

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]



Université d'Ottawa • University of Ottawa

Broadband Networks Design Models and Admission Control Algorithms

By

 **Miguel Ríos**

A thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Ottawa-Carleton Institute of Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa

Ottawa, Ontario, Canada

January 2000



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-48114-X

Canada

ABSTRACT

The development of efficient provisioning methods is one of the remaining challenges for the successful deployment of telecommunication networks. In this thesis, two of the many problems involved in telecommunication networks design are addressed: the **connection admission control problem (CAC)**, which refers to the set of actions, taken at connection time, to decide whether to accept or reject the request for a new call/connection. The second is the **network-wide link sizing problem**, which selects the topology of the (ideally) optimal network interconnecting the nodes, based on traffic characterization, nodal layout, routing and admission control procedures. These two problems are related since the CAC policy, together with the traffic characterization, nodal layout and routing, determines the internal traffic of the network which is used in the link sizing problem.

Two location models for the design of telecommunications networks, with constrained buffer length at switches, are introduced and discussed, which find optimal locations of switches on a network, and allocate users to them so as to minimize overall cost. Also, for a given buffer length, the models keep the desired cell-loss probability P smaller than or equal to α .

Four fault tolerant telecommunication networks design models are discussed. Relaxation techniques and heuristics are used to solve the models. The fundamental network design problem that is addressed consists in determining, given the location of the communication nodes, the traffic demands among pairs of origin-destination nodes, fault tolerance restrictions and the cost structure for the available technologies, the capacity to be assigned to each link, in such a way to satisfy the restriction while reaching a minimum cost network.

The thesis also focuses on the use of genetic algorithms in the design of telecommunications networks satisfying bi-connectivity and delay constraints. The design of a bi-connected telecommunication network, consists in the interconnection of a set of N nodes, in such a way there exist at least two alternative paths between any given pair of nodes and the cost of the network (considering routing, capacities and delay restrictions) is minimized.

An admission control and a traffic shaping method for BISDN/ATM networks are proposed. The method operates at the burst connection level based on the Sustainable data-transfer rate concept. The model treats all cells equally regardless of their associated service requirements.

ACKNOWLEDGEMENTS

This work started way back in August 1985. It is then understandable that many people deserve credit for helping me finishing this thesis just short of the next century. First of all I want to thank my supervisor and friend, Dr. Nicolas D. Georganas, for his everlasting patience, support and friendship.

I also have to thank my colleague and friend, Dr. Vladimir Marianov, who willingly accepted to act as an informal supervisor there in Chile where most of the research was done. Vladimir is responsible for directing me into the field of resource location analysis.

My special appreciation to the people at the Graduate School of the University of Ottawa, for their patience in allowing this soul to pursue this very unorthodox Ph.D. program. My deepest gratitude to the School of Engineering of the Universidad Católica de Chile, for supporting me these fourteen years. My special thanks to my former deans Dr. Alejandro Mackay, and Dr. Juan Enrique Coeymans, and to my present dean, Dr. Aldo Cipriano, for their continuing encouragement

On the personal side, for sure my family deserves very high credits. My cherished late wife Cristina who was my inspiration. My children Paola, Miguel Andres and Roxanna by understanding the long separation from them, because of the program. The rest of my family also deserves my thanks. They helped me by taking care of my children in those days when I was in Canada.

Special thanks to Dr. Georganas family: Jacynthe, Nikita and Emmanuel, who received me in Canada at their home as another member of the family: you made me feel at home and helped me cope with the absence of my children. I have also to thank my friend in Canada, Dr. Carlos Patrickson and family, who also always were beside me without asking anything in return.

Finally my special gratitude to Isabel, who pushed me into finishing this, up to the point where I seriously considered breaking our relationship. Now we will celebrate this accomplishment together.

TABLE OF CONTENTS

ABSTRACT	II
ACKNOWLEDGEMENTS	III
TABLE OF CONTENTS	IV
LIST OF TABLES	VII
LIST OF ILLUSTRATIONS	VIII
LIST OF ACRONYMS	X
CHAPTER 1: INTRODUCTION	1
1.1. NETWORK DESIGN METHODS	1
1.1.1. <i>Manual design</i>	2
1.1.2. <i>Heuristics</i>	2
1.1.3. <i>Formal optimization techniques</i>	2
1.1.4. <i>Bounds and relaxations</i>	4
1.2. DESIGN OF SURVIVABLE NETWORKS	5
1.2.1 <i>Introduction and Survey</i>	5
1.2.2. <i>Capacities</i>	6
1.2.3. <i>Survivability</i>	6
1.2.4. <i>Path length</i>	7
1.2.5. <i>Network Dimensioning Models</i>	7
1.2.5.1 No capacities and k-connectivity	7
1.2.5.1. <i>Continuous capacities and reservation</i>	7
1.2.5.3. <i>Discrete capacities and no survivability</i>	8
1.2.5.4. <i>Discrete capacities and diversification/reservation</i>	8
1.3. THE CONNECTION ADMISSION CONTROL PROBLEM	8
1.3.1. <i>Resource management in ATM networks</i>	9
1.3.2. <i>Quality of Service (QoS) Attributes</i>	10
1.3.3. <i>Service Classes</i>	11
1.3.4. <i>Other techniques and congestion control</i>	12
1.3.5. <i>Statistical multiplexing gain</i>	13
1.4. ORGANIZATION OF THE THESIS	14
1.5. THESIS CONTRIBUTIONS AND PUBLICATIONS	15
CHAPTER 2: LOCATION MODELS FOR THE DESIGN OF TELECOMMUNICATIONS NETWORKS WITH CONSTRAINED BUFFER LENGTH AT SWITCHES	17
2.1. INTRODUCTION	17
2.2. DEVELOPMENT OF A PROBABILISTIC PLANT LOCATION MODEL	20
2.3. DEVELOPMENT OF THE <i>P</i> -MEDIAN MODEL FOR CONGESTED SYSTEMS	23
2.4. COMPUTATIONAL EXPERIMENTS.....	24
2.5. CONCLUSIONS	28
CHAPTER 3: FAULT TOLERANT TELECOMMUNICATION NETWORKS DESIGN MODELS	29
3.1. INTRODUCTION	29
3.2. UNCAPACITATED ASSIGNMENT MODELS	30
3.2.1. <i>Network Modeling</i>	30
3.2.1.1. <i>Network Connectivity</i>	30
3.2.1.2. <i>Connectivity Levels</i>	30

3.2.1.3. Articulation Node.....	31
3.2.1.4. Cycles.....	31
3.2.1.5. How to prove network biconnectivity.....	32
3.2.1.6. Fault Tolerant Degree.....	32
3.2.2. <i>Survivable Uncapacitated Network Design Problem (SUND):</i>	33
3.2.2.1. First Model: SUND1.....	33
3.2.2.2. Second Model: SUND2.....	35
3.2.2.3. Model Complexity.....	37
3.2.2.4. Comparison between SUND1 and SUND2 Models.....	38
3.2.2.5. Heuristics to Solve SUND.....	39
3.2.2.6. Characterization of solutions networks for SUND.....	39
3.3. LINK CAPACITY ASSIGNMENT MODELS.....	41
3.3.1. <i>Survivable Capacitated Network Design SUCAN: Finite Link Capacity Assignment Fault Tolerant Network Design.</i>	41
3.3.2. <i>Strategies used in fault tolerant networks</i>	41
3.3.3. <i>Problem Characterization</i>	43
3.3.4. <i>Costs Structure</i>	44
3.3.5. <i>Previous work</i>	44
3.3.6. <i>Mathematical Models: SUCAN</i>	45
3.3.6.1. The SUCAN model.....	45
3.3.6.2. SUCAN2 Model.....	48
3.3.7. <i>Model Comparison</i>	50
3.4. METHODS FOR SOLVING THE FINITE ARC CAPACITY FAULT TOLERANT NETWORK DESIGN PROBLEM.....	51
3.4.1. <i>Lagrangean Relaxations</i>	51
3.4.1.1. Relaxation Strategies.....	52
3.4.1.2 Separation of the Relaxation problems.....	54
3.4.1.3. Solution to the relaxation problems.....	56
3.4.1.3.1. Solution to problems [PLI].....	56
3.4.1.3.2. Solution to subproblems [PLII].....	58
3.4.1.4. Lagrangean Relaxation Algorithm.....	59
3.4.1.5. Multiplier Adjusting Method.....	61
3.4.1.6. Heuristics for obtaining feasible solutions to the original problem from the Lagrangean Relaxation.....	62
3.4.1.7. Relative Positions between Different Values.....	62
3.4.1.8. Model Parameters.....	63
3.4.2. <i>Heuristic Methods</i>	65
3.4.2.1. Heuristics for searching feasible initial networks.....	65
3.4.2.2. Improvement Heuristics.....	69
3.5. COMPUTATIONAL EXPERIMENTS.....	70
3.5.1. <i>Generation of GEN networks</i>	71
3.5.2. <i>Generation of JER networks</i>	71
3.5.3. <i>Available capacity ranges</i>	73
3.5.4. <i>Lagrangean relaxations using different cost functions</i>	73
3.5.5. <i>Preliminary results</i>	75
3.5.6. <i>General Results</i>	76
3.5.7. <i>Problems with the Lagrangean Relaxation Method</i>	80
3.5.8. <i>Examples</i>	80
3.6 CONCLUSIONS.....	85

CHAPTER FOUR: USE OF GENETIC ALGORITHMS IN THE DESIGN OF TELECOMMUNICATIONS NETWORKS.....87

4.1. INTRODUCTION.....	87
4.2. PROBLEM FORMULATION.....	87
4.3. COMPLEXITY OF THE PROBLEM.....	89
4.4. GENETIC ALGORITHMS.....	90
4.4.1. <i>Genetic Algorithm Definition</i>	91
4.4.2. <i>Pseudo Code and Genetic Algorithm Components</i>	93

4.5. THEORETICAL ANALYSIS OF THE GENETIC ALGORITHMS.....	95
4.5.1. <i>Basic concepts</i>	95
4.5.2. <i>Reproductive growing equation of a scheme</i>	96
4.6. GENETIC ALGORITHM USED.....	97
4.6.1. <i>Initial population</i>	97
4.6.2. <i>Genetic Algorithm Implementation for Finding Good Solutions of the TSP</i>	98
4.6.3. <i>Ending criteria</i>	101
4.6.4. <i>Bi-connectivity checking</i>	103
4.6.5. <i>Routing</i>	103
4.6.6. <i>Capacity assignment</i>	103
4.6.7. <i>Average delay</i>	104
4.6.8. <i>Fitness function</i>	105
4.6.9. <i>Selection</i>	106
4.6.10. <i>Crossover or Recombination</i>	107
4.6.11. <i>Mutation</i>	107
4.6.12. <i>Penalizing Functions</i>	108
4.7. COMPUTATIONAL RESULTS.....	108
4.7.1. <i>Branch Exchange Algorithm</i>	109
4.7.2. <i>Exhaustive Search Method</i>	110
4.7.3. <i>Results for 10 node networks</i>	110
4.7.4. <i>Results for a 42 node network</i>	115
4.8. CONCLUSIONS.....	117
CHAPTER 5. BURST ADMISSION CONTROL AND INPUT TRAFFIC SHAPING FOR ATM NETWORKS.....	118
5.1. INTRODUCTION.....	118
5.2. MODEL DESCRIPTION.....	121
5.2.1 <i>Bit Rate Reduction Strategy</i>	123
5.3. ANALYSIS.....	124
5.3.1 <i>Resource sharing policy</i>	126
5.3.2 <i>Shaper Model</i>	127
5.4. SIMULATION RESULTS AND DISCUSSIONS.....	129
5.5. CONCLUSIONS.....	134
CHAPTER 6. CONCLUSIONS.....	136
REFERENCES.....	138

LIST OF TABLES

Table 2.1:	Test network.....	25
Table 2.2:	Maximum traffic arriving to each switch (in cells per second) for different values of α and b.....	26
Table 2.3:	Results for cell-loss probability of 10^{-3}.....	26
Table 2.4:	Results for cell-loss probability of 10^{-4}.....	27
Table 2.5:	Results for cell-loss probability of 10^{-5}.....	27
Table 2.6:	Results for cell-loss probability of 10^{-6}.....	27
Table 3.1:	Statistics in Model Solving.....	38
Table 3.2:	Comparison of Capacitated Models.....	51
Table 3.3:	Characteristics of GEN networks.....	71
Table 3.4:	Characteristics of the JER networks.....	72
Table 3.5:	Available ranges for the capacity assignment.....	73
Table 3.6:	Results for different relaxations.....	74
Table 3.7:	Costs for different network generation heuristics.....	75
Table 3.8:	Percentage decreasing of Total Cost by using different improvement heuristics.....	76
Table 3.9:	Results for different improvement methods (GEN networks).....	77
Table 3.10:	Results for different improvement methods (JER Networks).....	78
Table 4.1.	Solution space for different network sizes.....	90
Table 4.2.	Example of fitness calculation.....	106
Table 4.3.	Solution for 10 node networks for different traffic.....	112
Table 4.4.	10 node solutions with different locations.....	113
Table 4.5.	Capacity allocation for 42 node network.....	116

LIST OF ILLUSTRATIONS

Figure 2.1.	State transition diagram.....	22
Figure 2.2.	30-node network.....	24
Figure 2.3.	Locations of 4 switches, cell-loss probability of 10^{-6} and buffer length 100.....	28
Figure 3.1.	Arc or node disjoint paths.....	31
Figure 3.2.	A Cycle is a node biconnected network.....	32
Figure 3.3.	Five-node network.....	36
Figure 3.4.	Typical topology of a SUCAN network.....	40
Figure 3.5.	Two strategies for fault tolerant networks.....	42
Figure 3.6.	Arc v/s Capacity Costs.....	44
Figure 3.7.	Arc costs vs. arc length.....	45
Figure 3.8.	Cost functions.....	53
Figure 3.9.	Separation of the Original Problem.....	56
Figure 3.10.	Flow diagram for the general relaxation algorithm.....	60
Figure 3.11.	Relative position of different bounds of Lagrangean relaxation.....	62
Figure 3.12.	Link cost v/s total link flow.....	64
Figure 3.13.	Link cost v/s link length.....	65
Figure 3.14.	Typical topology for a hierarchical network.....	72
Figure 3.15.	Total cost for different solution method for GEN30.....	79
Figure 3.16.	GAP between BD and BLB for different number of special pairs....	80
Figure 3.17.	Optimum solution for a 10 node GEN network example.....	81
Figure 3.18.	Initial relaxation.....	81
Figure 3.19.	Improvement: first phase.....	82
Figure 3.20.	Improvement: second phase.....	82
Figure 3.21.	Initial solution for a 15 node GEN network example.....	83
Figure 3.22.	Improved solution.....	83
Figure 3.23.	Initial solution for a 20 node GEN network example.....	84
Figure 3.24.	Initial solution for a 20 node GEN network example.....	84
Figure 3.25.	Initial solution for a 15 node JER network example.....	85

Figure 3.26.	Improved solution.....	85
Figure 4.1.	Individual coding in chromosomes.....	91
Figure 4.2.	Crossover with gene interchange.....	92
Figure 4.3.	Individual mutation.....	93
Figure 4.4.	Genetic Algorithm Model.....	94
Figure 4.5.	Flow diagram of the network design algorithm.....	99
Figure 4.6.	Example of two infeasible chromosomes that recombine into a feasible one.....	100
Figure 4.7.	Flow diagram of the genetic algorithm.....	102
Figure 4.8.	Terminal equipment cost in terms of the capacity.....	105
Figure 4.9.	Single-point crossover.....	107
Figure 4.10.	Branch exchange example.....	110
Figure 4.11.	Solutions results when varying the traffic matrix.....	112
Figure 4.12.	Solution networks for <i>mtraffic1</i>	113
Figure 4.13.	Solution costs in terms of different locations.....	114
Figure 4.14.	Solution networks for <i>location3</i>	114
Figure 4.15.	42 node network solution.....	115
Figure 5.1.	Traffic shaping function environment.....	120
Figure 5.2.	Communication with Go-and-Store protocol.....	122
Figure 5.3.	Block diagram of the proposed model.....	123
Figure 5.4.	Bit Rate Reduction.....	124
Figure 5.5.	Shaper Model.....	128
Figure 5.6.	Voice transmission based on the sustainable data-transfer rate.....	130
Figure 5.7.	Back-off time Vs. Blocking probability.....	131
Figure 5.8.	Comparison of the three schemes.....	132
Figure 5.9.	Data blocking probability.....	133
Figure 5.10	Packetized voice blocking probability.....	134

LIST OF ACRONYMS

ABR	Available Bit Rate
ATM	Asynchronous Transfer Mode
BD	Best Discrete Bound
B-ISDN	Broadband Integrated Services Digital Network
BLB	Best Lower Bound
BT	Burst Tolerance
BUB	Best Upper Bound
CAC	Connection Admission Control
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CS	Complete Sharing
CTD	Cell Transfer Delay
EA	Arc Elimination
ES	Sequential Routing
GARANDOM	Design using Random Solutions through Genetic Algorithms
GATSP	Design using TSP Solutions through Genetic Algorithms
GCRA	Generic Cell Rate Algorithm
GEN	General Network
JER	Hierarchical Network
LAN	Local Area Network
LSCP	Location Set Covering Problem
MBS	Maximum Burst Size
MCLP	Maximal Covering Location Problem
MCR	Minimum Cell Rate
MS	Sequential Improvement
NMR	Network Resource Management
NNI	Network to Network Interface
NP	Non Polynomial Problem
NPC	Network Parameter Control
OD	Origin Destination
PCR	Peak Cell Rate
QoS	Quality of Service
RM	Minimal Routes
SCR	Sustained Cell Rate
SDH	Synchronous Digital Hierarchy
SMG	Statistical Multiplexing Gain
SQR	Square Root Rule
SUCAN	Survivable Capacitated Network Design Problem
SUND	Survivable Uncapacitated Network Design Problem
TASI	Time Assigned Speech Interpolation
TSP	Traveling Sales Problem

UB	Upper Bound
UBR	Unspecified Bit Rate
UNI	User to Network Interface
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC	Virtual Channel
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VP	Virtual Path
VPC	Virtual Path Connection
VPI	Virtual Path Identifier

CHAPTER 1: INTRODUCTION

The development of efficient provisioning methods is one of the remaining challenges for the successful deployment of telecommunication networks. In order to allocate network resources cost effectively, the network operators require provisioning tools that match expected traffic demands with the appropriate transmission and switching equipment.

In this thesis, two of the many problems involved in telecommunication networks design are addressed. The first is the **connection admission control (CAC)**, which refers to the set of actions, taken at connection time, to decide whether to accept or reject the request for a new call/connection. The decision is based on the connection's traffic characteristics, the required QoS and the network load. If the request is accepted, the necessary resources are allocated to the connection. The second problem is the **network-wide link sizing problem**, which selects the topology of the (ideally) optimal network interconnecting the nodes, based on traffic characterization, nodal layout, routing and admission control procedures. This problem is difficult because of the nonlinear relation between the statistical multiplexing gain and the bandwidth necessary to transport the traffic. The direct application of the conventional dimensioning methods available for circuit switching networks to an ATM network is therefore inappropriate.

These two problems are related since the CAC policy, together with the traffic characterization, nodal layout and routing, determines the internal traffic of the network which is used in the link sizing problem.

1.1. Network Design methods

In the designing process, it is important to distinguish between techniques used to design a network "from scratch" from those used for incremental design. Actually, most real network designs are incremental beginning with an existing network. Unfortunately, most of what is known and can be proven relates to techniques that design an entire network from a given set of requirements.

1.1.1. Manual design

A surprisingly large number of networks are still designed by hand calculations using rules of thumb, or even no "rules" at all. The attractive aspect of this approach is its flexibility: Unusual constraints and objectives can be taken into account. The designer can be responsive to changes in goals and requirements. Also, incremental as well as total designs can be done. This approach has several disadvantages, however. It is rarely quantitative. The designer makes decisions subjectively and inconsistently. It is difficult to repeat a successful design when similar circumstances arise and difficult to learn from previous mistakes.

1.1.2. Heuristics

Heuristics are design principles incorporated into algorithms. Like rules of thumb, they are good ideas embodying design experience. They, however, can be made quantitative and reproducible. Alternative heuristics can be compared by implementing them on the same problem, then observing which gives rise to the best solution. One of the most widely used heuristics is the **greedy algorithm**. Confronted by a series of choices, the greedy algorithm chooses the best one it can at each stage. The greedy algorithm is a broadly applicable heuristic based on the simple observation that inexpensive networks tend to contain inexpensive links. This algorithm can yield good networks. However, it will not yield the optimal solution. Heuristics are valuable because they allow the designer to obtain feasible solutions to difficult problems in a reasonable amount of time. Instead of an exhaustive exploration of the entire solution space, consideration is limited to solutions with characteristics that appear to be good. In this thesis several heuristics will be used to solve some of the problems addressed. In particular **genetic algorithms** will be used in the design of bi-connected networks.

1.1.3. Formal optimization techniques

Except for the smallest problems, it is not possible to enumerate all possible solutions and then choose the best one. A network with ten nodes, for example, has 45 potential links, each of which could be either included or excluded from the network. Even if it is assumed there is only one possible link speed, this gives rise to 2^{45} possible solutions, more than 10^{13} possibilities. The set of all possible solutions to a problem is referred to as the **solution space**. The notion of best is

expressed in formal optimization techniques by the **objective function**. The objective function associates a value with the design variables. Thus, for example, each link i can be included or excluded from a design. There is a cost associated with each link and to minimize cost, use the sum of the costs of the links chosen as the objective function.

While cost is the most commonly used objective function, it is not the only one. It may be wanted to maximize reliability. In this case, associate a reliability parameter with each possible node and link in the network, and then compute the reliability of candidate networks comprised of specific nodes and links. Unlike the case of cost, however, this may be a complex calculation in its own right. The best value of the objective function is referred to as the **optimum**.

There are some algorithms, which always produce optimal solutions. The problem with these, however, is that they only apply to a limited class of problems; for problems outside such class, they do not work at all. One such algorithm is the **simplex method**. This algorithm only works for the class of problems called **linear programming problems**.

Often, a problem can be phrased as a linear programming problem with the additional constraint that the variables must be integers. This situation arises frequently in network design; that is, when the decision is whether or not to include a link in the solution. The choice is to include the link ($x_i = 1$) or not to include it ($x_i = 0$). These **integer-programming problems** are much more difficult to solve than linear programming problems.

Another important class of problems arises when the solution space and the objective are **convex**. In a convex space, movement can be made from one feasible solution to another in small steps, without leaving the feasible region. This allows the use of simple, incremental methods to search for the optimal solution. A convex function is a function with the property that for any point c , between a and b , $f(c)$ lies below the line connecting $f(a)$ and $f(b)$. Convex functions have the property that their minimum can be found via a local search. Specifically, when a minimum is sought, the minimum can be found by starting at any point and moving in a direction where the function decreases (**descent methods**). This is much simpler than having to search through all possible values in the solution space. Descent methods can also be used on nonconvex functions but, in general, they converge to a **local minimum** (the bottom of a valley) instead of a **global minimum** (the bottom of the deepest valley.)

Note that linear functions are both convex and concave and thus their minima and maxima can be found relatively easily. This is part of the reason why linear programming problems are solvable in a reasonable amount of time. Integer programming problems, on the other hand, have solution spaces that are not convex but are isolated points at integer coordinates. None of the points on the line connecting two adjacent integers is in the solution space.

1.1.4. Bounds and relaxations

Sometimes a problem, P , is difficult to solve but is closely related to another problem, P' , which is easier to solve. P' may be derived from P with some constraints removed. Another possibility is that P' might have the same constraints as P but its objective function might be better behaved, perhaps convex. P' can then be solved in the hope of obtaining a solution, or a close approximation to a solution, to P .

Suppose P' is formed by substituting a well-behaved objective function, $g(x)$, for the original objective function, $f(x)$, from P . If $g(x)$ has the property $g(x) \leq f(x) \forall x$, then it is said that $g(x)$ is a lower **bound** for $f(x)$.

If P' is formed from P by dropping constraints and/or by substituting a lower bound for the objective function of P , then it is said that P' is a **relaxation** of P . Such P' has the useful property that the optimal solution to P' is no greater than the optimal solution to P . This can be seen by considering the value of x which is the optimal solution to P . x is also a solution to P' since x satisfies all the constraints of P' (which are a subset of the constraints of P). Furthermore, the value of $g(x)$ is no greater than $f(x)$, since $g(x)$ is a lower bound for $f(x)$.

Therefore, it is possible to solve P with a heuristic, that solves P' exactly (obtaining the optimal solution) and then compares the values of the two solutions obtained. If they are sufficiently close, say within an acceptable proximity of one another, no further search is necessary since at worst, the lower bound is within a specified deviation from the optimum. Note that even if the two solutions are far apart, it does not necessarily mean that the heuristic solution is far from the optimum; the lower bound may be loose. This approach can be used as the basis of a more general technique, known as **branch and bound**. There are also algorithms called **combinatorial algorithms**, which deal directly with discrete choices, typically variables that take the values 0 and

1. This thesis uses relaxation methods to solve some of the problems related with the design of survivable networks.

1.2. Design of survivable networks

Designing low-cost networks that survive certain failure situations is one of the prime tasks in the telecommunication industry. Algorithms integrating polyhedral combinatorics, linear programming, and various heuristic ideas can help solve real-world network dimensioning instances to optimality or within reasonable quality guarantees in acceptable running times.

The problem type addressed is the following. Let a communication demand between each pair of nodes of a telecommunication network be given. Consider the problem of choosing, among a discrete set of possible capacities, which capacity to install on each of the possible edges of the network in order to

- (i) satisfy all demands,
- (ii) minimize the building cost of the network.

In addition to determining the network topology and the edge capacities, it is required to provide for each demand a routing designation such that

- (iii) no path can carry more than a given percentage of the demand,
- (iv) no path in the routing exceeds a given length.

Also make sure that

- (v) for every single node or edge failure, a certain proportion of the demand is reroutable.

Moreover, for all failure situations feasible routings must be computed.

1.2.1 Introduction and Survey

A series of mathematical models have been developed in the recent years to describe and solve various telecommunication network design problems. Along with the solution methodology the users of these models have become more sophisticated, demanding the integration of tasks into one model that have traditionally been solved in a hierarchical fashion. A typical sequence of such decisions consists, among other issues to be considered, of the choice of technologies to be used, the topological design of the network, the planning of the capacities of the network components, a decision about routing strategies, and the treatment of failure situations. The problems considered

have the following in common. The input consists of two graphs on the same node-set V , the supply graph $G=(V; E)$ and the demand graph $H=(V; D)$. The set V consists of the nodes of the logical transport network. The edge-set E of the supply graph G is the set of all physical links that may potentially be used (in the planning period). Different types of links (representing different technologies, e.g., microwave connections, copper or fiber optic cables, leased lines, etc.) are represented by parallel edges. The demand graph H (for the planning period) contains an edge whenever there is a positive demand between its two end nodes. For each edge $uv \in D$ of the demand graph, the value $d_{uv} \in Z^+$ is the communication demand between nodes u and v . There are several levels of possible network design problems.

1.2.2. Capacities

It may be that the network designer is only interested in the topological structure and has decided to determine link capacities in a later stage. It may also be that capacities are no issue since the standard technology supplies enough for the application in question (Uncapacitated problem). If capacity planning is necessary, it may be selected arbitrarily from a certain range, or only finitely many choices may be available (continuous and discrete capacitated problems).

1.2.3. Survivability

It has become common to call a network survivable if it has been designed in such a way that the network is operational even if certain network components fail. A frequently used method to guarantee topological survivability is to require that the supply network to be designed contains, for each pair of nodes, a certain (node-pair dependent) number of node- and/or edge-disjoint paths between these nodes. It is said that k -connectivity is required. If routing is an integral part of the network design problem a reasonable strategy to keep the network "alive" in failure situations is to require, that for every pair of demand nodes, no path of the network carries more than a certain proportion of the total traffic between the two nodes. This concept runs under the name **diversification**. In case a model integrates capacity planning and routing, a natural variation of the k -connectivity concept is to require, that, for every pair of demand nodes and for every failure situation, a certain proportion of the traffic demand of the node pair can be routed. This concept is called **reservation**.

1.2.4. Path length

In case of very tight capacities, some connections, i.e., paths carrying traffic between demand nodes, turn out to be long. For various reasons (e.g., to reduce time delay or computer load) it may be advisable to restrict the length of communication paths. Thus there are models with and without path-length restrictions.

1.2.5. Network Dimensioning Models

Several authors considered various combinations of capacity and survivability models. These are discussed below.

1.2.5.1 No capacities and k-connectivity

The uncapacitated problem with connectivity requirements was one of the first network design problems investigated. It includes the Steiner-tree problem as a special case. Monma and Shallcross, 1989, present a heuristic approach for the 2-connectivity case. Their heuristics produce good solutions for the "LATA networks" of Telecordia in short running times. Theoretical investigations on the structure of optimal solutions can be found in Monma, Munson and Pulleyblank, 1990. Grotschel, Monma and Stoer, 1995, develop a framework (based on branch and cut methods) to solve the LATA networks of Bellcore for low- connectivity ($k < 3$) instances to optimality. Furthermore, Stoer, 1992, reports that special high-connectivity problems ($k > 2$) can be solved to optimality for up to 500 nodes.

1.2.5.1. Continuous capacities and reservation

Minoux, 1981, was apparently the first to consider survivability in a generalized multicommodity-flow model with continuous capacities. He reports that instances with up to 40 nodes can be solved within 5% of the optimum. In cooperation with France Telecom, Lisser, Sarkissian and Vial, 1995, develop another model including non-discrete capacities and survivability. Two survivability models are presented. In both models, part of the demand is routed in a separate network, called spare network, in case of a failure. The local survivability model routes only the failing flow, and the global survivability model routes only the affected demands in the spare network.

1.2.5.3. Discrete capacities and no survivability

Several models in the literature consider the installation of discrete capacities without addressing survivability issues. Moreover, these models restrict the possible capacities to multiples of one or two basic capacities. Bienstock et al, 1995, solve ATM network design problems with real-life data for instances of up to 16 nodes to optimality. Their model includes flow costs, and the capacities can be chosen as combinations of two basic technologies (OC3 and OC12 facilities). In another study with one basic technology Bienstock, Chopra, Gunluk, and Tsai , 1995, solve problems with up to 15 nodes and "Norwegian Problems" with up to 27 nodes almost to optimality. Magnanti, Mirchandani, and Vachani, 1995, investigate the same problem without flow costs and solve randomly-generated instances with up to 15 nodes with gaps about 10%.

1.2.5.4. Discrete capacities and diversification/reservation

Dahl and Stoer, 1994, were the first to consider a discrete capacity structure and survivability issues in the same model (for Norwegian Telecom Research). They also report considerable difficulties with some of the instances. This thesis presents a new formulation of the problem, which is a survivable discrete cost modified version of the one from Amiri and Pirkul, 1997. They present a continuous cost formulation for a multicommodity network flow problem but without survivability requirements.

1.3. The Connection Admission Control Problem

One of the key design objectives of B-ISDN is the provision of a wide range of services to a broad variety of users utilizing a limited set of connection types and multi-purpose user-network interfaces. The two prominent enabling technologies for the deployment of B-ISDN are fiber optics and the asynchronous transfer mode (ATM) network architecture. The ATM network is structured hierarchically in two levels: a) Virtual channel level and b) Virtual path level. The transmission medium (usually an optical fiber) provides a transport service to the virtual path (VP), and the VP provides a transport service to the virtual channel (VC). The VC describes a unidirectional communication capability for the transport of ATM cells, associated by a common unique identifier value (the VCI). A virtual channel link is defined as a mean of unidirectional transport of ATM cells between a point where a VCI value is assigned and the point where that value is translated or removed. A VC link is defined between two consecutive VC switching nodes

or between an ATM terminal and a VC switching node. VC links are concatenated to form a virtual channel connection (VCC) that keeps the order of cells as they are transferred. The VP describes a bundle of VC links that have the same endpoints. A specific VPI value is assigned each time a VP is switched in the network. A virtual path link is terminated by the points where a VPI value is assigned and translated or removed. A concatenation of VP links is called a virtual path connection (VPC). The physical layer offers a service that carries ATM cells belonging to a certain VP and VC. This layer may be viewed as an interface based on the synchronous digital hierarchy (SDH).

1.3.1. Resource management in ATM networks

The flexibility provided by ATM networks requires an adequate resource management to prevent congestion, have a good performance and optimize the network resources use. This requires different traffic controls: **connection admission control (CAC)**, **usage parameter control (UPC)**, **network parameter control (NPC)**, **cell-level quality control** and **congestion control**. The use of virtual paths allows for a simplified traffic control as only the aggregated traffic of an entire VP has to be handled.

CAC is the set of actions, taken at connection time, to decide whether to accept or reject the VCC or VPC requests. The decision is based on the connection's traffic characteristics, the required quality of service and the network load. If the request is accepted, the necessary resources are allocated to the connection.

UPC and **NPC** monitor the traffic at the user to network interface (UNI) and at the network to network interface (NNI) respectively, and adjust the traffic to conform to the connection accepted characteristics. The monitoring functions include the checking of the validity of VPI/VCI values, the checking of the traffic volume entering the network from all active VP and VC connections and the checking of the total volume of the accepted traffic on the access link.

Cell-level quality control refers to the time and space priority assigned to a cell. ATM cells have an explicit cell loss priority bit in the header, so at least two different priority classes can be distinguished.

Congestion control refers to the set of actions to reduce the spread and duration of congestion. It can employ connection admission and or UPC/NPC procedures to avoid overload situations.

1.3.2. Quality of Service (QOS) Attributes

The source traffic characteristics of an ATM connection are described by parameters such as *peak cell rate (PCR)*, *sustained cell rate (SCR)*, *burstiness*, *peak duration* and *source type*. The peak cell rate is specified at the physical layer service access point.

While setting up a connection on ATM networks, users can specify the following parameters related to the desired quality of service at source.

1. Peak Cell Rate (PCR): The maximum instantaneous rate at which the user will transmit. For bursty traffic, the inter-cell interval and the cell rate varies considerably. PCR is defined as the inverse of the minimum inter-cell interval.

