.

On IP network routing, for each datagram, the router searches in the routing table the network prefix that suits the same address, in order to determine the output link. As the network prefix has different sizes, the search time varies. The basic idea of MPLS is to increase the speed of processing IP datagrams, forwarding packets based on labels of short fixed size. The routing process based on label increases the processing speed, resulting in a better performance. MPLS network can offer

Quality of Service guarantees without dedicated links. It was designed to convert IP backbones (best effort) into business class transportation, capable of manipulating real time data (Andrade, 2008).

The packet forwarding through label is only performed within an MPLS domain. MPLS domain can be defined as a set of routers capable of loading MPLS packets. It consists of three basic entities. Two of them are located in the domain boundary and the third one is located inside the MPLS domain. The first boundary router inserts the label on the packet. The inside router is responsible for routing in the domain. The other boundary router removes the label packet (Andrade, 2008). When a packet enters the MPLS domain, the edge router, called LER (*Label Edge Router*), inserts labels on it. Thus, the router links the packet to a FEC (*Forward Equivalence Class*) class, which indicates the LSP (*Label Switched Path*) path end-to-end through which the packet should follow in the domain. FEC is defined as a class corresponding to a set of packets sent in the same way through the network. Each FEC explicitly defines the nodes that the packets of each class should follow. With FECs, it is possible to create a LIB (*Label Information Base*), that is kept in each LSR (*Label Switch Router*) (Dias, 2004; Awduche *et al.* , 2002).

Figure 2.5 represents a model of packet forwarding in the MPLS domain. Passing through the LER ingress, the IP packet is assigned to a FEC and it receives a label that indicates the path to be followed. Before the packet leaves the MPLS domain, the label that indicates the FEC is removed.

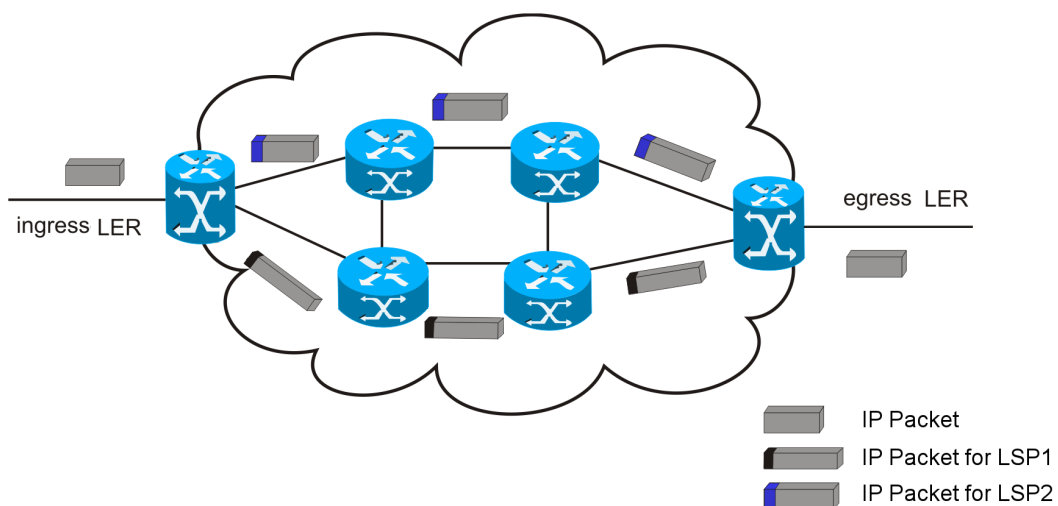


Figure 2.5: Routing packets in the MPLS domain. (Adapted from Maia (2006)). The cloud represents the MPLS domain. The ingress LER associates the IP packet to a FEC, explicitly stating the path to be followed - for instance, LSP1 or LSP2. Finally, the egress LER removes the label of the IP packet and it is forwarded on its way via standard IP routing.

MPLS is a very powerful technology for Internet Traffic Engineering because it supports explicit LSPs, i.e., dedicated paths (virtual circuits) can be created in a datagram network. The explicit routing allows constraint-based routing to be implemented efficiently in IP networks (Awduche *et al.* , 2002). Multiple paths can be used simultaneously to increase the performance of a given source to a given destination. Thus, load balancing can be performed using many routes for traffic flow.

As the network parameters can change at any time, schemes of routing and re-routing should be created to keep the contract between service provider and users. The Traffic Engineering with MPLS uses measurement systems coupled to routing algorithms and determines LSPs. The LSP selection should avoid re-routing, because it involves operation costs of management network. This process may be preventive, where path allocation prevents congestion in the network, or may be reactive, operating when a problem occurs (Andrade, 2008).

Embratel is an example of a company that provides the use of MPLS technology. The company offers to the customers network structures with this technology, that can ensure a suitable operation of their applications, allowing voice and video traffic over IP, for example, with the desired QoS (Embratel, 2013).

Related works

Current research indicate that many Computational Intelligence techniques are used to solve different problems related to the optimization of Routes on IP Networks. Among many problems, we can mention some of them that are investigated: the prediction of delay, loss of packets, congestion, routing, rerouting, load balancing and optimization of parameters such as available bandwidth. The literature review was based on a set of techniques and technologies used to develop the proposed algorithm in this thesis. The first part consists of network concepts, encompassing Traffic Engineering and MPLS. Then, we list research related to routing applied to IP networks. Subsequently, we list problems using techniques of multiobjective optimization of networks. Finally, we present different routing problems that use hybrid algorithms.

3.1 Traffic Engineering and MPLS Technology

Considering the Traffic Engineering and MPLS Technology , there is a set of research, such as Nogueira *et al.* (2006) and Cortez *et al.* (2006) that attempts to perform the detection of applications from the prediction of their traffic levels. The techniques used are based on Artificial Neural Networks with Multilayer Perceptron (MLP).

Bui *et al.* (2007) propose a hybrid approach for predicting end-to-end delays using Wavelet Transforms in combination with Artificial Neural Networks and Pattern Recognition techniques. It is interesting to our approach once that it can be useful

for the choice of requisitions sequence.

Onali & Atzori (2008) describe an algorithm for finding an optimal mapping of the Type of Service in each type of class and an optimal allocation of bandwidth for each one of them. The optimality is expressed in terms of the cost function in order to obtain a level of Quality of Service and utilization of network resources. From the improvement of techniques of Traffic Prediction, it is possible to create tools for fault detection and efficient Traffic Engineering techniques, resulting in economic benefits through an improvement of resource management. These studies may be useful to supplement this study, by carrying out the routing associated with the possible traffic or the best allocation of bandwidth with respect to the application. Other works that deal with prediction in order to control the size of the queue buffer, such as Sousa *et al.* (2006) also have significant importance to solve the delay caused by the congestion problem.

The research developed by Maia (2006) proposes a Traffic Engineering system capable of supporting mixed traffic (data, voice and video). It attempts to keep dynamically different levels of Quality of Service using MPLS, principles of Autonomic Computing and techniques of Computational Intelligence. This integration is achieved by combining different techniques of Computational Intelligence, such as Fuzzy Logic to implement the classification of routes, Genetic Algorithms for optimization and Artificial Neural Networks for the traffic prediction. The use of these different techniques allows networks to have an intelligent behavior with some features such as adaptability, fault tolerance and robustness. Autonomic Computing, on its turn, allows the network to respond automatically to changes in conditions that occur during its operation, presenting a self-management behavior. For the implementation of traffic control and QoS, they used the MPLS (Maia *et al.* , 2007). However, this work limits the construction of routes and treats them as pre-defined, creating genetic operators that act only on the same routes. It also considers only the shortest path as a criterion of convergence.

3.2 Routing applied to IP networks

In papers of routing applied to IP networks, Bagula (2006) attempts to maximize the use of network bandwidth and to reduce the delay through the GEP method (*Gene Expression Programming*).

Baguenine & Mellouk (2007) aim at finding a feasible path that satisfies QoS requirements, from the optimization of resources, by reducing the number of hops and delay. The metaheuristic used is the Ant Colony, which attempts to distribute the

traffic through the N-best paths according to the criterion of end-to-end delay.

Dias (2004) proposes mono-objective strategies to solve problems of TE in IP networks over MPLS. The method uses Lagrangian Relaxation, with a subgradient algorithm to obtain a feasible solution in polynomial time. Its objectives are intended to maximize the overall throughput of data flows injected into the network, considering the bandwidth of the links and the limit of end-to-end delay for each data stream. Although the subgradient algorithm finds a solution, there is no guarantee on its viability for a dynamic application.

The current study, however, involves the optimization of three parameters, which according Wang & Crowcroft (1996) renders problem the complexity NP-complete. For this reason, heuristic techniques become the most suitable ones.

Andrade (2008) proposes the provisioning of QoS in MPLS networks using bioinspired algorithms associated with local search methods such as Tabu Search and GRASP method (Greedy Randomized Adaptive Search Procedure) in an environment of autossimilar traffic. Santos (2009) proposes a dynamic evaluation of routing in an MPLS environment using multiobjective techniques, such as the NSGA-II algorithm. That work aims at minimizing the network cost, by responding to various user requests, ensuring the Quality of Service and performing load balancing of flows in the network.

3.3 Multiobjective optimization of networks

In the literature review, we noticed that the use of multiobjective methods applied to routing problems in IP networks is scarce. Most of the papers presented assume the existence of a single design goal. In situations in which more than one objective is considered, the mathematical formulation is developed as a weighted sum of the objectives. This results in a biased search for one particular purpose. For this reason, multiobjective strategies have also gained importance in the optimization of traffic flows in networks. As one of the advantages, the multiobjective approach presents a uniform treatment of all objectives. The decision maker is the responsible for choosing which goal will be prioritized in the set of final solutions. Among some studies involving QoS analysis, some papers describe the application of bioinspired methods, such as Ant Colony, Genetic Algorithms, Tabu Search and Simulated Annealing.

Alvarado *et al.* (2005) propose the application of evolutionary algorithms such as NSGA-II and SPEA2 for multiobjective optimization of multicast networks, aiming at minimizing the number of hops and the delay of transmission. Drummond *et al.* (2008) propose an approach based on Fuzzy Logic for dynamic allocation of band-

width for multimedia applications with high bandwidth requirement. The ability to dynamically adjust the bandwidth required by these applications allows a better offer of QoS to customers. Erbas & Erbas (2003) present a routing study with multiobjective MPLS networks. The conflicting objectives are the minimization of delay, load balancing and minimization of number of LSPs. An exact implementation of a Chebyshev method with lexicographic weights is used in that research, which leads to a high computational cost. In order to achieve similar results with smaller computational costs, Cerav-Erbas (2004) proposes the use of Simulated Annealing.

Banerjee & Kumar (2007) present a multiobjective approach applied to a realistic model of a network of Autossimilar Internet Traffic. That work optimizes the delay of packet delivery and the cost to build network topologies. The generated topologies for an autossimilar traffic have high delays if compared to the Poisson traffic model. In small networks, that work uses a deterministic algorithm with exhaustive search of the Pareto front and another algorithm based on the branch of the network. Due to the high computational cost, it can be applied only to small networks. The evolutionary algorithm was unable to find a good solution in regions of low cost.

3.4 *Hybridization of algorithms in routing problems*

Some works that involve the hybridization of algorithms in routing problems could be identified. Perboli *et al.* (2008) propose a hybrid method for a vehicle routing problem using Genetic Algorithms and Tabu Search. Each solution that was optimized by the GA is again optimized by Tabu Search, which combines two different neighborhoods in a process that increases the size of the neighborhood without increasing its complexity. To this end, they present a new operator for genetic mutation and a new procedure of movement. It is important to note that this model is applied to a mono-objective problem. de Freitas & Montané (2008) also investigate the vehicle routing with simultaneous pickup and delivery, by using metaheuristics that combine VND (Variable Neighborhood Descent) and GRASP (Greedy Randomized Adaptive Search Procedure). de Souza Filho (2007) describes a search for a pipeline distribution using the VNS. The main objective, in this case, is to reduce the costs involved in the logistics of pipeline transportation in the petroleum industry.

From hybrid methods, which put together Genetic Algorithms and Local Searches, it was observed that the use of population ensures the exploration of the search space. Beyond this, using local search techniques helps to quickly identify good areas in the search space. The success of these methods may be due to the balance between having a quick search and maintaining a diversity to avoid premature convergence

(Perboli *et al.* , 2008).

3.5 *Considerations*

Based on these research, we constructed the theoretical basis for our approach. We observed that most of the cited works perform the optimization of two parameters. Beyond this, it was not identified any published work that uses VNS as a basis of a hybrid algorithm for routing problems in telecommunications. Similar methods have shown good performances in other contexts of routing problems. We noted the existence of limitations to the problem of network design. Difficulties as the correction of infeasible routes, generated from the operations of crossover and mutation may increase the complexity and computational cost of this technique. These limitations need to be addressed with new approaches. We chose to investigate in this thesis a scenario that combines features of Multiobjective Genetic Algorithms and Variable Neighborhood Search in a specific encoding for routing on IP network, considering the MPLS as technology that may allow the QoS provisioning.

Optimization and Algorithms

Techniques of Traffic Engineering may use routing protocols in conjunction with optimization methods. Among different problems addressed from TE techniques, we can mention the prediction of delay and loss of packets, congestion, routing, load balancing and optimization of parameters, such as the proper use of available bandwidth. Although most of studies focus on optimization problems with a single goal, it was observed that network problems have multiobjective features, i.e., they have more than one objective to be minimized (e.g., reducing packet loss) or maximized (maximizing traffic flow). For example, while customers want faster and more reliable connections, network administrators prefer a stable and balanced network. Therefore, the objectives desired by customers conflict with the goals related to the performance of the network (Cerav-Erbas, 2004).

In the following sections, we will present some concepts about Multiobjective Optimization and Evolutionary Computation including description of Genetic Algorithms. Some discussion about Decision Making is also presented. Next, Local Search Algorithms such as Iterative Local Search and Variable Neighborhood Search are presented. Finally, some studies presented in literature are discussed.

4.1 *Optimization*

Optimization refers to the choice of the best elements in a set of available alternatives. In its simplest form, it consists of the minimization or maximization of a

function within a set of possible values.

Although most of the studies in operations research focus on optimization problems with a single objective, most real-world engineering optimization problems combine multiple objectives (Cerav-Erbas, 2004).

Early searches for optimal solutions of problems with simultaneous objectives, for instance products with low-cost production, low-cost materials, low energy consumption and, at the same time, high quality, high durability and reliability, treat them as single-objective problems. In such approaches, all the objective functions are aggregated in a single function, or all the objectives are transformed into constraints, except one (Parreiras, 2006). The feasible set, in this case, is totally ordered, i.e., we can rank all feasible solutions with regard to some objective function (Cerav-Erbas, 2004).

However, real problems do not present a natural way in which the several criteria which are involved in the design could be transformed in a single objective. Such an observation gave rise to the definition of multiobjective problems, that analyze several functions and lead to a set of solutions, instead of a single function with an optimal solution. In multiobjective problems, it rarely happens that all of the objectives can be optimized simultaneously; instead, it is generally the case that the objectives conflict with each other. It is not possible to obtain a total ordering of the feasible solutions, i.e., the feasible set is only partially ordered (Cerav-Erbas, 2004).

When two solutions of a multiobjective problem that belong to the Pareto front are compared, we observe that one achieves a better performance in one objective, while the other one is better for the other objective. A whole set of solutions can be generated in this way, describing the trade-off between objectives. Such a strategy has great advantage over single-objective approaches, due to the flexibility that it offers to the decision maker that can assess the relative importance of the different objectives in the moment of choosing a solution. Besides telecommunication problems, multiobjective techniques can be applied to various systems, for instance the transportation of petroleum (de Souza Filho, 2007), the design of power distribution systems (Carrano, 2007), and also epidemiological studies for finding irregularly shaped spatial clusters (Duczmal *et al.* , 2008).

4.1.1 Multiobjective optimization problems

A multiobjective optimization problem is defined as:

$$\begin{aligned} \min \mathbf{f}(\mathbf{x}), \quad \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_\ell(\mathbf{x})) \\ \text{subject to: } \mathbf{x} &= (x_1, \dots, x_n) \in \mathcal{G} \end{aligned} \tag{4.1}$$

where $\mathbf{x} \in \mathcal{X}$ is the *decision variable vector*, \mathcal{X} is the *space of decision variables*, $\mathcal{G} \subset \mathcal{X}$ is the *feasible set*, $\mathbf{f} \in \mathcal{F}$ is the *objective vector*, and \mathcal{F} is the *objective space*. Figure 4.1 illustrates an example of mapping between the space of decision variables and objective space. The objective functions perform the mapping between each point $x = (x_1, \dots, x_n)$ from the decision variable space and the respective image points $F(x) = (f_1, f_2, \dots, f_m)$ in the objective space.

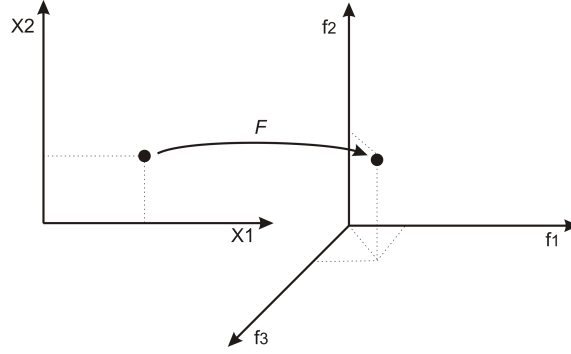


Figure 4.1: Mapping from decision variable space to the objective space (Parreiras, 2006).

The goal of some multiobjective optimization methods is to obtain estimates of the Pareto-optimal set (Ehrgott, 2000), which contains the set of non-dominated solutions of the multiobjective problem. In a minimization 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 relational operators \leq and \neq are defined as:

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

and

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

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:

$$\mathcal{P} = \{\mathbf{x}^* | \nexists \mathbf{x} : \mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}^*) \wedge \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}^*)\}. \quad (4.2)$$

All solutions which 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 multiobjective optimization

problem is the set of all Pareto-optimal solutions. The image of this set in the objective space is called the *Pareto front* ($f(\mathcal{P})$).

Figure 4.2 represents an example of non-dominated points. The points inside the cones represented in the figure are dominated by the point located in its vertex. It may be observed that the objective function f_1 is smaller for point A than for point B , while the objective function f_2 is smaller for point B than for point A . This means that there is not a dominance relationship between A and B . The same is valid for E , in relation to A and B . The points C and F are dominated by the point A . B dominates the points C , D and F . Among C , D and E there is no dominance relationship. The points C and D dominates F . The points E e F do not dominate any other point of the figure.

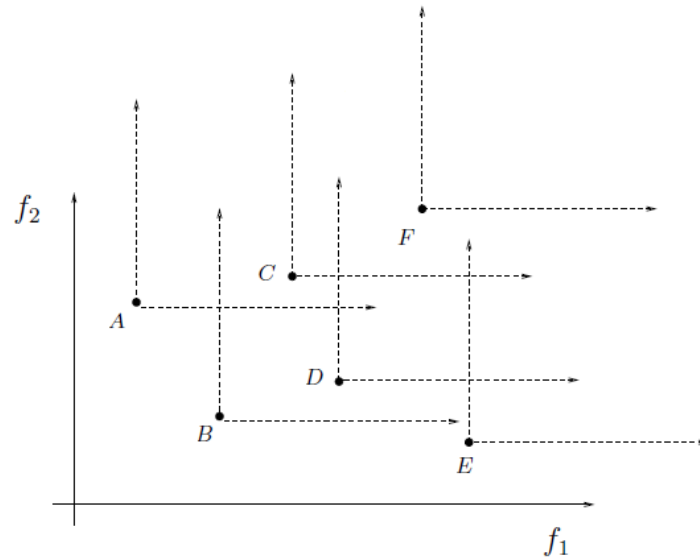


Figure 4.2: Non-dominated points. (i) There is no dominance relation between the points A , B and E . (ii) A dominates C and F . (iii) B dominates C , D and F . (iv) Between C , D and E there is no dominance relations. (v) C and D dominate F . (vi) E and F do not dominate any other point shown in Figure (Adapted from Takahashi (2007)).

4.2 Scalarization Methods

In general, scalarization methods reduce multiobjective problems to parameterized single objective problems. In those problems, the parameters can be systematically changed such that different solutions of the original multiobjective optimization problem can be found. Multiple single objective searches are carried out in order to generate the Pareto optimal solutions. (Cerav-Erbas, 2004; Takahashi, 2007).

The most popular methods include the weighted sum method, the ϵ -constraint

method, and the lexicographic weighted Chebyshev method (Cerav-Erbas, 2004). This work will employ the ϵ -constraint method in order to obtain a formulation of the problem studied here that can be solved by exact methods, at least for small problem instances.

4.2.1 The ϵ -constraint method

The ϵ -constraint method is an approach that transforms multiple objectives into a single objective by restating all but one objectives as constraints in the problem.

Theorem 1 *If $\mathbf{x}^* \in \mathcal{G}$ is efficient then there is an integer $i \in 1, 2, \dots, l$ and real numbers $\epsilon_j, j = 1, \dots, l$ ($j \neq i$) such that \mathbf{x}^* determines:*

$$\begin{aligned} \mathbf{x}^* = \arg \min f_i(\mathbf{x}) \\ \text{s.t.} \quad \begin{cases} f_j(\mathbf{x}) \leq \epsilon_j ; j = 1, \dots, l ; j \neq i \\ \mathbf{x} \in \mathcal{G} \end{cases} \end{aligned} \quad (4.3)$$

As a practical implication of the Theorem 1, it follows that by parametrically changing $\epsilon_j \forall i$ and $\forall j \neq i$, it is possible to completely generate the Pareto-optimal set \mathcal{P} .

In Figure 4.3, the Pareto-optimal set \mathcal{P} is represented by a continuous line. A and B are points of Pareto-optimal set, obtained with ϵ -constraint method, respectively: with the minimization of f_1 s.t. $f_2 \leq \epsilon_2$ and with the minimization of f_2 s.t. $f_1 \leq \epsilon_1$.

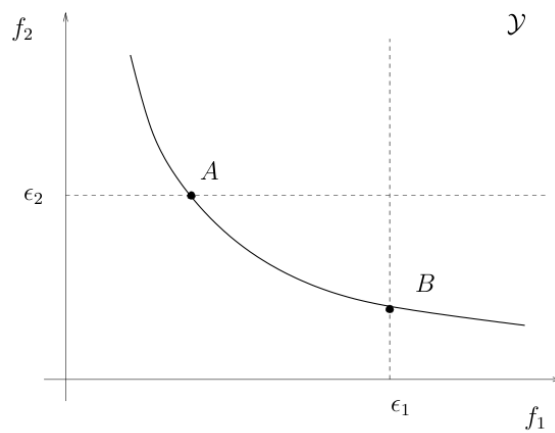


Figure 4.3: Representation of two points of the Pareto-optimal set \mathcal{P} attained with ϵ -constraint method. A is the minimization of f_1 s.t. $f_2 \leq \epsilon_2$. B is the minimization of f_2 s.t. $f_1 \leq \epsilon_1$ (Takahashi, 2007).

The ϵ -constraint method may fail in finding efficient points. However, it can guar-

antee at least a weak Pareto optimality of the solutions, such as illustrated in Figure 4.4. The optimal set of the problem includes the points lying on the line between z^a and z^b . On this line, z^b is the only efficient solution. However, the optimizer may end up with z^a whose corresponding decision vector is weakly Pareto optimal (Cerav-Erbas, 2004).

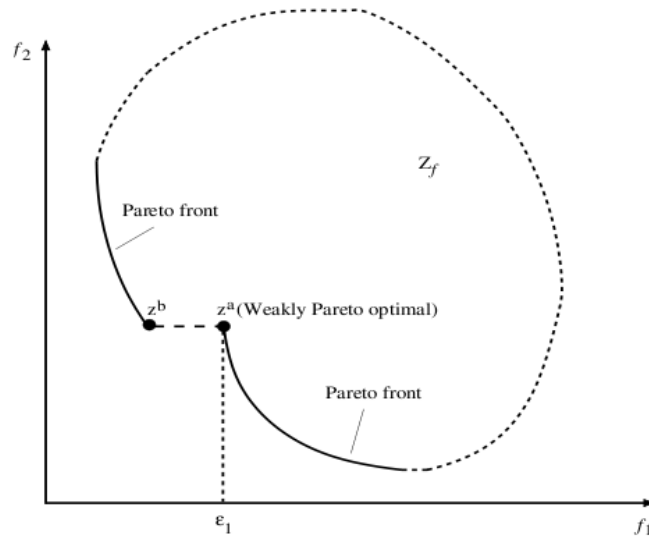


Figure 4.4: The ϵ -constraint method guarantees weak Pareto optimality (Cerav-Erbas, 2004).

4.3 Evolutionary Computation

Optimization problems of combinatorial nature may be NP-hard, which means that exact algorithms would require very long run times to obtain the optimal solution. Therefore, in several practical situations a better alternative is to employ heuristic algorithms, which attempt to find non-exact solutions as close as possible to the optimal one in a limited computational time budget (Cerav-Erbas, 2004). This class of algorithms include the Evolutionary Algorithms, which are stochastic search methods, originally inspired on the natural evolution (Parreiras, 2006). However, the term “evolutionary” has been expanded since the proposition of the earliest evolutionary algorithms, and now it is no longer limited to algorithms that mimic the natural evolution process described by Darwin. Nowadays it includes methods that, in general, produce random perturbations or recombinations in previous solutions, which are accepted or not accepted by a selection procedure which can have different degrees of stochasticity. Some examples of evolutionary algorithms are Genetic Algorithms (GA), Genetic Programming, Evolutionary Strategies, Ant Colony and Artificial Im-

mune Systems.

In this work, we focus on the description of Genetic Algorithms (GA) that are popular and easy to use, inspired by the theory of natural evolution and genetics (Parreiras, 2006). A justification for the use of GAs in routing is that they allow the creation of several sub-optimal solutions, which can be advantageous in relation to traditional algorithms which find routes by generating only one solution. In the case of the problem under consideration in this thesis, this feature allows the use of different routes for information delivery, keeping the lowest possible delay and performing load balancing, for example.

4.3.1 Genetic Algorithms

The first GA was presented by Holland in 1975 (Holland, 1975), but the technique was just popularized by Goldberg in 1989 (Goldberg, 1989). The GA is based on Darwin's theory of species evolution, that states that individuals less adapted tend to perish, while the most adapted ones continue to exist and are able to reproduce. Imitating this process, Genetic Algorithms are developed from an initial population and evolve through genetic operators such as selection, crossover and mutation (Goldberg, 1989).

According to Osman *et al.* (2005), GAs are robust, do not require auxiliary information and can offer significant advantages with respect to performance. GAs work with a set of encoded parameters, which means that they can easily handle discrete or integer variables. They use probabilistic transition rules instead of deterministic ones. Figure 4.5 presents a diagram for the development of this algorithm.

In first place, the individuals must be encoded for the desired application, such that a sequence of symbols represents a candidate solution of the problem. Analogously to the biological genetic code, the sequence of characters can be decoded into the problem variables (phenotype), thus being analogous to a chromosome. Each character of the chromosome corresponds to a gene. The concatenation of all chromosomes belonging to an individual represents the genotype. The decoded genotype represents the phenotype. Figure 4.6 illustrates these analogies.

A selection operation consists of choosing which individuals will have copies and which of them will disappear over the generations. Thus, it defines which individuals will participate of the next crossover and mutation operations to be performed. This choice is performed such that the best individuals are more likely to remain in population, and the less adapted ones are more likely to be discarded, in order to ensure the population quality (Parreiras, 2006; Carrano, 2007). Some of selection methods are the Roulette wheel and the Stochastic Tournament (Goldberg, 1989).

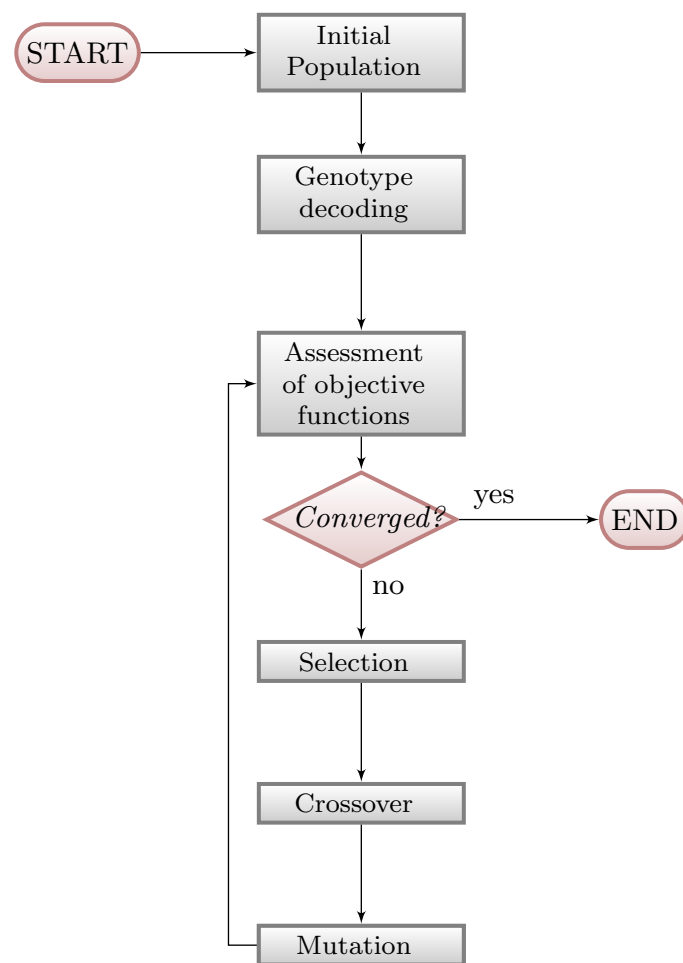


Figure 4.5: Diagram of Genetic Algorithm. It consists of an initial population that has its symbol representation decoded into variable values. With each variable, it is possible to assess the objective function of the problem. If it is according to a convergence criterion, the algorithm is concluded. Otherwise, it evolves again and again through genetic operators such as selection, crossover and mutation.

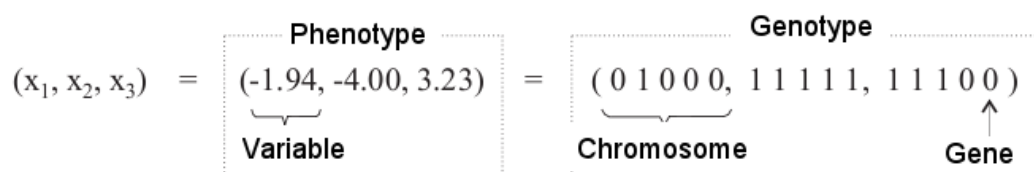


Figure 4.6: Genetic Encoding. The sequence of characters can be decoded into the problem variables (phenotype). Each character of the chromosome corresponds to a gene. The concatenation of all chromosomes belonging to an individual represents the genotype. The decoded genotype represents the phenotype (Parreiras, 2006).