2. Sustained Cell Rate (SCR): This is the average rate as measured over a long interval.

3. Cell Loss Ratio (CLR): The percentage of cells that are lost in the network due to error and congestion and are not delivered to the destination. It is the ratio of the number of lost cells to the number of transmitted cells.

Each ATM cell has a Cell Loss Priority (CLP) bit in the header. During congestion, the network first drops cells that have CLP bit set. Since the loss of CLP=0 cell is more harmful to the operation of the application, CLR can be specified separately for cells with CLP=1 and for those with CLP=0.

4. Cell Transfer Delay (CTD): The delay experienced by a cell between network entry and exit points is called the cell transfer delay . It includes propagation delays, queueing delays at various intermediate switches, and service times at queueing points.

5. Cell Delay Variation (CDV): This is a measure of variance of CTD. High variation implies larger buffering for delay sensitive traffic such as voice and video. CDV refers to the fact that the transport of cells does not maintain their intergeneration time mainly because of variation on the waiting times in buffers. This problem has to be considered by the CAC. The problem is that CDV may change the traffic characteristics or the bandwidth required by the connection.

6. Burst Tolerance (BT): This determines the maximum burst size that can be sent at the peak rate. This is the bucket size parameter for the leaky bucket algorithm that is used to control the traffic entering the network. The algorithm consists of putting all arriving cells in a buffer (bucket) which is drained at the sustained cell rate (SCR). The maximum number of back-to-back cells that can be sent at the peak cell rate is called maximum burst size (MBS).

7. Minimum Cell Rate (MCR): This is the minimum rate desired by a user. Only the first six of the above parameters were specified in UNI version 3.0 (ATM-Forum, 1993). MCR has been added after and appeared in the actual version of the traffic management document.

1.3.3. Service Classes

There are four classes of service, that have been proposed by the ATM Forum:

1. Constant Bit Rate (CBR): This class is used for emulating circuit switching. The cell rate is constant. Cell loss ratio is specified for CLP=0 cells and may or may not be specified for CLP=1 cells. Examples of applications that can use CBR are telephone, video conferencing, and television.

2. Variable Bit Rate (VBR): This class allows users to send at a variable rate. Statistical multiplexing is used and so there may be a small non-zero random loss. Depending up on whether or not the application is sensitive to cell delay variation, this class is sub-divided into two categories: Real time VBR and Nonreal time VBR. While cell transfer delay is specified for both categories. CDV is specified only for real-time VBR. An example of real-time VBR is interactive compressed video while that of nonreal time VBR is multimedia e-mail.

3. Available Bit Rate (ABR): This class is designed for normal data traffic such as file transfer and e-mail.

Although the standard does not require the cell transfer delay and cell loss ratio to be guaranteed or minimized, it is desirable for switches to minimize the delay and loss as much as possible. Depending up on the congestion state of the network, the source is required to control its rate. The user is allowed to declare a minimum cell rate, which is guaranteed to the VC by the network. Most VCs are likely to ask for an MCR of zero. Those with higher MCR may be denied connection if sufficient bandwidth is not available.

4. Unspecified Bit Rate (UBR): This class is designed for those data applications that want to use any leftover capacity and are not sensitive to cell loss or delay . Such connections are not rejected on the basis of bandwidth shortage (no connection admission control) and not policed for their usage behavior. During congestion, the cells are lost but the sources are not expected to reduce their cell rate. In stead, these applications may have their own higher-level cell loss recovery and retransmission mechanisms. Examples of applications that can use this service are e-mail, file transfer, news feed, etc.

Of course, these same applications can use the ABR service, if desired. Note that only ABR traffic responds to congestion feedback from the network.

1.3.4. Other techniques and congestion control

One other used technique in resource management is **traffic shaping**, where the traffic characteristics of a stream of cells on a VPC or VCC is actively altered in order to reduce the PCR, limit the burst length or reduce the CDV by suitable spacing of cells in time. Another technique is **fast resource management** that enables the network to immediately allocate capacity, such as bit rate or buffer space, to individual burst-type connections for the duration of a cell burst.

The CAC decision is based on the anticipated traffic characteristics of the connection, the QOS requirements and the current network load. The traffic characteristics are estimated from the traffic descriptor values and CDV values. The network load is estimated from the traffic descriptor values of the existing connections and a measurement of traffic.

The choice of traffic control algorithms directly affects the resource allocation strategy. If the PCR is considered for CAC and UPC, then the PCR should be allocated to the connection. However if the average bit rate is significantly lower than the peak rate then the efficiency of the network will suffer. Nevertheless a simple strategy can assist in quickly introducing ATM networks, especially if the knowledge about the traffic flows is limited. The objective of traffic control is to simultaneously achieve a good ATM network efficiency and meet the users' quality of service (QOS) requirements with a method that is generally applicable.

Different UPC mechanisms have been proposed, such as leaky bucket, sliding window, jumping window and exponentially weighted moving average. Leaky bucket and exponentially

weighted moving average seem to be the most promising with respect to flexibility and implementation complexity.

1.3.5. Statistical multiplexing gain

In ATM, the user will generate the necessary cells for the transmission of its data. When the channel resources are shared among independent users, the peak values of the user data are not likely to occur simultaneously and thus the network can work with reduced resources or accommodate more users with the same resources, and this is called *statistical multiplexing gain (SMG)*. Note that it is possible that at certain times the capacity of the channel be overloaded, and so cells can be delayed (by waiting in buffers) or lost (either because the buffer is full or non-existent or the delay is higher than the maximum allowable for the connection). Then in order to use this capability, it is necessary to model the incoming traffic, usually determining a measure that the connection will obey and that serves to calculate the performance measures of the system.

The SMG is also applicable to the call-level, especially when the sources can be modeled as ON-OFF sources.

A key performance measure in ATM networks is then the probability of losing information. The major approaches to determine the cell loss probability (CLP) of the system include the following:

a) The statistical multiplexer is treated as a discrete-time system modeled as a two-dimensional discrete-time Markov chain. The CLP is then expressed in terms of the Markov chain's limiting probabilities, which can be calculated by numerical methods. This approach yields an accurate estimate for the CLP but is computationally burdensome except for very simple problems.

b) The arrival process is approximated by a two-state Markov modulated Poisson process (MMPP) and the system solved as a MMPP/D/1/K queueing system, with the CLP expressed in terms of the limiting probabilities. This is computationally more efficient but the accuracy may not be reliable and depends on the parameter estimation of the two-state MMPP.

c) The arrival process is approximated by a stochastic fluid flow. The CLP can be approximated by the limiting probability of the buffer exceeding the buffer size. This is sufficiently

accurate for many cases but has the potential of breaking down for large problems due to numerical instability.

1.4. Organization of the thesis

The organization of this thesis is as follows: In Chapter 2 location models for the design of telecommunications networks, with constrained buffer length at switches, are introduced and discussed. Numerical examples are described. In ATM networks, when buffers at the switches are too long, and delay upperbound techniques are not being applied to delay-sensitive traffic, Cell Delay Variations occur that may seriously impair the communication. The design must be done in such a way that the cells do not form a long queue at each switch, reducing significantly the CDV. Two models are proposed, that find optimal locations of switches on a network, and allocations of users to them so as to minimize overall cost. Also, for a given buffer length, the models keep the desired cell-loss probability P smaller than or equal to α , the QoS objective.

In Chapter 3 several fault tolerant telecommunication networks design models are discussed. Relaxation techniques and heuristics are used to solve the models. Computer results are included. The fundamental network design problem that is addressed consists in determining the minimum cost network satisfying fault tolerance restrictions for a given topology, traffic demands and cost structure. This means to determine which nodes are to be connected (topological design), and with which bandwidth (capacity assignment), besides an efficient routing strategy which has to accommodate the traffic demands. The main objective is the unified formulation of mathematical models integrating the **Topology Design, Routing and Capacity Assignment** problems.

Chapter 4 focuses on the use of genetic algorithms in the design of telecommunications networks satisfying bi-connectivity and delay constraints. Numerical results are included. The design of a bi-connected telecommunication network, consists in the interconnection of a set of N nodes, so that there exist at least two alternative paths between any given two nodes and the cost of the network (considering routing, capacities and delay restrictions) is minimized. The complexity of this problem makes it very difficult to find the optimum solution even for networks with a few nodes. The Genetic Algorithms are used to solve this network design problem for fairly large networks (over 40 nodes).

Chapter 5 introduces a burst admission control and input traffic shaping for ATM networks. Computer simulation results are discussed. An admission control and a traffic shaping

method for BISDN/ATM networks are proposed. The method operates at the burst connection level based on the Sustainable Cell Rate concept. The model treats all cells equally regardless of their associated service requirements. This is done by operating the links for the most stringent delay and cell loss probability requirements of all the services. An adequate buffer dimensioning controls the cell delay. A new virtual path is established for every burst to be transmitted, causing the allocated resources to be released after the burst transmission, resulting in a higher network performance.

Finally, Chapter 6 presents the conclusions of this work.

1.5. Thesis contributions and publications

The main contributions of this thesis are:

- a) Formulation of two mathematical models for the design of telecommunications networks, where ATM sources are assigned to switches while maintaining controlled values for the cell-loss probability and the cell delay variation. The mathematical models choose optimum locations for the switches in such a way that the cell-loss probability is kept constrained below a maximum value, while maintaining a short length buffer to reach the cell delay variation objective. The models presented are based in mathematical models first published in Marianov; Ríos and ReVelle, 1990, then refined in Marianov; Pérez and Ríos, 1995 and finally presented in Marianov and Ríos, 1996 and in Ríos and Marianov, 1999.
- b) Formulation of four mathematical models for the design of survivable capacitated and un-capacitated networks, where given a supply network with point-to-point traffic demands, specified survivability requirements and capacity ranges, the models find the minimum cost solutions, such that the demands can be met even after a network component failure. Two efficient solution procedures are developed: one based on Lagrangean Relaxations and the other based on Sequential Heuristics. Results have been presented in Ríos; Gutierrez and Marianov, 1999.
- c) Formulation of a mathematical model for the design of a bi-connected telecommunication network (considering routing, capacities and delay restrictions), which is solved by using Genetic Algorithms. The GATSP heuristic provided results that compare favorably with those obtained through the use of the Branch Exchange heuristic, and also with the GARANDOM heuristic. In this sense, the use of initial solutions based in Traveling Salesman Problem (TSP) solutions, proved to perform better, both in cost and in convergence time than the

use of random initial solutions. The use of genetic algorithm to find fast and good solutions to the TSP problem is a key issue in making this method perform well. Results presented have been published in Alvarez; Ríos and Marianov, 1996, Alvarez and Ríos, 1997 and Ríos and Marianov, 1999.

d) Proposal of an admission control and a traffic shaping method for BISDN/ATM networks. The method operates at the burst connection level and is based on the Sustainable Cell Rate concept. The model complies with the Universal Performance Objective, by treating all cells equally regardless of their associated service requirements. This is done by operating the links for the most stringent delay and cell loss probability requirements of all the services. A new virtual path is established for every burst to be transmitted, causing the allocated resources to be released after the burst transmission, resulting in a higher network performance. The shaping or policing function is based on a set of traffic descriptors that the user specifies at the call establishment phase. Results presented have been published in Ríos and Cárdenas, 1995, and in Ríos, Marianov and Cárdenas, 1996.

CHAPTER 2: LOCATION MODELS FOR THE DESIGN OF TELECOMMUNICATIONS NETWORKS WITH CONSTRAINED BUFFER LENGTH AT SWITCHES

2.1. Introduction

In our time, telecommunication technology is playing a more important role than ever before, as demand for it grows and becomes diverse. Future telecommunication systems, in addition to currently available services, may have to provide for:

- a) Broadband telecommunication services, such as TV conference and visual information retrieval with transmission rates reaching 100 Mbps or more.
- b) Multimedia telecommunication services, such as motion pictures combined with high fidelity sound.
- c) Economical implementation for a diversity of services, that can be conveyed in a single integrated network with a unified interface.

The need for the deployment of a Broadband Integrated Services Digital Network (B-ISDN) has been widely recognized for years. At the same time, it has been known that the existing transfer modes (circuit switched and packet switched networks, because of their bandwidth or protocols used) are not suitable for the B-ISDN. Recent years have seen an increasingly growing work on the establishment of the B-ISDN based on the Asynchronous Transfer Mode (ATM) technology. The ATM has the following features:

- a) Information is sent in short fixed-length packets (53 bytes) called cells. The flexibility to support several variable transmission rates is provided by transmitting the necessary number of cells per unit time.
- b) The cell header (5 bytes) contains virtual channel/path identifiers (VCI/VPI), that facilitate switching and routing.
- c) Information transfer in an ATM network uses end-to-end flow control and error recovery, thus avoiding the complex procedures of protocols based on link procedures.

ATM thus has many advantages over both the circuit and packet modes. It combines the flexibility of a packet switched network with the facilities for circuit based switching, and can handle both variable bit rate (VBR) and constant bit rate (CBR) traffic.

The development of efficient provisioning methods is one of the challenges for the successful deployment of ATM or other broadband networks. In order to allocate network resources cost effectively, the network operators will require provisioning tools that match expected traffic demands with the appropriate transmission and switching equipment.

One of the major issues in the optimal design of circuit-switched telecommunications networks is the geographical location of the switches. Several optimization models have been presented to accomplish this task, as the ones presented in Gavish (1990a), Gavish (1991), Holmberg and Hellstrand (1994), Pirkul and Nagarajan (1992), Pirkul *et al* (1988). Usually, given a traffic demand matrix, these models solve the star-star concentrator problem, that is, determine the optimal number and location of switches so as to minimize overall cost, subject to traffic-capacity constraints at the switches. These capacity constraints are, in general, static. That is, there is no explicit probabilistic analysis of the traffic. Rather, some indication of the capacity of the switch is used as an upper bound for the sum of the average traffics offered by the users allocated to it. When using this approach, the required Grade of Service (GOS- the percentage of calls lost because of congestion) is not a design parameter. The technique is oriented to the design of circuit-switched networks carrying homogeneous traffic.

When dealing with circuit-switched networks carrying heterogeneous traffic, such as the ATM networks, besides cost, there are some other characteristics of the network, which become important, as the Quality of Service (QOS-the proportion of information-cells, lost in a communication. Usually expressed as cells lost per million cells transmitted).

In ATM networks, when buffers at the switches are too long, and delay upperbound techniques are not being applied to delay-sensitive traffic, the cell delay suffers variations (Cell Delay Variation, CDV) that may seriously impair the communication. Thus, this phenomenon must be taken into account in the design of the networks. The design must be done in such a way that the cells do not form a long queue at each switch, reducing significantly the CDV.

Two models are here proposed, that find optimal locations of switches on a network, and allocations of users to them so as to minimize overall cost. Also, for a given buffer length, if P is

the probability of the queues of cells at each switch of being longer than the length of the buffer, the models keep P smaller than or equal to α , the desired cell-loss probability or QOS. The first formulation, based on the plant location model, finds the optimal number of switches and their locations. The second one, based on the p -median model, locates a predetermined number of switches, minimizing the cost of connecting the users. Both models are also applicable to the design of networks carrying homogeneous traffic, in which case the traffic can be better characterized by its distribution, rather than by an average.

Both models assume that the system is a network of queues. Several location models have been presented which are intended for the design of spatial queueing systems. In general, these models are oriented to the design of emergency services, in which servers travel to the site of the emergency, as opposed to systems in which servers are fixed, as in most commercial telecommunications networks. Some of them assume a single server in the region under study. Berman, Larson and Chiu (1985b), develop a heuristic algorithm to locate optimally one server on a congested network. They formulate the Stochastic Queue Median. A model by Batta (1988a) considers the situation in which there might be a selective rejection of calls by the dispatcher. Batta, Larson and Odoni (1988b) present a model and an algorithm for locating one server when there are calls of different priorities. Batta (1989) presents a model to study the effect of using expected service time dependent queueing disciplines on optimal location of a single server. These models assuming single servers are, sometimes, used as building blocks for algorithms that locate more than one server, as in Berman, Larson and Parkan (1987). Berman and Mandowsky (1986) use the Stochastic Queue Median combined with a 2-server districting algorithm, to develop a general location - districting iterative algorithm for two units, and for n -nodes, m -server networks. All of these models are nonlinear, solved by heuristics, and their objective is to minimize expected response time of servers that travel to the site of an emergency. Also, all models use approximations in order to model the system.

Optimization models for location, derived from the Location Set Covering Problem (LSCP, Toregas *et al*, (1974), the p - median of Hakimi (1964) and ReVelle and Swain (1970), and the Maximal Covering Location Problem (MCLP, Church and ReVelle, 1974), are linear, and can be solved to optimality. To deal with congestion, very few of these optimization models make explicit use of queueing. The AMEXCLP, by Batta, Dolan and Krishnamurthy (1989), maximizes

coverage by an emergency service in a congested system. The QPLSCP, by Marianov and ReVelle (1994), and the QMALP, by Marianov and ReVelle (1996), assume a spatial queuing system in which emergency servers attend only demands within some distance. All of these models apply only to emergency services.

In Section 2.2, the probabilistic plant location model is developed. Section 2.3 is devoted to the p - median model. After formulating the models, some computational experiments are shown. Finally, some conclusions and ideas for future research are stated. Note in the following that the user is a traffic source/sink and not a person.

2.2. Development of a Probabilistic Plant Location Model

The probabilistic plant location model can be stated as:

"Locate the optimal number of switches and connect users to them so to assure that: i) every user will be connected to a switch, (possibly within a maximum distance from its location), and ii) at its arrival to the switch, every cell will find free space in a buffer of length b (will not be lost), with a probability of at least α ."

The formulation of the model is the following:

$$\text{Minimize } \sum_{j \in J} v_j y_j + \sum_{i,j} c_{ij} x_{ij} \quad (1)$$

$$\text{Subject to } \sum_j x_{ij} = 1 \quad \forall i \quad (2)$$

$$x_{ij} \leq y_j \quad \forall i, j, \text{ and} \quad (3)$$

$$P[\text{at least one of the } b \text{ positions of the buffer is free}] \geq \alpha \quad \forall j \quad (4)$$

$$x_{ij}, y_j = 0, 1 \quad \forall i, j$$

The 0 - 1 variable y_j is one if a switch is located at node j , and zero otherwise. The 0 - 1 variable x_{ij} is one if the user located at node i is connected to a switch located at node j , and zero otherwise. If it is required to force the connection of every user to a switch located within the pre-specified maximum distance S , a set N_i is defined as the set of candidate locations for switches that are within that distance from node i . Then, variables x_{ij} are only defined for the pair of

subscripts (i, j) such that $j \in N_i$, because there is no need to define variables that will never be equal to one. These are the variables that corresponds to connections longer than S . Objective (1) minimizes cost, which depends on the number of switches to be located in the region, and the cost of connection of a user to a switch. Constraint (2) forces the connection of every user i by exactly one switch, located inside of the set N_i . Constraint (3) states that a user i can not be connected to node j unless there is a switch at node j , and constraint (4) forces every switch to have less than, or at most, b cells on line in the buffer, with a probability of at least α . This constraint assures that, on its arrival to the switch, every cell will find, most of the time, free space in the buffer,.

In order to write constraint (4) in a tractable form, the usual assumption that cells appear at user node i according to a Poisson process with intensity f_i is used. Since each switch serves a set of user nodes, the input traffic at that switch is the sum of the traffic of the nodes in the set, and it can be described as another stochastic process, equal to the sum of several Poisson processes. This stochastic process can be easily shown to be also a Poisson process, with intensity λ_j equal to the sum of the intensities of the processes at the nodes served by the center. This set of nodes is not known before the solution of the mathematical programming problem is known. However, variables x_{ij} can be used in order to rewrite the parameter λ_j as

$$\lambda_j = \sum_{i \in I} f_i x_{ij}$$

Using this definition, if a particular variable x_{ij} is one, meaning that node i is connected to center j , the corresponding intensity f_i will be included in the computation of λ_j .

The model assumes an exponentially distributed service time (the output time from the buffer), with a service rate of μ_j . Here, $\mu_j \geq \lambda_j$, otherwise the buffers will always overflow. If steady state is assumed, the well known results for a M/M/1 queueing system can be used for each switch and its connected users.

If the state k of the system is defined as k users in the system (either being attended or in queue), the state transition diagram of the system is the one shown in Figure 2.1.

State zero corresponds to the buffer being idle, state 1 to one cell arriving and being output from the switch, state 2 to two cells at the switch: one of them being output and one in the buffer, and so on. The probability of having at most b cells in the buffer, plus one being processed for its

output at the switch, should be at least equal to α . If p_k represents the steady state probability of being in state k , the requirement can be written as:

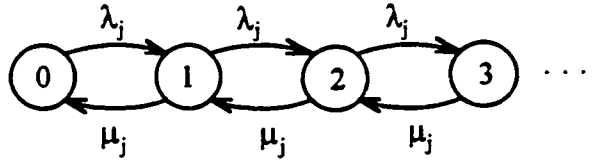


Figure 2.1. State transition diagram.

$$p_0 + p_1 + \dots + p_{b+1} \geq \alpha \quad (5)$$

Writing and solving the steady state balance equations of the M/M/1 system, the following expression for the steady state probabilities (Wolff, 1989) are obtained:

$$p_k = (1 - \rho_j) \rho_j^k.$$

where $\rho_j = \lambda_j / \mu_j$. Hence, equation (5) becomes:

$$(1 - \rho_j) + (1 - \rho_j) \rho_j + (1 - \rho_j) \rho_j^2 + \dots + (1 - \rho_j) \rho_j^{b+1} \geq \alpha,$$

or

$$(1 - \rho_j) \sum_{k=0}^{b+1} \rho_j^k \geq \alpha,$$

which is equivalent to

$$(1 - \rho_j) \frac{1 - \rho_j^{b+2}}{1 - \rho_j} \geq \alpha,$$

or

$$\rho_j \leq \sqrt[b+2]{1 - \alpha}.$$

Since $\rho_j = \lambda_j / \mu_j$,

$$\lambda_j \leq \mu_j \sqrt[b+2]{1 - \alpha}. \quad (6)$$

Equation (6) is equivalent to constraint (4). Using the relationship between the intensity of burst arrival at the switch and the intensities at the user nodes, constraint (4) is rewritten as:

$$\sum_{i \in I} f_i x_{ij} \leq \mu_j^{b+2} \sqrt[3]{1-\alpha}, \quad (7)$$

which is a linear, deterministic equivalent of constraint (4).

2.3. Development of the p -median Model for Congested Systems

The queueing p -median model can be stated as:

"Locate p switches and connect users to them, at the minimum cost, so that: i) connected users are allocated to a switch within a standard distance from its location, and ii) at the arrival of a cell to the switch, it will find free space in the buffer, with a probability of at least α ."

The formulation of the model is the following:

$$\text{Minimize } \sum_{i,j} c_{ij} x_{ij} \quad (8)$$

$$\text{subject to } x_{ij} \leq y_j \quad \forall i, j \quad (9)$$

$$\sum_j x_{ij} \geq 1 \quad \forall i \quad (10)$$

$$\sum_i y_i = p \quad (11)$$

$$P[\text{at least one of the } b \text{ positions of the buffer is free}] \geq \alpha \quad \forall j \quad (12)$$

$$x_{ij} = 0, 1, j \in N_j \quad \forall i, j \quad (13)$$

$$y_j = 0, 1 \quad \forall j$$

Variables and parameters are as defined before. The objective (8) minimizes connection cost. Constraint (9) states that it is not possible to connect a user i to a node j , unless there is a switch at this node. Constraint (10) forces every demand to be connected to at least a switch. Equation (11) limits the number of switches to be located. Constraint (12) determines the QOS. It is rewritten in the same form as before, in order to have a linear deterministic equivalent.

2.4. Computational Experiments

The p -median model was tested using a commercial integer-programming package, CPLEX. Several runs were made, with 4 to 6 switches located on a 30-node network. The cell-loss probability $(1 - \alpha)$ was set at 10^{-3} , 10^{-4} , 10^{-5} , and 10^{-6} , and the average service time (average rate of cells going out of the buffers) was set at 200,000 cells/sec. The test network is shown in Figure 2.2, where each user is also a potential switch location, and the distances are Euclidean. The connection cost was considered proportional to the distance. For the test runs, no distance limit for user-to-switch connections was assumed. Although we are dealing with burst of variable size, the experiment was done with a burst length of unity for simplicity.

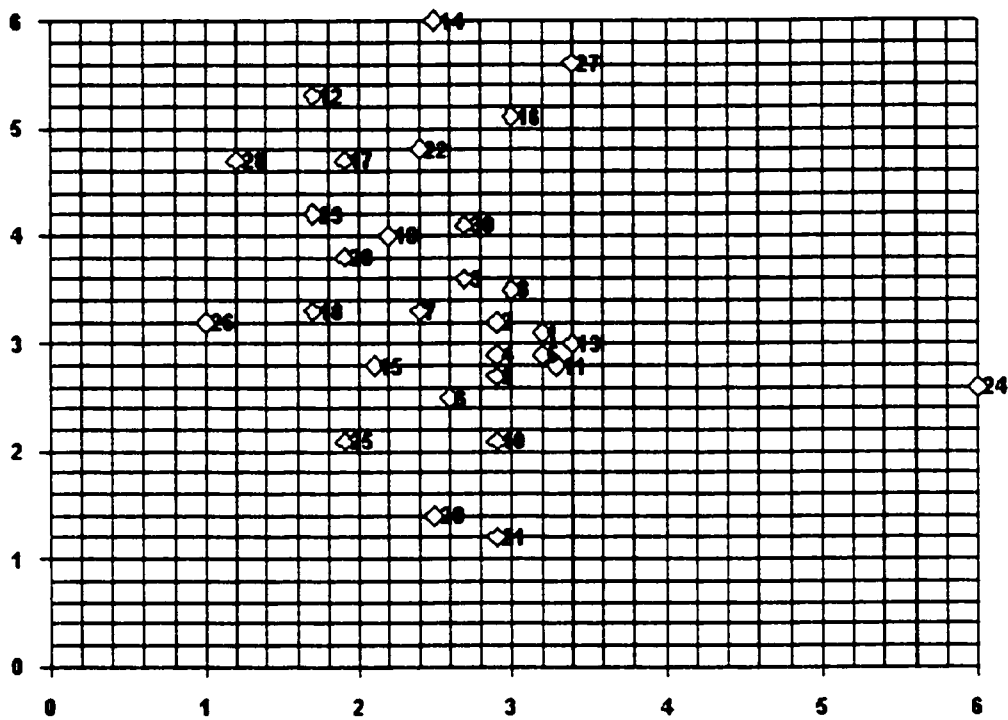


Figure 2.2. 30-node network

The traffic offered by each user, expressed as cells per second, as well as the coordinates of each user, are shown in Table 2.1.

Table 2.1: test network

Node	X	Y	Traffic
1	3.2	3.1	56800
2	2.9	3.2	49600
3	2.7	3.6	44800
4	2.9	2.9	31200
5	3.2	2.9	28000
6	2.6	2.5	16800
7	2.4	3.3	16000
8	3.0	3.5	15200
9	2.9	2.7	13600
10	2.9	2.1	13600
11	3.3	2.8	12800
12	1.7	5.3	12000
13	3.4	3.0	11200
14	2.5	6.0	9600
15	2.1	2.8	9600
16	3.0	5.1	8800
17	1.9	4.7	8000
18	1.7	3.3	8000
19	2.2	4.0	7200
20	2.5	1.4	7200
21	2.9	1.2	7200
22	2.4	4.8	6400
23	1.7	4.2	6400
24	6.0	2.6	6400
25	1.9	2.1	6400
26	1.0	3.2	5600
27	3.4	5.6	4800
28	1.2	4.7	4800
29	1.9	3.8	4800
30	2.7	4.1	4800

The values of the right hand side of equation (7) were computed (limit values of arrival rates to each switch, λ), for an average service rate μ of 200,000 cells per second, different values of $(1 - \alpha)$, and different values of buffer length b . These values are shown in Table 2.2.

The commercial Integer Programming software CPLEX 3.0, on a cluster of eight DEC 3000 - 700 AXP computers was used to solve the p -median model. The run time was between 1.5 and 17 seconds per run.

Tables 2.3 to 2.6 show the results obtained. The number of servers, as well as their locations, and the allocations of connections are shown.

Note that some cases are infeasible. This is because the total traffic exceeds the capacity of the switches, given the required cell-loss probability. Note also that, for the case of 4 servers, the solutions are extremely robust, in the sense that in all cases, the same locations are chosen by the model. However, the objective values are different.

Table 2.2: Maximum traffic arriving to each switch (in cells per second) for different values of α (cell-loss probability) and b (buffer length), for an average service rate of 200,000 cells per second.

$(1 - \alpha)$	b	λ
10^{-3}	100	186,904
	50	175,121
	20	146,105
10^{-4}	100	182,732
	50	167,536
	20	131,587
10^{-5}	100	178,653
	50	160,279
	20	118,511
10^{-6}	100	174,665
	50	153,336
	20	106,734

Table 2.3: Results for cell-loss probability of 10^{-3} .

Servers	b	Objective	Locations	Run time
4	100	20.7754	7,9,22,24	4.21
4	50	21.3789	7,9,22,24	4.09
4	20	Infeasible		
5	100	17.0796	1,6,22,24,29	1.82
5	50	17.3355	5,6,22,24,29	2.63
5	20	18.0469	1,6,22,24,29	7.23
6	100	14.8232	5,7,17,20,24,25	2.70
6	50	14.8714	5,7,16,20,23,24	2.20
6	20	15.2350	3,5,6,16,23,24	2.76

This is because when the cell loss probability is allowed to be high (10^{-3}), the incoming traffic at each switch can also be higher, and the users can then be connected to their closest switches. As the required cell-loss probability decreases (10^{-4} , 10^{-5} , and 10^{-6}), so does the traffic handling capacity of each of the switches. In this case, some users may be connected to switches which are not their closest switches, because if they were connected, the switch capacity would be exceeded, for the given cell-loss probability. In other words, instead of being connected to a switch because of its closeness, users are connected to a switch that has enough capacity, in terms of the cell-loss probability.

Table 2.4: Results for cell-loss probability of 10^{-4} .

Servers	b	Objective	Locations	Run time
4	100	21.1863	7,9,22,24	4.23
4	50	21.6020	7,9,22,24	4.33
4	20	Infeasible		
5	100	17.0796	1,6,22,24,29	2.58
5	50	17.3355	5,6,22,24,29	2.42
5	20	18.7872	1,6,22,24,29	14.10
6	100	14.8568	5,7,17,20,24,27	2.66
6	50	14.9129	3,5,10,16,23,24	1.95
6	20	15.8186	3,5,6,22,24,29	16.76

Table 2.5: Results for cell-loss probability of 10^{-5} .

Servers	b	Objective	Locations	Run time
4	100	20.6819	7,9,22,24	3.06
4	50	21.2823	7,9,22,24	3.62
4	20	23.2565	7,9,22,24	10.23
5	100	17.0796	1,6,22,24,29	1.72
5	50	17.0796	1,6,22,24,29	2.62
5	20	17.5355	5,6,22,24,29	4.39
6	100	14.6514	3,5,16,20,23,24	1.51
6	50	14.8714	5,7,16,20,23,24	3.13
6	20	15.0350	3,5,6,16,23,24	2.98

Table 2.6: Results for cell-loss probability of 10^{-6} .

Servers	b	Objective	Locations	Run time
4	100	21.2823	7,9,22,24	3.65
4	50	22.1784	7,9,22,24	7.13
5	100	17.0796	1,6,22,24,29	2.57
5	50	17.5355	5,6,22,24,29	3.46
6	100	14.8714	5,7,16,20,23,24	2.36
6	50	15.0350	3,5,6,16,23,24	2.94

Figure 2.3 shows the example case of 4 switches, cell-loss probability of 10^{-6} and buffer length 100. The arrows show the connections. Note that user located at node 3 is connected to the switch located at node 22, although node 3 is closer to the switch located at node 7. The same happens with the user located at node 1, which is connected to switch 7 instead of switch 9. Users located at nodes 7,9,22 and 24 are connected to the switches located at the same nodes.

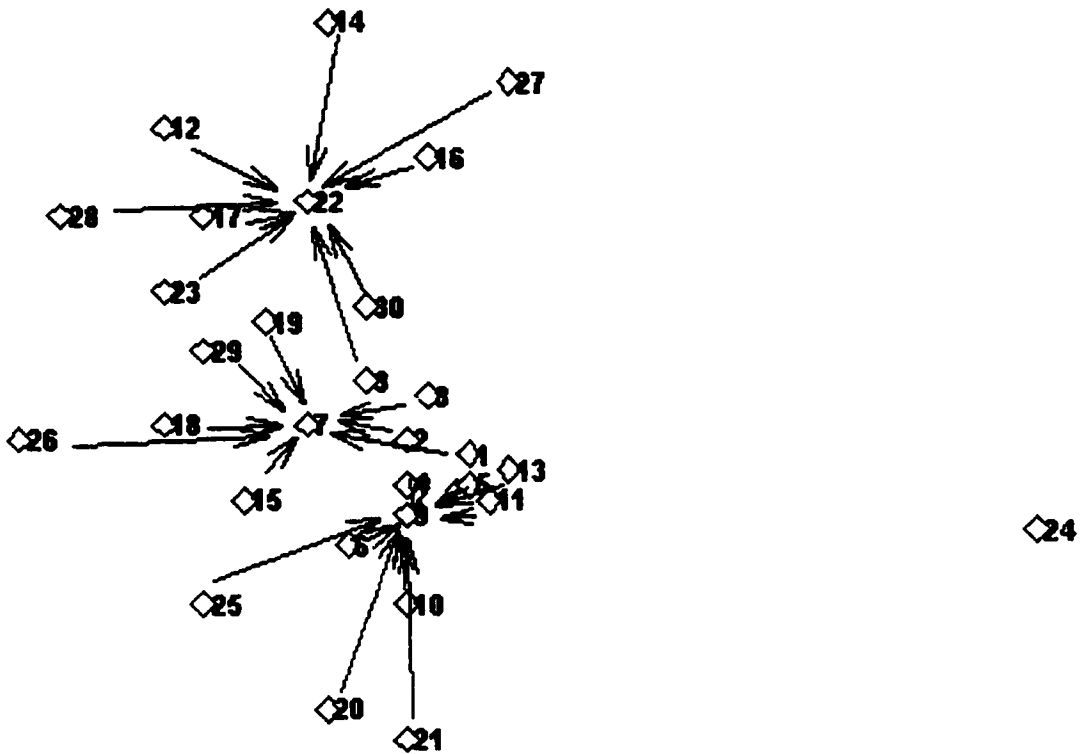


Figure 2.3. Locations of 4 switches, cell-loss probability of 10^{-6} and buffer length 100.