In Roulette wheel selection, the probability of each individual being selected is proportional to its relative fitness function, represented by a slice of a roulette wheel. The higher the fitness, the greater is the slice in roulette, and consequently, greater

is its selection probability (Maia, 2006). Tournament Selection, in turn, chooses the best individual among the limited set randomly selected.

The crossover operation consists of the exchange of genetic material between individuals, simulating the genetic reproduction process in nature. This genetic exchange can be implemented in various ways, such as Crossing with Multiple Cutting Points, Uniform Crossover, Crossover for each Variable. Considering the Crossover with one cut-off point, it selects two individuals randomly, called as parents, and picks up a random cut-off point in both of them. The chromosome parts after the selected points are exchanged, creating the children, as shown in Figure 4.7. Other schemes of recombination are also employed in AGs.

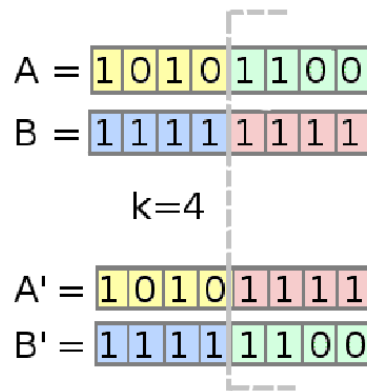


Figure 4.7: Crossover with one cutoff point. A and B represent the parents individual. A' and B' represent the offspring individual generated from the crossover of A and B. K indicates the cutoff point.

The mutation operation is the responsible for inserting new genetic characteristics in individuals of the population, in order to avoid premature convergence of the algorithm to local optima (Holland, 1975). Considering the mutation of one bit in a binary codification, a random bit is selected and its value is inverted, such as represented in Figure 4.8.

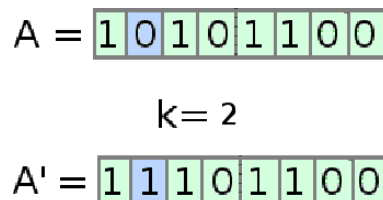


Figure 4.8: Mutation of one bit. A represents the initial individual. K indicates the random chosen bit. A' represents the individual after mutation.

4.3.2 Multiobjective Genetic Algorithms

Multiobjective Genetic Algorithms use the same structure of single-objective GAs, just differing in the operation of Selection. In single-objective selection, the individuals order is based on the value of the function to be optimized. If it is a minimization problem, the smaller the fitness function, the better the individual corresponding to this fitness (Parreiras, 2006). In multiobjective selection, the search algorithm tries to reduce the distance between the population and the Pareto-front of the problem, on the same time also trying to keep the maximum diversity of the samples. For this reason, the evolved population should result in a good distribution of solutions over the Pareto-front (Parreiras, 2006).

Several multiobjective evolutionary algorithms have been proposed, such as Non-Dominated Sorting Genetic Algorithm (NSGA) (Srinivas & Deb, 1994), Elitist Non-Dominated Sorted Genetic Algorithm (NSGA-II) (Deb *et al.* , 2002), Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler & Thiele, 1999), Strength Pareto Evolutionary Algorithm Improved (SPEA2) (Zitzler *et al.* , 2001) and Pareto Archived Evolution Strategy (PAES) (Knowles & Corne, 1999). By far, the most popular one currently is NSGA-II, which will be employed as the basis for the construction of the algorithm proposed in this work.

Figure 4.9 represents a diagram with a description of NSGA-II. Figure 4.10 also indicates the procedures of NSGA-II. We will describe the t th generation of this algorithm. It starts with the generation of a population P_t that must be sorted by non-dominance. This population represents the parents. On this, selection is performed with Tournament method, which uses the lowest rank and the greater crowding distance as choice criteria. Subsequently, operations of crossover and mutation are performed to generate the offspring population Q_t .

In a next step, the P_t population is joined to Q_t to generate a new group of individuals: $R_t = P_t \cup Q_t$. The new population R_t is sorted by non-dominance to obtain the non-dominated front.

After finding the non-dominated front, individuals are sorted again by rank value, which is equal to their non-dominance front level. The front 1 has 1 as rank, the front 2 has 2 as rank, successively. The best individuals are those with lowest rank (Deb *et al.* , 2002). Again, it obtains the crowding distances decreasingly ordered. Individuals with greater distances are then copied to the new population. The others are discarded. Crowding distance is an approximation of the perimeter formed by the cuboid whose vertices are their nearest neighbors. The first and last front individual receive infinite distance. This operation is used with the aim at maintaining the population diversity.

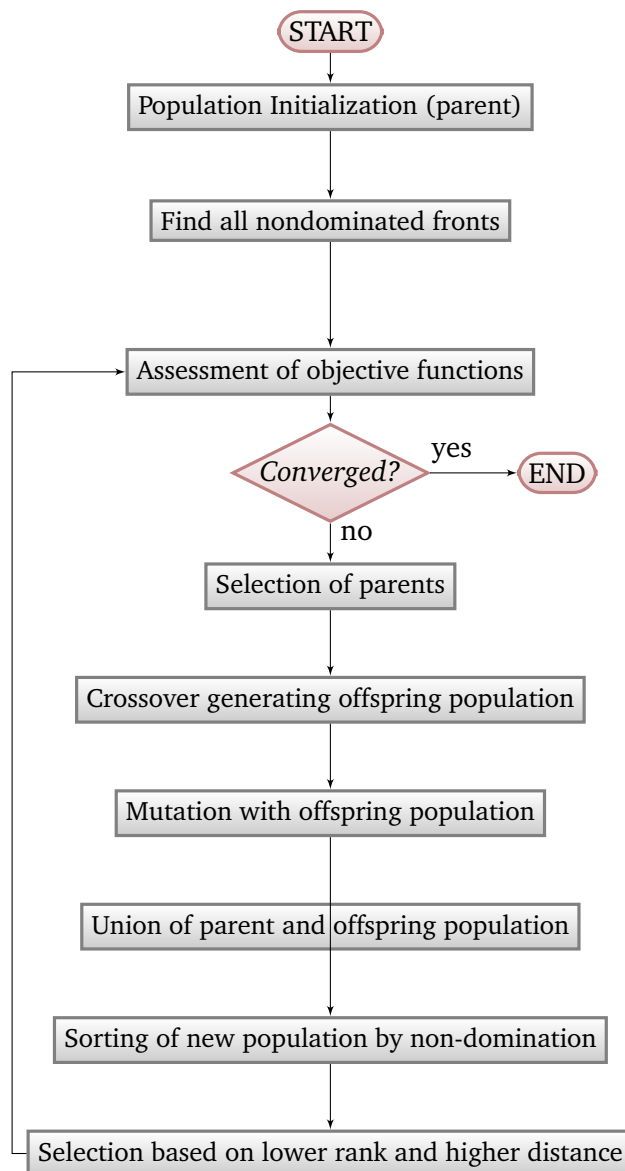


Figure 4.9: Diagram showing the steps of NSGA-II

Finally, individuals are selected with the lowest rank and greater distance and they replace the initial population.

4.3.3 Decision Making

Considering real applications, even if a multiobjective technique is applied, just one optimal solution should be selected and executed. As the final result of a multiobjective problem is a set of feasible Pareto-optimal solutions, another multicriteria decision problem is originated: the choice among many Pareto-optimal solutions, considering simultaneously several criteria. This choice may consider the preferences of someone who deeply knows the problem in question. Thus, a multicriteria decision

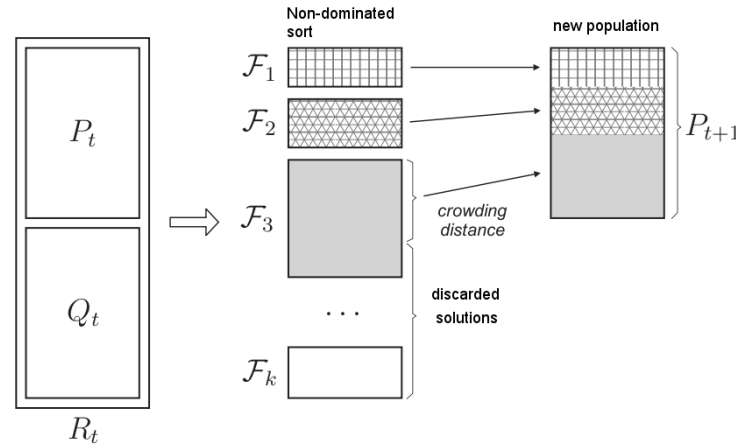


Figure 4.10: NSGA-II procedure (Deb *et al.* , 2002).

method is also necessary to solve the problem (Parreiras & Vasconcelos, 2007).

In order to determine a solution for multiobjective optimization problems, there are three ways of combining decision methods with optimization algorithms: a priori, a posteriori and progressive decision-making.

In the a priori decision-making approach, the objectives are aggregated into a single objective function where the decision maker preferences are represented. The decision-maker is consulted before the optimization process, and his preferences are used to guide the search toward the favorite solution from the Pareto front.

In the a posteriori decision-making approach, decision techniques are applied after the attainment of efficient solutions. The feasible set is searched with no previous informations about preferences. A multiobjective optimization algorithm is executed, resulting in a discrete approximation of the Pareto front. Thereafter, the decision-maker may use a decision method to compare the available alternatives and choose a unique final solution (Parreiras & Vasconcelos, 2007; Cerav-Erbas, 2004).

In the progressive decision-making approach, the information about decision-maker preferences is obtained in the course of the iterative optimization process in order to guide the search algorithm. At each iteration, the search result is evaluated by the decision-maker in order to update the preferences. The search region is then reduced and the search direction is restricted to some particular regions according to the preferences of the decision maker (Parreiras & Vasconcelos, 2007; Cerav-Erbas, 2004).

This study does not deal with previous or progressive preferences. The final set is an approximation to the Pareto optimal solutions which allows a decision maker to see the trade-offs among objectives and then, make a choice.

4.4 Local Search Algorithms

Metaheuristic methods have gained great attention in large combinatorial problems, that emerge in industrial or academic level (Paquete, 2010). Among various existing metaheuristics, we highlight some methods that perform the neighborhood search. Those methods start with a feasible solution and iteratively try to improve this solution. These algorithms can also be called local search algorithms. They are important for finding local optima, allowing the algorithm to deliver better solutions. As examples of those algorithms, there is Iterated Local Search (ILS) (Lourenço *et al.*, 2001), Tabu Search (TS) (Glover, 1989) and Variable Neighborhood Search (VNS) (Mladenovi & Hansen, 1997).

With these metaheuristics, it is assumed that there is a heuristic that makes local decisions based the knowledge of the problem. Metaheuristics guide such heuristic in the search space based on an appropriate choice of certain methods and parameters. These approaches are conceptually simple to be parameterized and quite effective, once a good heuristic for the problem is found (Paquete, 2010).

Both VNS and ILS are approaches conceptually simple to parameterize and they are effective. For these reasons, these metaheuristics were chosen as a basis for the development of this work. Their principles are presented as follows, based on the descriptions of Paquete (2010).

4.4.1 Iterated Local Search

Several researchers have proposed ILS independently, under names such as *large-step Markov chains* (Martin *et al.*, 1991, apud Paquete, 2010, p.126) and *iterated Lin-Kernighan* (Johnson and McGeoch, 1997, *idem*). A group of researchers have observed that these approaches follow very similar principles. Thus, the name ILS was proposed in order to unify the terms. This method has demonstrated high performance in classical problems such as Traveling Salesman, Graph Coloring problem and problem escalation.

According to Paquete (2010), ILS is a metaheuristic that iteratively tries to construct a sequence of solutions generated by a subordinate heuristic. At each iteration, it causes a perturbation in the solution of previous iterations.

Formally, given a minimization function f , the candidate solutions are called s , belonging to the set S . ILS tries to do a walking between two basins of attraction in a stochastic and heuristic way in each iteration.

A local search uses a neighborhood structure, in which a move can be performed from one solution s to another better solution s^l , through a smart way. The simplest

way of trying an enhancement in the solution is to repeat the local search starting from another point. Thus, if the search is performed on S^l , it avoids large search spaces. However, the biggest problem is how to define the neighborhood of S^l in order to be numbered and accessed efficiently.

Thus, to explore S^l without the notion of neighborhood, a disturbance is applied in the local optimal $s^l \in S^l$ which leads to an intermediate state $s \in S$. One applies a local search in s , generating another local optimal $s^{l*} \in S^l$. At the end, following an acceptance criterion, one chooses which of the local optima, s^l or s^{l*} , will be subject to a perturbation in the next iteration. Clearly, ILS performs the search in S^l , but without an explicit notion of neighborhood between basins of attraction.

Algorithm 1 presents the pseudocode of ILS. In order to avoid cycles, it uses *memory* as a data structure capable of returning information about visited solutions, improving the performance of ILS.

Algorithm 1 Iterated Local Search

```

1:  $s \leftarrow \text{Generate}()$ 
2:  $s^l \leftarrow \text{LocalSearch}(s)$ 
3: repeat
4:    $s \leftarrow \text{Perturbation}(s^l, \text{memory})$ 
5:    $s^{l*} \leftarrow \text{LocalSearch}(s)$ 
6:    $s^l \leftarrow \text{Acceptance}(s^l, s^{l*}, \text{memory})$ 
7: until Stop condition is met
  
```

Paquete (2010) highlights the modularity as a great advantage because it allows different configurations by changing the ILS components *Generate*, *Perturbation*, *Acceptance* and *LocalSearch*. For this reason, the best combination of these components should be found. They are detailed as follows.

Generate: This component generates initial solutions, that may be based on a greedy heuristic or randomly. The greedy heuristic is preferred by some authors because it already has some level of quality and, in general, it is efficient in terms of computational time. However, there are situations, as described by Johnson and McGeoch (1997) (apud Paquete (2010)) in the traveling salesman problem, in which the greedy heuristic returns solutions that are not possible to be improved through the component *LocalSearch*. In this case, it is preferable that the initial solution be generated randomly.

Perturbation: Perturbation is used to exit from local optima. This choice is related to the choice of heuristic to *LocalSearch*. An essential requirement is to choose a perturbation that does not return a solution in a previously visited basin of attraction. The perturbation must not be extremely strong. Otherwise, the algorithm

would reduce to a random reboot. Thus, an experimental analysis should be performed on the problem to be solved in order to find good shaking and acceptable values for the force parameter, which govern the intensity of the perturbation. Perturbation may vary, for example, according to the computational time spent. Thus, the ILS approaches to other searches, such as VNS (Mladenovi & Hansen, 1997) or Reactive Search (Battiti, 1996).

Acceptance Criteria: This criterion determines which local optimal should be chosen to be shaken, either the optimal current or the optimal returned by `LocalSearch` in the previous iteration. This procedure allows to obtain a balance between exploration and intensification of the ILS. For example, accepting the best local optimum for perturbation favors intensification. However, the acceptance of the latest local optimal favors the exploration of the search space. In any case, the procedure must save the best solution found so far, because this is the solution that should be returned by ILS.

Local Search: In each problem, it is possible that different notions of neighborhood exist and, in each of these neighborhoods, several ways of exploiting them. However, it is often necessary to do experiments because we do not know which is the best choice for the exploration of this neighborhood, that can use an exhaustive or random way. Moreover, the computational cost available for performing an ILS must be observed, since in many situations, it is preferable to obtain a lower quality result, but with an acceptable computational cost instead of a better local optimization with higher computational cost.

4.4.2 *Variable Neighborhood Search*

VNS can be described as a particular case of ILS. It consists of an implementation of subordinate heuristics sequences, alternated with small perturbations in the returned solutions by these heuristics, exploring systematically the idea of modifying the neighborhood structure in the search.

The main difference between VNS and ILS approaches is that VNS explores the notions of neighborhood in a more explicit manner. This method was proposed by Mladenovi & Hansen (1997) and thanks to its simplicity of implementation, it has shown promising results in combinatorial problems, such as optimization problems in clustering, the problem of p-medians, among others.

Contrary to most local search methods, VNS does not follow a trajectory, but it explores increasingly the distance of neighborhoods of a given current solution and jumps from there to a new one, if and only if an improvement has been done. In this way, the favorable characteristics of a current solution, for instance, that most of

the variables are already in their optimal values, would be used to obtain promising neighborhood solutions (Mladenovi & Hansen, 1997). In other words, the idea of VNS is to have two or more neighborhood functions, defined by different points of view, and switch between them.

There are many variants of VNS in literature that consider various orders in the sequence of neighborhoods, either from simple or more complex changes, which generate more distant neighborhoods and involve higher computational cost followed by higher accuracy in the solutions. Other versions of VNS consider different acceptance criteria.

The first version of VNS performed by changing the neighborhood on each iteration of the search method, following a previously defined sequence. The stopping condition happened when it was not possible to find a neighbor solution with a smaller cost.

In a version called Reduced VNS, the change of the neighborhood would be done if there were no neighborhood solution with lower cost. In general, the neighbor solution is randomly selected. If a solution with a lower cost were found, the local search method would use again the neighborhood that was ranked before, which had occupied the first position in the sequence of neighborhoods. The advantage of this approach over the previous one is in terms of computational time, because it prevents the full exploration of the neighborhood and it allows the first neighborhood, usually smaller, to be used more often. The stopping criterion may be the number of iterations, the computational time or some other criteria set by user, since it is not possible to identify a local optimum if the neighbor solution is randomly chosen.

The basic version of VNS is currently the most known and is the closest one to ILS. Initially, a set of neighborhoods $N = N^k, k = 1, \dots, kmax$ and an initial solution x are defined, generated by a random procedure or by a heuristic way. This solution will be used by the local search method, which uses the neighborhood N^1 and returns a local optimum. This procedure is repeated through several iterations.

The initial solution is randomly selected in the neighborhood of the local optimal of the previous iteration in accordance with the neighborhood N^k previously selected. The acceptance criteria will be used to choose the solution with smallest cost, between the new and the previous local optima. If the new local optimum is chosen, then the selected neighborhood for the next iteration will be the one that occupies the first position. Otherwise, the neighborhood will be the following in the sequence. It is noteworthy that the process of generating initial solutions in each iteration in the basic VNS is just the process of perturbation, typical of ILS.

The basic algorithm, proposed by Hansen & Mladenovi (2001), is described in

Algorithm 2.

Algorithm 2 Basic VNS

```

1: Initialization. Select the set of neighborhood structures  $N^k(k = 1, \dots, kmax)$  that
   will be used in the search; find an initial solution  $x$ ; choose a stopping condition;
2: Repeat the following until the stopping condition is met:
3: repeat
4:   Set  $k \leftarrow 1$ ;
5:   while  $k \neq kmax$  do
6:     Shaking. Generate a point  $x'$  at random from the  $k^{th}$  neighborhood of  $x$ 
       ( $x' \in N^k(x)$ );
7:     Local search. Apply some local search method with  $x'$  as initial solution;
       denote with  $x''$  the so obtained local optimum;
8:     Move or not. If this local optimum is better than the incumbent, move there
        $x \leftarrow x''$ , and continue the search with  $N^k, k \leftarrow 1$ ; otherwise, set  $k \leftarrow k + 1$ .
9:   end while
10: until

```

Based on Variable Neighborhood Search, the purpose of this research is to integrate these concepts into the Genetic Algorithm metaheuristic, by developing a new algorithm with features of both of them. During the process of evolution of the Genetic Algorithm, crossover and mutation operators are defined under the perspectives of different neighborhoods, which are called *Level 1* and *Level 2*. The search on each level is aided by a refinement performed by the other level through their alternate application, which explores different search spaces. In a next chapter, we describe these different operations proposed at each level.

The Variable Neighborhood Multiobjective Genetic Algorithm: VN-MGA

This chapter presents the optimization model for the problem of ensuring QoS on IP networks. It also presents the genetic operators implemented and the definition of *levels*, inspired by methods of VNS, used to enhance the exploration of the search space.

5.1 *Problem definition*

The domain of discussion of this study is the TE intra-domain, directed for a problem of choosing routes in a scenario of a corporate IP network with some specific technology that allows the explicit routing, such as choosing LSP on IP networks with the technology MPLS (Multiprotocol Label Switching). The routing should consider some QoS requirements according to the type of application. Each QoS metric is modeled as one objective. The proposal is to minimize the network cost, to respond for simultaneous user's requests, ensuring the Quality of Service and to provide a load balancing in the network.

This research employs, in the formulation of the case studies, some classical network topologies, analyzed in a similar context, but with different goals, by authors such as Santos (2009) and Andrade (2008). These topologies are shown in Figure 5.1. Simple topologies such as #a, #b and #c allow the validation of the results

of the proposed algorithm, comparing them with results obtained using other techniques. Topologies as #d, #e and #f are of a similar scale of the networks that appear in real world applications, such as metropolitan networks or Internet backbones.

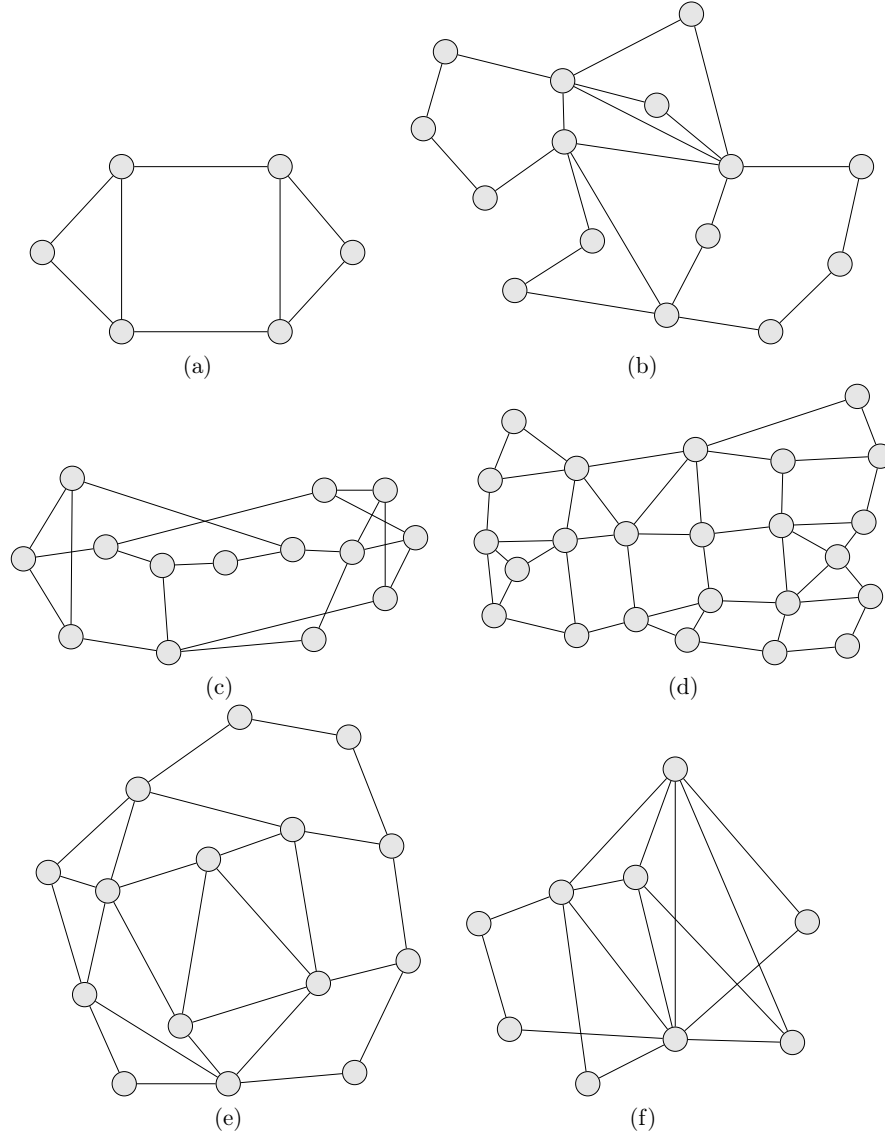


Figure 5.1: Instances of networks (Santos, 2009).

5.2 Model Formulation

The network model is represented by the graph $G = (V, A)$, where $V = \{1, 2, \dots, v\}$ indexes the set of routers (nodes) in the MPLS domain and $A = \{(i, j) \mid i, j \in V, i \neq j\}$ is the set of links, in which the link (i, j) connects node i to node j . The bandwidth of each link (i, j) is represented by B_{ij} . Each user's request is represented by the triple (o^k, d^k, b^k) , where $o^k \in V$ and $d^k \in V$ 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 indexed by $\Theta = \{1, 2, \dots, m\}$.

Objective functions and constraints are described by equations from (5.1) to (5.4), respectively, which are based on the work of Santos (2009).

$$\min \begin{cases} F_1 = \sum_{k \in \Theta} \sum_{(i,j) \in A} z_{ij}^k \\ F_2 = \sum_{k \in \Theta} (1 - a^k) \\ F_3 = \alpha \end{cases} \quad (5.1)$$

s.t.

$$\sum_{j \in V, j \neq i} z_{ij}^k - \sum_{l \in V, l \neq i} z_{li}^k = \Gamma_i^k, \forall i \in V, \forall k \in \Theta \quad (5.2)$$

$$\sum_{k \in \Theta} b^k z_{ij}^k \leq \alpha B_{ij}, \forall (i, j) \in A \quad (5.3)$$

$$\sum_{k \in \Theta} a^k \geq C \quad (5.4)$$

in which

$$z_{ij}^k \in \{0, 1\}, \forall (i, j) \in A, \forall k \in \Theta \quad (5.5)$$

is a variable which is equal to 1 when the link (i, j) is used to attend request k , and 0 otherwise,

$$a^k \in \{0, 1\}, \forall k \in \Theta \quad (5.6)$$

is a variable that becomes equal to 1 if request k is attended and 0 otherwise, Γ_i^k , given by

$$\Gamma_i^k = \begin{cases} a^k & ; \text{if } i = o^k; \\ -a^k & ; \text{if } i = d^k; \\ 0 & ; \text{otherwise} \end{cases} \quad (5.7)$$

indicates if the node i is the source, the destination, or none of these, for the request k , and

$$\alpha \in [0, 1] \quad (5.8)$$

is a variable which, being minimized under constraint (5.3) becomes equal to the maximum relative bandwidth occupation considering all links.

Objective function F_1 represents the number of links that are used in the paths of all the accepted requests. The fewer links are used the smaller is the delay for the data to travel from origin to destination. F_2 represents the number of rejections of

requests. The amount of rejection of requests is related to the admission control of new connections, which determines whether a connection can be admitted or not, according to the network load condition and the amount of requested bandwidth. The minimum number of requests that must be responded is represented by C , shown in Equation (5.4). In F_3 , α represents (in relative terms) the load of the most used link, with values varying from 0 to 1. Minimizing the amount of data traffic on the links means that the load is evenly distributed and consequently the network is balanced. The constraint (5.2) represents the classical flow conservation. According to constraint (5.3), the requested bandwidth (b^k) for a link (i, j) must be less than or equal to the available bandwidth.

5.3 The Implementation

The proposed multiobjective optimization approach to the problem of routing in IP networks, with the specific features that are necessary to deal with QoS parameters, is depicted in this section. The proposed algorithm, the Variable Neighborhood Multiobjective Genetic Algorithm (VN-MGA), is endowed with a new way to perform a search, considering different neighborhoods. It can initially be described as an implementation of the NSGA-II. The innovation concerns on the way the genetic operators work, in a switching mode between two different neighborhoods (levels).

Figure 5.2 presents a sketch of the main structure of VN-MGA. In this figure, it can be seen that the idea is to perform a whole genetic search in one of the levels (which means: a search considering the decision variables of that level) until a switch indicator is met, and to change to the other level, and so forth, until a global stopping criterion is reached. The switch indicator is defined here as the attainment of a fixed number of generations, $NumGen$, within that level or as a number of generations with no longer solutions enhancement.

5.3.1 Solution Encoding

The structure of an individual is described by an ordered set of m requests, along with its corresponding routes, as shown in Figure 5.3.

The concept of *level* refers to the neighborhoods considering two different subsets of decision variables, the *low-level* encoding, which represents explicitly the routes that are followed by each request of service, and the *high-level* encoding, that represents the permutations of several requests of service, defining the order in which they are included into the solution. The alternate employment of those encodings in

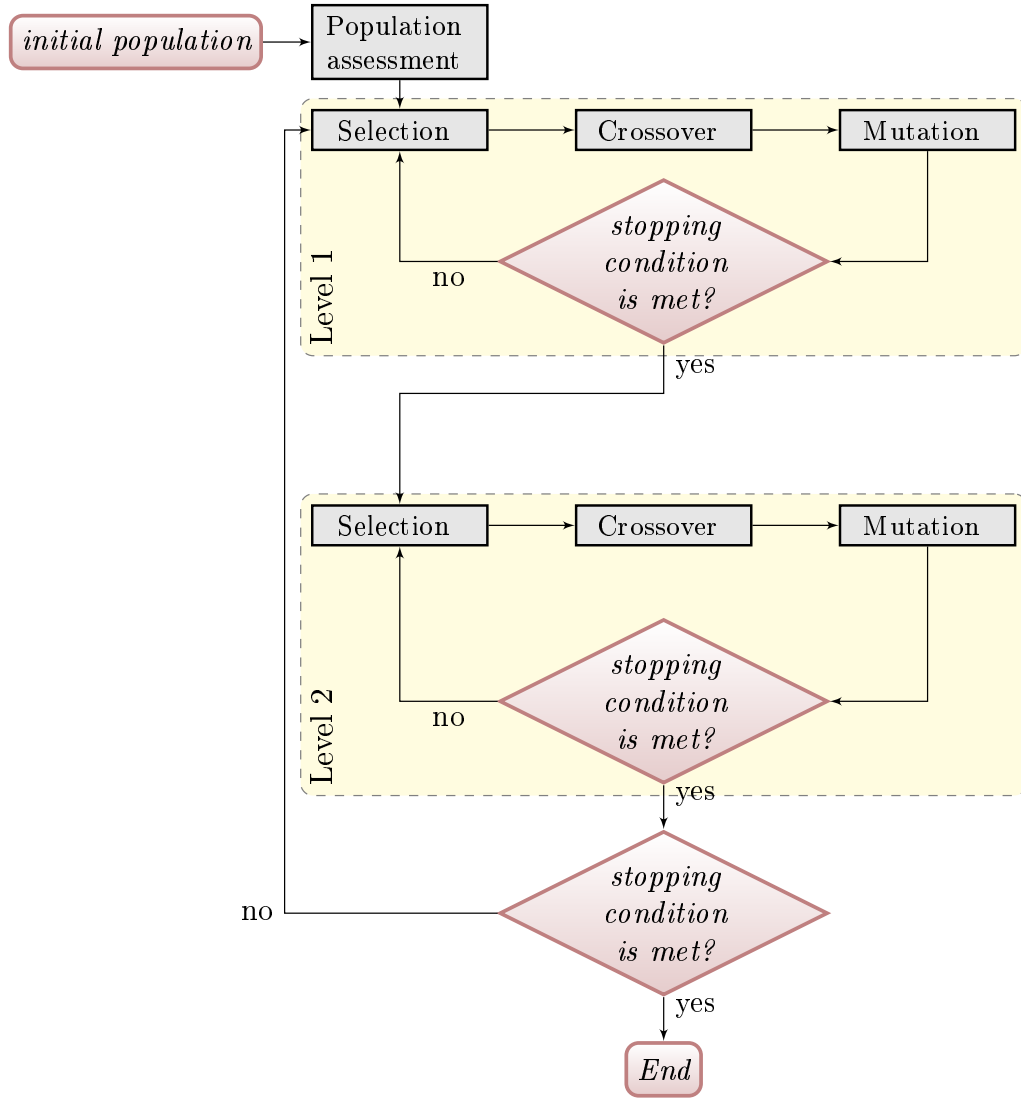


Figure 5.2: Flowchart of VN-MGA

the VN-MGA, as long as they induce different neighborhoods, becomes similar to the search principle of VNS.