2.5. Conclusions

Two mathematical models have been formulated for the design of telecommunications networks, whose novel characteristic is the explicit treatment of the Cell-Loss Probability and the Cell Delay Variation.

Normally, a short length buffer is located at each switch, so that the cell-delay variation is reduced. However, shortening the buffer length increases the cell-loss probability.

The mathematical models choose optimum locations for the switches in such a way that the cell-loss probability is kept constrained below a maximum value, while maintaining a short length buffer to reach the cell delay variation objective.

Computational experience is offered with one of the models showing that, as the cell-loss probability is decreased, the locations of the switches do not change significantly, while the allocations of users to certain switches do change.

CHAPTER 3: FAULT TOLERANT TELECOMMUNICATION NETWORKS DESIGN MODELS

3.1. Introduction

The design of cost-efficient survivable telecommunications networks is a major task with great economic impact. Nowadays, the extensive use of high capacity physical media like optical fiber increases the potential damage to network services due to failures in links or nodes (cable cuts, electronic failures on switching centers, etc.). Since quality of service has become a major competition advantage for services in the industry, planners are looking for end-to-end survivable designs that are robust with respect to failure in network components.

Network reliability refers to the probability that a network works according to a previously specified way. Fault tolerance is the capability of a network to continue the service, in a specified way, after a failure has occurred. While reliability is a characteristic depending on the external environment of the network, fault tolerance is an internal characteristic, and so it is a variable to be considered in the design process.

The fundamental network design problem that will be addressed in this part of the thesis, consists of, given the location of the communication nodes, the traffic demands among pairs of origin-destination nodes, fault tolerance restrictions and the cost structure for the available technologies, to determine the capacity to be assigned to each link, in such a way to satisfy the restriction while reaching a minimum cost network. This requires determining which nodes are to be connected (topological design), with which bandwidth (capacity assignment), and an efficient routing strategy which has to accommodate the traffic demands.

The main objective of this part of the thesis is the unified formulation of mathematical models integrating the **Topology Design, Routing and Capacity Assignment** problems. Four models are presented, three of them being new formulations to the aforementioned problems. The mathematical models correspond to the class of Mixed Variables Linear Programming Problems, where the objective and the restrictions are a linear function of the problem variables, and some of the variables can only take integer values. Generally, this type of formulation can be solved to optimality, by using a Branch and Bound algorithm, for very small sized networks. The

formulations belong to the NP-hard type of problems. For instance, the topology design problem for N nodes, requires to find the optimum among all the possible networks of N nodes, which are $2^{N(N-1)/2}$. For $N=12$, the number of candidate networks is around 10^{20} , which requires of around 31,000 years of CPU time, assuming each alternative can be processed in 10^{-8} seconds.

A second objective of this part of the thesis corresponds to the development of efficient procedures to solve the problems for bigger networks. Two methods were investigated: Lagrangean Relaxations and Heuristics, proposing some new elements.

3.2. Uncapacitated Assignment Models

In Section 3.2.1 some basic concepts are introduced referring to the modeling and the characteristics of a fault tolerant network. In Section 3.2.2 the mathematical models for the uncapacitated fault tolerant network design are presented.

3.2.1. Network Modeling

A fault tolerant communication network design problem can be modeled as a linear programming problem that uses real and integer variables. A graph $G = (V, E)$ is used, where V is the set of nodes representing the communications offices that are to be connected. E is the set of arcs connecting different pairs of nodes. The elements of E represent potential links to be considered in the network under design. Unidirectional links allow the flow of data in only one direction, while bidirectional links allow the flow of information in both ways. In this thesis, each arc allows traffic flow in both directions (i.e. the graph is not directional). If, for instance, any arc $e \in E$ has a cost C_e , the total cost of building a network $N=(V,F)$, where $F \subseteq E$, is:

$$CT(N) = \sum_{e \in F} C_e$$

3.2.1.1. Network Connectivity

The connectivity of a node pair in a network, is defined as a function of the number of existing routes that connect that node pair.

3.2.1.2. Connectivity Levels

A node s is one-connected to a node v if at least one route exists in the network between both nodes. If every network node pair is connected, the network is said to be a *one-connected*

network. A node v is *two-connected* to a node s if at least two disjoint paths between v and s exist.

The disjoint paths can be either:

- Arc-disjoint paths: these paths do not have any arc in common.
- Node-disjoint paths: these paths do not have any intermediate node in common. Two node-disjoint paths are also arc-disjoint paths.

In case a) the nodes are said to be *arc biconnected*, while in case b) the nodes are said to be *node biconnected*. If all the network nodes are node (arc) biconnected, the network is said to be *node(arc) biconnected* or has *two-connectivity*.

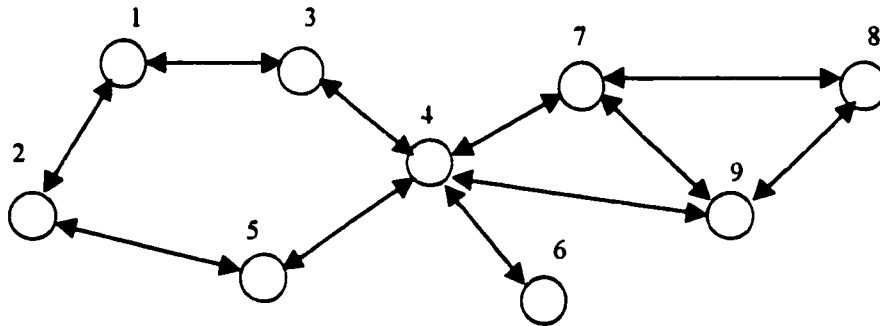


Figure 3.1: Arc or node disjoint paths

In the network of figure 3.1, node 6 is one-connected to all the network nodes, while it is not two-connected to the network nodes. Node 1 is node biconnected with node 4 through paths 1-3-4 and 1-2-5-4 which are node disjoint. Nodes 2 and 9 are arc biconnected through path 2-1-3-4-9 and 2-5-4-7-9, which share node 4 but do not have common arcs. It can be said that the sub-network of nodes 1,2,3,4 and 5 is node biconnected, as well as the sub-network of nodes 4,7,8 and 9. The sub-network formed by all nodes but node 6, is arc biconnected. The whole network is one-connected.

3.2.1.3. Articulation Node

A node v is called an *articulation node* if after eliminating it from the network (with all its arcs), the remaining network is composed of two *one-connected* sub-networks (for example, node 4 in figure 3.1).

3.2.1.4. Cycles

A cycle is a path starting and ending in the same node, and that may visit other nodes without repeating any arc. In figure 3.1 several cycles can be identified, such as 2-5-4-3-1-2, 9-8-

7-4-9, or 9-8-7-9. The importance of cycles is that all nodes in a cycle are node biconnected, and so in the case any node or arc would fail, there is still a path available between the nodes.

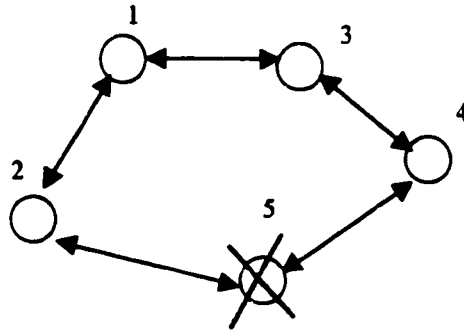


Figure 3.2: A Cycle is a node biconnected network

In figure 3.2, if node 5 fails, node 2 can communicate with node 4 through path 2-1-3-4.

3.2.1.5. How to prove network biconnectivity

In order to test a network biconnectivity, arcs (or nodes) are taken out from the network, one at a time, while observing if the remaining nodes are still one-connected.

3.2.1.6. Fault Tolerant Degree

Fault tolerance of a given network can be measured as the proportion of total traffic that can be carried after the failure occurs. In particular:

- a) Arc Fault Tolerance: it is the proportion of point to point traffic that can be carried after a failure in only one of the arcs of the network .
- b) Node Fault Tolerance: it is the proportion of point to point traffic (not going to or departing from the failed node) that can be carried after a failure in only one of the nodes of the network.

In this way, for each given fault, a fault tolerance proportion is obtained. Normally the network is given an Average Fault Tolerance degree (arithmetic mean of different proportions) upon a simple fault, or sometimes it is given the Worst Fault Tolerance degree (lowest proportion).

It is also useful to have a measure of the biconnectivity in the network (Biconnectivity Degree), defined as the point to point traffic percentage that can be carried through two or more (node or arc) disjoint paths in the network. As an example, a ring network has a full Node and Arc

Fault Tolerance, and a full Biconnectivity Degree. Note that these measures assume that each arc of the network has an infinite capacity, and also no re-routing algorithms are given.

3.2.2. Survivable Uncapacitated Network Design Problem (SUND):

Two models are now presented, which are of interest in the design of fault tolerant networks. The problem can be stated as:

Survivable Uncapacitated Network Design Problem: given the node topology of the network, the arc fixed costs and the fault tolerance restrictions, the problem is to choose which arcs are to be installed which produce the minimum cost network

In this problem, for each arc a 0-1-decision variable exists, which determines if the arc is present or is not in the final solution. There are no variables assigning capacities to those links. The links then are assumed to be of infinite capacity. Besides, it is assumed that arc $e \in E$ has a fixed cost C_e . Since there is no limit to the amount of traffic in the arcs, it is of no relevance in these models the traffic demand for each origin-destination pair.

3.2.2.1. First Model: SUND1

This first model is due to Grötschel, Monma and Stoer (1992). The formulation is based in “cut inequalities” in the network’s graph. Given a graph $G=(V,E)$, a minimum cost network is to be found that satisfies the following connectivity restrictions.

In the network there exist three node types. Each node s is characterized by a *connectivity requirement* or *type* represented by $r_s \in \{0,1,2\}$ for all $s \in V$. In this way, the node possessing $r_s=2$ is said to be of type 2. Each type 2 node is a special node requiring of at least two disjoint paths (in arcs or nodes) to all the other type 2 nodes in the network. These nodes may correspond to economically important network nodes, because of heavy traffic (or any other characteristic) handled by them that justifies the implementation of a fault tolerant system. The cost/benefit analysis deciding which nodes in the network are declared of type 2 should be done by experts and based in own economical parameters. Each node v with $r_v=1$, it is said to be a “common node” and requires at least one path to each one of the other type 1 or 2 nodes. Finally, type 0 nodes are those nodes that may or may not be connected to other nodes (representing interconnection

boxes). In other words, the minimum cost network $N=(V,F)$ should contain at least $r(s,t):=\min\{r_s,r_t\}$ disjoint paths between each node pair $s,t \in V$. Furthermore, for every node set $W \subseteq V$ the W induced cut is defined as the set of arcs represented by:

$$\delta_G(W):=\{ij \in E \mid i \in W, j \in V \setminus W\}$$

i.e., all those arcs possessing an end in some node in W and the other end in some node in V that is not in W (that is, in $V \setminus W$). In the same way, if $G - v$ is the graph obtained by eliminating node v and all its connected arcs from graph G , a cut is obtained given by:

$$\delta_{G-v}(W):=\{ij \in E \mid i \in W, j \in V - v \setminus W\} \text{ for every node set } W \subseteq V - v.$$

The concept of connectivity requirement is extended to

$$r(W):=\max\{r_s \mid s \in W\} \quad \forall W \subseteq V \text{ and defining}$$

$$con(W):=\max\{r(s,t) \mid s \in W, t \in V \setminus W\}$$

$$:=\min\{r(W), r(V \setminus W)\}.$$

If the following variables are used:

$$Y_{ij} = \begin{cases} 1 & \text{if arc } ij \text{ is selected as part of the network.} \\ 0 & \text{otherwise} \end{cases}$$

and the operator $y(F) = \sum_{e \in F} Y_e$ is defined for every arc subset $F \subseteq E$, then the *Survivable Uncapacitated Network Design Problem (SUNDI)* is stated as follows:

[P1]

$$\text{Minimize } \sum_{ij \in E} C_{ij} Y_{ij}$$

Subject to

$$\begin{aligned} y(\delta_G(W)) &\geq con(W) && \forall W \subseteq V \\ &&& \phi \neq W \neq V \end{aligned} \quad (1a)$$

$$\begin{aligned} y(\delta_{G-z}(W)) &\geq 1 && \forall z \in V \\ &&& \forall W \subseteq V \setminus \{z\} \\ &&& \phi \neq W \neq V \setminus \{z\} \\ r(W) &= 2 \\ r(V \setminus (W \cup \{z\})) &= 2 \end{aligned} \quad (1b)$$

$$0 \leq Y_{ij} \leq 1 \quad \forall ij \in E \quad (1c)$$

$$Y_{ij} \text{ integer} \quad \forall ij \in E \quad (1d)$$

Restrictions (1a) are called *cut restrictions* and make sure at least two disjoint paths exist between each type 2-node pair. These restrictions make the number of arcs in the cut to be greater or equal to the one required by the nodes with greater requirements in the cut's departing and arriving sets. This also makes the rest of type 1 nodes to be connected to the network. Type (1b) restrictions are called *node cut restrictions* and make that two arcs going out of a node subset W will not go into a one arrival node in the set \bar{W} . These restrictions make sure the existence of at least two disjoint paths between each special node pair. Restrictions (1c) and (1d) make sure the arc assigning variables Y_{ij} will be binary.

If only restrictions (1a),(1c) and (1d) are considered, a design model for an arc biconnected network for type 2 nodes results (model SUND1E). If restrictions (1b) are also considered, then a design model for a node biconnected for type 2 nodes results (model SUND1N).

3.2.2.2. Second Model: SUND2

The following new model is based in the classical network design formulation. This model considers that each type 1 node sends one unit of communication (its requirement) to each one of the other nodes in the network. Special type 2 nodes send 2 units of communication to each of the other type 2 nodes, but using arc disjoint paths. Finally, type 0 nodes do not send communication units and only function as transfer nodes.

For each node pair m requiring a connection, an origin and destination is established. Since the network is bi-directional in all its links, for sake of simplicity, it is assumed that a communication pair has its origin in the lower numbered node and its destination in the higher numbered node. In this way, only the pair $m_1=(N_1, N_7)$ would be considered while omitting pair (N_7, N_1) . Functions $O(m)$ and $D(m)$ define respectively the origin and destination of pair m (for example, $O(m_1) = N_1$ and $D(m_1) = N_7$).

Let M be the set of origin-destination (OD) pairs m_i . M contains all the possible OD pairs. Additionally, for each pair $m_i \in M$ there exists a pair requirement $rp_i = \min\{r_{O(m_i)}, r_{D(m_i)}\}, \forall i$. The set of all rp_i is called R . For example, in the network of figure 3.3:

$$M = \{(N_1, N_2); (N_1, N_3); (N_1, N_4); (N_2, N_3); (N_2, N_4); (N_3, N_4)\}$$

$$R = \{1; 1; 1; 2; 2; 2\}$$

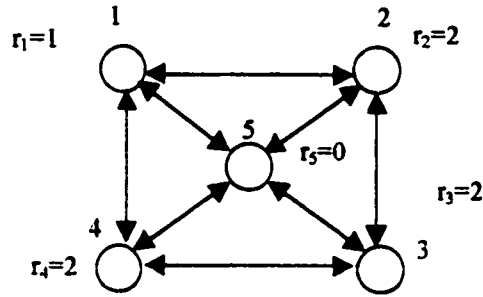


Figure 3.3: Five-node network.

Variables for this model are:

$$Y_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is selected in the solution} \\ 0 & \text{otherwise.} \end{cases}$$

X_{ij}^m = Flow of pair m going from node i to j .

The model SUND2 is then:

[P2]

$$\text{Minimize } \sum_{ij \in E} C_{ij} Y_{ij}$$

Subject to

$$\sum_{\{j:(i,j) \in E\}} X_{ij}^m - \sum_{\{j:(j,i) \in E\}} X_{ji}^m = \begin{cases} rp_m & \text{if } i = O(m) \\ -rp_m & \text{if } i = D(m) \\ 0 & \text{otherwise; } \forall i \in V \end{cases} \quad \forall m \in M \quad (2a)$$

$$X_{ij}^m \leq 1 \quad \forall (ij) \in E \quad \forall m \text{ such that } rp(m) = 2 \quad (2b)$$

$$\sum_{j:(i,j) \in E} X_{ij}^m \leq 1 \quad \forall i \in V - \{O(m), D(m)\} \quad \forall m \text{ such that } rp(m) = 2 \quad (2c)$$

$$X_{ij}^m \leq Y_{ij} \quad \forall (ij) \in E \quad \forall m \in M \quad (2d)$$

$$X_{ij}^m \geq 0 \quad \forall (i,j) \in E$$

$$\forall m \in M \quad (2e)$$

$$Y_{ij} = 1 \text{ or } 0 \quad \forall (i,j) \in E \quad (2f)$$

Restrictions (2a) indicate the existence of one unit of outgoing flow in each origin node of the OD pairs with $rp = 1$, and two units of flow for all OD pairs with $rp = 2$. Similarly, each destination node absorbs the same number of units. For all the rest of the nodes, the input flow is equal to the output flow (including the case where there is no flow). This restriction ensures the continuity in the search for paths. Restrictions (2b) make impossible the flow of pair m to be greater than one unit, for all the special OD pairs, and so oblige to the search of an alternate arc disjoint path. Restrictions (2c) make sure that the output flow in the node will be less or equal than one unit, and so making it necessary for a second node disjoint path to exist. Restrictions (2d) enforce the selection of arc (i,j) whenever there is flow in that arc. Restrictions (2e) and (2f) express the non-negativity and the binary nature of variables.

Sub-problems can also be defined. SUND2E is the design problem with arc disjoint paths, and it consists of [P2] with restrictions (2a), (2b), (2d)..(2f). SUND2N is the design problem with node and arc disjoint paths, and it consists of [P2] with restrictions (2a)...(2f).

Finally, it is interesting to note that every type 1 origin node that has a path to a type 1 or type 2 destination node, it is automatically connected physically to all those nodes that are also connected to the same destination node. This allows reducing the size of sets M and R , and so reducing the number of variables.

3.2.2.3. Model Complexity

SUND1 model belongs to the NP-hard category of problems, since the Steiner Tree problem, which is a particular case of [P1], belongs to that category. SUND2 also belongs to the same category, since it generalizes the classical network design problem, which is also NP-hard. In practical terms, this means there is no known algorithm that can solve the problems in polynomial time (Ahuja, Magnanti and Orlin (1993)).

3.2.2.4. Comparison between SUND1 and SUND2 Models

Due to the very high complexity of both models, it is very difficult to solve to optimality networks of more of around 20 nodes.

As a first experience, 10 instances of each were solved for 5,10,15 and 20 node networks. The CPLEX 6.0.1 solver was used, running under the AMPL modeling language in a Digital Alpha 700 (225 MHz) workstation with Digital Unix V4.0B. Execution times obtained are average values of the 10 instances.

Table 3.1: Statistics in Model Solving.

		Net5	Net10	Net15	Net20	Net25
Total number of nodes	n	5	10	15	20	25
Number of type 2 nodes	n2	3	5	5	8	10
Number of type 1 nodes	n1	1	4	5	9	10
Number of type 0 nodes	n0	1	1	5	3	5
Number of edges	e	10	24	38	70	100
Number of Comm. Pairs	K	6	36	45	136	190
SUNDP1E	1/0 Var.	10	24	38	70	100
SUNDP2E	1/0 Var.	10	24	38	70	100
SUNDP1E	Real Var.	0	0	0	0	0
SUNDP2E	Real Var.	60	864	1710	9520	19000
SUNDP1E	Total Restrict.	15	511	16383	524287	2E+07
SUNDP2E	Total Restrict.	120	1464	2765	14200	28250
SUNDP1E	CPU (sec)	0.003	0.1	25	-	-
SUNDP2E	CPU (sec)	0.25	190	8100	72100	-

It is observed that by using model SUNDP1E, considerably shorter execution times are obtained. This is due to the model being more packed for the Branch and Bound algorithm that CPLEX uses. However, the number of restrictions of model SUND1 grows exponentially in the number of nodes, while model SUND2 grows only in function of each model restrictions. This makes the use of the SUND1 model very costly for networks of more than 20 nodes, while model SUND2 still has a manageable number of restrictions. The main advantage of the model SUND2 is that it contains variables X of information about paths or routes, which the OD traffic should follow. This characteristic allows extending the model to one considering capacity assignment and routing.

Model SUND1 has been used by Grötschel, Monma and Stoer (1992) as a base for the development of a network complexity reduction procedure, which afterwards makes use of an iterative algorithm for the solution of a LP problem, where the restrictions are added one by one. A similar procedure is used by Clarke and Anandalingam (1995) to find initial feasible solutions in a network with only type 2 nodes, and later uses heuristics to improve on the initial solutions.

3.2.2.5. Heuristics to Solve SUND

Several heuristics that can be used to find solutions to this problem are found in Monma and Shalcross (1989). They are divided in two stages: in the first several procedures are explained to build an initial feasible network and, in the second, some heuristics are presented that improve those initial solutions. These heuristics are efficient in terms of the quality of the found solutions, considering the short execution time needed.

3.2.2.6. Characterization of solutions networks for SUND

The resulting topology for the solution network of the SUND problem consists of one or more cycles covering all the type 2 nodes and some type 1 nodes. Around these cycles some node “ears” are located, which are groups of nodes forming a linear path starting in one of the cycle nodes and ending in another of them. Additionally, some tree branches connect the rest of type 1 nodes, usually through minimum spanning tree segments. Figure 3.4 illustrates a typical topology. A cycle called Main Ring is seen, which is formed by special nodes. To this ring, ears of nodes are connected. Each ear that joins a cycle generates an alternate node and arc disjoint path (biconnectivity) among all nodes in the ear and those nodes in the cycle segment. Finally in figure 3.4, several tree branches join type 1 and type 0 nodes with the rest of the network.

This type of network is a good design alternative when fixed costs and infinite capacity are assumed for the arcs. This allows, for example, to minimize the Euclidean total length of the network links, or the total cost of installed fiber in ducts. Whenever the cost structure is proportional to the link length, the optimum network tends to have a reduced number of links. For other applications, where a communication demand for OD pairs exists, it is often desired to design a network that complies with the fault tolerant characteristics, but with the difference that arcs have finite capacities and costs dependent on that installed capacity. In this scenario, models like the ones discussed so far are very expensive since the optimum network results with many

high capacity links. For instance, if the two nodes with the highest traffic in the network, are located in the opposite extremes of the Main Ring, it will be necessary to use high capacities in all the ring arcs instead of considering a pair of routes with less intermediate nodes and shorter in length. Discussed models are useful in the design of networks where the fixed cost structure is dominated by the length variable and does not depend on the link installed capacities. For example, this could be the situation of building an underground network in a city, where ducts represent most of the construction costs. Similarly, the model can be useful to those networks where there exist a large number of OD pairs requiring two or more paths to route their traffic, or situations where all OD demands are very similar and flows are very small compared to the available capacities.

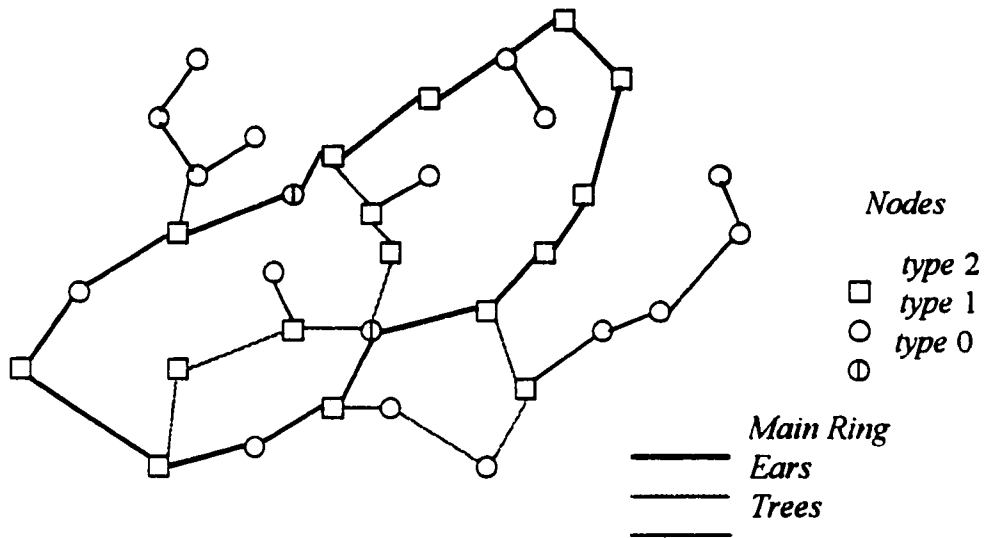


Figure 3.4: Typical topology of a SUND network

3.3. Link Capacity Assignment Models

3.3.1. Survivable Capacitated Network Design SUCAN: Finite Link Capacity Assignment Fault Tolerant Network Design.

This Section presents two new models that represent the finite link capacity assignment fault tolerant network design problem. This problem can be stated as follows:

Finite Link Capacity Assignment Fault Tolerant Network Design Problem : given the network OD traffic demands, find which links, and with which capacities, should be used such that the demands are satisfied for every single fault case.

Between the network nodes there are traffic demands, measured in specific traffic units. For each network arc, the amount of traffic that can be transported is limited by the arc capacity. There exist some special OD pairs that require that their traffic survive any fault that may occur in some node or arc of the network. The objective is to find the minimum cost network satisfying the fault tolerance restrictions for the different traffic demands in each fault situation. There are two different routing and capacity assignment strategies, which are discussed now.

3.3.2. Strategies used in fault tolerant networks

This section explains two different procedures used to reach fault tolerance in a network. The first is called Diversification Routing; the second is called Capacity Reservation [Alevras et al (1998)]. Figure 3.5 shows the two procedures.

In figure 3.5, 100 traffic units must be routed between an OD pair. In the first method called *D-Survivability*, the demand is separated into two or more groups and channeled using two or more physical disjoint paths. This allows the survival of a certain percentage of the traffic, upon a failure in an arc or a node. In figure 3.5, 50% of the traffic remains untouched after a link failure.

The second strategy uses reservation capacity or *R-Survivability*, consisting of providing the network with standby equipment (or excess capacity) such that the traffic can be re-routed through those facilities upon a failure. Figure 3.5 shows the re-routing of the 50% of the traffic after a link failure. This strategy requires a dynamic routing algorithm at the nodes and

management of routing tables for each one of the possible failure cases. Formally, these strategies can be defined in terms of the following parameters:

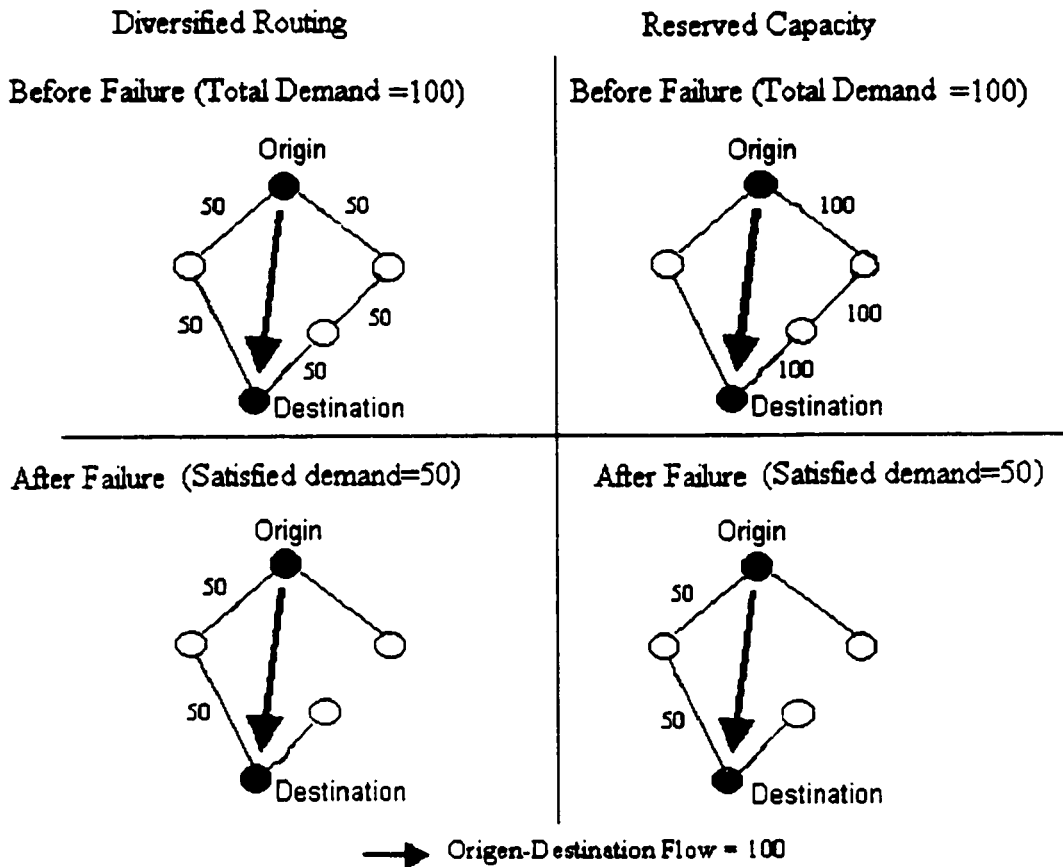


Figure 3.5. Two strategies for fault tolerant networks.

d_m : demand between nodes $O(m)$ and $D(m)$.

σ_m : *diversification parameter*. It corresponds to the maximum fraction of the demand d_m passing through an arc or a node.

ρ_m : *reservation parameter*. It corresponds to the minimum fraction of the traffic demand that can be transported after a single node or arc failure.

If the parameter σ_m is fixed, no more of $\sigma_m d_m$ traffic units can be routed in each arc. Similarly, no more of $\sigma_m d_m$ traffic units are allowed to go out of a given node. Using $\sigma_m < 1$ implies the necessity of finding two or more disjoint paths. For instance, if $0.5 \leq \sigma_m < 1$ at least two disjoint paths are required to route the demand of pair m , while if $\sigma_m = 0.49$, three disjoint

paths are required as a minimum. Values of σ_m lower than 0.33 require four or more disjoint paths which usually is not practical from the management point of view (too many physical routes to monitor). Obviously, a full fault tolerant network can not be achieved by using only a routing diversification method. It is necessary to add some reservation capacity. The reservation parameter ensures that a certain capacity will be available to re-route a minimum of $\rho_m d_m$ traffic units after the failure. It can be appreciated that for a given σ_m , an amount of $(1-\sigma_m)d_m$ traffic units survive the failure without re-routing. This means only those values of ρ_m such that $\rho_m > (1-\sigma_m)$ are relevant.

3.3.3. Problem Characterization

In terms of input and output, the problem may be characterized by:

Input: (conditions known before solving the problem)

1) Original graph $G=(V,E)$ where nodes represent demanding agents and arcs represent the possible links among nodes to be considered. It is also known the position and so the distance among nodes are also known.

2) Demands d_m for every pair of nodes m .

3) Parameters σ_m and ρ_m .

4) Different network states s , where every state s identifies a simple node or arc fault situation. Each state s is defined by the graph $G_s=(V_s,E_s)$, where either the set V_s or E_s has a missing element with respect to the original graph, representing the faulted element. The state $s = 0$ is defined as the normal state of the network, where no fault exists.

5) Set of discrete capacities to be assigned to each arc and their costs.

Output: (resulting from the problem solution)

1) Assigned arcs capacities.

2) Routing for every OD pair demand in each fault situation.

The problem considers a discrete cost structure proportional to the arc capacity and length.

This problem integrates in a single formulation the topology design, capacity assignment and traffic routing problems. Given this characteristic, the formulation is rather general and very hard to solve to optimality. Besides, it is an NP-Hard type of problem, since the uncapacitated

network design problem (which is known to be NP-hard) can be formulated as a special case of this model.

3.3.4. Costs Structure

It has been mentioned that the arc cost is a discrete stepped function depending on the arc capacity and length. This cost structure is typical of those networks where the link services are leased, by paying a fixed cost for a given capacity. It also represents the case where a new network is being installed. Figure 3.6 shows the cost in terms of the capacity:

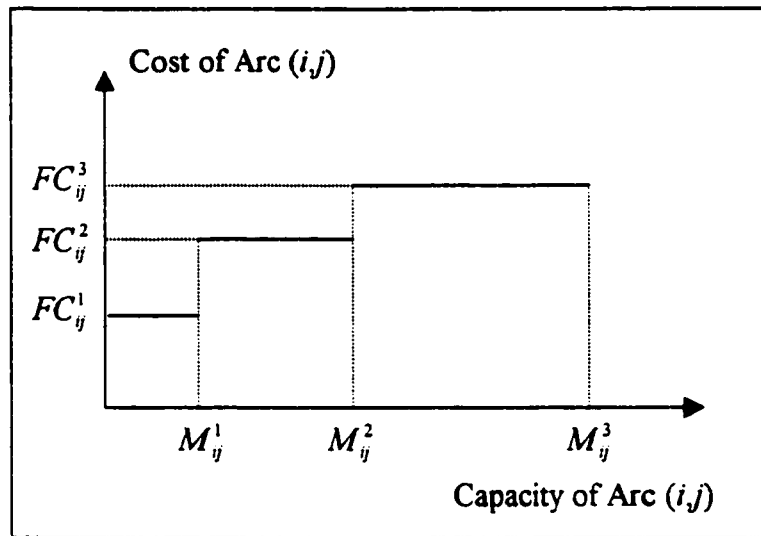


Figure 3.6. Arc v/s Capacity Costs

Figure 3.6 shows the cost of a given link (i,j) in terms of its capacity. Every arc (i,j) of range r supports a total traffic f such that $M_{ij}^{r-1} \leq f \leq M_{ij}^r$ and with a fixed cost FC_{ij}^r which does not depend on f . The total cost of the network is the sum of the costs of each one of the arcs. It is assumed that higher capacity links are cheaper, in a per traffic unit cost point of view. There are also some scale economies in terms of the link lengths, as it is shown in figure 3.7.