When the search focuses on the routes (Level 1), the request sequence becomes fixed and the sequences of graph edges for each pair source-destination compose the decision variable vector, in the search for the request paths. The individual, denoted by I_1, I_2, \dots, I_N , is represented by a sequence of requests, which is kept fixed, and by a group of edges associated with each request, which constitute the decision variables within this level. Each request has a specific requirement of bandwidth according to its application (requested bandwidth). This means that the requested bandwidth must be less than or equal to the available one, when this request is examined. If the request cannot be met, i.e. the requested bandwidth is greater than the available one, then that request is rejected.

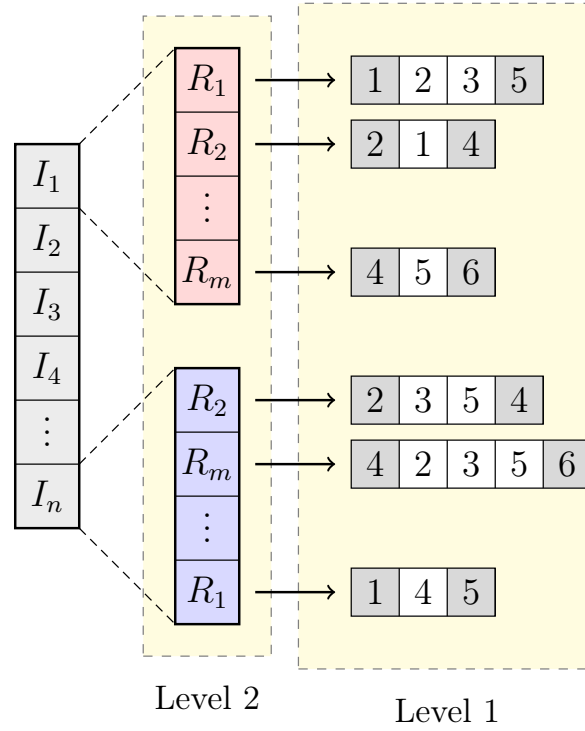


Figure 5.3: Representation of solution encoding in two operation levels. In this figure, the individual I_1 processes a sequence of requests starting with R_1 followed by R_2 , and individual I_n processes a sequence of requests that starts with R_2 followed by R_m . In individual I_1 the request R_1 has origin in node 1 and destination in node 5, with a path that comprises nodes 2 and 3. In individual I_n the request R_1 has origin in node 1 and destination in node 5 (the origin and destination nodes are necessarily the same ones), with a path that comprises only node 4.

When the search focuses on the request sequence (Level 2), the paths associated with each request are kept fixed, and the sequence of requests constitutes the decision variable. The decision variables in this case are represented by the ordered sets of requests, indicated by R_1, R_2, \dots, R_m , which define a priority order for request answering.

Several former works consider the routing as the unique space of search – this is called here the *single-level approach*. The proposed algorithm (the two-level approach) implements the *Level 1* and the *Level 2* encodings, which allow the construction of routing solutions that are built according to different request sequences. Although, in the case of solutions that accommodate all requests, both approaches are able to find the same solutions, it should be noted that the two-level approach is able to deal with partial solutions which may be rather useful in the construction of the final solution. This means that the two-level approach is likely to become more efficient in the search than the single-level approach. Of course, in the case of those

problems in which some requests are not attended, the single-level approach is not applicable.

Describing an example of instance represented by Figure 5.4 (a), a possible request R_1 of an individual I_1 may be composed of nodes 1 and 5 as source and destination respectively (Figure 5.4 (b)). It means that a potential path, represented in Figure 5.4 (c), may be composed of nodes 2 and 3. Another potential path for the same request R_1 associated with an individual I_n , represented in Figure 5.4 (d), may be composed of node 4.

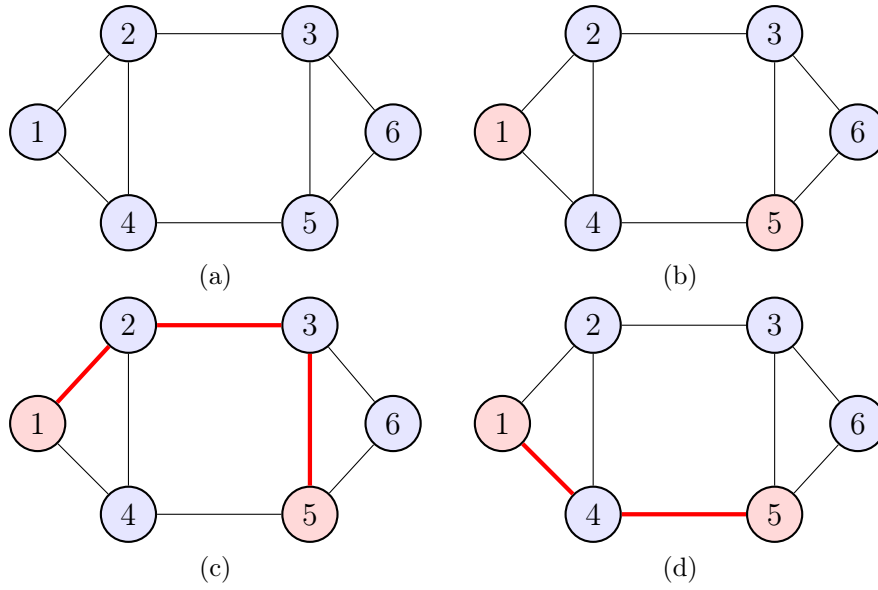


Figure 5.4: Representation of candidates paths in a given instance. In (a), the representation of the network instance. In (b), nodes 1 and 5 are source and destination respectively. In (c), 1 - 2 - 3 - 5 is a candidate path. In (d), 1 - 4 - 5 is another candidate path.

Considering another request, R_m for the same individual I_1 , for the same instance represented by Figure 5.5 (a), nodes 4 and 6 are defined as source and destination respectively (Figure 5.5 (b)). A potential path, represented in Figure 5.5 (c), may be composed of node 5. Another candidate path may be composed of nodes 2, 3 and 5 (Figure 5.5 (d)).

5.3.2 Solution Decoding

The solution decoding procedure evaluates an individual according to the steps described in Algorithm 3. The evaluation of an individual starts with the request which is situated on the top of the request list, proceeding to the next one immediately below, and so forth. It should be noted that if in any link the available bandwidth is

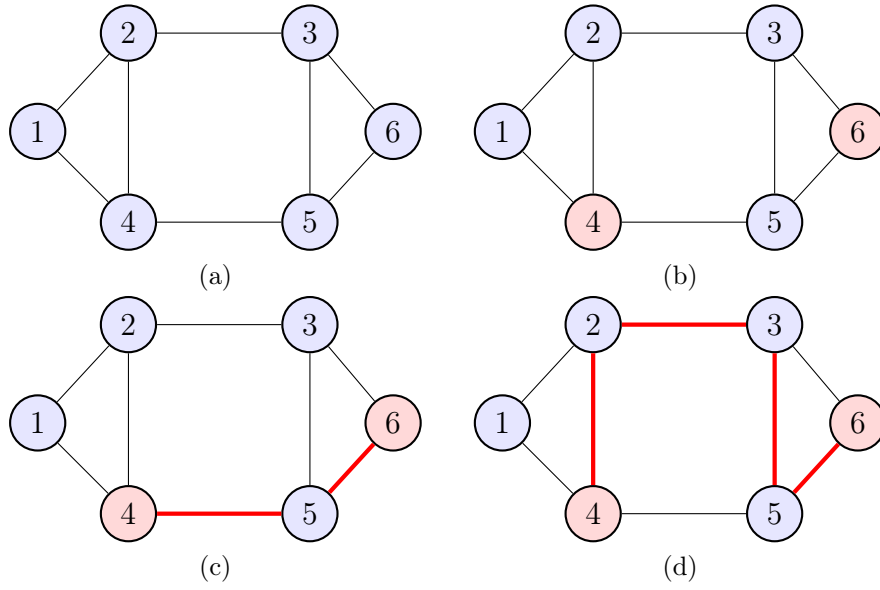


Figure 5.5: Representation of candidates paths in a given instance. In (a), the representation of the network instance. In (b), nodes 4 and 6 are source and destination respectively. In (c), 4 - 5 - 6 is a candidate path. In (d), 4 - 2 - 3 - 5 - 6 is another candidate path.

not enough, then a local search is performed within the decoding procedure, which may modify the individual.

Algorithm 3 Solution Decoding

```

1: for  $k \leftarrow 1$  to  $m$  do
2:   if there is available bandwidth in all links of request  $k$  then
3:     • request  $k$  is marked as "attended"
4:     • the available bandwidth in each link of the path is reduced by the amount  $b^k$ 
5:   else
6:     • a maximum path search is conducted from the origin to the destination of the request, considering the value of the available bandwidth in each link as the corresponding edge length
7:     if the request can be attended in this new path then
8:       • the path sequence is updated
9:       • the request is marked as "attended"
10:      • the available bandwidth in each link of the new path is reduced by the amount  $b^k$ 
11:     else
12:       • the request is marked as "rejected"
13:     end if
14:   end if
15: end for

```

After the evaluation of an individual, the objective function values can be estimated for such an individual. The objective function F_1 accumulates the number of nodes between the source and the destination of all attended requests; the objective function F_2 accumulates the number of rejected requests; and the objective function F_3 registers the percentual of occupation of the most loaded link in the network.

5.3.3 Initial population

The initial population of VN-MGA is composed of N individuals. The ordering of the requests in each individual of this initial population is randomly chosen. The initial paths are chosen such that: (i) 10% of the individuals are constructed with a minimum path algorithm, considering the number of links as the path length; (ii) 10% of the individuals are constructed with a minimum path algorithm, considering the percentual of bandwidth occupation of each link as the edge length; (iii) 10% of the individuals are constructed with a maximum path algorithm, considering the number of links as the path length; (iv) 70% of the individuals are created with a minimum path algorithm, considering random weights in the links.

It should be noted that the individuals created in steps (iii) and (iv) are relevant in order to provide a diversity of the solutions in the genetic pool of the initial population.

5.3.4 Crossover Operators

For both Levels 1 and 2, two different crossover operators are implemented. They were proposed under a general context. It means that they can be used in another graph problems, considering one or both levels, depending on the adopted encoding. This way, it may guarantee the population factibility.

Level 1 crossover consists of the exchange of genetic material between individuals concerning the encoded routes. This crossover, which is illustrated in Figures 5.6 and 5.7, is implemented as described in Algorithm 4.

Level 2 crossover performs combinations of individuals concerning the request sequence. This crossover operator, which is illustrated in Figure 5.8, is presented in Algorithm 5. Both crossover operators deliver only one offspring.

5.3.5 Mutation Operators

The mutation operators are responsible for the insertion of new genetic characteristics in the population. As in the case of crossover operators, the mutation operators are also implemented in two levels. The Level 1 Mutation is presented in Algorithm

Algorithm 4 Level 1 Crossover

```

1: • Two individuals  $I_i$  and  $I_j$  are randomly selected;
2: if  $I_i \prec I_j$  then
3:    $O \leftarrow I_i$ 
4: else if  $I_j \prec I_i$  then
5:    $O \leftarrow I_i$ 
6: else
7:   for  $k \leftarrow 1$  to  $m$  do
8:     if the paths of request  $k$  in both individuals share some nodes then
9:       • randomly choose a common node  $n_c$ ;
10:      • the path of request  $k$  in offspring solution  $O$  starts in node  $o^k$ , using the
        path of individual  $I_i$  until node  $n_c$ , and then proceeds with the path of
        individual  $I_k$  until node  $d^k$  (see figure 5.6);
11:     else if there is a node  $n_d$  which is reachable, in one step, from some node of
        the path of individual  $I_j$ , and which either belongs to the path of individual
         $I_k$  or reaches this path in one step then
12:       • the path of request  $k$  in offspring solution  $O$  starts in node  $o^k$ , using
        the path of individual  $I_i$  plus one step outside this path, until node  $n_d$ ,
        and then proceeds with the path of individual  $I_j$  until node  $d^k$ , eventually
        employing one step outside this path (see figure 5.7);
13:     else
14:       • the offspring solution  $O$  inherits the path of request  $k$  either from  $I_i$  or
        from  $I_j$ , randomly chosen.
15:     end if
16:   end for
17: end if
18: return offspring  $O$ 

```

Algorithm 5 Level 2 Crossover

```

1: Two individuals are randomly selected;
2: A request is randomly chosen;
3: The two routes that are associated with that request in the different individuals
   are swapped, generating a new individual.

```

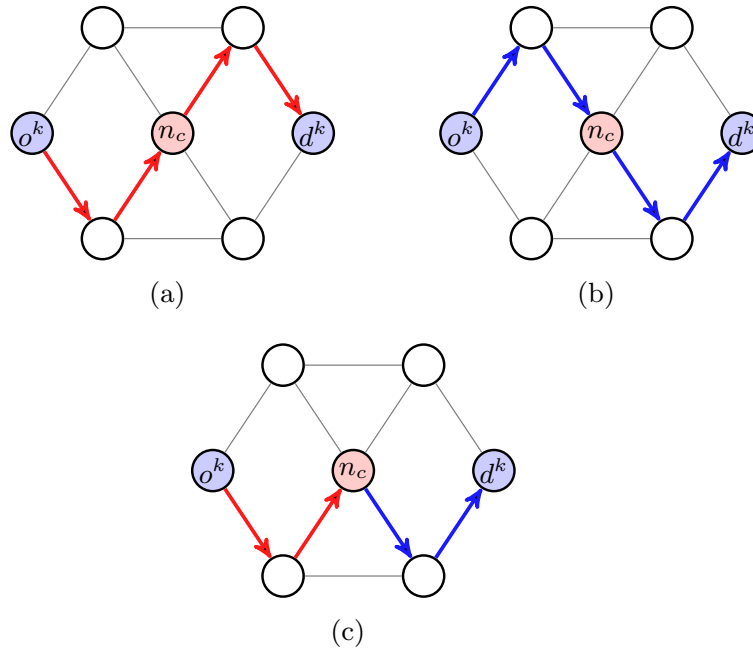


Figure 5.6: Level 1 Crossover: a common node n_c exists (Santos, 2009). (a) The route of parent I_1 for request r . (b) The route of parent I_2 for request r . (c) The resulting route in the offspring for request r .