3.3.5. Previous work

Most of the previous work on network design with capacity assignment has focussed in the use of non discrete cost structures (most of them linear) and fault tolerance with reservation

capacity, or else in stepped cost structures but without fault tolerance. Minoux (1981) presents a continuous linear cost model and R-survivability. Gavish et al. (1989), use a step cost function but they do not use diversified routing, and they solve the model to optimality for small instances. Lee and others (1995) present a study of how to add capacity to an existing network, using two capacities ranges attending to fault tolerance restrictions. A more general formulation is found in Dahl and Stoer (1998), where a fault tolerant network is designed by a combination of reservation capacity and physical route diversification. They use step cost functions and solve the problem by using a cutting plane algorithm similar to the one presented by Grötschel et al. (1992). A variation of this work can be found in Alevras and others (1998), where restrictions in the maximum number of arcs that a route can have are considered.

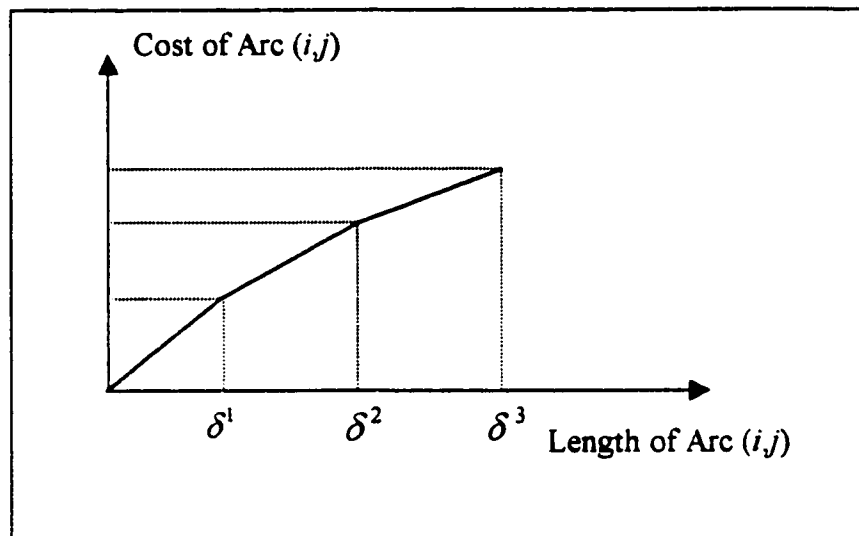


Figure 3.7: Arc costs vs. arc length

3.3.6. Mathematical Models: SUCAN

The following section presents two new mathematical formulations for the Survivable Capacitated Network Design Problem SUCAN.

3.3.6.1. The SUCAN model

This new model is formulated as an extension of the SUND2 model and uses the same notation for the OD pairs. The model considers fault tolerance with physical route diversity and reservation capacity. The parameters σ_m and ρ_m restrict the fault tolerance degree. Each failure situation is identified starting from a state s characterized by the graph $G_s=(V_s, E_s)$, where V_s and E_s

represent nodes and arcs that are working after the failure. The state $s = 0$ is defined as the operation without failures state, with $G_0 = G = (V, E)$, the original graph. For instance, it could be considered the state $s = 1$ as the one representing a failure in link $(2,5)$, resulting in $V_1 = V$ and $E_1 = E - \{(2,5)\}$. The decision of which state s to consider, can be done by a failure probability analysis of the main components of the network, including those nodes and arcs that are more at risk. The following parameters are defined:

S : set of all states s .

K_s : set of all OD pairs with demand d_m without considering those with origin or destination in a faulted node in the state s . $K = K_0$ defines the set of OD pairs in the normal operation state (no failures).

ρ_m^s : is valued 1 for $s = 0$ and ρ_m for $s \neq 0$.

$r \in R = \{1, 2, \dots, r_{\max}\}$, enumerates the type or ranges of available capacities.

M_{ij}^r : capacity associated to the range r for the arc (i, j) . It is measured in traffic units.

FC_{ij}^r : cost of giving capacity range r to arc (i, j)

$$Y_{ij}^r \begin{cases} 1 & \text{if capacity } r \text{ is selected in arc } (i, j) \\ 0 & \text{otherwise.} \end{cases}$$

W_{ij}^{ms} : flow of pair m in arc (i, j) going from i to j for the state s of the network.

The mathematical formulation of the model is then:

[P3]

$$\text{Minimize } \sum_{ij \in E} \sum_{r \in R} FC_{ij}^r Y_{ij}^r$$

subject to

$$\sum_{(j|i, j) \in E_s} W_{ij}^{ms} - \sum_{(j|j, i) \in E_s} W_{ji}^{ms} = \begin{cases} \rho_m^s d_m & \text{if } i = O(m) \\ -\rho_m^s d_m & \text{if } i = D(m) \\ 0 & \text{otherwise; } \forall i \in V_s \end{cases}$$

$$\forall m \in K_s,$$

$$\forall s \in S.$$

(3a)

$$W_{ij}^{m0} \leq \sigma_m d_m \quad \forall (i,j) \in E$$

$$s = 0$$

$$\forall m \in K \quad (3b)$$

$$\sum_{j:(i,j) \in E} W_{ij}^{m0} \leq \sigma_m d_m \quad \forall i \in V - \{O(m), D(m)\}$$

$$s = 0$$

$$\forall m \in K. \quad (3c)$$

$$\sum_{m \in K} W_{ij}^{ms} \leq \sum_{r \in R} M_{ij}^r Y_{ij}^r \quad \forall (i,j) \in E$$

$$\forall s \in S \quad (3d)$$

$$\sum_{r \in R} Y_{ij}^r \leq 1 \quad \forall (i,j) \in E \quad (3e)$$

$$W_{ij}^{ms} \geq 0 \quad \forall (i,j) \in E$$

$$\forall m \in K$$

$$\forall s \in S \quad (3f)$$

$$Y_{ij}^r \in \{0,1\} \quad \forall (i,j) \in E$$

$$\forall r \in R \quad (3g)$$

Restrictions (3a) ensure the continuity of a certain minimum flow from the origin to destination, for each one of the possible states in the network. For $s = 0$ the routing equations for all the traffic demand result (normal operation), while for other states the equations ensure a certain proportion of the original demand reaches its destination, regardless of the failure. This allows fault tolerance by capacity reservation. Restrictions (3b) ensure that no more than $\sigma_m d_m$ traffic units may flow in a given arc, obliging the existence of two or more disjoint OD paths. Restrictions (3c) similarly control the total output flow in each node, resulting in node disjoint paths. Restrictions (3b) and (3c) are those determining the fault tolerance by physical route diversity and are defined only for the normal state $s = 0$. Restriction (3d) makes the total flow in a given arc no larger than the capacity assigned to that arc, for all possible states. Finally, restrictions (3e) ensure that only one capacity range will be selected for each arc.

The model SUCANE (with arc disjoint routes) is defined as the one using restrictions (3a),(3b),(3d)...(3g). Similarly, model SUCANN (with node and arc disjoint routes) corresponds to the model using all the restrictions (3a)...(3g).

This model is useful for the design of a network in which some special nodes exist, with very important traffic responsibilities, requiring for instance that at least 80% of the traffic survives a simple failure. In this case, for instance, set $\sigma_m = 0.5$ and $\rho_m = 0.8$ for the special nodes and $\sigma_m = 1$, $\rho_m = 0$ for the rest.

Parameter ρ_m ensures the minimum surviving traffic fraction upon any node or arc failure. In practice, computer results indicate the survival level is higher, because the arc capacities are assigned using inequalities (3d) using worst situation. So it is possible to send a higher traffic for most of states s without violating the capacity constraint. Variables X_{ij}^{ms} represent flows that should exist in each arc such that the network cost is minimized. For the normal state $s = 0$, this variables indicate how the routing between each OD pair should be done. However, for most of the failure states $s \neq 0$, there will exist alternative routings to the one indicated by variables X which allow to transport a flow higher than the required minimum.

In general, to obtain a fault tolerance level ρ_m for each pair in the network, it may be necessary to re-route, after a failure, all the network OD pairs. This requires dynamic routing in the network. Using a more pure approach of fault tolerance by diversity in physical paths, while costlier, may require less sophisticated hardware and simpler management, resulting in a more viable design. The following formulation uses this latter approach.

3.3.6.2. SUCAN2 Model

This second model only considers fault tolerance by physical route diversity. Hence, it is only necessary to define the normal state $s = 0$. Besides, $\rho_m = 1$, $\forall m$. While these characteristics can be incorporated in the [P3] formulation, the following formulation has some characteristics that will prove useful in the methods discussed in the following chapter. The model is a modified version of a one due to Amiri and Pirkul (1997).

For this model the decision variables are:

$$Y_{ij}^r \begin{cases} 1 & \text{if capacity } r \text{ is selected in arc } (i, j) \\ 0 & \text{otherwise.} \end{cases}$$

X_{ij}^{mr} : flow of pair m in arc (i, j) going from i to j , assuming that the total arc flow is within range r .

W_{ij}^m : flow of pair m in arc (i,j) going from i to j .

The mathematical formulation of the model is:

[P4]

$$\text{Minimize } \sum_{ij \in E} \sum_{r \in R} FC_{ij}^r Y_{ij}^r$$

subject to

$$\sum_{\{j|(i,j) \in E\}} W_{ij}^m - \sum_{\{j|(j,i) \in E\}} W_{ji}^m = \begin{cases} d_m & \text{if } i = O(m) \\ -d_m & \text{if } i = D(m) \\ 0 & \text{otherwise; } \forall i \in V \end{cases} \quad \forall m \in K \quad (4a)$$

$$W_{ij}^m \leq \sigma_m d_m \quad \forall (i,j) \in E \quad \forall m \in K \quad (4b.e)$$

$$\sum_{j|(i,j) \in E} W_{ij}^m \leq \sigma_m d_m \quad \forall i \in V - \{O(m), D(m)\} \quad \forall m \in K \quad (4b.n)$$

$$W_{ij}^m \leq \sum_{r \in R} X_{ij}^{mr} \quad \forall (i,j) \in E \quad \forall m \in K \quad (4c)$$

$$X_{ij}^{mr} \leq d_m Y_{ij}^r \quad \forall (i,j) \in E \quad \forall m \in K \quad \forall r \in R \quad (4d)$$

$$\sum_{i=1..K} X_{ij}^{mr} \leq M_{ij}^r Y_{ij}^r \quad \forall (i,j) \in E \quad \forall r \in R \quad (4e)$$

$$\sum_{i=1..K} X_{ij}^{mr} \geq M_{ij}^{r-1} Y_{ij}^r \quad \forall (i,j) \in E \quad \forall r \in R \quad (4f)$$

$$\sum_{r \in R} Y_{ij}^r \leq 1 \quad \forall (i,j) \in E \quad (4g)$$

$$Y_{ij}^r \in \{0,1\} \quad \forall (i,j) \in E \quad \forall r \in R \quad (4h)$$

$$\begin{aligned}
 X_{ij}^{mr} &\geq 0 && \forall (i,j) \in E \\
 &&& \forall m \in K \\
 &&& \forall r \in R && (4i)
 \end{aligned}$$

$$\begin{aligned}
 W_{ij}^m &\geq 0 && \forall (i,j) \in E \\
 &&& \forall m \in K && (4j)
 \end{aligned}$$

This model adds variable X which indicates the amount of flow of pair m going through (i,j) , when it is assumed that flow is between the maximum and minimum values of the capacity of type r . This allows the formulation of restrictions (4e) and (4f) which control the total flow in each arc between a maximum and a minimum value. Restrictions (4c) oblige the existence of a value for X in some of the ranges, if there exist a non-zero value for the flow W . Restrictions (4d) oblige to select an arc in the range r (making $Y=1$) in the case some flow exist in that range for some OD pair. Similarly to the previous model, SUCAN2E is defined as the case with arc disjoint paths with restrictions (4a),(4b.e),(4c)...(4j) and SUCAN2N defines the one with arc and node disjoint paths, using restrictions (4a),(4b.e),(4b.n),(4c)...(4j).

3.3.7. Model Comparison

The very high complexity of the problems makes the optimal solution with a large number of nodes difficult to realize within a reasonable execution time. Some computer tests were performed using 10 networks of 10 nodes each, using CPLEX version 6.0.1 under AMPL, in a Digital Alpha 700 (225 MHz) machine running Digital Unix V4.0B. Tests were done using networks with different parameters. In all the tests, the optimal value found was the same for problems [P3] and [P4]. The networks used had a certain number of special OD pairs with $\sigma = 0.5$ and $\rho = 0$, and the rest had $\sigma = 1$ and $\rho = 0$. For the special pairs it was set $d_m = 2$, and for the rest $d_m = 1$. Results were also obtained for a LP relaxation of the problem. Average values obtained were the following:

Table 3.2: Comparison of Capacitated Models

Nodes	10
Arcs	38-55
OD pairs ($ K $)	10-20
Special pairs (% over $ K $)	40-100
Optimum	205.03
SUCAN LP relaxation value	117.5
SUCAN2 LP relaxation value	181.4
SUCAN CPU (sec.)	67.4
SUCAN2 CPU (sec.)	256.12

It can be observed that model SUCAN2 uses a CPU time 3.8 times higher than SUCAN to reach the optimum, due to its higher number of variables and restrictions. The main result from table 3.2, is the fact that the relaxation values, which are lower bounds of the solution, are closer to the optimum in the SUCAN2 model. This is due to the restrictions (4d), (4e) and (4f), which make the model more tight than restrictions (3c) of model SUCAN. This characteristic will be used in the relaxation algorithms described in the next chapter.

3.4. Methods for Solving the Finite Arc Capacity Fault Tolerant Network Design Problem

Given the complexity of the problem, it is very hard to obtain optimal solutions in reasonable times, for networks of moderate sizes. This chapter discusses several methods that can obtain solutions to larger instances of the capacity assignment fault tolerant network design problem. The methods are divided into two groups: Lagrangean Relaxations and Improvement Heuristics.

3.4.1. Lagrangean Relaxations

The basic idea of this method is to relax one or more of the restrictions of the linear formulation, by incorporating them into the objective function, and using penalizing constants. The optimal solution of the relaxed problem with its penalized objective is always a **lower bound** value of the solution of the original problem, for any set of positive penalizers. The objective then, is to

relax the problem in such a way the relaxed problem is easier to solve than the original, and its optimal solution can be found in a relatively short time (Ahuja, Magnanti and Orlin (1993)).

3.4.1.1. Relaxation Strategies

In the SUCAN2 model, a Lagrangean relaxation is obtained by dualizing the restriction group (4c) penalized by multipliers α_{ij}^m . This restriction is relaxed because it is the only one linking the X and W variables, and this allows to separate the relaxed problem into simpler subproblems. Amiri and Pirkul (1997) have used this strategy in a non-fault tolerant network with good results. In this way, the term $\sum_{ij \in B} \sum_{m \in K} \alpha_{ij}^m (W_{ij}^m - \sum_{r \in R} X_{ij}^{mr})$ is added to the objective function of model SUCAN2, obtaining the new relaxed problem

[PL]

$$Z_L = \text{minimize} \quad \sum_{ij \in B} \sum_{r \in R} \sum_{m \in K} (-\alpha_{ij}^m X_{ij}^{mr}) + \sum_{ij \in B} \sum_{r \in R} FC_{ij}^r Y_{ij}^r + \sum_{ij \in B} \sum_{m \in K} \alpha_{ij}^m W_{ij}^m$$

subject to (4a),(4b),(4d)..(4j).

Problem [PL] does not satisfy the integrity property, that is, if the restriction (4h), obliging variables Y to be binary, is eliminated, then the solution of [PL] is not necessarily integer, which means the Lagrangean relaxation delivers lower bounds equal or better than the direct LP relaxation of problem [P4].

In the following, two additional formulations are presented. Both are analogous to SUCAN2 and deliver good lower and upper bounds after being relaxed as problem [PL]. Consider figure 3.8:

In figure 3.8 three alternatives are shown that lead to three different problems, each one with different objectives but with the same set of restrictions (4a)...(4j). These problems are:

$$[\text{Pstep}] \text{ minimize} \quad \sum_{ij \in B} \sum_{r \in R} FC_{ij}^r Y_{ij}^r \quad \text{subject to (4a)...(4j)}$$

$$[\text{Plinear}] \text{ minimize} \quad \sum_{ij \in B} \sum_{r \in R} \sum_{m \in K} C_{ij}^r X_{ij}^{mr} + \sum_{ij \in B} \sum_{r \in R} F_{ij}^r Y_{ij}^r \quad \text{subject to (4a)...(4j)}$$

$$[\text{Pquad}] \text{ minimize} \quad \sum_{ij \in B} \sum_{r \in R} A_{ij}^r \left(\sum_{m \in K} X_{ij}^{mr} \right)^2 + \sum_{ij \in B} \sum_{r \in R} \sum_{m \in K} B_{ij}^r X_{ij}^{mr} + \sum_{ij \in B} \sum_{r \in R} D_{ij}^r Y_{ij}^r \quad \text{subject to (4a)...(4j)}$$

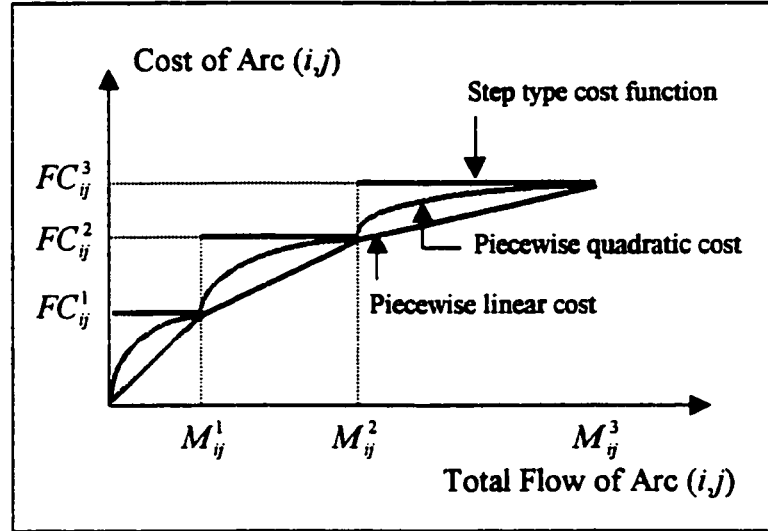


Figure 3.8. Cost functions.

These formulations correspond, respectively, to the cases with Piecewise Step cost function (Pstep), Piecewise Linear cost function (Plinear) and Piecewise Quadratic cost function (Pquad). Values for A , C , D and F are obtained in such a way of adjusting each one of the curves. For each one of the problems, relaxations similar to [PL] can be obtained, by penalizing restriction (4c) in the objective. The relaxations are then:

$$[\text{PL}_s] \quad Z_{L_s} = \text{minimize} \quad \sum_{ij \in E} \sum_{r \in R} \sum_{m \in K} (-\alpha_{ij}^m X_{ij}^{mr}) + \sum_{ij \in E} \sum_{r \in R} FC_{ij}^r Y_{ij}^r + \sum_{ij \in E} \sum_{m \in K} \alpha_{ij}^m W_{ij}^m \quad \text{subject to} \\ (4a), (4b), (4d) \dots (4j).$$

$$[\text{PL}_L] \quad Z_{L_L} = \text{minimize} \quad \sum_{ij \in E} \sum_{r \in R} \sum_{m \in K} (C_{ij}^r - \alpha_{ij}^m X_{ij}^{mr}) + \sum_{ij \in E} \sum_{r \in R} F_{ij}^r Y_{ij}^r + \sum_{ij \in E} \sum_{m \in K} \alpha_{ij}^m W_{ij}^m \quad \text{subject to} \\ (4a), (4b), (4d) \dots (4j).$$

[PL_Q]

$$Z_{L_Q} = \text{minimize} \quad \sum_{ij \in E} \sum_{r \in R} A_{ij}^r \left(\sum_{m \in K} X_{ij}^{mr} \right)^2 + \sum_{ij \in E} \sum_{r \in R} \sum_{m \in K} (B_{ij}^r - \alpha_{ij}^m X_{ij}^{mr}) + \sum_{ij \in E} \sum_{r \in R} D_{ij}^r Y_{ij}^r + \sum_{ij \in E} \sum_{m \in K} \alpha_{ij}^m W_{ij}^m \\ \text{subject to } (4a), (4b), (4d) \dots (4j).$$

Since the piecewise linear and quadratic cost functions are always under the step function, then every lower bound obtained by solving [PL_L] or [PL_Q] is also a lower bound to the original SUNDP2 problem.

3.4.1.2 Separation of the Relaxation problems

Each problem Z_L can be separated into two subproblems types [PLI] and [PLII]. Thus, for example, the relaxation [PL_L] results into:

$$[\text{PL}_{L\text{I}}] \text{ minimize } \sum_{ij \in E} \sum_{r \in R} \sum_{m \in K} (C_{ij}^r - \alpha_{ij}^m) X_{ij}^{mr} + \sum_{ij \in E} \sum_{r \in R} F_{ij}^r Y_{ij}^r \text{ subject to (4d)..(4i)}$$

$$[\text{PL}_{L\text{II}}] \text{ minimize } \sum_{ij \in E} \sum_{m \in K} \alpha_{ij}^m W_{ij}^m \text{ subject to (4a),(4b),(4j)}$$

Problem [PL_LI] can be decomposed into one subproblem for each arc ij , and so obtaining $|E|$ subproblems of type [PL_LI_i] (each one with fixed (i,j)):

$$\text{Minimize } \sum_{r \in R} \sum_{m \in K} (C_{ij}^r - \alpha_{ij}^m) X_{ij}^{mr} + \sum_{r \in R} F_{ij}^r Y_{ij}^r$$

subject to

$$X_{ij}^{mr} \leq d_m Y_{ij}^r \quad \forall m \in K \quad \forall r \in R \quad (5a)$$

$$\sum_{m \in K} X_{ij}^{mr} \leq M_{ij}^r Y_{ij}^r \quad \forall r \in R \quad (5b)$$

$$\sum_{m \in K} X_{ij}^{mr} \geq M_{ij}^{r-1} Y_{ij}^r \quad \forall r \in R \quad (5c)$$

$$\sum_{r \in R} Y_{ij}^r \leq 1 \quad (5d)$$

$$Y_{ij}^r \in \{0,1\} \quad \forall r \in R \quad (5e)$$

$$X_{ij}^{mr} \geq 0 \quad \forall m \in K \quad \forall r \in R \quad (5f)$$

In this problem, the trivial solution $Y_{ij}^r = 0, X_{ij}^{mr} = 0, \forall m \in K, \forall r \in R$ is feasible.

Similarly, problem [PL_LII] can be decomposed into k_1 subproblems [PLII₁] for every OD pair requiring only one path to route its traffic ($\sigma = 1$), and k_2 subproblems [PLII₂] for those OD pairs requiring two or more paths ($\sigma \neq 1$). These subproblems are stated as follows:

[PLII₁]

$$\text{Minimize } \sum_{ij \in E} \alpha_{ij}^m W_{ij}^m$$

subject to

$$\sum_{\{j:(i,j) \in E\}} W_{ij}^m - \sum_{\{j:(j,i) \in E\}} W_{ji}^m = \begin{cases} d_m & \text{if } i = O(m) \\ -d_m & \text{if } i = D(m) \\ 0 & \text{otherwise; } \forall i \in V \end{cases} \quad (6a)$$

$$W_{ij}^m \geq 0 \quad \forall ij \in E \quad (6b)$$

[PLII₂]

$$\text{Minimize } \sum_{ij \in E} \alpha_{ij}^m W_{ij}^m$$

subject to

$$\sum_{\{j:(i,j) \in E\}} W_{ij}^m - \sum_{\{j:(j,i) \in E\}} W_{ji}^m = \begin{cases} d_m & \text{if } i = O(m) \\ -d_m & \text{if } i = D(m) \\ 0 & \text{otherwise; } \forall i \in V \end{cases} \quad (7a)$$

$$W_{ij}^m \leq \sigma_m d_m \quad \forall (i,j) \in E \quad (7b)$$

$$\sum_{j:(i,j) \in E} W_{ij}^m \leq \sigma_m d_m \quad \forall i \in V - \{O(m), D(m)\} \quad (7c)$$

$$W_{ij}^m \geq 0 \quad \forall (i,j) \in E \quad (7d)$$

For each cost function considered, that is S, L or Q, their Lagrangean relaxations are decomposed into similar problems as stated before. Figure 3.9 illustrates the separation described.

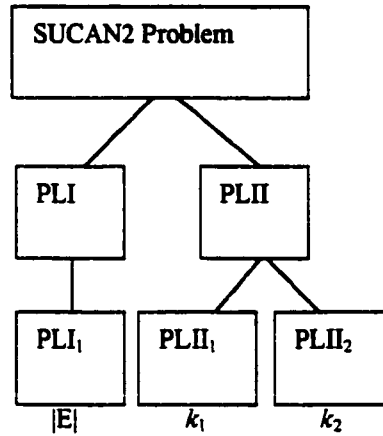


Figure 3.9. Separation of the Original Problem.

3.4.1.3. Solution to the relaxation problems

This section addresses the solution algorithms to each one of the problems resulting from the separation of the relaxed formulations [PL_S], [PL_L] and [PL_Q].

3.4.1.3.1. Solution to problems [PLI]

This problem depends on the cost function considered in each relaxation. In the piecewise linear cost function [PL_LI], the following *greedy* algorithm can be used:

Algorithm I_L

Step 0: choose an arc (i, j)

Step 1: order the OD pairs m_k in terms of decreasing values of α_{ij}^m . Pairs are re-indexed in such a way the m_k pair with the greatest α value gets index m_1 .

Step 2: For $m = 1$ to $m = |K|$, perform:

$$X_{ij}^{mr} = \begin{cases} P_1 & \text{if } C_{ij}^r - \alpha_{ij}^m \leq 0 \\ P_2 & \text{otherwise} \end{cases}$$

where

$$S_m = \sum_{k < m} X_{ij}^{kr} ,$$

$$P_1 = \min\{d_m, M_{ij}^r - S_m\},$$

$$P_2 = \min\{d_m, \max\{0, M_{ij}^{r-1} - S_m\}\}$$

$$M_{ij}^0 = 0, \forall ij$$

This procedure progressively assigns the greater possible values to variables X while satisfying restrictions (5a) and (5b), which obliges X to take lower values than its demand or the total traffic to be lower than the arc capacity r . By ordering the OD pairs into decreasing values of α , an ascending order in the X multipliers is obtained in the objective, corresponding to the values $C_{ij}^r - \alpha_{ij}^m$ which, every time they are negative, will be multiplied by the largest possible X , reducing in this way the objective. Furthermore, equation (5d) requires to select at most one range r for each arc. If this procedure is used for all possible ranges r then, for each arc (i,j) , there exists a specific range r^* which minimizes the objective

$$Z_{ij}^r = \sum_{m \in K} (C_{ij}^r - \alpha_{ij}^m) X_{ij}^{mr} + F_{ij}^r$$

If $Z_{ij}^{r^*} < 0$, then $X_{ij}^{mr} = 0$ and $Y_{ij}^r = 0$, $\forall r \neq r^*$, $\forall m \in K$ and so results $Y_{ij}^{r^*} = 1$ and $X_{ij}^{r^*}$ the value obtained in algorithm I.

If $Z_{ij}^{r^*} > 0$, then the trivial solution $X_{ij}^{mr} = 0$ and $Y_{ij}^r = 0$, $\forall m \in K$ is used.

Summarizing, problem [PL_LI] is solved by separating it into $|E|$ subproblems, one for each arc (i,j) . Each one of these subproblems is solved by iterating Algorithm I for each range r . After that, the range r^* which minimizes objective Z_{ij}^r is selected and verified if it is negative. If negative, the values of X_{ij}^{mr} obtained from Algorithm I for r^* are used and make $Y_{ij}^{r^*} = 1$, giving a zero value to the rest of the variables. Otherwise, all variables are made zero and a null capacity is assigned to all the arcs (trivial solution).

To solve [PL_SI] with a step cost function, $Z_{ij}^r = \sum_{m=1..K} (-\alpha_{ij}^m \cdot X_{ij}^{mr}) + FC_{ij}^r$ is used and then a similar procedure to the one previously described is followed, but modifying Algorithm I, by making $C_{ij}^r = 0$ in each one of the terms $C_{ij}^r - \alpha_{ij}^m$ of Step 2, which means that value P_1 is always obtained in that step.

To solve [PL_QI] a similar procedure to the one detailed in [PL_LI] is used, but Algorithm I is changed to:

Algorithm I_Q

Step 0: choose an arc (i,j)

Step 1: order the OD pairs m_k in terms of decreasing values of α_{ij}^m . Pairs are re-indexed in such a way the m_k pair with the greatest α value gets index m_1 .

Step 2: For $m = 1$ to $m = |K|$, perform:

$$X_{ij}^{mr} = \begin{cases} P_1 & \text{if } \{X_Q \leq 0\} \text{ or } \{0 < X_Q < d_m \text{ and } A_{ij}^r(P_1)^2 + P_1(2A_{ij}^r S_m + B_{ij}^r - \alpha_{ij}^m) < 0\} \text{ or} \\ & \{0 < X_Q < d_m \text{ and } M_{ij}^{r-1} - S_m \geq 0 \text{ and } M_{ij}^{r-1} - S_m > P_1\} \\ P_2 & \text{otherwise} \end{cases}$$

where

$$S_m = \sum_{k < m} X_{ij}^{kr}$$

$$X_Q = -S_m + (\alpha_{ij}^m - B_{ij}^r)/(2A_{ij}^r)$$

$$P_1 = \min\{d_m, M_{ij}^r - S_m\},$$

$$P_2 = \min\{d_m, \max\{0, M_{ij}^{r-1} - S_m\}\},$$

$$M_{ij}^0 = 0, \forall ij$$

Conditions to choose P_1 or P_2 are determined by minimizing the piecewise quadratic function for each range r , and the restrictions related to the flow X and total flow in the link. The procedure is identical to the one used in solving [PL_LI], but with the objective being

$$Z_{ij}^r = \text{minimize } A_{ij}^r \left(\sum_{m \in K} X_{ij}^{mr} \right)^2 + \sum_{m \in K} (B_{ij}^r - \alpha_{ij}^m) X_{ij}^{mr} + D_{ij}^r Y_{ij}^r$$

3.4.1.3.2. Solution to subproblems [PLII]

It has been pointed out that each problem PLII is decomposed into k_1 subproblems of type PLII₁ and k_2 subproblems of type PLII₂. PLII₁ and PLII₂ do not depend upon the cost function considered in the original problem. Problem PLII₁ for an OD pair m can be solved as a shortest path problem using the positive multipliers α_{ij}^m as positive fixed costs for each arc. For solving it a label-correcting algorithm has been used (see Ahuja *et al* 1993, pp.141).

To solve PLII₂ is not enough to solve a shortest path problem for the OD pair $O(m), D(m)$ since two or more disjoint paths has to be found minimizing the cost. This problem is solved by finding the $\lceil 1/\sigma_m \rceil$ shortest paths among origin and destination. The algorithm is as follows:

Step 1: $t := 1; \alpha'_{ij} = \alpha_{ij}^m$

Step 2: **While** $t \leq T$ **do**

- Find the shortest path C_t between origin and destination by using the label correcting algorithm and the costs α' ;
 - For every arc $(i,j) \in C_t$ and for every node $i \in C_t - \{O(m), D(m)\}$ do $\alpha'_{ij} = \infty$;
 - $t := t + 1$;
- end;**

3.4.1.4. Lagrangean Relaxation Algorithm

The previous section discussed the separation and decomposing of the relaxed problems and the procedures to solve them to optimality. In each of those, the multipliers α_{ij}^m are considered fixed. If $Z_L(\alpha)$ is the optimal value obtained in a relaxation with multipliers α , the General Relaxation Algorithm looks for an α^* such that $Z_L(\alpha^*) = \max_{\alpha} \{ Z_L(\alpha) \}$, in such a way to obtain the best possible lower bound of the solution to the original problem. In general, obtaining a set of optimum multipliers α^* is of high computational cost, so a subgradient method is used instead, giving good results though not assuring optimal multipliers. A flow diagram of the algorithm is shown in figure 3.10.

The algorithm starts by separating the original problem SUCAN2. Then the multipliers α_{ij}^m are initialized, solving afterwards each one of the subproblems as stated before. As a result, a lower bound Z_L and values for the variables X, Y, W . From the flows W , resulting from solving [PLII₁] and [PLII₂], a feasible solution to the original problem can be built, by using a heuristic, to obtain an upper bound (UB). If this upper bound is lower than the best upper bound (BUB) found so far, it replace that value and it is compared with the current lower bound Z_L . If they are equal, the optimal solution has been found. Otherwise, the ending criteria is checked. If the criteria is met, the algorithm stops. Else the α_{ij}^m multipliers are adjusted and the procedure is repeated again.

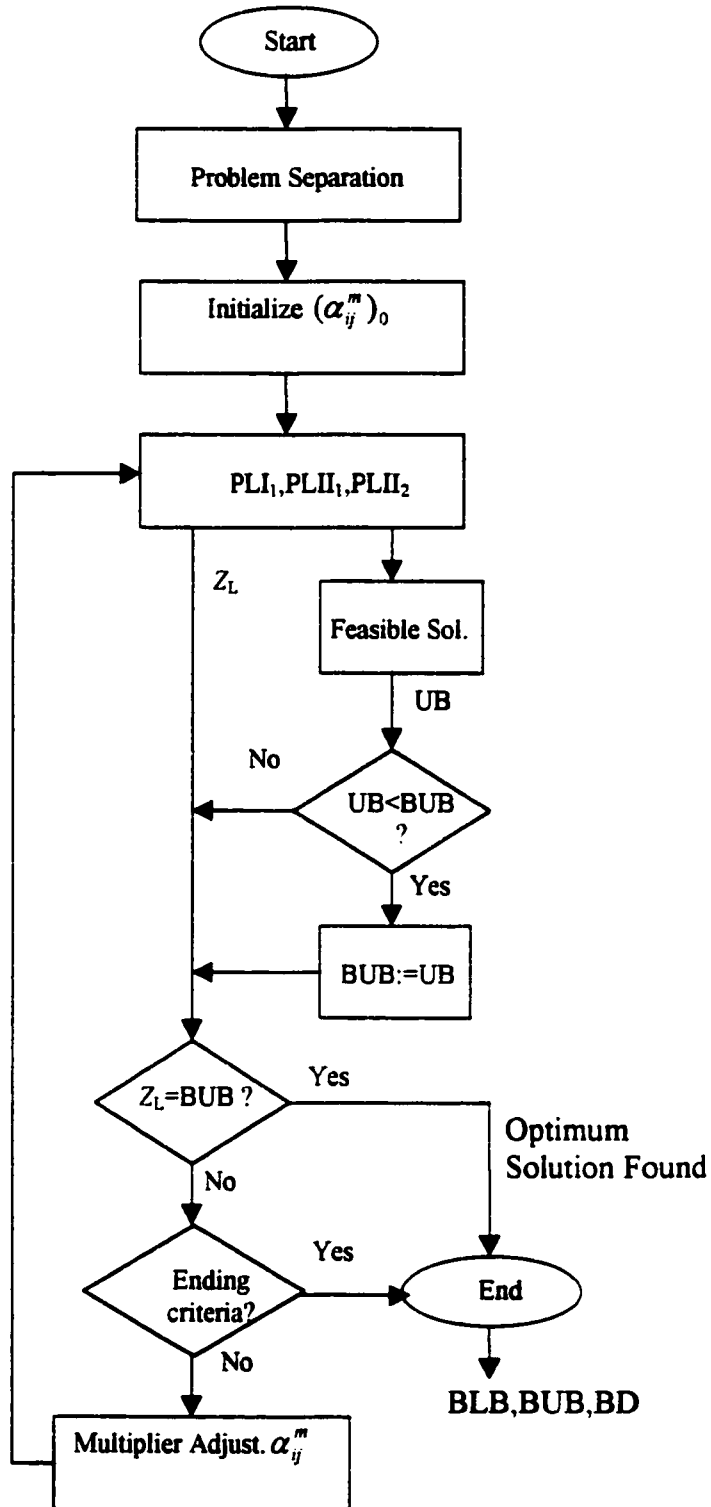


Figure 3.10. Flow diagram for the general relaxation algorithm.

Ending criteria are:

- No more than 1000 iterations
- During 50 consecutive iterations, the lower bound value Z_L does not improve more than 0.01% with respect to the previous iteration.

Finally, the algorithm delivers:

BLB (best lower bound): corresponds to the higher valued Z_L found in all the iterations.

BUB (best upper bound): corresponds to the lowest of the upper bounds (UB) found in all the iterations.

BD (best discrete bound): corresponds to the best upper bound found for the Step cost function. Evaluating the generated feasible solution (UB) in the step cost function in each iteration generates this.

BLB and BUB depend on the cost function of the original problem. However, BD is always calculated with the Step cost function.

3.4.1.5. Multiplier Adjusting Method

A subgradient method [Ahuja, Magnanti and Orlin, 1993] has been used to adjust the multipliers. The method consists in actualizing the n iteration multipliers in terms of their values on the $n-1$ iteration, using the formula:

$$(\alpha_{ij}^m)_{n+1} = (\alpha_{ij}^m)_n + t_n ((W_{ij}^m)_n - \sum_{r \in R} (X_{ij}^{mr})_n)$$

where $(W_{ij}^m)_n$, $(X_{ij}^{mr})_n$ are the values obtained upon solving the relaxation [PL] using the multipliers $(\alpha_{ij}^m)_n$ and t_n is a positive constant which determines the step size. It has been shown that $Z_L(\alpha^n) \rightarrow Z_L(\alpha^*)$ if $t_n \rightarrow 0$ and $\sum_{n=0}^{\infty} t_n \rightarrow \infty$, but the convergence rate is very sensitive to the way the step t_n is chosen. Here the following step has been used:

$$t_n = \frac{\mu_n (BUB - Z_L(\alpha^n))}{\sum_{ij \in B} \sum_{m \in K} \left((W_{ij}^m)_n - \sum_{r \in R} (X_{ij}^{mr})_n \right)^2}$$

where BUB is the best feasible solution (upper bound) found until iteration n and μ_n is a scalar greater than zero and which reduces its value while the number of iterations increase. An initial

value $\mu_0 = 2$ has been used and different strategies of reduction of μ_n have been tested. The best compromise between convergence speed and quality of the obtained bounds is reached by reducing μ_n to half each time the Z_L value does not surpasses the BLB during 25 iterations. As initial multipliers, $(\alpha_{ij}^m)_0 = F_{ij}^{r_{max}} / M_{ij}^{r_{max}}$ are used.

3.4.1.6. Heuristics for obtaining feasible solutions to the original problem from the Lagrangean Relaxation

In each iteration n of the general Lagrangean relaxation algorithm, after solving [PLII₁] and [PLII₂], flows $(W_{ij}^m)_n$ are obtained that satisfy the routing restrictions (6a),(7a),(7b) and (7c). In case the total flow in each arc is lower than the maximum range capacity, that is, $\sum_{m \in K} (W_{ij}^m)_n \leq M_{ij}^{r_{max}}, \forall ij \in E$, then a feasible solution of the original problem can be found, using those flow values (it only remains to assign the capacities). This is done through the following procedure:

Feasible Solution Generation Algorithm

Step 1: For each arc (i,j) , compute the total traffic $T_{ij} = \sum_{m \in K} W_{ij}^m$

Step 2: For each arc (i,j) with $T_{ij} > 0$, select the minimum capacity r^* which supports the total flow in the arc, that is, $M_{ij}^{r^*-1} < T_{ij} \leq M_{ij}^{r^*}$ and assign $Y_{ij}^{r^*} = 1$, and $Y_{ij}^r = 0 \forall r \neq r^*$. Furthermore make $X_{ij}^{mr^*} = W_{ij}^{r^*}, \forall m \in K$, and $X_{ij}^{mr} = 0 \forall r \neq r^*, \forall m \in K$. If $T_{ij} = 0$, then no capacity is assigned and all variables X and Y in the arc are zeroed.

3.4.1.7. Relative Positions between Different Values

After solving each one of the three relaxation problems, with the same data, different values are obtained, which are now analyzed. The typical order of those values are shown in figure 3.11.

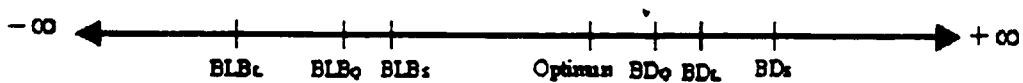


Figure 3.11. Relative position of different bounds of Lagrangean relaxation.

In that numerical scale, the optimum value for the design problem using step costs is shown. The lower bounds are found to the left of that optimum and the best discrete bounds (BD) are found to the right. As it was pointed out before, lower bounds in relaxations, with linear and quadratic costs, are also lower bounds to the original problem with step costs. Usually, the lowest bound is obtained by using piecewise linear costs (BLB_L), and to the right of this BLB_Q and BLB_S are located. To the right of the optimum, upper bound (feasible solution costs) are located. The goal is to get lower and upper bounds close together and to the optimum.

3.4.1.8. Model Parameters

In figures 3.5 and 3.6 the problem cost structure was established, and in figure 3.8 different cost functions were presented. In this section, the values of the costs parameters of the model are discussed:

If l_{ij} is the Euclidean distance between nodes j and i , and define $F_{ij}^0 = 0, \forall ij$; $M_{ij}^0 = 0, \forall ij$, then:

$$F_{ij}^{r+1} = F_{ij}^r + (C_{ij}^r - C_{ij}^{r+1})M_{ij}^r$$

$$FC_{ij}^{r+1} = FC_{ij}^r + (M_{ij}^{r+1} - M_{ij}^r)C_{ij}^{r+1}$$

In this way, piecewise linear and step cost functions parameters are determined in terms of the values of C_{ij}^r . In experimental tests, 4 capacity ranges were used ($r = 1, 2, 3, 4$) and a slope recursion C_{ij}^r similar to the one used by Amiri and Pirkul (1997) was used, with the following values:

$$C_{ij}^1 = \hat{l}_{ij}, C_{ij}^2 = 0.6C_{ij}^1, C_{ij}^3 = 0.7C_{ij}^2, C_{ij}^4 = 0.8C_{ij}^3$$

where \hat{l}_{ij} represents the savings over the Euclidean distance l_{ij} of the link illustrated in figure 3.6. For this purpose, the distance axis is divided in segments of 20 units, and a piecewise linear function is built with negative slopes. Then for every arc (i, j) verifying $\delta_{q-1} < l_{ij} \leq \delta_q$, the segment q is valid where \hat{l}_{ij} is:

$$\hat{l}_{ij} = \lambda_q(l_{ij} - \delta_{q-1}) + \beta_{q-1}$$

with

$$\beta_q = \beta_{q-1} + (\delta_q - \delta_{q-1})\lambda_q$$

$$\beta_0 = 0; \delta_0 = 0; \delta_1 = 20; \delta_2 = 40; \delta_3 = 60; \delta_4 = 80; \delta_5 = 100; \delta_6 = 120; \delta_7 = 140$$

$$\lambda_1 = 1/4; \lambda_{q+1} = 0.9\lambda_q$$

Once the parameters C, F, FC , are calculated, parameters A, B, D can be computed. These values represent the characteristic of a parabola for each segment, and they are obtained by fixing the conditions:

- (i) $FC_{ij}^r = A_{ij}^r (M_{ij}^r)^2 + B_{ij}^r M_{ij}^r + D_{ij}^r$, which is equivalent to fix the coordinates of the highest point in each parabola segment.
- (ii) $FC_{ij}^{r-1} = A_{ij}^r (M_{ij}^{r-1})^2 + B_{ij}^r M_{ij}^{r-1} + D_{ij}^r$, which is equivalent to fix the coordinates of the lowest point of each parabolic segment.
- (iii) $2A_{ij}^r M_{ij}^r + B_{ij}^r = 0$, which is equivalent to say that the derivative of each parabolic segment, in its highest point is zero.

Using (i), (ii) and (iii), then A, B, D are given by:

$$A_{ij}^r = -(FC_{ij}^r - FC_{ij}^{r-1}) / (M_{ij}^r - M_{ij}^{r-1})^2$$

$$B_{ij}^r = -2A_{ij}^r M_{ij}^r$$

$$D_{ij}^r = FC_{ij}^{r-1} - A_{ij}^r (M_{ij}^{r-1})^2 - B_{ij}^r M_{ij}^{r-1}$$

Figures 3.12 and 3.13 show the cost structure for the piecewise linear cost function:

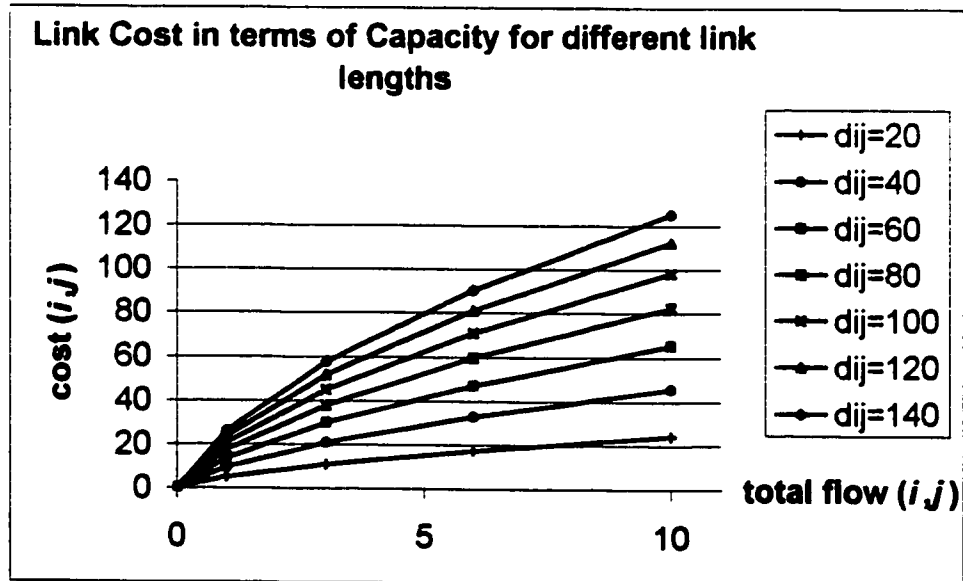


Figure 3.12. Link cost v/s total link flow

3.4.2. Heuristic Methods

This section presents some heuristics to solve the network design problem. While these methods do not guarantee an optimal solution to the problem, they provide good results in reasonable times even for very large problems. Some of these methods are based in heuristics discussed in Goldstein (1983). The proposed method consists of two parts: in the first, heuristics to search for feasible initial networks are introduced and, in the second, heuristics to improve the cost of the feasible solutions are presented.

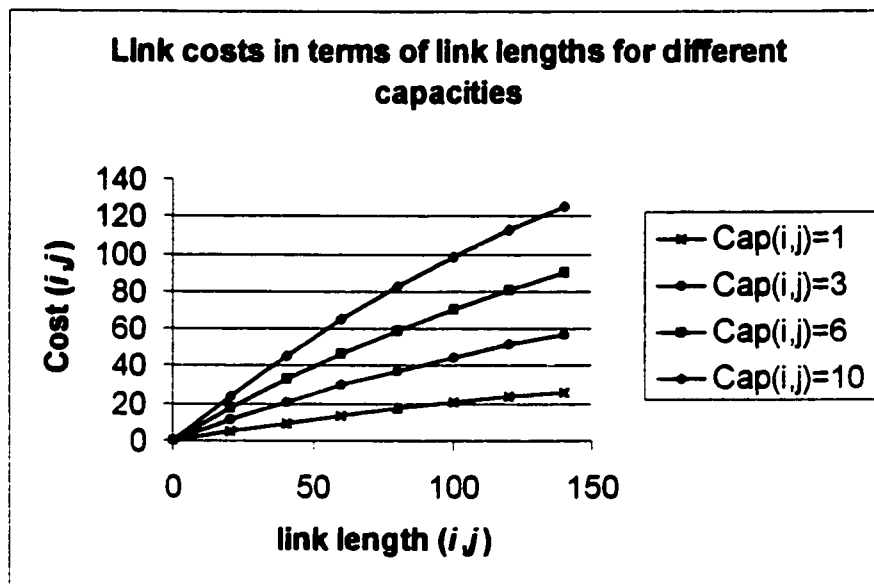


Figure 3.13. Link cost v/s link length

3.4.2.1. Heuristics for searching feasible initial networks

Three alternative heuristics are presented to build a feasible initial network.

a) Sequential routing

The method consists in ordering each one of the OD pairs m into a list, and then process each one of the demands in that order. The ordering can be done by some of the following criteria, where d_m is the demand measured in traffic units for the OD pair m , and l_m is the total distance of the shortest path between $O(m)$ and $D(m)$, considering the Euclidean distance as the arc cost.:

- i) Decreasing d_m .
- ii) Increasing l_m .

- iii) Increasing l_m/d_m .
- iv) Increasing d_m .
- v) Decreasing l_m .
- vi) Decreasing l_m/d_m .
- vii) Decreasing $l_m d_m$.
- viii) Increasing $l_m d_m$.
- ix) Random.

The sequential routing heuristic follows the following steps:

First Part

Step 1: Pairs m are ordered according to some of the listed criteria.

Step 2: For $s = 0$. The pair m in the list corresponding to the order is considered.

Using as a base the network already built in the previous iterations, the route going from $O(m)$ to $D(m)$ is selected, in such a way that it has the lowest incremental cost in adding d_m units of traffic to each arc of the route. All the demand is channeled through this route and the minimum capacities to support the flows are recalculated. For all those OD pairs verifying $\sigma_m < 1$, a search is done for the group of disjoint routes (arc and node disjoint) which minimize the incremental cost. This step is repeated for the next pair m in the list.

If the problem only considers fault tolerance with route diversity, then these are the only necessary steps. For the problem considering fault tolerance with capacity reservation the following additional steps must be performed:

Second Part

Step 3: For each arc $(i,j) \in E$, calculate the *Totalflow* number and for each node $n \in V$ calculate $(Total_input_flow + Total_output_flow)$ number. The demands channeled in step 2 are taken out, maintaining the assigned capacities and total costs.

Step 4: Order the failure states $s \neq 0$ according to some criteria. It can be done in terms of the increasing order of the numbers calculated in step 3. It can be considered two blocks of data, with the first representing all the possible arc failures and the second with the node failures.

Step 5: For each one of states s , the following is done. The next state is taken, according to the order established in step 4, and the demands, characterized by $\rho_m > 0$ and not having origin nor destination in faulted nodes, are routed sequentially. This routing is done by making sure the faulted element of state s is not used and procuring the least possible incremental cost. The capacities are updated to accommodate the flows for each one of the previously processed states s (including $s = 0$) and the current state s ; the total costs are recalculated. The routing sequence is done by the order of one of the criteria (i)...(ix), taking into account that, instead of the demand d_m , it must be used $\rho_m d_m$. For the first routed demands, it may be possible that several zero incremental cost routes exist, so in those cases the route with the least number of arcs is selected. Finally, all the demands channeled for the current state s are taken out, maintaining the capacities already assigned and their costs, and the step is repeated with the next state s . The final network is the one with the arc capacities given by the last iteration of this step.

If in the design characteristics, it is considered that ρ_m is not much larger than $(1-\sigma_m)$ and there is a reduced number of OD pairs with $\rho_m > 0$, then the second part of the heuristic just adds a small (or even null) percentage to the total cost.

Computational experience showed that the problem is much more sensitive to step 2, because if an OD route is found having excess capacity in all its arcs, the demand will be routed through that path, and will have a null incremental cost

The decision regarding which ordering criteria to use is not obvious. If, for example, the order (i) is selected, the heuristic tries first to minimize the costs of routing the high traffic demands, and later takes charge of lower traffic demands, which can be routed using the excess capacity of the network. In this way, the order (i) is more convenient when the network has some demand that are much higher than the rest, such as concentrator traffic. Experimental results showed that criteria (i), (ii) and (iii) gave better results, and so they were selected for the computational experience.

One of the problems of this heuristic corresponds to the search, in each step, of the lowest incremental cost route. For instance, in a 30 node network, having each node with n incoming arcs, there may exist routes from one arc (when a direct arc exist) to $n-1$ arcs, which total a number of routes in the order of n^{29} . For example if $n = 5$, the total number of routes can be up to $5^{29} = 1.8E20$, which is computationally impossible to manage. In practice, the lowest incremental cost was searched for all those routes with less than 8 arcs in the path.

b) Shortest Route Routing

This procedure consists in adding to the network the traffic of each pair m , through the shortest route joining the origin with the destination, considering as the arc cost the Euclidean distance. The heuristic performs the following steps:

Step 1: The state $s = 0$ is considered, and the minimum cost route is found for all the OD pairs requiring of only one path to route their traffic. For all those pairs with $\sigma < 1$, the traffic is channeled through the shortest group of disjoint paths.

Step 2: For each arc, the minimum arc capacity supporting the flow is selected.

Similarly to the previous heuristic, if some OD pairs have $\rho_m > 0$, a second part is performed following the steps 3, 4 and 5 of the sequential routing heuristic.

This heuristic obliges to find the shortest paths for all OD pairs. This can be done by using an algorithm of the type "all pairs shortest path". Besides, a group of minimal paths have to be found for those pairs requiring them through their diversification parameter.

c) Hierarchical Trees

This heuristic uses the following steps:

Step 1: For each node i calculate the *total flow in the node* $= fm = \sum d_m$ such that $O(m) = i$, or $D(m) = i$. The nodes are ordered in a list L by their decreasing order on fm . The list is divided into blocks $L_1, L_2..L_M$, where L_1 corresponds to the first t_1 elements of L , L_2 corresponds to the following t_2 elements, etc..

Step 2: A minimum spanning tree T_1 is built over the L_1 elements of list L , considering the arc Euclidean distance as a cost.

Step 3: For $i = 2 \dots M$, the minimum spanning trees T_i are built, covering the nodes present in $L_1 \cup L_2 \cup L_3 \dots L_i$, considering with null cost those arcs present in tree

T_{i-1} . At the end of this process, a tree covering all the network nodes is obtained.

Step 4: For each OD pair, a single route exists in the tree. Each pair requiring only one path is routed through the tree route. For those pairs requiring disjoint paths, the first route is channeled through the tree, and the rest is done through the set of routes of lowest incremental cost.

Similarly to the previous heuristic, if some OD pairs have $\rho_m > 0$, a second part is performed following the steps 3, 4 and 5 of the sequential routing heuristic. In practice, the heuristic was implemented separating the nodes into three lists $L_1 \dots L_3$, with L_1 representing 30% of nodes with higher f_m , L_2 the 40% and L_3 the remaining 30% of nodes.

3.4.2.2. Improvement Heuristics

This section describes different procedures to reduce the total cost of a feasible initial network. For those designs using reservation capacity fault tolerance, that is $\rho_m > 0$ for some OD pair m , it is assumed that the flow in each arc for each pair m in each state s is known (defined by f_{ij}^{ms}).

a) Sequential Improvement

Similarly to the Sequential Routing Heuristic, the idea is to follow a certain order in the processing of the demands. The steps are:

Step 1: It is defined the *original_cost* as the feasible cost of the network before applying the improvement heuristic. The demands are ordered following some of the criteria (i)...(ix) of section 3.4.2.1.a.

Step 2: This step is performed for the state $s = 0$. Following the order of step 1, define as $cap_{a_{ij}}$ the present capacities in each arc of the network. The present cost c_a of the network is calculated, and then the demand d_m of all those arcs with flow (those arcs ij where $f_{ij}^{m0} > 0$) is taken out. The new minimum capacities supporting the total flow for state $s = 0$ are re-calculated. The demand d_m is re-routed in such a way to have the lowest incremental cost (taking care of finding as many paths as required by the parameter σ_m). The minimum capacities $cap_{n_{ij}}$ are re-calculated supporting all the network flows. If this

procedure indeed produces a decreasing in the costs, then $cap_{n_{ij}} \neq cap_{a_{ij}}$ for some $(i,j) \in E$.

Step 3: If the network has some OD pairs with $\rho_m > 0$, then steps 3, 4 and 5 of the second part of heuristic described in section 3.4.2.1.a. are followed

Step 4: If new_cost is defined as the one delivered by the heuristic so far, the changes are performed only if $new_cost < original_cost$.

b) Arc Deleting

This heuristic is based on deleting the more expensive arcs, and it can be described as:

Step 1: For each arc (i,j) , calculate the merit figure $arc_cost(i,j)/total_flow(i,j)$. A decreasing list using the merit figure is built.

Step 2: The first arc is taken from the list and the demands flowing through that arc are identified. All those demands and the arc are taken out from the network. The minimum capacities and total cost are re-calculated. All the retired demands are re-routed through the shortest path (without using the retired arc). The minimum capacities and total cost are re-calculated. If the total cost decreases, the heuristic goes back to step 1, otherwise the network previous to the retirement of the arc is restated and a new arc is tested.

For those networks with reservation capacity fault tolerance, steps 3 and 4 of the previous heuristic are performed. This heuristic concentrates in the elimination of those arcs that are more expensive, considering the amount of traffic they transport.

3.5. Computational Experiments

This section presents computer results obtained in the design of fault tolerant networks that use physical route diversity.

Two types of networks were generated. The first one corresponds to a general type of network, where most of the parameters are selected at random, identified as GEN. The second corresponds to hierarchical networks and it is identified as JER. For both families, there have been considered that some special OD pairs exist, which require fault protection through traffic routing using two or more node and arc disjoint paths ($\sigma_m = 0.5$). The rest of the OD pairs m are

considered normal and have $\sigma_m = 0$. For all pairs $m \in K$, $\rho_m = 0$. Ten different networks were generated, with 10, 15, 20 and 30 nodes.

3.5.1. Generation of GEN networks

First the node coordinates are assigned. This is done by randomly placing the nodes in a 100x100 point plane, for example. Each coordinate is chosen through a random number uniformly distributed in the range. The set of possible arcs E is generated by using the *arc_density* parameter, which defines the probability of having an *arc* between a given node pair. As the number of *arcs* grows as the square of the number of nodes, for the networks of 20 and 30 nodes, *arcs* has been generated by using a lower *arc* density and by limiting the number of *arcs* going into a given node to a number *max_arcs*. OD pairs m have been generated by choosing randomly the origin and destination (for different nodes) among the network nodes. They have been generated $|K|$ different OD pairs for each network. Capacities can be assigned over a total of 4 available ranges. In order to compare the results with those published in the literature, it has been used a demand $d_m = 1$ for all the normal pairs and demand $d_m=2$ for the special pairs. Table 3.3 shows the characteristics of the generated networks:

Table 3.3: Characteristics of GEN networks

	GEN10	GEN15	GEN20	GEN30
<i>number of nodes</i>	10	15	20	30
<i>Arc_density</i>	0.5	0.5	0.4	0.4
<i>max_arcs</i>	-	-	5	5
<i>number of arcs</i>	38-52	58-94	88-98	134-142
<i>Special pairs (% over K)</i>	40%	40%	40%	40%
<i>Number of pairs (K)</i>	10	20	30	40
<i>Special pair d_m</i>	2	2	2	2
<i>Normal pair d_m</i>	1	1	1	1

3.5.2. Generation of JER networks

Two hierarchical levels were considered for the nodes. The first corresponds to gateway nodes (or traffic concentrators) with a high traffic demand. These are the most important nodes in

the network and they are considered as special nodes, which require fault protection, and have $\sigma_m = 0.5$. The second hierarchical level correspond to the hub category. Hubs are connected to a single gateway. A number of hubs close to a gateway form a group and each group connects to a single gateway. Each pair m of type (hub,gateway) is defined as normal with $\sigma_m=0$. For each communication pair m its traffic density d_m is chosen randomly. For those pairs of type (hub,gateway) it is chosen between $d_m = 1$ or 2, while for the pairs (gateway,gateway) it is chosen between $d_m = 4$ or 6. The set of arcs E of the network is chosen considering those linking hubs with their gateways, and gateways among themselves. Coordinates of each node is chosen randomly in a 100x100 points graph (similarly to GEN). Figure 3.14 shows a typical scheme of a hierarchical network.

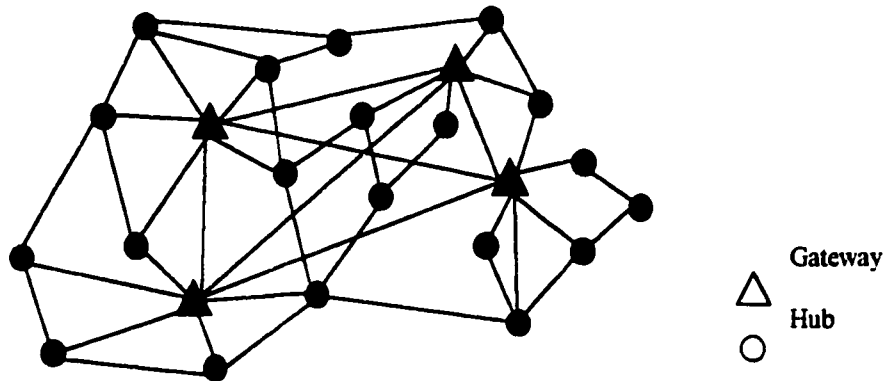


Figure 3.14. Typical topology for a hierarchical network

Table 3.4 shows the characteristics of the JER hierarchical generated networks.

Table 3.4: Characteristics of the JER networks

	JER10	JER15	JER20	JER30
<i>number of nodes</i>	10	15	20	30
<i>number of arcs</i>	46-52	66-72	92-98	136-146
<i>number of gateways</i>	2	3	4	5
<i>number of hubs per gateway</i>	4	4	4	5
<i>Special pairs</i>	1	3	6	10
<i>Number of pairs (K)</i>	9	15	22	35
<i>Special pair d_m</i>	4 or 6	4 or 6	4 or 6	4 or 6
<i>Normal pair d_m</i>	1 or 2	1 or 2	1 or 2	1 or 2

3.5.3. Available capacity ranges

Four capacity ranges were selected ($|R| = 4$). These are shown in Table 3.5:

Table 3.5: Available ranges for the capacity assignment

Network type	$r = 1$	$r = 2$	$r = 3$	$r = 4$
GEN10 or JER10	1	3	6	10
GEN15 or JER15	1	3	6	15
GEN20 or JER20	1	4	9	20
GEN30 or JER30	1	4	9	30

In finding the solution, in very few cases was the maximum available capacity used. For the OD pair traffic generated, the solution algorithms placed the traffic in various channels of smaller capacities. In general it was found that the quality of the relaxation techniques and heuristics worsens when a greater number of available capacities ranges is used, as well as when the capacity range step is too high. It is convenient to fix the first capacity range as a unit of traffic, because the algorithms find better results.

3.5.4. Lagrangean relaxations using different cost functions

All the tests were performed by using MATLAB version 5.1.0.421 running in a Digital DEC 433 (64MB RAM and 433 MHz) under Digital Unix V4.0B.

The objective of this section is the application of the Lagrangean relaxation for the different cost functions and discuss their advantages. The best lower bound (BLB) and upper bound (BUB) are obtained using different costs, according to the relaxation. The best discrete bound (BD) is obtained as the best feasible solution found using discrete costs. Clearly, for step cost functions, it follows that $BUB=BD$.

Relaxations were applied, using each one of the cost functions considered (step, quadratic and linear), to a group of networks. Result are shown in Table 3.6, which are the average results for 10 networks:

Table 3.6: Results for different relaxations.

	GEN10	GEN15
BD_S (step)	173.0	340.8
BD_Q (quadratic)	172.2	333.9
BD_L (linear)	171.8	338.9
Optimum	169.7	-
BLB_S	150.4	284.7
BLB_Q	150.1	280.6
BLB_L	142.3	271.9
N_S (Number of Iterations)	263	293
N_Q	267	270
N_L	258	255
CPU_S (seconds)	117	779
CPU_Q (seconds)	186	735
CPU_L (seconds)	118	677
GAP_S to optimum %	1.88	-
GAP_Q to optimum %	1.64	-
GAP_L to optimum %	1.28	-
GAP_S to BLB %	14.86	19.75
GAP_Q to BLB %	14.93	19.18
GAP_L to BLB %	20.65	24.85

In table 3.6, N represents the number of iterations for the relaxation convergence. GAP to optimum represents the percentage deviation from the optimum calculated as $(BD - \text{Optimum}) / \text{Optimum}$. Similarly, GAP to BLB is the ratio $(BD - BLB) / BLB$ for each case.

For networks of 10 nodes, the relaxations resulted in $BD = \text{Optimum}$ in 6 out of 10 tested networks, using piecewise quadratic costs. While linear cost relaxation gives good results for 10 and 15 nodes networks, the lower bounds are poor (which is due to the cost nature). As a conclusion, the step and quadratic cost function relaxations bring similar results. However, the

best results in terms of obtaining a good compromise between minimum BD, maximum BLB and GAP to minimum BLB, is obtained through the use of a piecewise quadratic cost function, and the rest of the tests were done by using that cost function.

3.5.5. Preliminary results

Each one of the heuristics discussed in section 4 were tested. Results for GEN10 are presented in Table 3.7:

Table 3.7: Costs for different network generation heuristics

GEN10	Total Cost (CT)	(CT-Optimum)/Optimum %
Sequential Routing ($d_m \downarrow$)	178.2	5.06
Sequential Routing ($l_m \uparrow$)	176.3	3.95
Sequential Routing ($l_m/d_m \uparrow$)	179.5	5.83
Minimal Routes	181.1	6.89
Hierarchical Trees	221.4	31.0

\uparrow : increasing; \downarrow : decreasing.

From these results it was decided to discard the Hierarchical Trees heuristic, due to its high cost. Next, improvement heuristics were used in networks with different number of nodes. Several improvement combination were tested for 20 and 30 node networks.

It was also tested the use of two improvement heuristics in sequence. The first improves the feasible initial network, while the second improves the network resulting from the first heuristic. Table 3.8 shows the values corresponding to the percentage of improvement in the total cost relative to the initial cost:

Table 3.8: Percentage decreasing of Total Cost by using different improvement heuristics

First Heuristic.	Second Heuristic	Number of Network Tested	Total Cost Improvement after first heuristic (%)	Total Cost Improvement after second heuristic (%)
MS ($d_m \downarrow$)	MS ($l_m \uparrow$)	50	5.27	1.1
MS ($d_m \downarrow$)	MS ($l_m/d_m \uparrow$)	50	5.27	0.58
MS ($l_m \uparrow$)	MS ($l_m \uparrow$)	40	3.88	1.39
MS ($l_m \uparrow$)	MS ($l_m/d_m \uparrow$)	40	3.88	0.48
MS ($l_m/d_m \uparrow$)	MS ($l_m \uparrow$)	60	3.23	0.41
EA	MS ($l_m \uparrow$)	60	6.66	1.80
EA	MS ($l_m/d_m \uparrow$)	50	6.66	1.78

MS: Sequential Improvement; EA: Arc Elimination

From these preliminary results, the conclusions are the following:

- The best improvement strategy corresponds to Arc Elimination (EA) followed with Sequential Improvement (MS).
- The order in which the best results were obtained corresponds to decreasing d_m , increasing l_m and increasing l_m/d_m .
- It was found that the initial networks were more sensitive to the first improvement heuristic than to the second heuristic.
- On applying Sequential Improvement as a second heuristic, better results do tend to depend on increasing l_m .
- Applying a third improvement heuristic does not result in any significant solution improvement.