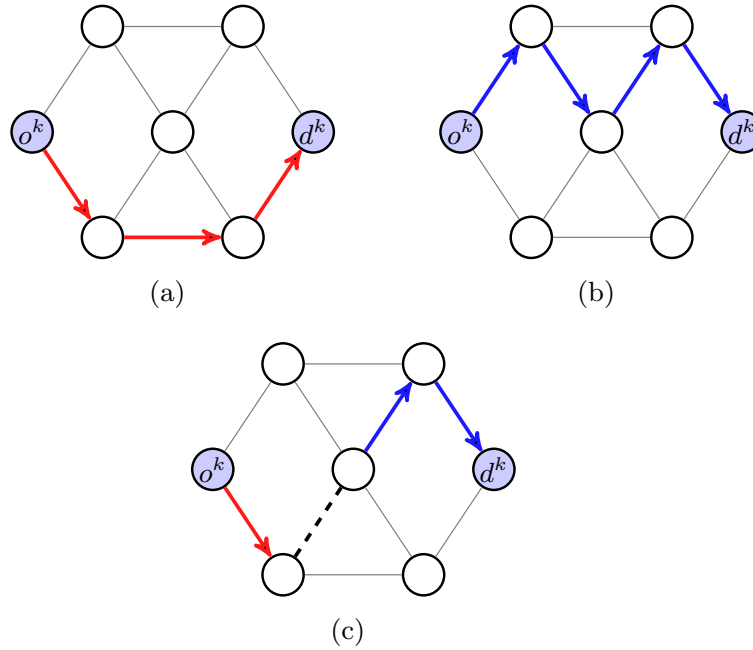


Figure 5.7: Level 1 Crossover: no common node (Santos, 2009). (a) The route of parent I_1 for request r . (b) The route of parent I_2 for request r . (c) The resulting route in the offspring for request r .

6. In the Level 1 Mutation, a new section of the mutated route is created. This search is performed 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), alter-

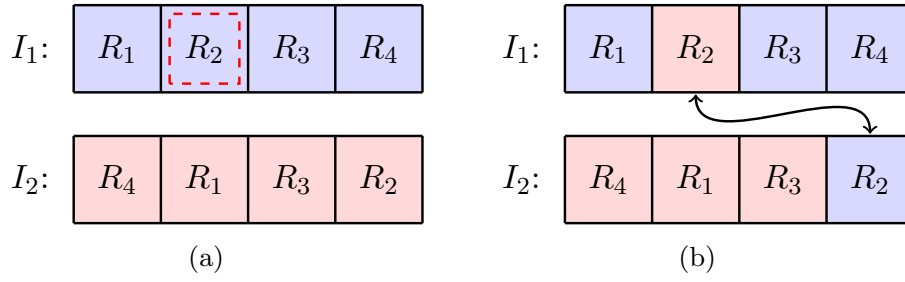


Figure 5.8: Level 2 Crossover. (a) Request R_2 is randomly selected. (b) The routes of request R_2 in I_1 and I_2 are interchanged, generating the offspring.

nately, avoiding any bias in this search. Figure 5.9 illustrates this mutation process.

Algorithm 6 Level 1 Mutation

- 1: • an individual is randomly chosen;
 - 2: **for** $k \leftarrow 1$ to m **do**
 - 3: • p is chosen from a uniform distribution in the interval $[0, 1]$;
 - 4: **if** $p < pmut$ **then**
 - 5: • Two cutoff points are randomly chosen in the path of request k ;
 - 6: • The Dijkstra algorithm looks for a new path considering random costs distributed on the links, joining the first cutoff point to the second one.
 - 7: **end if**
 - 8: **end for**
-

The Level 2 Mutation is presented in Algorithm 7. It just performs a swap of the position of requests in the precedence list. This mutation operator is illustrated in Figure 5.10.

Algorithm 7 Level 2 Mutation

- 1: Choose randomly an individual;
 - 2: Select randomly two requests;
 - 3: Swap the position of those requests in the sequence.
-

5.3.6 VN-MGA Structure

The basic structure of the Variable Neighborhood Multiobjective Genetic Algorithm (VN-MGA) proposed here is similar to 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 of sorting the solutions according to the non-dominance ranking. An individual belonging to rank 1 is not dominated by any

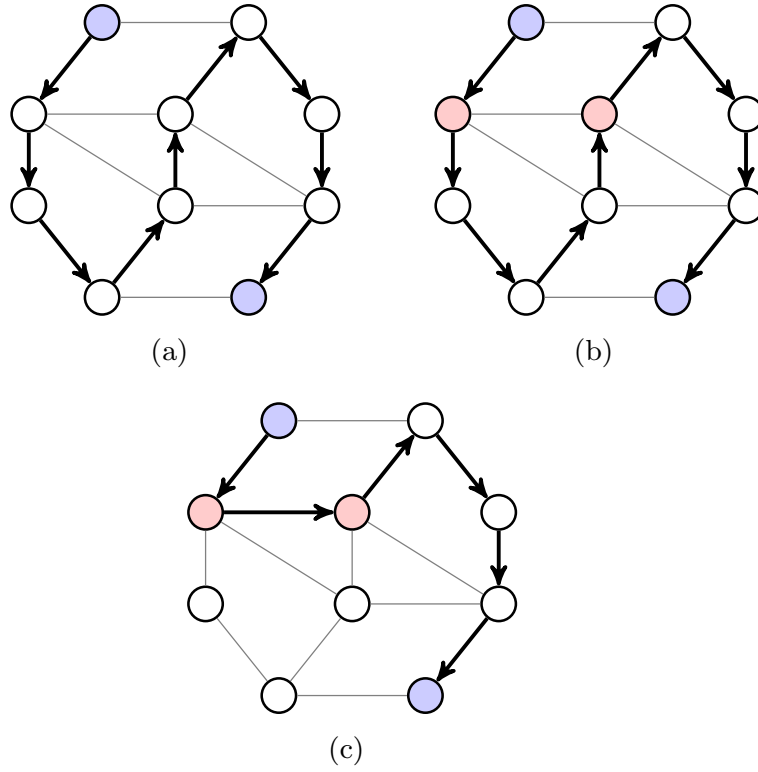


Figure 5.9: Level 1 Mutation. (a) The route for request r . (b) Random selection of two intermediate nodes which belong to the route for a new sub-route. (c) Mutated individual.

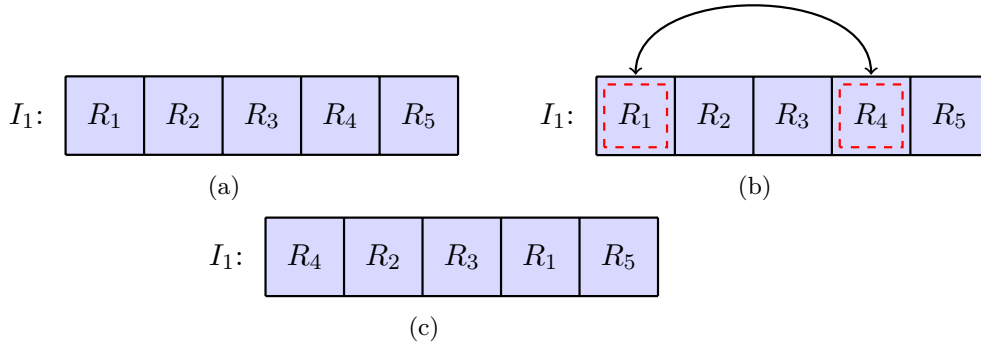


Figure 5.10: Level 2 Mutation. (a) Sequence of requests $R_1 \dots R_5$ from an individual I_1 . (b) Selected requests for mutation: R_1 and R_4 . (c) Mutated individual.

solution, while an individual belonging to rank q is dominated by at least one individual that belongs to rank $q - 1$ and by no solution belonging to rank q or greater. 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 edge lengths of a hypercube with vertices situated on the ℓ

nearest solutions (in which ℓ stands for 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-optimal set.

3. *Binary tournament*: consists of randomly choosing two individuals and comparing them according to a 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.

The VN-MGA procedure is presented in Algorithm 8, which describes in more detail the flowchart of Figure 5.2.

Algorithm 8 VN-MGA pseudocode

-
- 1: Generate an *initial population* Q with N individuals;
 - 2: Evaluate the objective functions on Q ;
 - 3: Assign the *non-dominated sorting ranking* to the individuals of Q ;
 - 4: Assign the *crowding distances* to the individuals of Q ;
 - 5: **while** not stop condition 0 **do**
 - 6: **while** not stop condition 1 **do**
 - 7: Using a *binary tournament*, considering the non-dominated sorting rankings and the crowding distances, select $0.75N$ individuals from Q , forming population C , and select other $0.75N$ individuals from Q , forming population M
 - 8: Perform *Level 1 crossover* on the individuals of C , generating a subpopulation \bar{C} with a number of individuals equal to $0.75N$;
 - 9: Perform *Level 1 mutation* on the individuals of M , generating a subpopulation \bar{M} with a number of individuals equal to $0.75N$;
 - 10: Evaluate the objective functions on the individuals of \bar{C} and \bar{M} ;
 - 11: Join the population Q with subpopulations \bar{C} and \bar{M} , forming population $\bar{Q} = Q \cup \bar{C} \cup \bar{M}$;
 - 12: Assign the *non-dominated sorting ranking* to the individuals of \bar{Q} ;
 - 13: Assign the *crowding distances* to the individuals of \bar{Q} ;
 - 14: Select deterministically N individuals from \bar{Q} , considering the non-dominated sorting rankings and the crowding distances, forming population Q ;
 - 15: **end while**
 - 16: **while** not stop condition 2 **do**
 - 17: Using a *binary tournament*, considering the non-dominated sorting rankings and the crowding distances, select $0.75N$ individuals from Q , forming population C , and select other $0.75N$ individuals from Q , forming population M
 - 18: Perform *Level 2 crossover* on the individuals of C , generating a subpopulation \bar{C} with a number of individuals equal to $0.75N$;
 - 19: Perform *Level 2 mutation* on the individuals of M , generating a subpopulation \bar{M} with a number of individuals equal to $0.75N$;
 - 20: Evaluate the objective functions on the individuals of \bar{C} and \bar{M} ;
 - 21: Join the population Q with subpopulations \bar{C} and \bar{M} , forming population $\bar{Q} = Q \cup \bar{C} \cup \bar{M}$;
 - 22: Assign the *non-dominated sorting ranking* to the individuals of \bar{Q} ;
 - 23: Assign the *crowding distances* to the individuals of \bar{Q} ;
 - 24: Select deterministically N individuals from \bar{Q} , considering the non-dominated sorting rankings and the crowding distances, forming population Q ;
 - 25: **end while**
 - 26: **end while**
-

Results

With the purpose of determining the effect produced by the two-level encoding, which was presented in the VN-MGA, some experiments have been performed. Initially, we validate the implementation of the proposed algorithm. Afterwards, we validate the functionality of the proposed approach. Those tests were performed with some network instances which are often used in the evaluation of algorithms for telecommunication routing problems.

6.1 Validating the implementation

The algorithms presented in this work need to be validated. Generally, the validation of algorithms is done by comparing with other similar algorithms, previously created to solve the same problem. In this context, the term numerical validation is used to represent observation procedures, comparison and evaluation of algorithms with their technical specifications. Methods are considered validated if they can solve, within the accuracy limits defined, the proposed test problems.

Aiming at validating the implementation of the proposed algorithm, initially, we use a very simple network instance, represented in Figure 5.1(e). An implementation of the ϵ -constraint method in a ILP solver looked for values of two objective functions, F_1 and F_3 , equating F_2 to zero. After, the proposed algorithm was tested.

The tests considered 10, 20, 30 and 40 simultaneous requests, with 150 generations and 100 individuals, ignoring results with F_2 different from zero. The proposed

algorithm reached the same values of the exact method, providing support to the proposed method. Such results are represented in Figure 6.1.

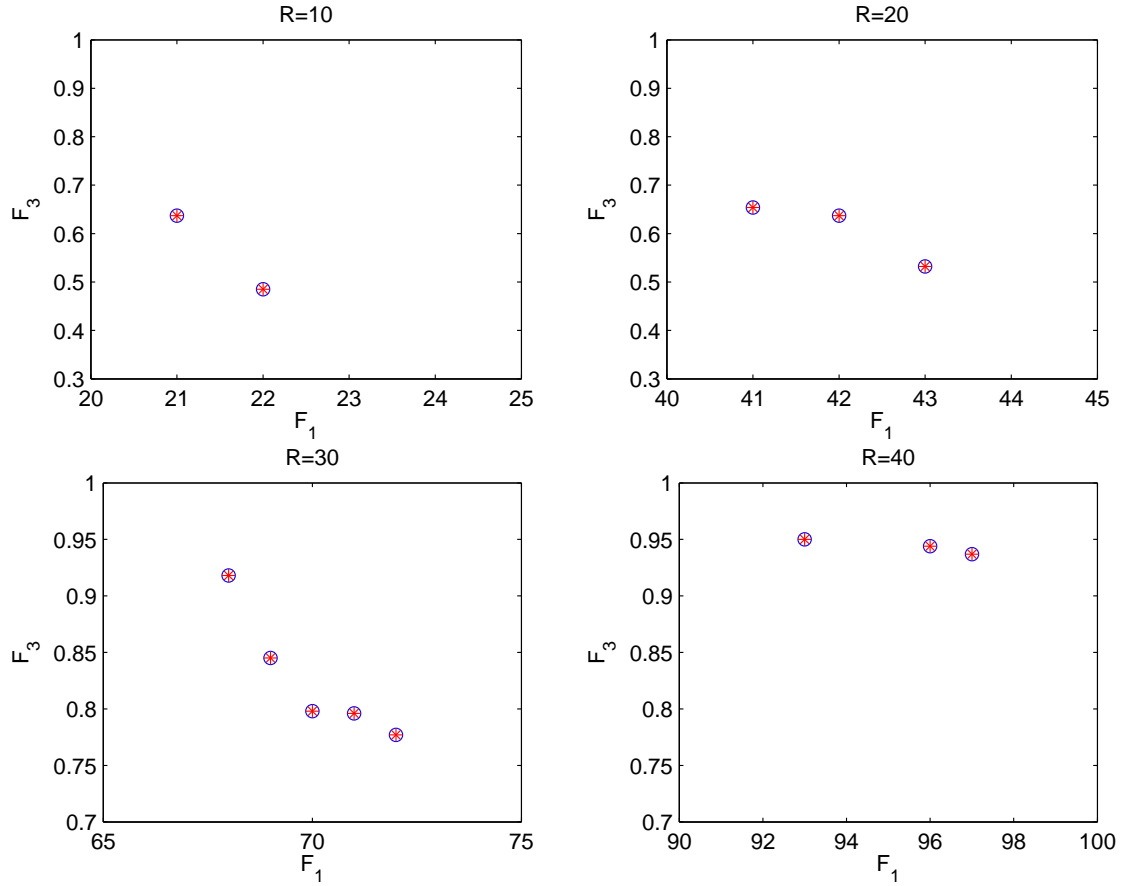


Figure 6.1: Comparison between VN-MGA (represented by asterisks) and ILP ϵ -constraint (represented by circles) results for 10, 20, 30 and 40 simultaneous requests for the network instance represented in Figure 5.1(e).