3.5.6. General Results

This section presents the results obtained by combining the different methods discussed in section 4. For each network, the cost obtained by using an initial network generation algorithm and a second value, corresponding to the costs obtained after applying some improvement heuristics, are presented. The improvement strategies used were:

a) Initial Network \rightarrow Arc Elimination \rightarrow Sequential Improvement (increasing l_m).

- b) Initial Network \rightarrow Sequential Improvement (decreasing d_m) \rightarrow Sequential Improvement (increasing l_m).
- c) Initial Network \rightarrow Sequential Improvement (increasing l_m) \rightarrow Sequential Improvement (increasing l_m).
- d) Initial Network \rightarrow Sequential Improvement (increasing l_m/d_m) \rightarrow Sequential Improvement (increasing l_m).

For each network, the final cost assigned by the improvement strategy delivering the best result was considered. The results are shown in Table 3.9 and 3.10.

Table 3.9: Results for different improvement methods (GEN networks)

	Method	GEN10	GEN15	GEN20	GEN30
Initial Total Cost	Relaxation Q	172.19	333.93	551.88	870.45
	ES (Decreasing d_m)	178.16	363.85	625.97	981.79
	ES (Increasing l_m)	176.33	360.13	628.15	987.91
	ES (Increasing l_m/d_m)	176.03	358.03	632.43	994.81
	RM	181.14	327.92	631.16	981.71
Improved Total Cost	Relaxation Q	170.12	330.54	535.50	832.72
	ES (Decreasing d_m)	173.51	348.07	557.97	893.24
	ES (Increasing l_m)	174.95	348.92	566.10	900.29
	ES (Increasing l_m/d_m)	174.11	344.82	572.53	903.54
	RM	173.68	333.68	552.09	856.42
CPU(sec)	Relaxation Q	186.25	734.75	1827.50	7101.82
	ES (Decreasing d_m)	3.09	8.50	13.83	35.37
	ES (Increasing l_m)	2.95	8.54	12.88	36.32
	ES (Increasing l_m/d_m)	3.03	8.60	13.45	35.12
	RM	0.10	0.25	0.58	4.42
	BLB RelaxationQ	150.06	280.59	442.08	663.78
	Optimum	169.66	-	-	-
Initial GAP to BLB	Relaxation Q	14.93	19.18	27.73	34.72
	ES (Decreasing d_m)	18.78	29.90	41.46	48.10
	ES (Increasing l_m)	17.54	28.94	42.26	48.83
	ES (Increasing l_m/d_m)	17.31	27.60	43.06	49.87
	RM	20.88	27.95	42.82	48.11
Improved GAP to BLB	Relaxation Q	13.40	18.03	21.35	25.62
	ES (Decreasing d_m)	15.53	23.76	26.80	34.59
	ES (Increasing l_m)	16.53	24.86	28.76	35.63
	ES (Increasing l_m/d_m)	16.03	22.89	29.51	36.12
	RM	15.66	18.99	25.45	29.17

ES: Sequential Routing; RM: Minimal Routes; RelaxationQ: Piecewise quadratic cost function relaxation.

Table 3.10: Results for different improvement methods (JER Networks)

	Method	JER10	JER15	JER20	JER30
Initial Total Cost	Relaxation Q	138.60	291.05	475.33	775.30
	ES (Decreasing dm)	143.35	293.95	478.16	772.94
	ES (Increasing Lm)	142.88	295.55	477.99	787.83
	ES (Increasing Lm/dm)	142.57	297.58	473.10	835.86
	RM	143.39	306.82	502.92	824.66
Improved Total Cost	Relaxation Q	138.14	287.40	433.84	704.44
	ES (Decreasing dm)	139.11	291.40	452.55	750.08
	ES (Increasing Lm)	139.28	291.79	453.99	751.38
	ES (Increasing Lm/dm)	139.53	287.56	453.02	772.96
	RM	139.55	287.10	435.36	715.26
CPU(sec)	Relaxation Q	196.07	500.60	1321.44	8735.44
	ES (Decreasing dm)	5.40	11.42	16.20	52.00
	ES (Increasing Lm)	5.34	11.63	16.11	51.37
	ES (Increasing Lm/dm)	5.28	11.76	16.01	51.04
	RM	0.08	0.12	0.26	4.84
	BLB RelaxationQ	115.28	235.26	353.52	506.32
	Optimum	133.96			
Initial GAP to BLB	Relaxation Q	20.17	23.69	34.27	53.09
	ES (Decreasing dm)	24.45	24.95	34.81	52.30
	ES (Increasing Lm)	23.94	25.63	35.21	55.60
	ES (Increasing Lm/dm)	23.37	26.46	33.77	63.84
	RM	24.34	30.43	42.32	62.28
Improved GAP to BLB	Relaxation Q	19.77	22.19	23.02	38.72
	ES (Decreasing dm)	20.57	23.88	28.33	47.71
	ES (Increasing Lm)	20.82	24.03	28.42	48.40
	ES (Increasing Lm/dm)	20.96	22.22	28.51	51.87
	RM	20.97	22.02	23.49	40.85

ES:

Sequential Routing; RM: Minimal Routes; RelaxationQ: Piecewise quadratic cost function relaxation.

From these results, it can be concluded that:

- The proposed relaxation method delivers the best results, with a percentile deviation (GAP) to the lower bound varying between 15% and 35% for general networks and between 20 and 50% for hierarchical networks. Results compare favorably with values up to 50% obtained in Dahl and Stoer (1998), and Alevras et al (1997).
- For 10 nodes networks, the optimum is at a 13.1% from BLB for GEN networks and 16.2% for JER networks. From these percentages it can be estimated that the deviation to the optimum, obtained through relaxations, is around a 16% for 30

nodes GEN initial networks, and up to 11% for the improved networks. The estimated values are respectively 32% and 20% for hierarchical networks.

- For larger networks, the improvement stage is quite important, delivering a reduction of up to 22% in the GAP.
- While using Sequential Routing, the total cost solution does not depend significantly from the type of ordering used. This heuristic has proved to be, generally, superior to the one of Minimal Routes. However, if the cost after the improvement stage is considered, the latter method is clearly superior to the former. An explanation is that RM finds an initial solution routing the traffic through a greater number of arcs than ES (this explains the greater initial cost), which allows a larger number of alternatives at the improvement stage. It can be studied if this situation is maintained when there is a greater variance in the traffics d_m of different OD pairs.
- On the average, the quality of solutions obtained through relaxation worsens when the variance of the demands d_m increase.

Figure 3.15 shows differences of the costs obtained through each method for a 30 node GEN network.

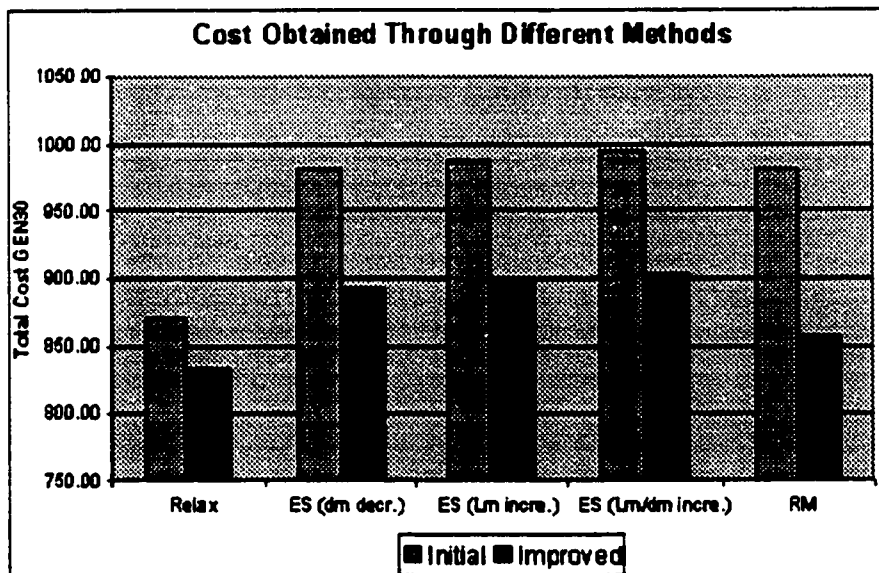


Figure 3.15. Total cost for different solution method for GEN30.

In Figure 3.15, it is observed that the relaxation method delivers initial networks of lower costs. Furthermore, it is apparent the importance of the improvement stage on the final solution.

3.5.7. Problems with the Lagrangean Relaxation Method

The computed GAP, between the best feasible solution found in all iterations for step costs (BD) and the best lower bound (BLB), is due to the weakness of the lower bounds. Figure 3.16 shows that GAP for a number of special pairs (with fault tolerance $\sigma_m = 0.5$). The GAP is computed as $GAP = (BD - BLB) / BLB \%$.

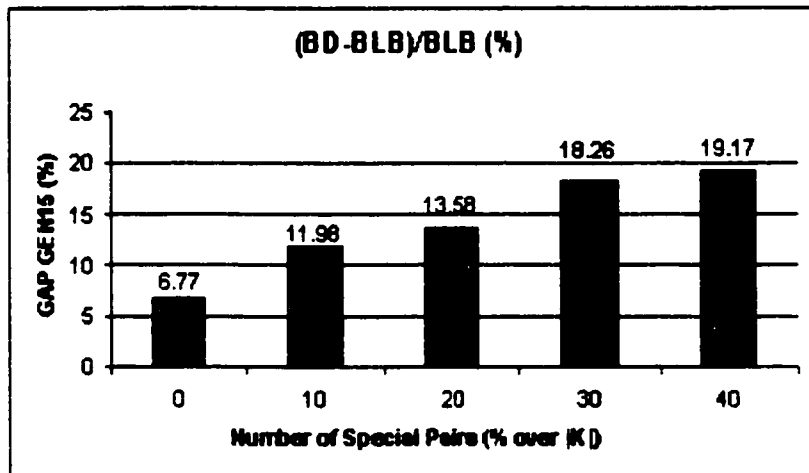


Figure 3.16. GAP between BD and BLB for different number of special pairs

Figure 3.16 shows the deviation close to 7% for networks without fault tolerance, result similar to the one obtained by Amiri and Pirkul (1997). The GAP almost doubles when it is required that the 10% of OD pairs have a 50% fault tolerance.

3.5.8. Examples

Next figures show some examples of networks obtained through the different methods discussed. In the figures CT denotes the total cost. For the 10 node GEN network, OD pairs 8-10 and 4-9 require bi-connectivity, while OD pairs 5-6, 1-4, 3-7, 1-2, 2-3 and 4-7 do not.

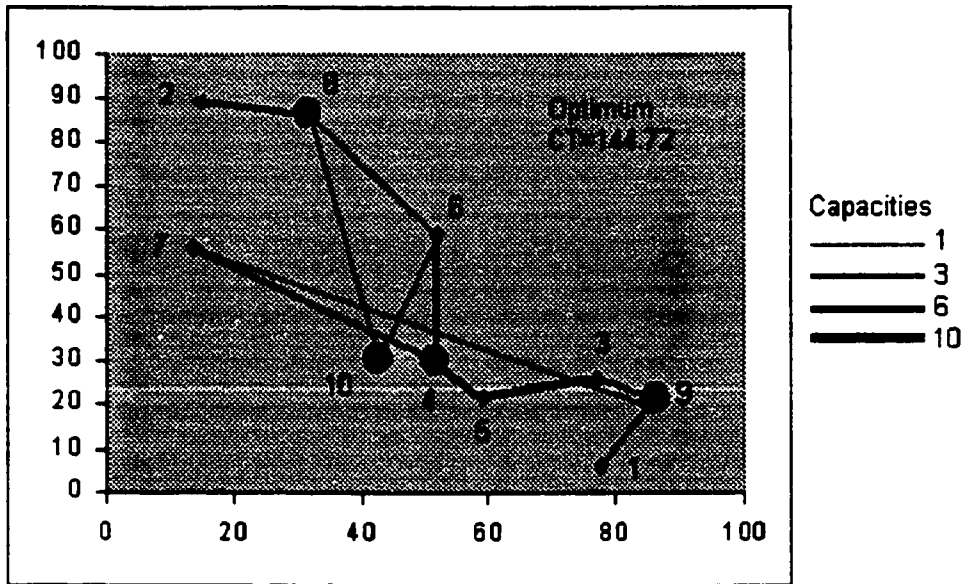


Figure 3.17. Optimum solution for a 10 node GEN network example.

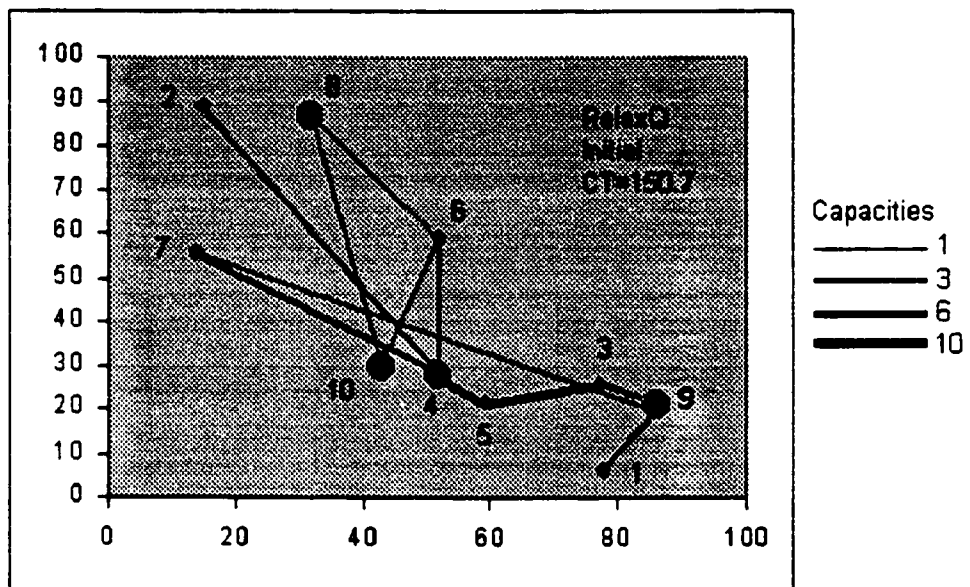


Figure 3.18. Initial relaxation.

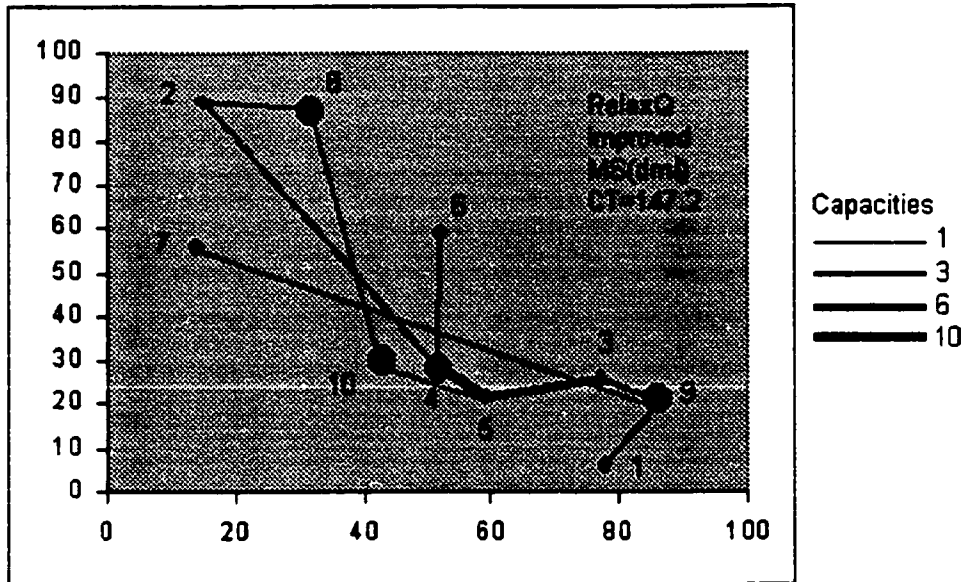


Figure 3.19. Improvement: first phase.

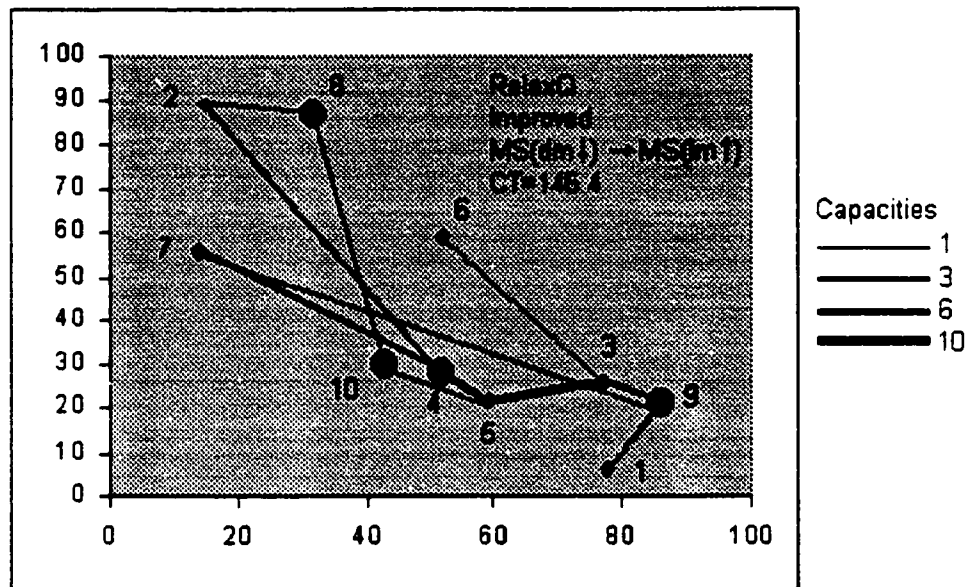


Figure 3.20. Improvement: second phase

For the 15 node GEN network, OD pairs requiring bi-connectivity are 4-9, 4-11, 1-14, and 3-9. OD pairs that do not require bi-connectivity are: 3-15, 7-8, 3-7, 5-12, 3-11, 7-9, 8-14, 1-11, 1-7, 8-13, 8-10 and 9-10.

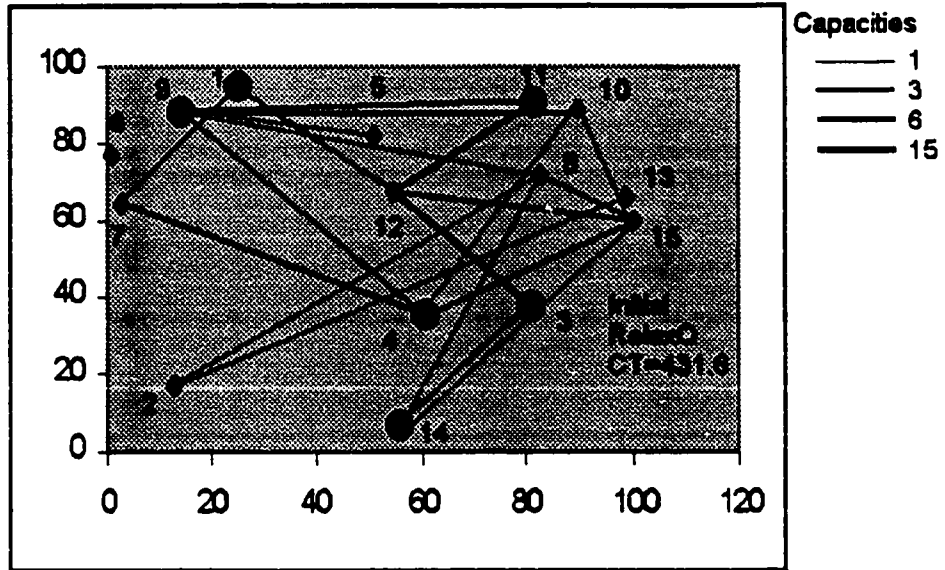


Figure 3.21. Initial solution for a 15 node GEN network example.

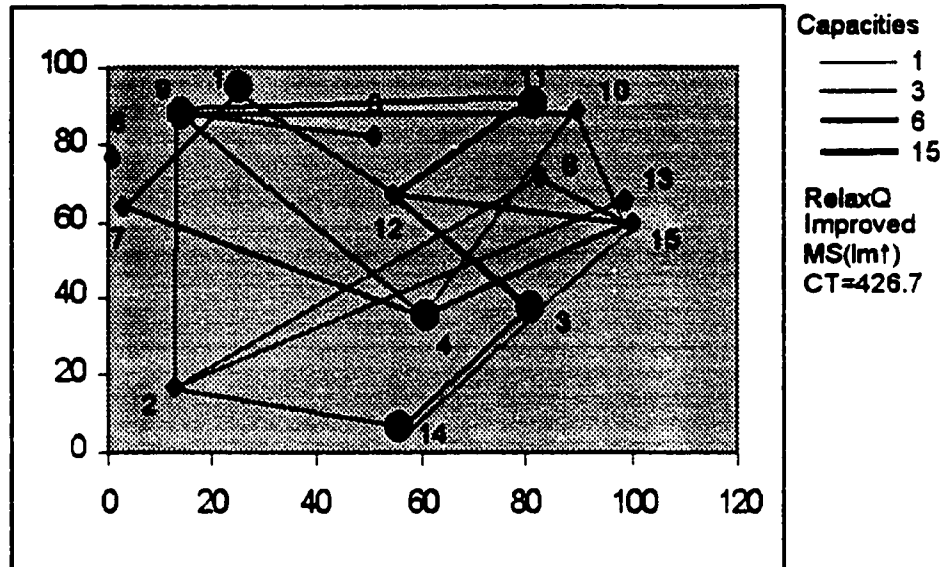


Figure 3.22. Improved solution.

For the 20 node GEN network, OD pairs requiring bi-connectivity are: 3-5, 4-17, 3-7, 1-12, 8-12 and 11-13. OD pairs that do not require it are: 8-9, 12-14, 9-17, 16-20, 10-20, 3-20, 8-11, 3-8, 7-20, 1-7, 18-20, 15-19, 10-15, 13-16, 1-5, 2-7, 3-11 and 12-18.

Finally, for the 15 node JER network, concentrators are located at nodes 9, 11 and 15. The traffic carried between OD pairs is the following:

11-15→6 9-15→4 9-11→6 2-15→2 3-15→1 13-15→1

14-15→2

8-11→2

1-11→1

6-11→2

11-12→1

4-9→1

7-9→1

5-9→1

9-10→2

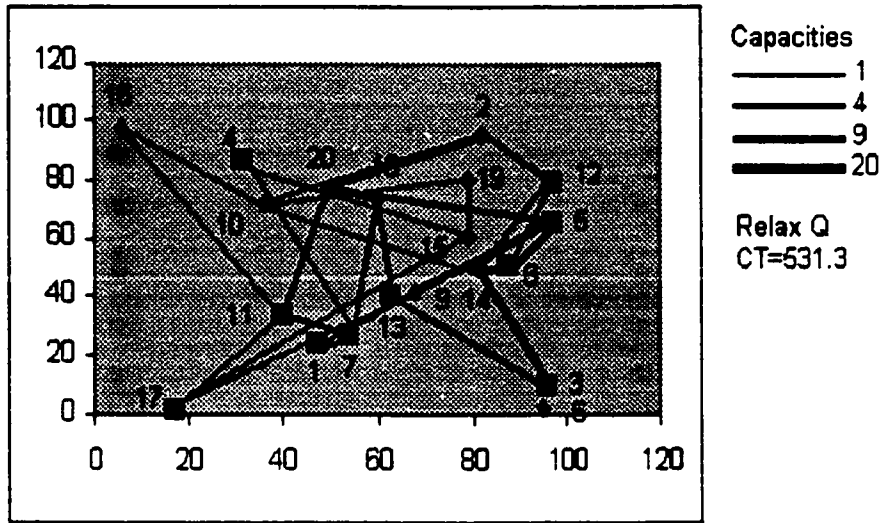


Figure 3.23 Initial solution for a 20 node GEN network example.

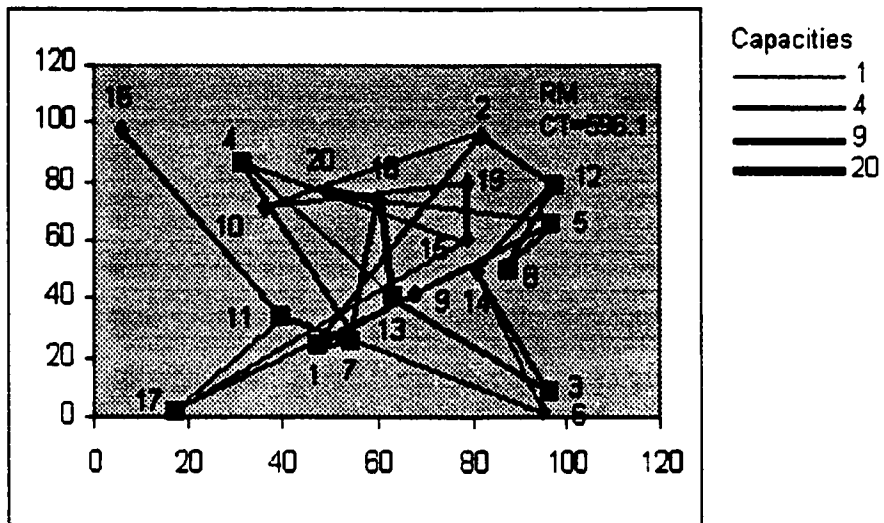


Figure 3.24. Initial solution for a 20 node GEN network example.

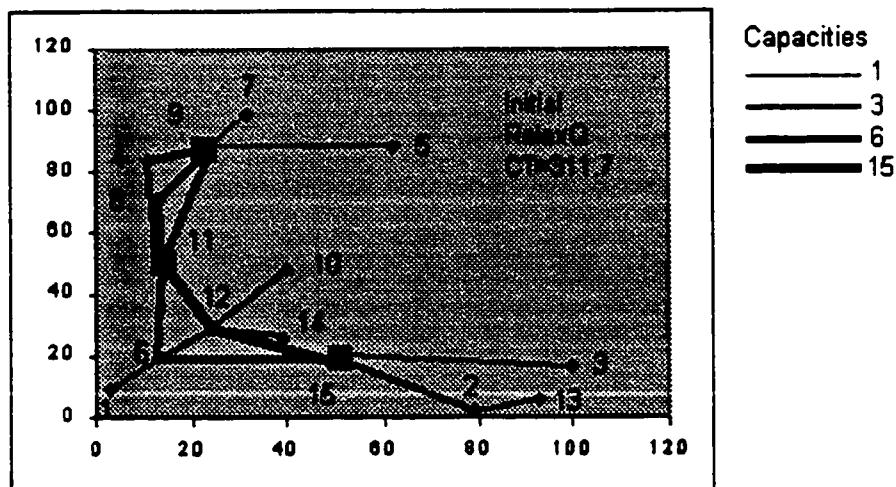


Figure 3.25. Initial solution for a 15 node JER network example.

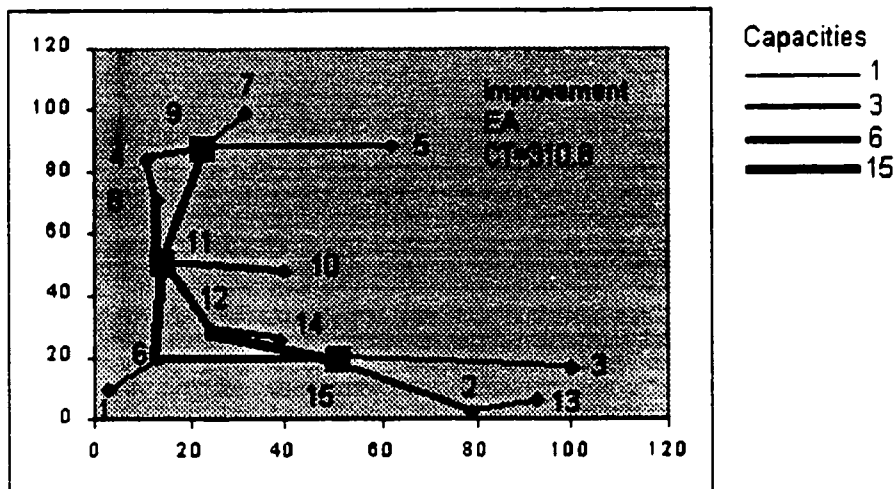


Figure 3.26. Improved solution.

3.6 Conclusions

This chapter has presented a complete coverage of the Fault Tolerant Network Design Problem. It was mentioned the importance of giving fault tolerance to the main traffic of the network. Two fault tolerance strategies have been discussed: physical route diversity and reservation capacity. Physical route diversity, while being costlier than reservation capacity, is

easier to coordinate and manage, resulting in architectures of lower hardware and software complexity.

Four mixed variable linear mathematical formulations have been presented for the analysis of the Fault Tolerant Network Design Problem. Three of them are new (SUND2, SUCAN1 and SUCAN2). Uncapacitated and Capacitated network design problems have been considered. The uncapacitated model is useful for those situations where the cost structure strongly depends on the distance, or the traffic demands are low compared with the available capacities. Models were compared by solving to optimality by computer (using Branch and Bound) networks of up to 20 nodes.

Due to the problem high complexity, solving to optimality larger instances of it is very difficult. With that purpose a Lagrangean Relaxation and Heuristics methods have been presented to solve the problem these larger instances. A piecewise step cost function has been used. It is well known that using step cost functions generate results that are poorer than those obtained by using linear costs. The computational experience showed the superiority of the relaxation method, achieving a % deviation to the lower bound of less than 35% (for 30 node random networks). This value can be reduced down to 25% after applying improvement heuristics on the initial feasible solutions.

CHAPTER FOUR: USE OF GENETIC ALGORITHMS IN THE DESIGN OF TELECOMMUNICATIONS NETWORKS

4.1. Introduction

The design of a bi-connected telecommunication network, consists in the interconnection of a set of N nodes, in such a way there exist at least two alternative paths between any given pair of nodes. Given a graph $G(V,E)$, where V is the set of nodes and E is the set of arcs interconnecting those nodes, the objective is to find a subgraph $g \in G$ over a set $e \in E$ of possible arcs connecting the nodes, such as the cost of the network (considering routing, capacities and delay restrictions) is minimized. The complexity of this problem makes it very difficult to find optimum solutions even for networks with a few nodes. In this chapter it will be shown how the use of the Genetic Algorithms can help to solve the network design problem for fairly large instances. The solution suggested employs several different algorithms working together, in order to design the network to minimum cost, while satisfying bi-connectivity, defining the routing and assigning discrete capacities to the links. The chapter is organized as follows. First the model to be solved is presented. Then, the genetic algorithms are reviewed. After that, the solution method is discussed in detail. Finally, the chapter presents the computer results for some example networks, comparing them with other well known solving methods.

4.2. Problem Formulation

The network design problem, to be discussed in this chapter, can be stated as follows:

"Given the nodes location of a network and the costs of the infrastructure components, find the minimum cost network topology, satisfying the traffic and delay restrictions among the nodes, while maintaining network bi-connectivity, defining the routing and assigning discrete capacities to the network links."

The mathematical formulation of the model is the following:

Given:

1. The **Costs** of equipment (c^k), infrastructure (c_2) and transmission media (c_3).
2. The **Network topology**, that is the geographical location of all the nodes that will be connected by the network. This allows the calculation of the distance matrix (***mdist***) among all node pairs.

$$mdist = \begin{bmatrix} 0 & d_{12} & d_{13} & d_{1n} \\ d_{21} & 0 & d_{23} & d_{2n} \\ d_{31} & d_{32} & 0 & d_{3n} \\ d_{n1} & d_{n2} & d_{n3} & 0 \end{bmatrix}$$

It is assumed that the matrix is symmetric, that is $d_{ij} = d_{ji}$. If a link (i,j) is considered as infeasible for its use in the final solution, it can be penalized by making its distance value in the matrix, for instance, 100 times its real value.

3. The **Traffic among nodes**. The traffic demands in the network are given by the traffic matrix (***mtraffic***), where t_{ij} represents the traffic between nodes i and j :

$$mtraffic = \begin{bmatrix} 0 & t_{12} & t_{13} & t_{1n} \\ t_{21} & 0 & t_{23} & t_{2n} \\ t_{31} & t_{32} & 0 & t_{3n} \\ t_{n1} & t_{n2} & t_{n3} & 0 \end{bmatrix}$$

Again, this matrix is assumed to be symmetrical, that is, $t_{ij} = t_{ji}$.

$$\text{Minimize } C = \left(\sum_{i \in E} \sum_{j \in E} \sum_{k \in K} c_1^k n_{i,j}^k + c_2 \sum_{i \in E} \sum_{j \in E} d_{i,j} + c_3 \sum_{i \in E} \sum_{j \in E} d_{i,j} \sum_{k \in K} n_{i,j}^k \right) \quad \dots (4.1)$$

where:

- C = Total cost of the network.
- E = Set of arcs (links) in the network.
- K = Set of discrete capacities of different links.
- $n_{i,j}^k$ = Number of links of type k in the edge i,j .
- $d_{i,j}$ = Euclidean distance of link i,j .
- c^k = Terminal equipment cost for type k .

c_2 = Infrastructure cost, that is, ducts and support buildings.

c_3 = Transmission media cost.

Subject to:

- 1) The network is bi-connected.
- 2) Each link has k available discrete capacities. This is defined by the vector “*Discap*”, for example, *Discap* = [55 155 622] Mbps.
- 3) The mean delay in the network is lower than a given value τ_{max} , such that:

$$\frac{1}{\gamma} \sum_{i \in E} \sum_{j \in E} \left(\frac{f_{i,j}}{c_{i,j} - f_{i,j}} \right) \leq \tau_{max}$$

where:

$f_{i,j}$ = Flow in link i,j , that is the data being transported through the link i,j .

$c_{i,j}$ = Total capacity of link i,j .

γ = Total flow in the network.

Over the design variables:

- 1) Topology
- 2) Routing
- 3) Discrete capacity assignment

4.3. Complexity of the problem

This problem is NP-complete, where the total number of solutions (both feasible and infeasible) is given by the expression:

$$Ns = \sum_{i=0}^T \binom{T}{i} L^i$$

where:

T = Total number of links in the network. This is $n(n-1)$ for directed links and $n(n-1)/2$ for undirected links, for n nodes.

L = Number of discrete link capacities available.

The number N_s explodes for a relatively small number n , as seen in Table 4.1, so only for the very small networks, is finding the optimum feasible.

Table 4.1. Solution space for different network sizes.

Network Size	N_s for $L=1$	N_s for $L=2$	N_s for $L=3$
$n=4 \Rightarrow T=6$	64	729	4026
$n=10 \Rightarrow T=45$	3.5184E+13	2.9543E+21	1.2379E+27
$n=15 \Rightarrow T=105$	4.0565E+31	1.2524E+50	1.6455E+63
$n=20 \Rightarrow T=210$	1.6455E+63	1.5684E+100	2.7077E+126
$n=24 \Rightarrow T=276$	1.2142E+83	4.8469E+131	1.4742E+166
$n=42 \Rightarrow T=861$	1.5375E+259	?	?
$n=46 \Rightarrow T=1035$?	?	?

As it is observed in the table, for a network of 42 nodes, the number of possible solutions is enormous. So an exhaustive search for the optimum solution network is well beyond the capability of current computer hardware. For this reason, a heuristic method can be the only way of approaching the optimum solution. Further complications reside in the fact that in the solution space there are many local optima, so the heuristic must have the ability of searching for these minima, and also to jump out of these local minima to explore and try to obtain better solutions. This chapter will explore the use of genetic algorithms for that purpose.

4.4. Genetic Algorithms

In large size problems, with combinatorial complexity and with several local optimums and non-linear restrictions, the exact optimization algorithms do not yield good results because of the huge solution space, which has to be evaluated. In these problems, being close to a good solution is often the only possible way. An additional difficulty is that these problems have many local minima, so finding the global minimum may be very hard. Stochastic strategies are among the best approaches for these kind of situations. These strategies search the optimum solutions by using all the feasible solution space. The greater the dispersion on the tested solutions is, the greater the

probability of finding a good solution will be. Among these stochastic methods it can be mentioned the following: Simulated Annealing, Tabu Search, Random Search, N-opt Switching and Genetic Algorithms. In this thesis the use of genetic algorithms is studied for the solution of the link sizing problem for ATM networks.

A genetic algorithm is a heuristic-stochastic method for searching optimum solutions to a problem (Holland, 1992). It is based in the Natural Selection process of the species (commonly known as Evolution). In the process the stronger individuals have a better chance of surviving (due to genetic characteristics). The genetic algorithm is based on the simultaneous evaluation of a set of feasible solutions (the population), from which a new set of feasible solutions is determined by using a Selection/Evolution process. The genetic algorithm allows the searching for solutions in all the feasibility zone and the detection of promising areas in which to search for optimum solutions. It has been used with success in combinatorial problems of medium to large sizes (traveling salesman, location problems, etc.). In addition the algorithm may be implemented by using parallel processing.

4.4.1. Genetic Algorithm Definition

Several authors have defined the genetic algorithms (Davis, 1987, Goldberg, 1989, Fogel, 1995), which can be described as follows: Each individual or solution is characterized by a **chromosome**, where all the variables of the problem are represented. This chromosome is composed of a series of genes, each one representing a continuous or discrete variable (Figure 4.1.).

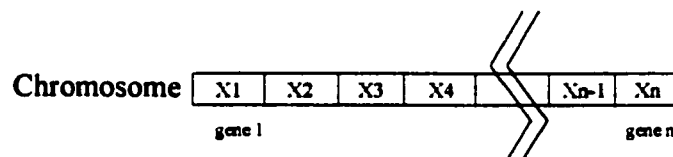


Figure 4.1: Individual coding in chromosomes.

The value of the solution each individual (chromosome) represents, is called the **fitness** or **capacity** of the individual. The higher the fitness value is, the higher the individual surviving

probability will be. The coding process for determining a feasible solution is the most delicate step of the algorithm and determines the convergence of the process. A set P of solutions or individuals, the **population**, is defined and kept constant in size along the iteration process (or **generations**). Empirical tests have found that the higher the population size is, the higher the feasible zones explored are and so the higher the probability of detecting good solutions is. Experiments have determined that the population size should be higher than the number of genes of each individual.

The algorithm works with an individual population (solution set P_k) on each generation k . The first stage of the algorithm is to evaluate the individual fitness of the current population and then sort the individuals according to their fitness value. The individual evaluation process requires the verification of the problem restrictions and the validation of the solution feasibility. Once the individuals are sorted, some of them are to be selected to pass to the next generation. This selection is a stochastic process where the probability of an individual being selected is proportional to its fitness. The selection process is such that an individual may be chosen more than once to survive to the next generation $P_{k'}$. The selection process may be implemented through charged roulettes, Bernoulli processes, etc., but keeping in sight that the higher the fitness is, the higher the surviving probability will be.

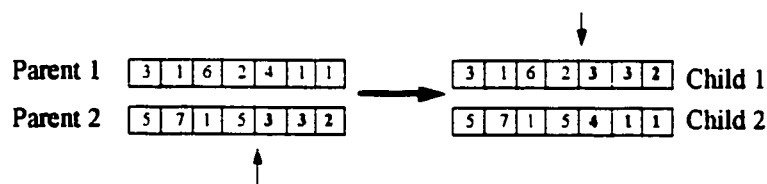


Figure 4.2: Crossover with gene interchange.

After that, from the new population $P_{k'}$, a set of C individuals (parents) are chosen. The parents will mate (**crossover**) each other to generate a new set of individuals (children), which will form a new population $P_{k''}$. The mating process is done by the interchange of one or several genes between the parents (Figure 4.2.). This can be done in several ways, such as mixing all the genes, adding them, or interchanging them from some point in the chromosome, etc. The next stage is to

randomly select a set S of individuals ($S \ll P$) from the generation P_k and pass them through a **mutation** process (Figure 4.3.). This mutation changes one or more genes of the chromosome. This process allows for the introduction of some characteristics that were not previously present in the individuals, and that may contribute to a better solution (value of fitness).

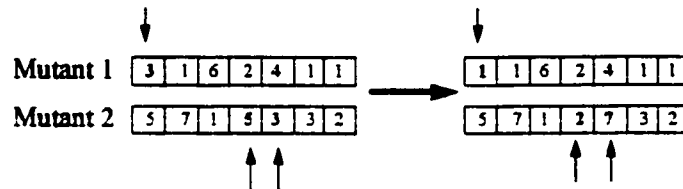


Figure 4.3: Individual mutation.

Once the mutation process is completed, the next population P_{k+1} will be well defined and the evaluation process of its individuals may be started. The process repeats itself until the ending criterion is met, for instance, when the maximum number of iterations is reached, or when no further improvement in the solution is realized. The following sections discuss some of the genetic algorithm characteristics

4.4.2. Pseudo Code and Genetic Algorithm Components

The usual Genetic Algorithm works using a cycle as the one shown in figure 4.4:

1. An initial population (the current generation), composed by chromosomes (individuals) is created. These are the individuals that will evolve in such a way that the future generations will improve their capacity (objective function) with respect to the previous generations.
2. Each chromosome is evaluated by calculating its fitness (objective value). The chromosome is usually a solution (feasible or infeasible) to the problem.
3. If one (or more) of the chromosomes satisfies the optimality criterion (for instance its fitness can not be improved or the total number of iterations allowed have been performed), then that is the best found solution to the problem.

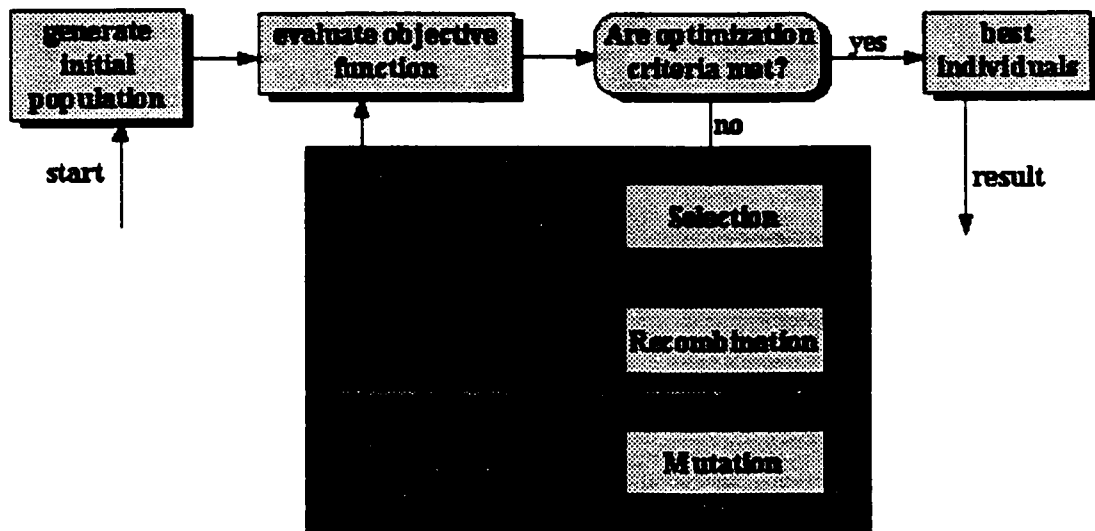


Figure 4.4. Genetic Algorithm Model.

4. If the optimality criterion is not reached, then a new generation is generated through crossover or mutation processes. This new generation replaces the old one and the genetic algorithm returns to step 2.

The pseudo code that shows the operation of a genetic algorithm is the following:

```

//Initialize the generation counter
  g=0;
//Initialize the first generation
  initpopulation P(g);
//Evaluate the fitness of all the generation chromosomes
  evaluate P(g);
//Test for the ending criterion (usually a maximum number of generations)
  while g ≤ MAXGEN do
//Increment the generation counter
    g = g+1;
//Select a subpopulation P' of P(g) (the parents) according to their fitness
    P'=selectparents P(g);
//Crossover the subpopulation P'
    crossover P'(g);
//Stochastically Mutate the subpopulation P'(g)
    mutate P'(g);
//Evaluate the fitness of all children chromosomes
    evaluate P'(g);
//Select the new generation from the parents and children in terms of their fitness
    P(g+1) = survive (P'(g), P(g));
  Loop
End
  
```

4.5. Theoretical Analysis of the Genetic Algorithms

This section discusses why do the genetic algorithms work to provide better solutions to a given problem. It will turn out that the genetic algorithms process similarity patterns among chromosomes, and since each chromosome fits well in many patterns at the same time, the search efficiency is increased.

4.5.1. Basic concepts

A *scheme S* is a similarity pattern which is built by introducing the sign “*” of optionality or indifference. In this way a scheme represents all the individuals fitting its pattern. For instance, the scheme (* 1 * 1 0 1 0) represents the following chromosomes:

0101010 0111010 1101010 1111010

It turns out that a population of size *Popsize* chromosomes, each of length *Len* contains between 2^{Len} and $Popsize * 2^{Len}$ different schemes.

The *order of a scheme o(S)* is the number of specified positions (1 or 0) in the scheme. It measures how specific is the scheme

The *characteristic length of a scheme δ(S)* is the distance, in digits, between the extreme fixed positions. For instance, the scheme (***101*1**) has $o(S)=4$ and $\delta(S)=10-4=6$.

Given a population of binary chains $P(g)$ and a scheme *S*, the *presence of S in P(g)*, $\xi(S,P(g))$ is the number of chains of $P(g)$ that fit in the scheme *S*.

The *fitness of scheme S in P(g)*, $Fitness(S,P(g))$ is the average of all the fitness of all chains of population that fit in the scheme in the generation *g*. If chains of $P(g)$ that fit in *S* are enumerated as $c_1, c_2, c_3, \dots, c_p$, where $p=\xi(S,P(g))$, then

$$Fitness(S,P(g)) = \frac{1}{p} * \sum_{i=1}^p Fitness(c_i)$$

The average fitness of population in generation *g*, $AveFitm(P(g))$, is the average of the fitness of all chains of population in generation *g*, that is

$$AveFitm(P(g)) = Fitness((*...*),P(g)) = \frac{1}{PopSize} * \sum_{c_i \in P(g)} Fitness(c_i)$$

The *relative fitness of S in P(g)*, $AveFitn(S, P(g))$ is the ratio between the scheme fitness in the population and the average fitness of the population in generation g, that is

$$AveFitn(S, P(g)) = \frac{Fitness(S, P(g))}{AveFitn(P(g))}$$

4.5.2. Reproductive growing equation of a scheme

Consider $P(g)$ of $PopSize$ elements of length Len , in which $PopSize$ is large enough. A scheme S in population $P(g)$ evolves according to the formula

$$\xi(S, P(g+1)) = \xi(S, P(g)) * k_g * k_s \quad \dots\dots\dots (4.2)$$

where k_g is the *growing factor* of S in the population, and k_s is the *survival factor*. The first measures the tendency of the scheme to increase its presence in the intermediate generations, while the second measures the probability of the scheme surviving into the following generation. The *selection* operator determines the increasing presence of schemes in the intermediate generations. Hence, the scheme presence in an intermediate population $Q(g)$ evolves on the average according to:

$$\begin{aligned} \xi(S, Q(g)) &= \xi(S, P(g)) * AveFitn(S, P(g)) \\ &= \xi(S, P(g)) * \frac{Fitness(S, P(g))}{AveFitn(P(g))} \end{aligned}$$

from where it results $k_g = AveFitn(S, P(g))$. It is possible that the schemes will be destroyed by the genetic operators before passing to the next generation, which is defined by the k_s factor. This factor is difficult to calculate, but it can be bounded by

$$k_s \geq \left(1 - \frac{p_{xov} \delta(S)}{Len - 1}\right) (1 - p_{mut})^{o(S)}$$

The first term measures the probability of surviving a crossover (of probability p_{xov}), while the second measures the probability of the scheme of being affected by a mutation (of probability p_{mut}). Hence, replacing all factors in formula (4.2), the presence of a scheme S in population $P(g)$ evolves on the average according to the following:

$$\xi(S, P(g+1)) \geq \xi(S, P(g)) * AveFitm(S, P(g)) * \left(1 - \frac{p_{cov} \delta(S)}{Len-1}\right) * (1 - p_{mut})^{o(S)}$$

And since $p_{mut} \ll 1$

$$\xi(S, P(g+1)) \geq \xi(S, P(g)) * AveFitm(S, P(g)) * \left(1 - \frac{p_{cov} \delta(S)}{Len-1} - p_{mut} * o(S)\right)$$

From this inequality, it follows that the *selection* operator is in charge of geometrically increase the presence of the good schemes ($k_g > 1$) and reducing the presence of the bad schemes ($k_g < 1$). However, selection does not introduce new schemes.

The *crossover* operator allows the structured interchange of useful information among the individuals. This is an essential operator for the efficiency of genetic algorithms. Finally, the *mutation* operator introduces diversity in the schemes of a population, and provides insurance against the loss of valuable information.

Holland, 1992 and Goldberg, 1989, proved that genetic algorithms, instead of processing n chromosomes during a given generation, really process $O(n^3)$ schemes per generation, which is known as the implicit parallelism of genetic algorithms.

Summarizing, in successive generations, the presence of a scheme S in the population $P(g)$ evolves statistically in an exponential mode. So why do the genetic algorithms work?. Because they give exponentially growing opportunities to the fitter schemes and exponentially decreasing opportunities to the less fit ones.

4.6. Genetic algorithm used

The algorithm used in the design of the network, consists of different algorithms that work together, as seen in figure 4.5.

4.6.1. Initial population

In figure 4.5, the initial population was chosen to be formed 50% by solutions to the Traveling Salesman Problem (TSP), all of them feasible, and 50% by random solutions (which can be either feasible or infeasible solutions). The reason for introducing infeasible solutions in the initial population is that it creates diversity in the solutions, which allows to jump out from local minimum solutions. It also can be explained because two infeasible solutions, after being combined may or may not produce a feasible solution, as seen in figure 4.6.

4.6.2. Genetic Algorithm Implementation for Finding Good Solutions of the TSP

As mentioned before, the use of TSP solutions as initial feasible solutions to the network design problem makes the heuristic to work faster. In this section, the use of genetic algorithms to find those initial feasible solutions is discussed. The traveling salesman problem (TSP) is a widely studied problem in combinatorial optimization. It consists in finding the minimum traveled distance to visit a collection of N cities, returning to the starting point (Lowler 1985). Any city can only be visited once. Solutions to this problem are based in a) optimal procedures, such as cutting planes or branch and bound or b) heuristics, such as tour construction, tour improvement or composite procedures. Because TSP is a NP-complete type of problem, optimal procedures demand very large computer capacities, even for moderate sizes. Heuristics have been used to obtain solutions to large problems.

The focus of this section is to present a genetic algorithm that rapidly converges to the optimal solution (Rios, 1999). Simulated annealing and genetic algorithms have been generally considered as slow convergence methods (Davis 1987). The proposed genetic algorithm can provide excellent solutions in a relatively very short time.

The genetic algorithm implemented follows the steps shown in figure 4.7. An integer representation was used for the chromosomes; that is, the elements of each chromosome are the numbers, representing the visited cities. Thus a tour that visits 5 cities in the order 1-5-3-2-4-1 may be represented as a chromosome 1/5/3/2/4, or also as the chromosome 3/2/4/1/5, etc.

The initial population was generated randomly. Parasites are random solutions that are generated as needed. The importance of parasites is fundamental in making the genetic algorithm work fast. In a given generation, each chromosome is evaluated by calculating its fitness. If there are redundant chromosomes (that is several chromosomes with the same fitness value), only one of them is kept, replacing the others with parasites. Parasites thus give diversity to the population, allowing the algorithm to look for better solutions.

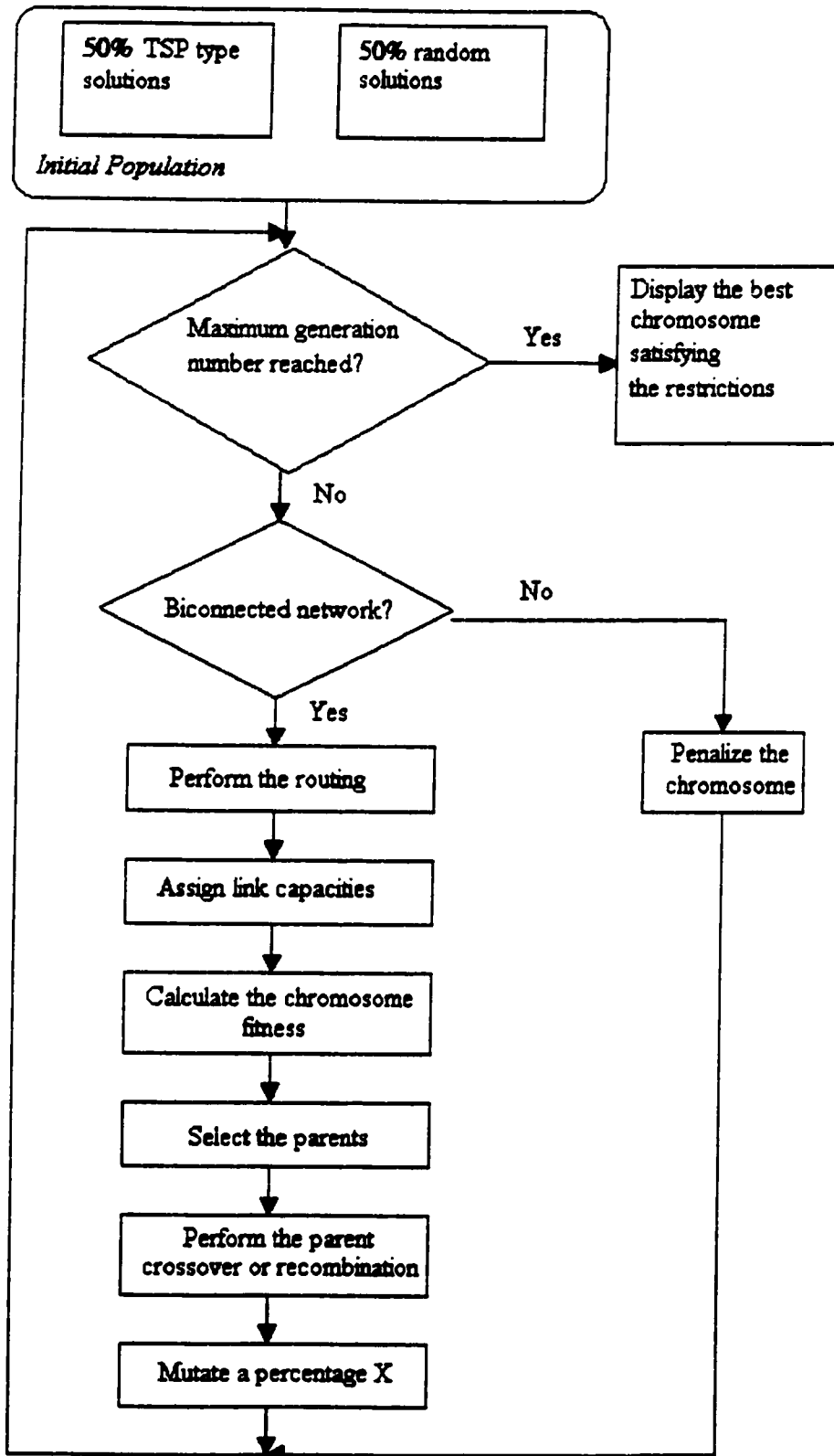


Figure 4.5. Flow diagram of the network design algorithm.

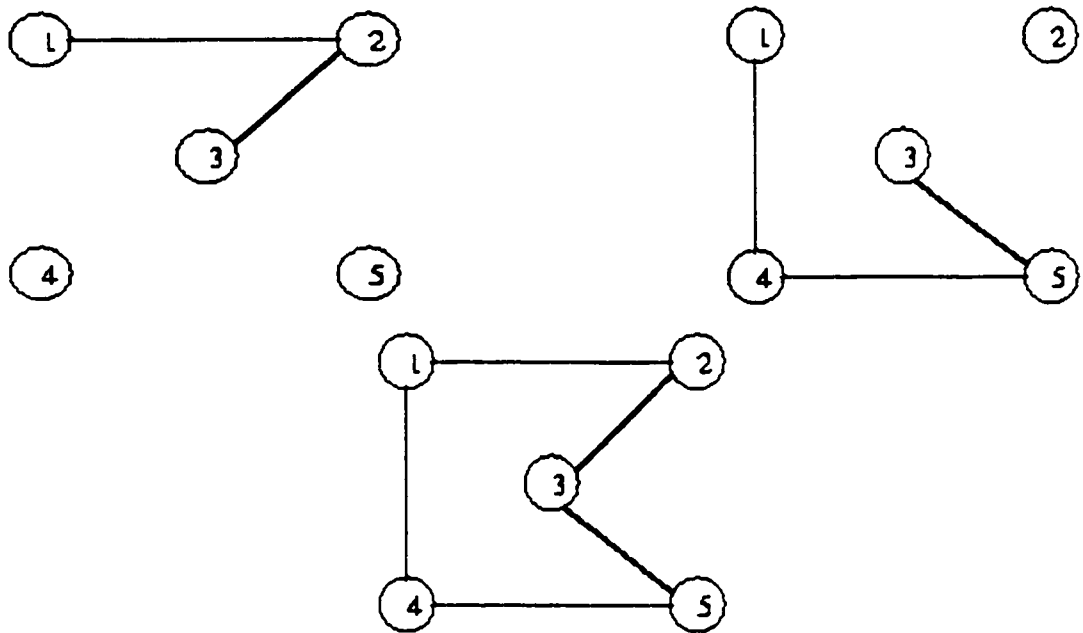


Figure 4.6. Example of two infeasible chromosomes that recombine into a feasible one.

The use of an integer representation for the chromosomes makes it necessary to modify the crossover method, since the normal method would produce infeasible solutions. A special crossover algorithm was developed, which consists in inherit to the children the largest possible amount of useful information from each parent. This is illustrated by the example below:

Parent1: (1 2 4 5 3) ← this is the chromosome

Parent2: (2 3 4 5 1)

We want to generate two children with these two parents. The algorithm is as follows:

- 1) Select a gene (or city) as a base node. This number is randomly chosen. Assume gene 3 is chosen.
- 2) Rotate one of the parents, such that the chosen gene appears in the same position in both parents:

Parent1: (5 3 1 2 4) ← this is the rotated parent

Parent2: (2 3 4 5 1)

- 3) Start building the children by using the base node and gene located to the right of the base node on each parent. Child1 uses parent1 and child2 uses parent2.

Child1: (..3 1..)

Child2: (..3 4..)

- 4) To child1 add gene located to the left of base node of parent2. To child2 add gene located to the left of base node of parent1.

Child1: (.. 2 3 1..)

Child2: (.. 5 3 4..)

- 5) Return to point 3 only if the next gene located to the right is not already present in the children. If it is present, then fill the rest randomly.

Child1: (2 3 1 5 4) ← last two genes filled randomly

Child2: (5 3 4 2 1)

The method then preserves a good part of the tour already present on each parent. In the mutation part of the genetic algorithm, two-opt method was used, which detects if there are crossed links between cities. If so, the algorithm eliminates the crossed links replacing them with non-crossing links (on the same nodes). This mutation always improves the chromosome fitness.

4.6.3. Ending criteria

The ending criteria corresponds to the *maximum number of generations*, that is after the genetic algorithm runs through that number of generations it stops, giving as a result the following:

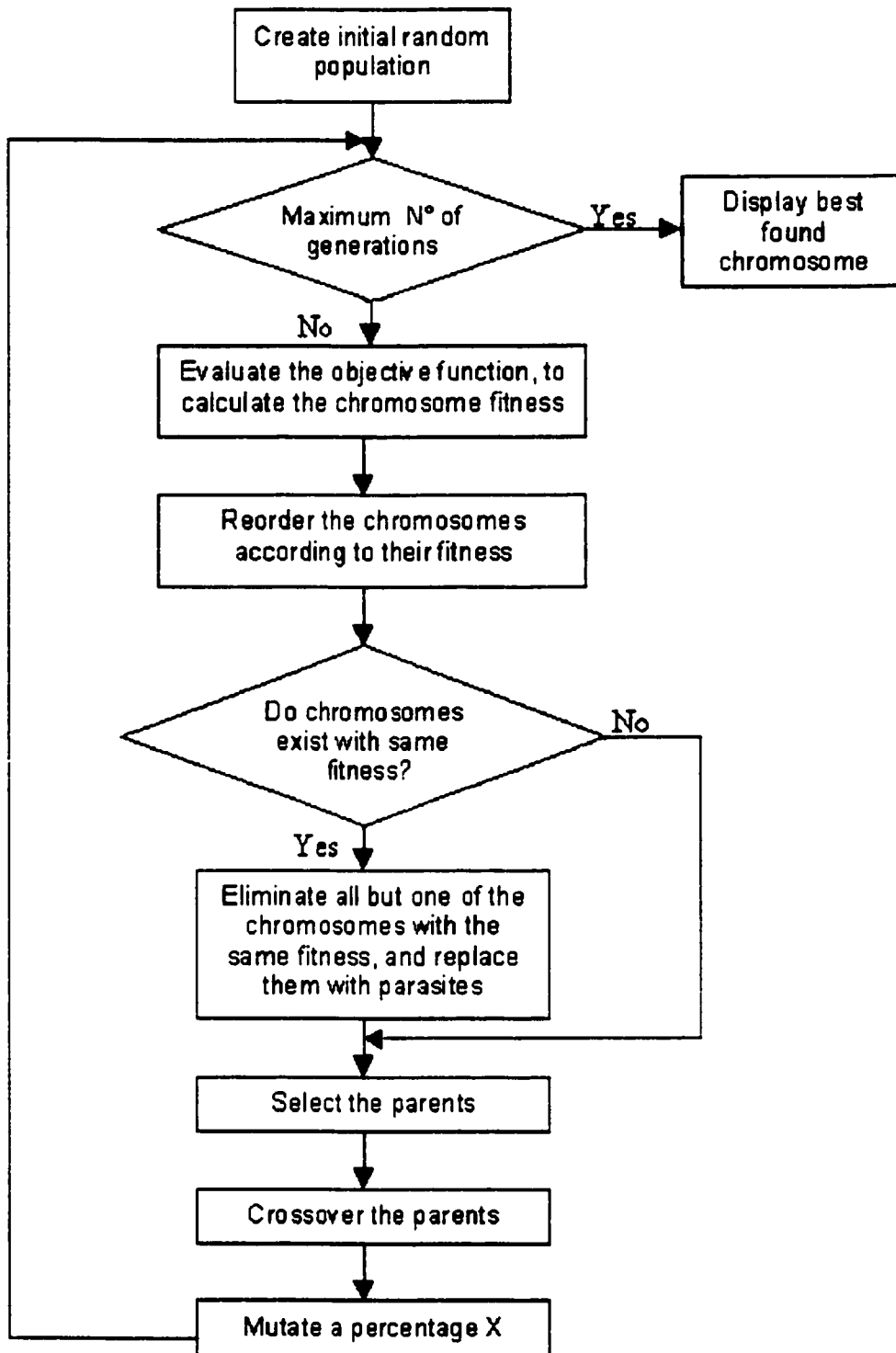


Figure 4.7. Flow diagram of the genetic algorithm.

- 1) The “best” chromosome found after the evolution, satisfying the restrictions.
- 2) Graph for the topology representing the best found chromosome (i.e. the solution network).
- 3) Routing matrix among all node pairs.
- 4) Flow matrix, allowing to observe the total flow in link ij .
- 5) Link capacity matrix, showing the total capacity of link ij .
- 6) Three matrices (of dimension $n \times n$), showing the number e of links of type k , which have been assigned to link ij .
- 7) The average delay delivered by the network that the “best” chromosome represents.

4.6.4. Bi-connectivity checking

To verify if the chromosome complies with the two-connectivity restriction, the **Depth First Search** algorithm was used (Sedgewick, 1992). This algorithm gives:

- 1) A vector showing the articulation nodes, which are those nodes that after being eliminated divide the network into two or more parts.
- 2) The number of subnetworks formed after eliminating the different articulation nodes.

This data was used to penalize those networks that were not two-connected.

4.6.5. Routing

A static routing method was used, since it is easy to describe and allows the direct calculation of flows in the links. The **Floyd Warshall** algorithm (Cormen, 1990, Ahuja, 1993) was used, which finds the shortest path between two given nodes. The algorithm has a $O(n^3)$ complexity. The algorithm considers all the intermediate nodes in a route. The algorithm delivers as a result:

- 1) The routing table.
- 2) The total flow in each link of the network.

4.6.6. Capacity assignment

A non-linear concave discrete cost function (see figure 4.8) was used to determine the capacity of a given link. Typical broadband capacities were used (55, 155 and 622 Mbps). A

greedy algorithm (of $O(n^2)$ complexity) was then used to assign the capacity (Kereshbaum, 1993) which can be described as follows:

- 1) Given the network and the flow matrix, assign to each available link the minimum available capacity, or a linear combination of the available discrete capacities.
- 2) Calculate the **delay** as a function of the capacity and flow.
- 3) While **delay** \geq **restriction delay** (see problem formulation)

Do {

- a) Find the link (i,j) with the lowest difference after calculating the matrix difference between the Capacity and Flow matrices.
 - b) Increase the capacity of that link (i,j) .
 - c) Recalculate the delay.
- }

The algorithm delivers as results the following:

- 1) The total number of terminal equipment or links needed. This number can be a linear combination of the discrete available capacities. For instance, to satisfy a flow of 650 Mbps, the best solution would be to put a link of 622 plus a link of 55 Mbps
- 2) Three matrices (of dimension $n \times n$) giving the number of links of each type.

4.6.7. Average delay

An M/M/1 queueing system was assumed, using the Kleinrock independence assumption (Bertsekas 1992, Gerla, 1977), which states:

- 1) Packets arrive according to a Poisson distribution.
- 2) The packets are stored in infinite buffers at intermediate nodes (no packet loss at buffers).
- 3) The routing is fixed (assuming an M/M/1 network of queues).
- 4) The transmission channel is assumed error-free.
- 5) There is no processing delay at buffers.

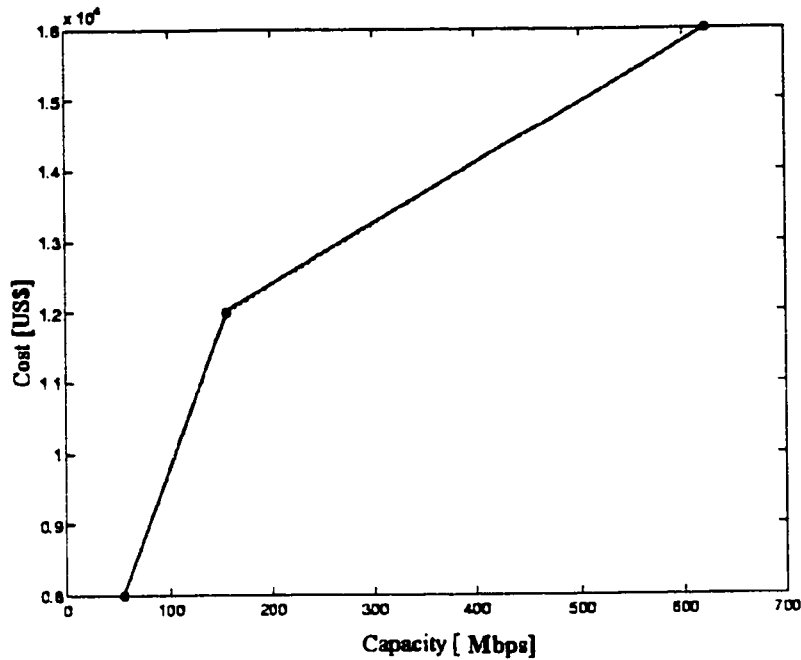


Figure 4.8. Terminal equipment cost in terms of the capacity.

4.6.8. Fitness function

Having a chromosome with a very high fitness value, compared with those of the rest of the population, may lead the algorithm to converge, prematurely, to a local minimum. To avoid this problem, a function is used which calculates the chromosome fitness as follows (Whitley, 1989, Back, 1991):

$$Fitness(Pos) = 2 - SP + \frac{2(SP - 1)(Pos - 1)}{Nind - 1}$$

where:

Nind = Number of chromosomes (individuals) in the population.