6.2 Validating the proposed algorithm

Some experiments have been performed in order to compare the proposed approach with other alternative formulations. Three analysis were performed. The first one consists of a comparison among the proposed algorithm, the basic VNS and an exact solution obtained from an Integer Linear Programming solver. The second experiment compares the performance of VN-MGA with single-level encodings of the same Genetic Algorithm and with the basic VNS. The third one analyzes the quality of the solutions delivered by the algorithms, after 20 executions of each algorithm, using the Hypervolume Metric (Zitzler, 1999).

These tests considered an available bandwidth for each link of 1024 kbps. Each request required a bandwidth of 200 kbps or 400 kbps (randomly chosen). Source

and destination of each request are randomly assigned. Different scenarios, with different numbers of requests were considered in the tests.

6.2.1 Comparison with Integer Linear Programming (ILP) results

In first place, the VN-MGA was compared with the results obtained by Integer Linear Programming (ILP) solvers, which provide exact solutions of the problem. The ILP formulation determined the solutions of the multiobjective problem by using a scalarization approach, the ϵ -constraint method (Ehrgott, 2000). In this approach, the multiobjective optimization problem is converted into several single-objective problems, each one with only one objective function to be optimized. The other objective functions are treated as constraints. Each solution of such a problem is at least a weakly non-dominated solution (Ehrgott, 2000). By varying the values ϵ_i of the constraints, it is possible to generate all the solutions belonging to the Pareto-optimal set of the problem. In this way, a multiobjective problem as defined in (4.1) can be modeled in terms of ℓ problems:

$$\begin{aligned} \mathbf{x}^* &= \arg \min f_i(\mathbf{x}) \\ \text{s.t. } &\begin{cases} f_j(\mathbf{x}) \leq \epsilon_j ; j = 1, \dots, \ell ; j \neq i \\ \mathbf{x} \in \mathcal{X} \end{cases} \end{aligned} \quad (6.1)$$

In order to define an ϵ -constraint version of problem (5.1), only F_1 and F_3 objective functions were considered, and F_2 function was fixed in zero (no request rejection). In this way, the multiobjective problem studied the trade-off between cost of time delay in the network and the relative occupation of the most loaded link. In first place, F_1 function was minimized without a constraint in F_3 value. Afterwards, starting from the optimal value of F_1 , the ILP solver minimized F_3 objective function, with F_1 employed as a constraint which was relaxed, from that minimal value until the minimum of F_3 was reached. The ILP tools FICOTMXPress Optimization and CPLEX solver were used in this procedure, leading to identical results.

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

Another comparison which was performed at the same time was with the results provided by the original VNS algorithm, as defined by Algorithm 2, using the same Level 1 and Level 2 neighborhoods. This algorithm employed the same ϵ -constraint

scheme, generating only one estimate of a Pareto-optimal solution for each algorithm run.

Some fixed parameters of the VN-MGA are described in Table 6.1. The choice

Table 6.1: Parameters for the VN-MGA algorithm

Mutation Probability ($pmut$)	0.4
Number of generations	150
Number of individuals (N)	100

of those parameters was performed, as a preliminary step, considering a rough grid, in which the $pmut$ value had resolution of 0.1 and both the number of generations and the number of individuals had a resolution of 20. Each combination was then executed. The final values were chosen as the ones that produced accurate results within reasonable computational time.

It can be inferred, from Table 6.1, that the number of function evaluations in each run of VN-MGA with this configuration is 45000. In order to provide a comparison which becomes even unfavorable for the VN-MGA algorithm, the basic VNS was also run with 45000 function evaluations *for each* ϵ value, which means 630000 function evaluations for the search of the whole Pareto-optimal set.

The test instance 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 ILP ϵ -constraint method. For 20 and 30 simultaneous requests, not all the values were reached. In some cases, the results by VN-MGA presented a gap in relation to the exact solution. The algorithm was not able to cover the whole Pareto front. Nevertheless, it clearly outperforms the basic VNS algorithm both in terms of convergence and diversity. Figure 6.2 shows the results for 30 requests, with the best values reached by the VN-MGA and the basic VNS, after 21 runs, against the values provided by the ILP approach.

Although the ILP approach has delivered the best solutions for the problem under consideration, there are some drawbacks with this approach which may render it impracticable. The ILP solver has to be executed once for generating each solution of the Pareto-optimal set. Beyond that, it requires a simplification of the problem, namely, the use of only two objective functions. But the main inconvenience is that its computational complexity is exponential in the number of decision variables. Therefore, this kind of approach is not suitable for even slightly larger problem instances. An example of problem instance for which the proposed method works and the ILP formulation does not run, in the same computer environment, is also presented in the

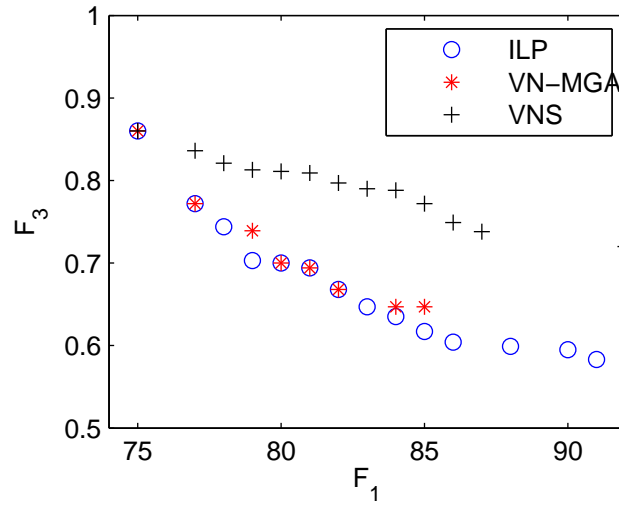


Figure 6.2: Comparison among ϵ -constraint, VNS and VN-MGA algorithm for $R=30$. Circles represent the solutions achieved by the ϵ -constraint ILP technique. The solutions delivered by VNS are represented by black crosses and the solutions of the proposed VN-MGA are represented by red asterisks. (Horizontal axis: F_1 . Vertical axis: F_3).

comparison.

6.2.2 Comparison among VN-MGA, single-level encoding GAs and the basic VNS

This experiment compares the performance of VN-MGA with single-level encodings of Genetic Algorithms and the basic VNS. Those single-level versions are built by simply replacing the section in Algorithm 8 which performs search in a level by a corresponding section which performs search in the other level. In this way, the algorithm performs twice the search in the same level, instead of changing the level. In this case, differently from the first experiment, the whole problem is considered, with three objective functions. The same instance represented in Figure 5.1(d) is considered, with 40 simultaneous requests. The same set of parameters presented in Table 6.1 is employed again.

As can be observed in Figure 6.3, the variable encoding represented by VN-MGA delivers the best results among all algorithms. With few exceptions, most of the solutions found by the other algorithms were dominated by some solution provided by VN-MGA.

Interestingly, the second best algorithm was the Level 2 MGA. It suggests that the greedy operators, which provide the routes in the initial population plus the local search included in the solution decoding routine, are able to reach good solutions in

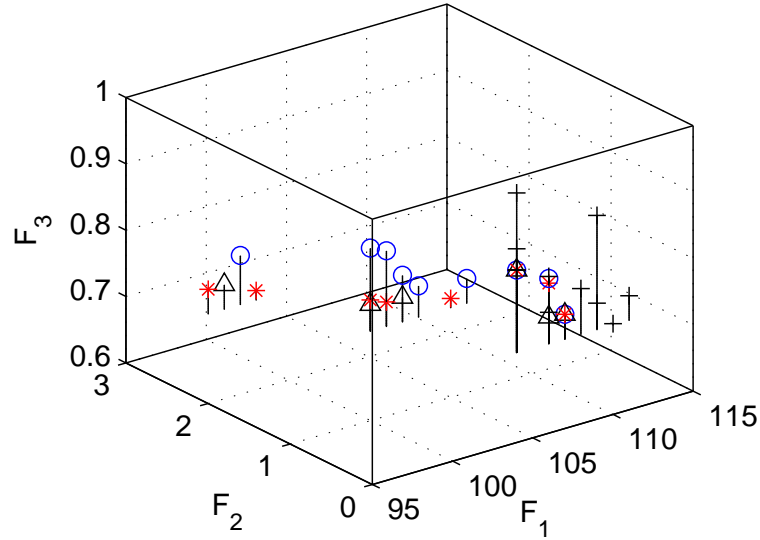


Figure 6.3: Comparison among VN-MGA, single-level encoding GAs and the basic VNS for the network instance of Figure 5.1(d). Circles represent solutions achieved by Level 1 GA. Triangles represent solutions obtained by Level 2 GA. Crosses indicate solutions provided by VNS. Asterisks indicate solutions provided by VN-MGA.

this problem.

Likewise, another experiment considered 40 simultaneous requests, 150 generations and 100 individuals. The same instance represented in Figure 5.1(e) is considered. The purpose of this experiment is the comparison between single-level encoding GAs and the proposed VN-MGA with three objective functions, as shown in Figure 6.4. The results are represented by circles (Level 1), triangles (Level 2) and asterisks (VN-MGA). The figure shows the domination of VN-MGA solutions, also presenting a better spread of solutions. This indicates the better performance of the two-level encoding.

6.2.3 Hypervolume metric

The hypervolume metric is addressed to measure the proximity of solution sets to the Pareto front and to quantify the uniformity of solution distribution over the front. This indicator is defined as follows: Each point of the solution set A defines a vertex of a hypercube hc_i . The opposite vertex, considering the main diagonal, is a fixed point whose coordinates may be defined as the worst values attained by each objective for all solution samples in the Pareto-front. Considering a solution set A , the union of the regions covered by all such hypercubes generates a figure whose hypervolume defines the hypervolume metric. Equations 6.2 and 6.3 illustrate this metric.

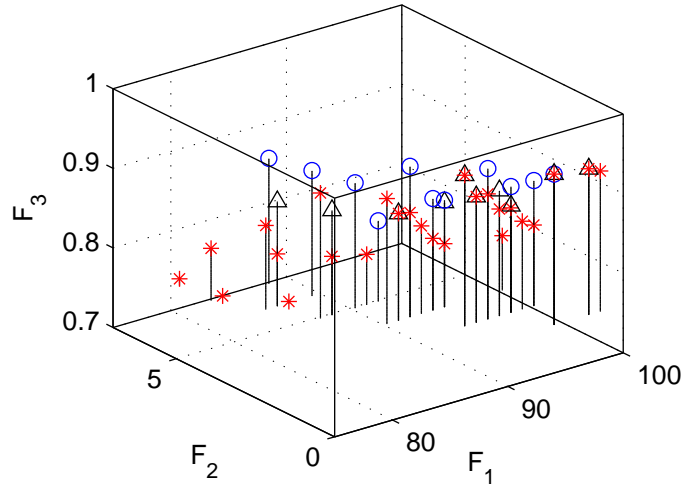


Figure 6.4: Comparison among VN-MGA, single-level encoding GAs and the basic VNS for the network instance of Figure 5.1(e). Circles represent solutions achieved by Level 1 GA. Triangles represent solutions obtained by Level 2 GA. Asterisks indicate solutions provided by VN-MGA.

$$HV_A = volume(\cup_{i=1}^{|A|} hc_i) \quad (6.2)$$

$$HVR_A = HV_A / HV_{Y^*} \quad (6.3)$$

Hypervolume metric (Zitzler, 1999) was employed here in order to compare the basic VNS and the proposed VN-MGA algorithm. Also the single-level GAs involving only Level 1 and Level 2 operations are included, with the aim of quantifying the gain of the combination of levels. For the hypervolume computation, the package described by Fonseca *et al.* (2006) was used. Table 6.2 shows the values of hypervolume metric calculated for the Pareto fronts produced by each algorithm with the set of parameters of Table 6.1, after 20 runs, using the instance of Figure 5.1(d) with 40 simultaneous requests.

These results are also shown in the boxplot of Figure 6.5. Comparing them, one observes that the VN-MGA attains significantly better hypervolume values than the other algorithms. Even its worst value is better than the best value provided by the other ones.

The problem instance represented in Figure 5.1(d) with 40 simultaneous requests is, indeed, near the edge of problem sizes which are solvable by the ILP formulation in computational environments such as the one used in the comparisons performed

Table 6.2: Hypervolume metric after 20 runs of algorithms with Level 1, Level 2, VNS and VN-MGA, for 40 simultaneous requests.

Hypervolume	Level 1	Level 2	VNS	VN-MGA
Best value	27.73	26.22	19.83	44.64
Worst value	19.18	21.76	14.20	30.39
Average value	23.75	25.31	16.97	35.20
Standard Deviation	2.02	1.23	1.66	5.11

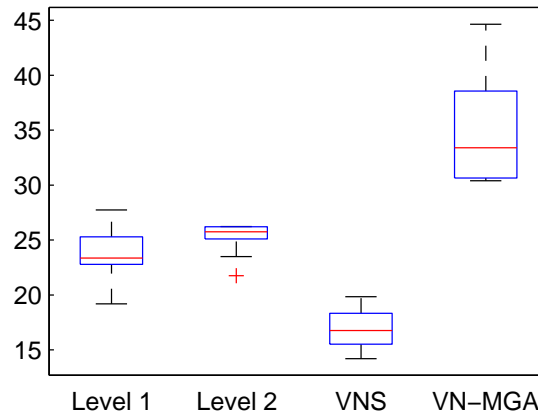


Figure 6.5: Boxplot results of the hypervolume metric after 20 runs of algorithms with Level 1, Level 2, VNS and VN-MGA, for 40 simultaneous requests.

here¹. The same network topology with 50 simultaneous requests no longer runs in that environment. In order to illustrate the performance of the proposed methodology in that problem instance, Table 6.3 presents the hypervolumes of the solution sets achieved by VN-MGA, Level 1 and Level 2 algorithms. Those results were obtained with 15 runs of each algorithm.

Table 6.3: Hypervolume metric after 15 runs of algorithms with Level 1, Level 2, and VN-MGA, for 50 simultaneous requests.

Hypervolume	Level 1	Level 2	VN-MGA
Best value	19.13	31.13	37.40
Worst value	13.67	24.35	25.86
Average value	17.53	26.76	31.91
Standard Deviation	1.66	1.94	3.16

¹Hardware: Intel Quad Core 2.66 GHz 64bits with 2GB of DDR-400 RAM. Software: Windows XP SP3 and FICO XPress V. 7.01.

Conclusions and Future work

7.1 *Conclusion*

This thesis proposed an algorithm to optimize the multiple objectives that represent Quality of Service indices on IP networks, in the context of packet routing in MPLS-IP networks. It aimed at finding LSPs in MPLS domain. This way, the routing process tried to minimize the delay, keeping the load balancing of the network and minimize the rejection of simultaneous requests.

The multiobjective techniques applied are based on genetic algorithms due to good results observed in similar applications problems, besides the advantage of producing a number of different solutions. The concept of walking between borderer basins of attraction in a stochastic and iterative way acts as an effective solution that comes from the concepts of VNS techniques. The proposed optimization model relies on the definition of specific Genetic Algorithms operators and the study of different alternatives for the choice of routes on IP networks.

The presented algorithm, VN-MGA, is a Genetic Algorithm based on the NSGA-II with inspirations on VNS. Genetic operators were implemented in two levels that encode the same individual in different ways to the same problem. In the first level, named Level 1, each node is analyzed in order to establish the routes to be followed under a "microscopic" point of view. The solution is encoded considering as decision variables the edges that form the routes to be followed by each request. In the second level, named Level 2, the operators focus on requests. The solution is encoded

with the routes considered as fixed, and the sequence of requests considered as the decision variable. The treatment of each request is based on the arrival order, i.e., there is no priority ranking between simultaneous requests. This level considered routing under a "macroscopic" scenario that uses pre-defined routes of Level 1.

The modification of search space, aided by the levels change, generates different local optima for each neighbor. It makes possible to jump from one attraction basin to another, avoiding premature convergence of the algorithm. The expansion of the search space allows a greater diversity of solutions belonging to the Pareto Front. It results in a large range of options that may be considered by 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.

From results presented in Chapter 6, in comparison with single levels, the combination of levels generates non-dominated solutions, which thus confirms the efficiency of the proposed method. It can be stated that the implementation with a new kind of encoding combined in a Multiobjective Genetic Algorithm with features based in concepts of VNS, the results show an increasing of the convergence and diversity of solutions. Solutions not envisioned by other algorithms that dealt with the same problem have been achieved. These results are innovative facing an IP network problems with particular features. It is important to say that the proposed algorithm gives solutions comparable to those obtained by exact methods. Furthermore the algorithm is able to run larger instances than those obtained by commercial solvers. It means that the method allows indications of high quality results in feasible computational time.

7.2 *Limitations*

It is important to note the methodological limitations of the studies involved in this thesis. Although this research was conducted to indicate routes to provide QoS parameters, it did not examine a very important QoS parameters, that is the loss of packets. However, an investigation in a dynamic environment should be carefully analyzed due to the limitations of simulation softwares, such as the number of possible paths as described by Andrade (2008). It can be investigated in a future work.

It should be highlighted that, if scenarios of routers or backbones are analyzed, for example, the instances used in Chapter 6 can be classified as medium size. However, with these medium size instances, unlike the proposed algorithm, commercial solvers

were not capable of running.

Another limitation of this study is that there were no previous works with the same configuration to be used as benchmarks. For this reason, this research extracted reference values until the limitation of the commercial solvers to validate the proposed algorithm. It can be seen as a limitation but also as an advantage of an innovative approach.

7.3 Future works

The good behavior presented by the proposed VN-MGA, outperforming some algorithm versions that do not employ variable neighborhood 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?

There are two principal directions for further development of the approach proposed here. The first one considers the specific problem of routing. The second one considers the theoretical problem of employing variable neighborhoods or different structures encoding in generic evolutionary algorithms.

Concerning the routing 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.

Concerning the theoretical problem of studying variable neighborhoods in generic evolutionary algorithms, there are several open issues. We intend to tackle, in the near future, some issues related to the usage of encodings that allow metric operations (Moraglio *et al.* , 2007; Carrano *et al.* , 2010).

7.4 Publications

ONETY, R. E. ; TADEI, R. ; MAGELA NETO, O. ; TAKAHASHI, R. H. C.. Multiobjective Optimization of MPLS-IP Networks with a Variable Neighborhood Genetic Algorithm. In: Applied Soft Computing, 2013. v. 13, p. 4403-4412.

ONETY, R. E. ; MOREIRA, G. J. P. ; MAGELA NETO, O. ; TAKAHASHI, R. H. C.. Variable neighborhood multiobjective genetic algorithm for the optimization

of routes on IP networks. In: 6th International Conference on Evolutionary Multicriterion Optimization, 2011, Ouro Preto. Evolutionary Multi-Criterion Optimization (LNCS 6576). Berlin: Springer-Verlag, 2011. v. 1. p. 433-447.

ONETY, R. E. ; MOREIRA, G. J. P. ; ERRICO, L. . Abordagem multiobjetivo baseada em AG e VNS para otimização de rotas em redes IP. In: 2a. Escola Luso-Brasileira de Computação Evolutiva (ELBCE), 2010, Guimarães, Portugal.

Bibliography

- ALVARADO, C., HERAZO, I., ARDILA, C., & DONOSO, Y. 2005. Aplicación de NSGA-II y SPEA-II para la optimización multiobjetivo de redes multicast. *Ingeniería y desarrollo*, **17**, 28–53.
- ANDRADE, ALESSANDRO VIVAS. 2008 (December). *Provisionamento de Qualidade de Servico em Redes MPLS utilizando Algoritmos Bio-inspirados em um Ambiente de Trafego Auto-Similar. (in Portuguese)*. Ph.D. thesis, Universidade Federal de Minas Gerais.
- AWDUCHE, D., MALCOLM, J., AGOGBUA, J., O'DELL, M., & MCMANUS., J. 1999. Requirements for traffic engineering over MPLS. *Request for comments*, **2702**(Sep).
- AWDUCHE, D., CHIU, A., NETWORKS, CELION, & ELWALID, A. 2002. Overview and Principles of Internet Traffic Engineering. *Request for comments*, **3272**(May).
- BAGUENINE, F., & MELLOUK, A. 2007. N-Best Optimal Path Ant Routing Algorithm for State-Dependent N-Best Quality of service Routes in IP Networks. *Pages 747–754 of: 32nd IEEE Conference on Local Computer Networks, 2007. LCN 2007*.
- BAGULA, AB. 2006. Traffic Engineering Next Generation IP Networks Using Gene Expression Programming. *Pages 230–239 of: 10th IEEE/IFIP Network Operations and Management Symposium, 2006. NOMS 2006*.
- BANERJEE, N., & KUMAR, R. 2007. Multiobjective network design for realistic traffic models. *Pages 1904–1911 of: Proceedings of the 9th annual conference on Genetic and Evolutionary Computation*. ACM Press New York, NY, USA.
- BATTITI, ROBERTO. 1996. Reactive search: Toward self-tuning heuristics. *Modern heuristic search methods*, 61–83.

- BRADEN, ROBERT, ZHANG, LIXIA, BERSON, STEVE, HERZOG, SHAI, & JAMIN, SUGIH. 1997. *Resource reservation protocol-version 1 functional specification*. Tech. rept. RFC 2205, September.
- BUI, V., ZHU, W., PESCAPE, A., & BOTTA, A. 2007. Long Horizon End-to-End Delay Forecasts: A Multi-Step-Ahead Hybrid Approach. *Pages 825–832 of: 12th IEEE Symposium on Computers and Communications, 2007. ISCC 2007*.
- CARRANO, E. G., SOARES, L. A. E., TAKAHASHI, R. H. C., SALDANHA, R. R., & 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., & NETO, O. M. 2010. Non-linear network topology optimization - An embedding vector space approach. *IEEE Transactions on Evolutionary Computation*, **14**, 206–226.
- CARRANO, EDUARDO GONTIJO. 2007. *Algoritmos Evolucionários eficientes para Otimização de Redes. (in Portuguese)*. Ph.D. thesis, Universidade Federal de Minas Gerais.
- CERAV-ERBAS, SELIN KARDELEN. 2004. *Traffic engineering in mpls networks with multiple objectives: Modeling and optimization*. Ph.D. thesis, RWTH Aachen, Germany.
- CORTEZ, P., RIO, M., ROCHA, M., & SOUSA, P. 2006. Internet Traffic Forecasting using Neural Networks. *Pages 2635–2642 of: Neural Networks, 2006. IJCNN'06. International Joint Conference on*.
- DE FREITAS, LIA MARA BORGES, & MONTANÉ, F. A. T. 2008. Metaheurísticas VNS-VND e GRASP-VND para Problemas de Roteamento de Veículos com Coleta e Entrega Simultânea. (in Portuguese). *In: XI Simpósio de Pesquisa Operacional e Logística da Marinha*.
- DE GIOVANNI, L., DELLA CROCE, F., & 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, ERITO MARQUES. 2007. *Variable Neighborhood Search (VNS) aplicado ao problema de distribuição dutoviária. (in Portuguese)*. M.Phil. thesis, Universidade Federal do Rio de Janeiro.
- DEB, K., PRATAP, A., AGARWAL, S., & MEYARIVAN, T. 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, **6**(2), 182–197.

- DIAS, ROBERTO ALEXANDRE. 2004. *Engenharia de Trafego em Redes IP sobre Tecnologia MPLS: Otimização Baseada em Heurísticas*. Ph.D. thesis, Universidade Federal de Santa Catarina.
- DRUMMOND, AC, DA FONSECA, NLS, & DEVETSIKIOTIS, M. 2008. A multiobjective fuzzy bandwidth partitioning model for self-sizing networks. *European Journal of Operational Research*, **191**(3), 1161–1174.
- DUCZMAL, LUIZ, CANÇADO, ANDRÉ L. F, & TAKAHASHI, RICARDO H. C. 2008. Delin-eation of irregularly shaped disease clusters through multiobjective optimization. *Journal of Computational and Graphical Statistics*, **17**(1), 243–262.
- EHRGOTT, M. 2000. *Multicriteria optimization*. Springer-Verlag.
- EMBRATEL. 2013. *Rede Unica Dados MPLS*. (in Portuguese). http://portal.embratel.com.br/mge/solucoes_dados_rud_mpls.jsp. (Accessed: 07/2013).
- ERBAS, S.C., & ERBAS, C. 2003. A multiobjective off-line routing model for MPLS networks. *Pages 471–480 of: Providing Quality of Service in Heterogeneous Environments-Proc. of the 18th International Teletraffic Congress (ITC-18)*, vol. 16. Citeseer.
- FONSECA, CARLOS M., PAQUETE, LUÍS, & LÓPEZ-IBÁÑEZ, MANUEL. 2006 (July). An improved dimension-sweep algorithm for the hypervolume indicator. *Pages 1157–1163 of: IEEE Congress on Evolutionary Computation*.
- GIRISH, MUCKAI K, ZHOU, BEI, & HU, JIAN-QIANG. 2000. Formulation of the traffic engineering problems in mpls based ip networks. *Pages 214–219 of: Computers and Communications, 2000. Proceedings. ISCC 2000. Fifth IEEE Symposium on*. IEEE.
- GLOVER, FRED. 1989. Tabu search-Part I. *ORSA Journal on computing*, **1**(3), 190–206.
- GOLDBERG, D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
- HANSEN, P., & MLADENOVI, N. 2001. Variable Neighborhood Search: Principles and applications. *European Journal of Operational Research*, **130**(3), 449–467.
- HOLLAND, J. H. 1975. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA.

- IETF. 2013. *Internet engineering task force*. <http://www.ietf.org>. (Accessed: 07/2013).
- ITU-T REC.X.200. 1994. *Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*. Tech. rept. International Telecommunication Union ITU-T.
- KNOWLES, JOSHUA, & CORNE, DAVID. 1999. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1. IEEE.
- LOURENÇO, HELENA R, MARTIN, OLIVIER C, & STUTZLE, THOMAS. 2001. Iterated local search. *arXiv preprint math/0102188*.
- MAIA, NILTON ALVES. 2006. *Engenharia de Tráfego em domínio MPLS utilizando técnicas de Inteligência Computacional*. (in Portuguese). Ph.D. thesis, Universidade Federal de Minas Gerais.
- MAIA, NILTON ALVES, DE ERRICO, LUCIANO, & CAMINHAS, WALMIR MATOS. 2007. Combining MPLS, Computational Intelligence, and Autonomic Computing into a Self-Managing Traffic Engineering System. *Proceedings of the Second Latin American Autonomic Computing Symposium LAACS 2007*, 1–6.
- MLADENOV, N., & HANSEN, P. 1997. Variable neighborhood search. *Computers & operations research*, **24**(11), 1097–1100.
- MORAGLIO, ALBERTO, KIM, YONG-HYUK, YOON, YOURIM, & MOON, BYUNG-RO. 2007. Geometric crossovers for multiway graph partitioning. *Evolutionary Computation*, **15**(4), 445–474.
- NOGUEIRA, A., SALVADOR, P., & VALADAS, R. 2006. Detecting Internet Applications using Neural Networks. *Pages 95–95 of: Networking and Services, 2006. ICNS'06. International conference on*.
- ONALI, TATIANA, & ATZORI, LUIGI. 2008. Traffic classification and bandwidth management in diffserv-aware traffic engineering architectures. *Pages 70–74 of: Communications, 2008. ICC'08. IEEE International Conference on*. IEEE.
- ONETY, RENATA E., MOREIRA, GLADSTON J. P., NETO, ORIANE M., & TAKAHASHI, RICARDO H. C. 2011. Variable neighborhood multiobjective genetic algorithm for the optimization of routes on IP networks. *Pages 433–447 of: Proceedings of the 6th International Conference on Evolutionary Multicriterion Optimization*. (LNCS 6576), vol. 1. Berlin: Springer.

- ONETY, RENATA E, TADEI, ROBERTO, NETO, ORIANE M, & TAKAHASHI, RICARDO H. C. 2013. Multiobjective Optimization of MPLS-IP Networks with a Variable Neighborhood Genetic Algorithm. *Applied Soft Computing*.
- OSMAN, MS, ABO-SINNA, MA, & MOUSA, AA. 2005. An effective genetic algorithm approach to multiobjective routing problems (MORPs). *Applied mathematics and computation*, **163**(2), 769–781.
- PAQUETE, LUÍS. 2010. Algoritmos de Pesquisa Local Iterativa e Pesquisa de Vizinhança Variável. (in Portuguese). Pages 125–136 of: *Manual de Computação Evolutiva e Metaheurística*.
- PARREIRAS, R.O., & VASCONCELOS, J.A. 2007. A multiplicative version of promethee ii applied to multiobjective optimization problems. *European Journal of Operational Research*, **183**, 729–740.
- PARREIRAS, ROBERTA OLIVEIRA. 2006. *Algoritmos Evolucionários e Tecnicas de Tomada de Decisão em Análise Multicritério*. (in Portuguese). Ph.D. thesis, Universidade Federal de Minas Gerais.
- PAUL, P., & RAGHAVAN, SV. 2002. Survey of QoS routing. Page 50 of: *Proceedings of the international conference on Computer communication*, vol. 15. Citeseer.
- PERBOLI, G., PEZZELLA, F., & TADEI, R. 2008. EVE-OPT: a hybrid algorithm for the capacitated vehicle routing problem. *Mathematical Methods of Operations Research*, **68**(2), 361–382.
- SANTOS, FERNANDO AFONSO. 2009 (March). *Otimização Multiobjetivo aplicada a alocação dinâmica de rotas em redes de telecomunicações*. (in Portuguese). M.Phil. thesis, Universidade Federal de Minas Gerais.
- SOUSA, LMC, VIEIRA, FHT, LEE, L.L., & DECOM, C. 2006. A fuzzy approach for adaptive control of MPLS network traffic flows. Pages 29–34 of: *Telecommunications Symposium, 2006 International*.
- SRINIVAS, NIDAMARTHI, & DEB, KALYANMOY. 1994. Muultiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, **2**(3), 221–248.
- TAKAHASHI, RICARDO H. C. 2007. *Otimização escalar e vetorial*. (in Portuguese). Tech. rept. Volume 3: Otimização vetorial. Universidade Federal de Minas Gerais.
- TANENBAUM, ANDREW S. 1984. *Structured computer organization*. Prentice Hall PTR.

- TANENBAUM, ANDREW S. 1989. *Computer networks*. Vol. 4. Prentice-Hall Englewood Cliffs (NY).
- WANG, Z., & CROWCROFT, J. 1996. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected areas in communications*, **14**(7), 1228–1234.
- ZITZLER, E., LAUMANN, M., THIELE, L., *et al.* . 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. *Pages 95–100 of: EUROGEN*.
- ZITZLER, ECKART. 1999. *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Ph.D. thesis, Swiss Federal Institute of Technology (ETH), Zurich.
- ZITZLER, ECKART, & THIELE, LOTHAR. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, **3**(4), 257–271.