Pos = Position of the individual in the population (the more fit is in the first position, the least fit in the *Nind* position).

SP = Is the selective pressure, which is the probability that the fittest chromosome will be selected, as compared with the average probability of any chromosome of being selected.

In Table 4.2, an example shows how the fitness is calculated for *Nind*=11 for two values of the selective pressure in a minimization problem. Note linear ranking allows values of the selective pressure in [1.0, 2.0]. As seen there, the fitness value of each chromosome depends only on the

chromosome position (and not on the objective value). Note how with a selective pressure value of 1.1, the least fit individual (with a fitness value of 0.9) will have a good chance of surviving to the next generation, while with a selective pressure value of 2.0, it will not survive (its fitness value drops to 0.0).

Table 4.2. Example of fitness calculation.

objective value	fitness value (parameter: selective pressure)		
	linear ranking		no ranking
	2,0	1,1	1,0
1	2.0	1.10	1,0
3	1.8	1.08	1,0
4	1.6	1.06	1,0
7	1.4	1.04	1,0
8	1.2	1.02	1,0
9	1.0	1.00	1,0
10	0.8	0.98	1,0
15	0.6	0.96	1,0
20	0.4	0.94	1,0
30	0.2	0.92	1,0
95	0.0	0.90	1,0

4.6.9. Selection

A truncated selection was used (Blickle, 1995). In truncation selection individuals are sorted according to their fitness. Only the best individuals are selected for parents. These selected parents produce uniform at random offsprings. The parameter for truncation selection is the truncation threshold *Trunc*. *Trunc* indicates the proportion of the population to be selected as parents and takes values ranging from 50%-10%. Individuals below the truncation threshold do not produce offsprings. The term selection intensity is often used in truncation selection. The selection intensity is defined as the expected average fitness value of the population after applying a selection method to the normalized Gaussian distribution. Table 4 shows the relation between both. Only the best chromosomes (with a maximum of 50%) were considered as parents of the next generation.

truncation threshold	1%	10%	20%	40%	50%	80%
selection intensity	2.66	1.76	1.2	0.97	0.8	0.34

4.6.10. Crossover or Recombination

To generate new chromosomes from the selected parents, the *single point crossover* method was used (Pohlheim, 1996). In single-point crossover, one crossover position $k[1,2,\dots,Nvar-1]$, where $Nvar$ is the number of variables of an individual, is selected uniformly at random and the variables exchanged between the individuals about this point. This produces two new offsprings. Figure 4.9 illustrates this process.

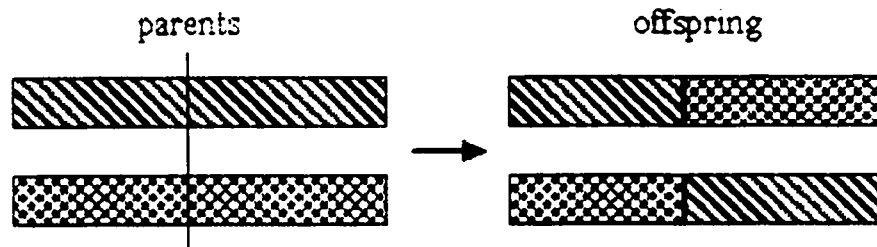


Figure 4.9. Single-point crossover

Consider the following two individuals with 11 binary variables each:

individual 1 0 1 1 1 0 0 1 1 0 1 0

individual 2 1 0 1 0 1 1 0 0 1 0 1

Suppose the chosen crossover position is: 5. After crossover the new individuals are created:

offspring 1 0 1 1 1 0 | 1 0 0 1 0 1

offspring 2 1 0 1 0 1 | 0 1 1 0 1 0

4.6.11. Mutation

Mutation ensures all chromosomes of the solution space are reachable. For instance, if the optimum solution requires that the first position of the solution chromosome be a “1”, it could happen that the initial population does not have chromosomes with that characteristic. Mutation can provide for such chromosomes. For binary valued individuals mutation means flipping of variable values. For every individual the variable value to change is chosen uniformly at random. Next example shows a binary mutation for an individual with 11 variables (genes), where variable 4 is mutated.

before mutation 0 1 1 1 0 0 1 1 0 1 0

after mutation 0 1 1 0 0 0 1 1 0 1 0

4.6.12. Penalizing Functions

These functions penalize the networks, represented by their chromosomes that do not satisfy the restriction of being bi-connected. After some testing, the most natural penalizing function is the one that takes into account the number of subnetworks formed after eliminating an articulation node, and also the one that considers the number of articulation nodes in a chromosome. Hence, if some chromosomes do not comply with the restriction on bi-connectivity, their objective value is calculated as:

$$objval = \sum_{k \in K} c_1^k n_{i,j} + c_2 d_t + c_3 d_t n_{i,j}$$

where:

$$d_t = \max(\text{mdistan}) * (N + \text{artnode} + \text{subgraph})$$

n_{ij} = Number of links of type k in arc i,j

K = Number of discrete capacities available ($k \in \{1, 2, 3\}$)

C_1^k = Cost of terminal equipment k

C_2 = Cost of ducts

C_3 = Cost of transmission media

N = number of nodes

$Artnode$ = number of articulation nodes

$Subgraph$ = Total number of subnetworks

$Mdistan$ = $N \times N$ matrix representing the distance among all node pairs.

The equation penalizes each term as a function of the parameters of the chromosome. For instance, if a chromosome presents more articulation nodes, its objective value will be greater

4.7. Computational Results

Several experiments were carried out to test the genetic algorithmic approach. To verify the quality of the results, the method was compared with the Branch Exchange (Kershenbaum, 1993) and the Exhaustive Search methods. The software was implemented using Matlab 5.2.0.3084, and run in Digital Alpha 700 and Intel Pentium workstations. Networks with 4, 10 and 42 nodes were tested, the first one was used in order to validate the results.

Two types of genetic algorithms were used in the results. The first one is a new development (Rios 1999) and it is very good in finding good solutions to the TSP. This TSP

solutions are then used as part of the initial population which feeds the second genetic algorithm. The second genetic algorithm is a toolbox developed for Matlab 4.2 (Polheim 1996), and it was modified to run in Matlab 5.2 and to use automatically the results obtained in the first genetic algorithm.

4.7.1. Branch Exchange Algorithm

This is a widely used method for network design. However it has some disadvantages (Kershenbaum, 1993):

- 1) Since the routing has a complexity of $O(n^3)$, and it has to be calculated inside a loop that searches for merit figures in a $O(n^2)$ square matrix, it will have a complexity of $O(n^5)$, which is bad for networks of more than about 50 nodes.
- 2) The algorithm has the tendency of finding a local minimum close to the region of the initial solution. Hence it is very sensitive to variations on the initial solution. If the initial solution is poor, the final solution will also be poor.

The algorithm used in this thesis can be stated as follows:

- 1) Given a set of nodes V and an initial graph G , which connects every node i with every other node j (i.e. a fully meshed graph), verify if it satisfies the bi-connectivity requirements.
- 2) Calculate the routing using the shortest path method.
- 3) Assign capacities using the greedy algorithm.
- 4) While restrictions are satisfied and cost (equation 4.1) does not increase

Do {

- a) Calculate the routing using the shortest path method.
 - b) Assign capacities using the greedy algorithm.
 - c) Eliminate the link having the highest merit figure $Cost/bit$
- }

where $Cost/bit = link(i,j) \text{ cost} / \text{total flow in link}(i,j) = C_{i,j} / f_{i,j}$

The link cost $C_{i,j}$ is calculated using the following equation:

$$C_{i,j} = \sum_{k \in K} c_1^k n_{i,j}^k + c_2 d_{i,j} + c_3 \sum_{k \in K} d_{i,j} n_{i,j}^k \forall i, j$$

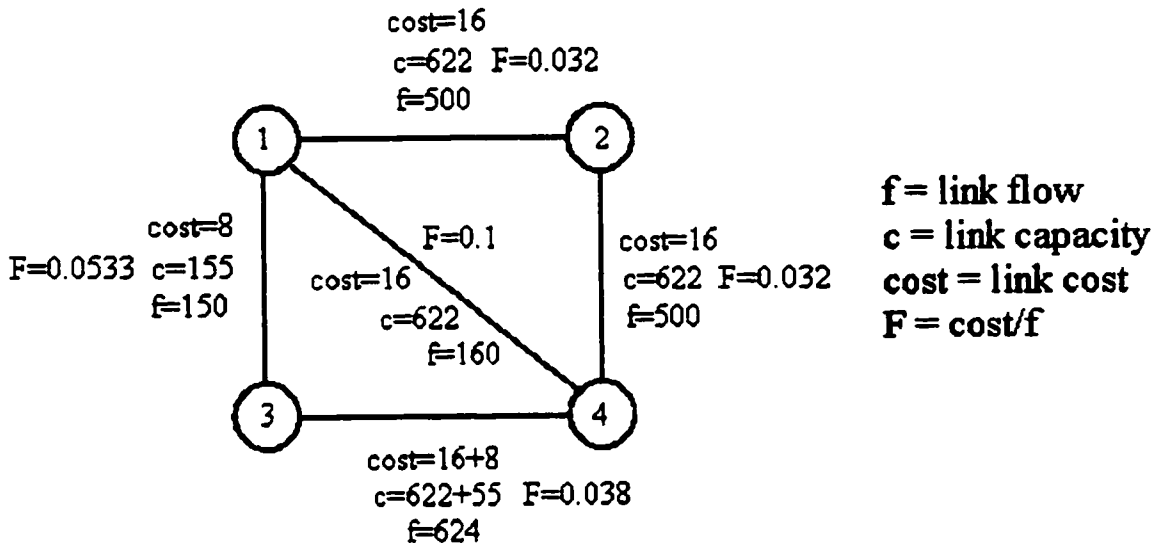


Figure 4.10. Branch exchange example.

For instance, in figure 4.10 the network is bi-connected and the merit figure is the link cost/link flow. As seen the highest merit figure is 0.1, so the link joining nodes 1 and 4 would be eliminated *if and only if* the total cost of the network does not increase and the bi-connectivity restriction is satisfied.

4.7.2. Exhaustive Search Method

This method consists in analyzing all the possible network topologies, to find a subset G of the networks satisfying the restrictions of the model. From this subset the optimum solution is searched. Clearly while this method finds the optimum solution, it can only do that for very small networks. In the thesis it was only used to validate the genetic algorithm results for 4 node networks. Results for 10 and 42 nodes were not obtained by this method.

4.7.3. Results for 10 node networks

Two types of heuristics using genetic algorithms were used. In the first (called GATSP), the initial population is conformed with 50% of TSP solutions for the n nodes, and 50% of random solutions (some of them infeasible). The second heuristic (called GARANDOM) uses a 100% random initial population. The total space of solutions is equal to $1.2379e+27$ networks.

The values used for the model variables were the following :

a) Infrastructure

- i) $c_1^1 = \text{Cost of the 55 Mbps equipment} = 3,728,000$
 $c_1^2 = \text{Cost of the 155 Mbps equipment} = 5,592,000$
 $c_1^3 = \text{Cost of the 622 Mbps equipment} = 7,456,000$
- ii) $c_2 = \text{Cost of the ducts} = 12,000,000 \text{ per Km.}$
- iii) $c_3 = \text{Cost of the transmission media} = 4,660,000 \text{ per Km.}$
- b) Node location. They were expressed as a 2x10 location matrix, where the first row indicates the x coordinate (in Km.) of the node and the second row indicates the y coordinate.

$$[\text{Location}] = \begin{bmatrix} 0 & 1 & 0 & 3 & 4 & 6 & 6 & 8 & 7 & 4 \\ 2 & 4 & 7 & 6 & 5 & 6 & 8 & 3 & 0 & 2 \end{bmatrix}$$

- c) Available link capacities. Given as a vector in Mbps: [55 155 622]
- d) Traffic matrix among the nodes, expressed in Mbps. For example:

$$m_{\text{traffic1}} = \begin{bmatrix} 0 & 300 & 100 & 500 & 100 & 50 & 200 & 500 & 100 & 300 \\ 300 & 0 & 200 & 100 & 300 & 60 & 200 & 50 & 100 & 150 \\ 100 & 200 & 0 & 200 & 100 & 150 & 250 & 50 & 130 & 100 \\ 500 & 100 & 200 & 0 & 100 & 170 & 210 & 120 & 40 & 250 \\ 100 & 300 & 100 & 100 & 0 & 150 & 50 & 0 & 60 & 50 \\ 50 & 60 & 150 & 170 & 150 & 0 & 90 & 220 & 140 & 270 \\ 200 & 200 & 250 & 210 & 50 & 90 & 0 & 60 & 120 & 210 \\ 500 & 50 & 50 & 120 & 0 & 220 & 60 & 0 & 350 & 0 \\ 100 & 100 & 130 & 40 & 60 & 140 & 120 & 350 & 0 & 90 \\ 300 & 150 & 100 & 250 & 50 & 270 & 210 & 0 & 90 & 0 \end{bmatrix}$$

- e) Maximum average delay restriction. This delay was set to

$$\text{Delay} \leq 0.01 \text{ seconds}$$

In five experiments all the previous data was maintained, changing only the traffic matrix. The GATSP and GARANDOM heuristics and the Branch Exchange method were used to find solutions, which are summarized in Table 4.3.

Table 4.3. Solution for 10 node networks for different traffic.

	GATSP	GARANDOM	BRANCH	Improvement	Improvement
	Cost	Cost	Cost	GATSP %	GARANDOM %
mtraffic1	1.04E+09	1.10E+09	1.09E+09	3.80	-1.11
mtraffic2	8.38E+08	8.75E+08	8.56E+08	2.16	-2.11
mtraffic3	5.74E+08	6.29E+08	6.79E+08	15.48	7.34
mtraffic4	5.88E+08	6.29E+08	7.93E+08	25.93	20.74
mtraffic5	6.91E+08	7.12E+08	7.05E+08	1.96	-0.96
			Average	9.87	4.78

It is observed in the results, that the proposed GATSP heuristic finds consistently better results than the Branch Exchange method. GARANDOM is not that good in that respect, which is due to the use of purely random initial solutions, which slow the convergence to feasible solutions. Figure 4.11 shows graphically these results.

Figure 4.12 shows the solution networks obtained with the three heuristics for the *mtraffic1* case. The number over the links denotes the capacity assigned (left number) and the flow (right number). It is observed that the GATSP solution, while having a larger number of links than the Branch Exchange solution, has some links with lower capacities and so the total cost is lower.

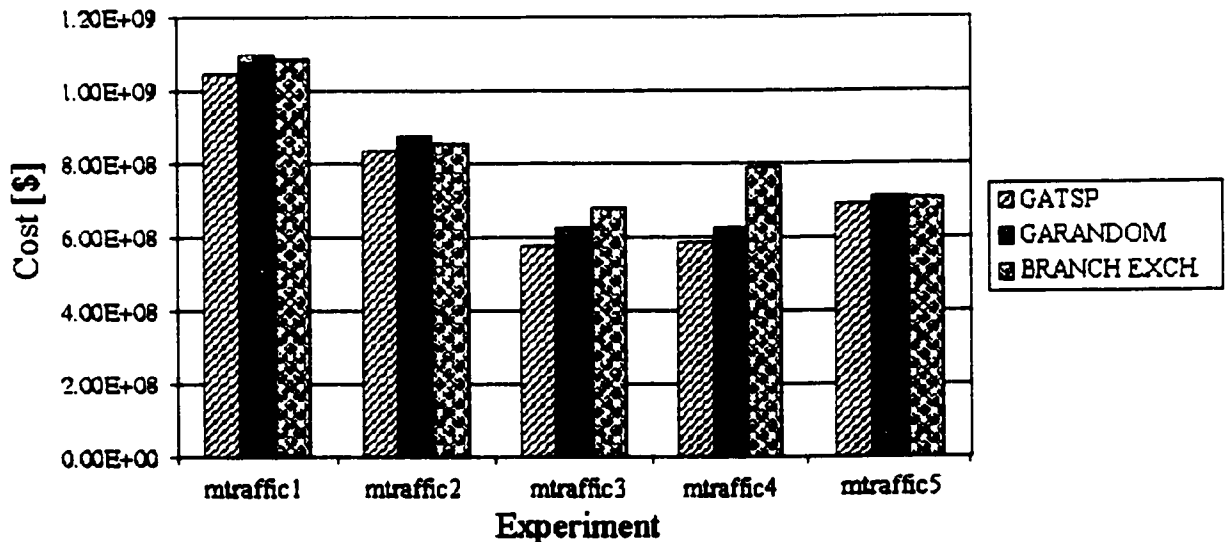


Figure 4.11. Solutions results when varying the traffic matrix.

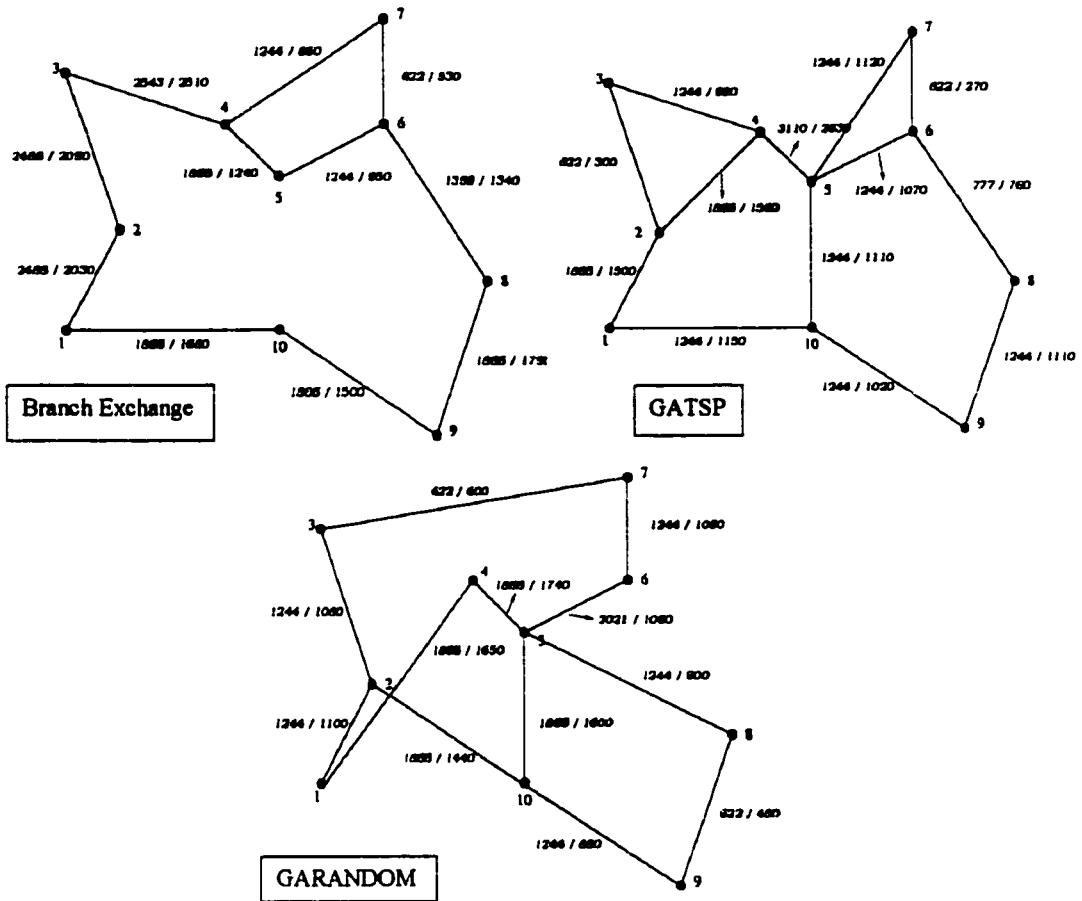


Figure 4.12. Solution networks for *mtraffic1*.

The next experiments were done by changing the node locations, while maintaining the rest of the variables. Table 4.4 shows the results obtained.

Table 4.4. 10 node solutions with different locations

	GATSP	GARANDOM	BRANCH	Improvement	Improvement
	Cost	Cost	Cost	GATSP %	GARANDOM %
Location1	1.22E+10	1.40E+10	1.22E+10	0.00	-14.94
Location2	6.71E+09	7.00E+09	6.97E+09	3.83	-0.29
Location3	3.34E+09	3.48E+09	3.43E+09	2.61	-1.48
Location4	3.09E+10	3.54E+10	3.11E+10	0.61	-13.90
Location5	1.43E+10	1.50E+10	1.54E+10	7.43	2.65
			Average	2.89	-5.59

Figure 4.13 shows graphically this results. Again it is evident that the proposed GATSP heuristic performs better than Branch Exchange or GARANDOM heuristics.

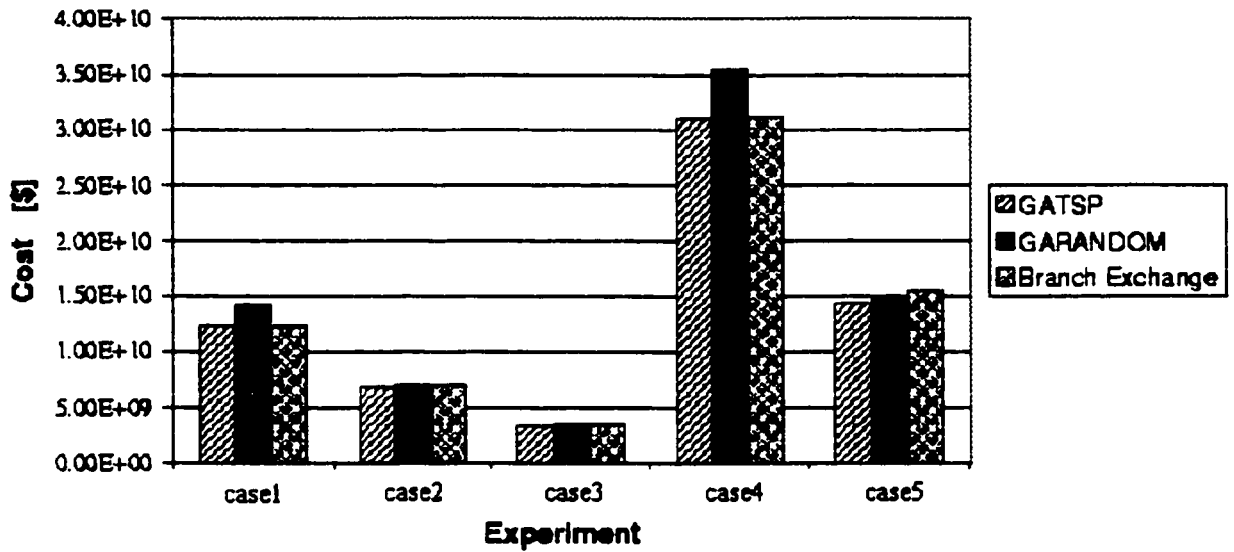


Figure 4.13. Solution costs in terms of different locations.

Figure 4.14 shows the solution networks found for *location3*. Again a better selection of links by the GATSP heuristic results in a lower total cost of the solution, as compared with the Branch Exchange and GARANDOM heuristics.

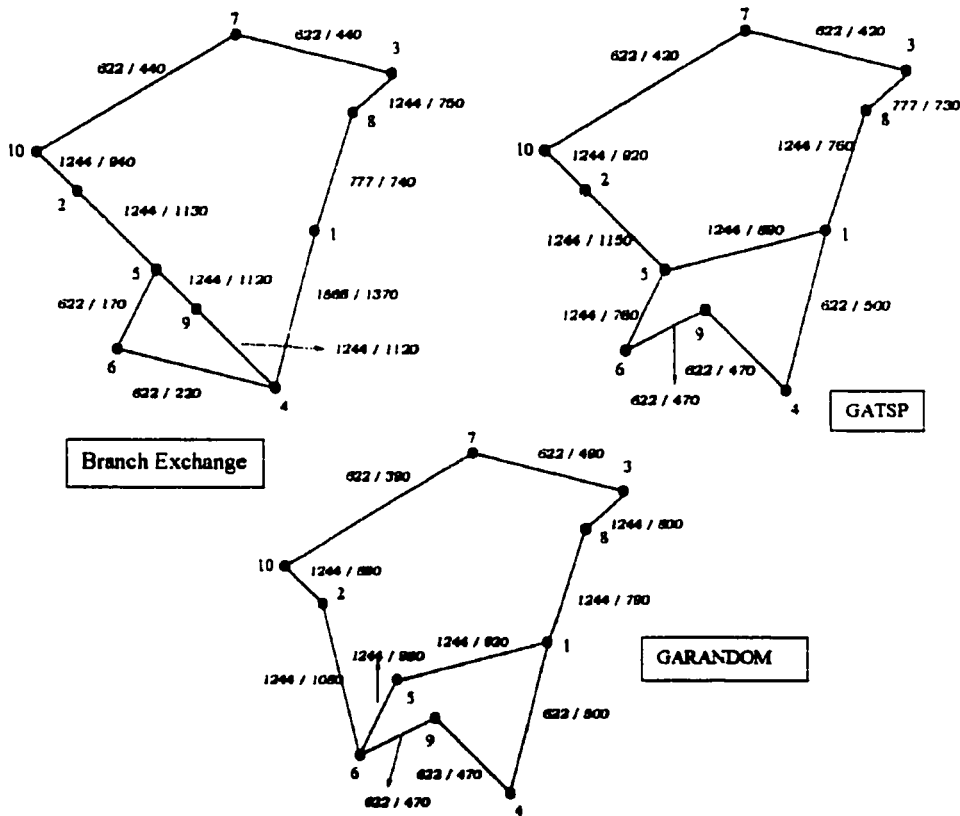


Figure 4.15. Solution networks for location3.

4.7.4. Results for a 42 node network

Since the GATSP heuristic performed so well for 10 node networks, it was tried in the design of a 42 node network, which is a better test for the algorithm. Variable values are the same as those used in the 10 node networks, with the exception of the location matrix (now 2x42) and the traffic matrix. Figure 4.15 shows the solution network, which has a total cost of $9.8e+9$, and provides an average delay of 0.0099 seconds. Table 4.5 shows the capacities and flows of each link of the solution.

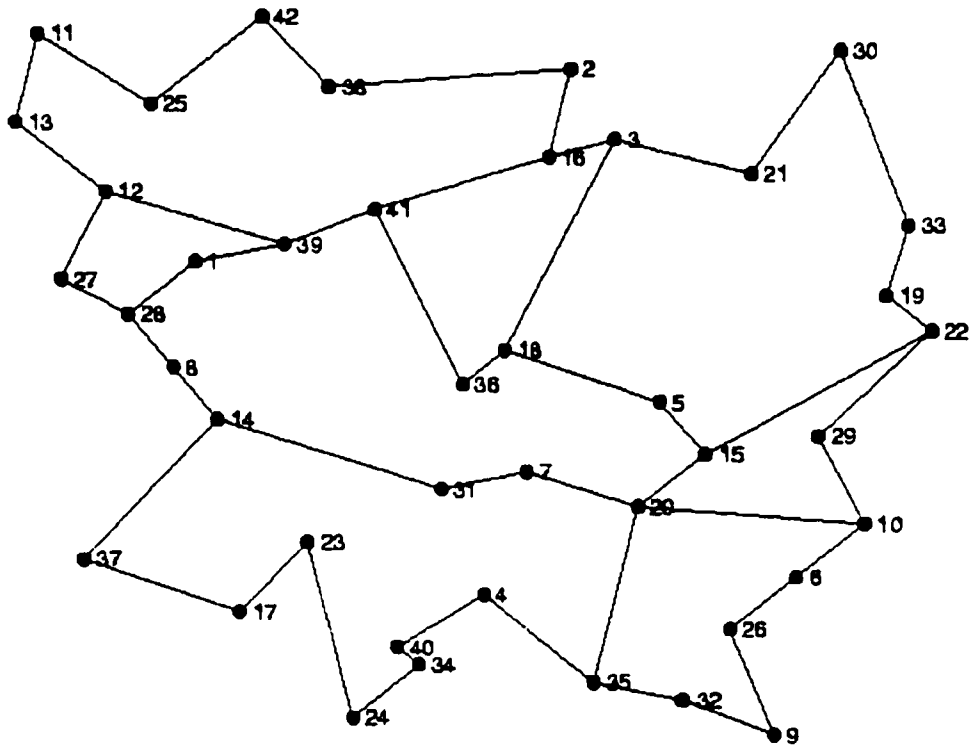


Figure 4.15. 42 node network solution.

Table 4.5. Capacity allocation for 42 node network.

Link i - j	Flow Mbps	Capacity Mbps	N° of links		
			622	155	55
1_28	2239	2488	4		
1_39	2064	2488	4		
2_16	2114	2488	4		
2_38	1264	1299	2		1
3_16	2880	3110	5		
3_18	1764	1866	3		
3_21	1595	1866	3		
4_35	2357	2488	4		
4_40	1715	1866	3		
5_15	3164	3265	5	1	
5_18	3009	3110	5		
6_10	1372	1399	2	1	
6_26	1016	1244	2		
7_20	3546	3732	6		
7_31	3247	3265	5	1	
8_14	4286	4354	7		
8_28	4133	4354	7		
9_26	493	622	1		
9_32	927	1244	2		
10_20	2086	2488	4		
20_29	963	1244	2		
11_13	1558	1866	3		
11_25	973	1244	2		
12_13	2122	2488	4		
12_27	1788	1866	3		
12_39	917	1244	2		
14_31	3048	3110	5		
14_37	2217	2488	4		
15_20	3991	4354	7		
15_22	1423	1866	3		
16_41	1696	1866	3		
17_23	1276	1299	2		1
17_37	1690	1866	3		
18_36	1504	1866	3		
19_22	1762	1866	3		
19_33	1413	1866	3		
20_35	3448	3732	6		
21_30	1226	1244	2		
22_29	545	622	1		
23_24	1145	1244	2		
24_34	1316	1399	2	1	
25_42	577	622	1		
27_28	2300	2488	4		
30_33	1232	1866	3		
32_35	1472	1866	3		
34_40	1509	1866	3		
36_41	1145	1244	2		
38_42	706	777	1	1	
39_41	2770	3110	5		

4.8. Conclusions

A novel model for the design of telecommunications networks has been presented and solved with the help of genetic algorithms. The model considers delay, routing, discrete capacity assignment and bi-connectivity.

The GATSP heuristic provided results that compare favorably with those obtained through the use of the Branch Exchange heuristic, and also with the GARANDOM heuristic. In this sense, the use of initial solutions based on TSP solutions, proved to perform better, both in cost and in convergence time than the use of random initial solutions as expected. TSP solutions were used because they are feasible solutions to the network problem.

The use of genetic algorithm to find fast and good solutions to the TSP problem was a key issue in making this method to work well.

CHAPTER 5. BURST ADMISSION CONTROL AND INPUT TRAFFIC SHAPING FOR ATM NETWORKS

5.1. Introduction

This chapter proposes an admission control and a traffic shaping method for BISDN/ATM networks. The method operates at the burst connection level and is based on the Sustainable Cell Rate concept, as defined by the ATM Forum.

The model complies with the Universal Performance Objective, by treating all cells equally regardless of their associated service requirements. This is done by operating the links for the most stringent delay and data loss probability requirements of all the services. An adequate buffer dimensioning controls transfer delay.

A new virtual path is established for every burst to be transmitted, causing the allocated resources to be released after the burst transmission, resulting in a higher network performance.

The shaping or policing function is based on a set of traffic descriptors that the user specifies at the call establishment phase. The appropriate minimum set of traffic descriptors is proposed, studied and discussed.

The performance analysis of the proposed model is done by discrete event simulation. The Quality of Service (QOS) offered to the network is evaluated, as well as the effect of the shaping function on the resource management of the network.

The future ATM network is expected to satisfy a number of key objectives, which may influence directly its topology and, therefore, its performance. Some of the goals to be provided are:

- **Flexibility**, to support a wide range of service types and mixes of them “on demand”.
- **Wide range of uses**, in the sense of accepting and managing new service types as they appear.
- **Simplicity**, by avoiding the need of multiple operations over the cells and several managing, maintenance, flow and congestion control schemes.

- **Efficiency**, which will be allowed by an adequate management of resources and the allocation policy implementation.

Typically there are two main congestion control mechanisms used in networks: **preventive control** which acts at the call establishment phase and **reactive control**, which includes congestion smoothing policies operating at the cell level, such as the leaky bucket type mechanisms among others.

In order to prevent congestion and control the network's traffic load, the best decision is to use preventive control. Since ATM is a connection oriented scheme, then the access control is the most effective method, because the decision of accepting or rejecting a new call attempt is based on the knowledge of the required resources, bandwidth availability and virtual paths. A preventive congestion control module like this should perform the following functions:

- a) Call acceptance control.
- b) Bandwidth management policy.
- c) Bit rate control.

When simplicity, flexibility and wide range of uses are key concepts of the network design, the network will comply with the Universal Performance Objective (UPO) [Woodruff, 1988], which indicates that a truly integrated transport facility should treat all cells equally, regardless of their associated service requirements. The UPO is defined as:

- The most stringent transfer delay requirement of all the services. and
- The most stringent data loss probability requirement of all delay-sensitive services.

Given this performance objective, the transfer delay can be controlled by an adequate buffer dimensioning, since propagation delay over transmission links are fixed and independent of the traffic characteristics and call admission control policy. Considering the ideal condition where the only limiting parameter on the data loss probability is given by the link error probability, it can be concluded that a minimal buffering will be needed.

Once a connection has been admitted into the network, the traffic flow of the connection must be monitored and enforced to ensure that it conforms to its allocated virtual bandwidth, or equivalently, to its traffic descriptor values. Whenever a difference between the offered traffic and

the expected traffic is detected, it will be necessary to adjust the traffic flow to match its declared traffic descriptors. In this sense, the use of a shaping function determines a gain in efficiency. This shaping function may be applied at the cell generation phase as illustrated in figure 5.1.

Inside the network, the intermediate nodes will be aware of the contracted bandwidth. To meet the performance/design objectives already mentioned, the ATM-Forum [1993] defines the following set of functions to manage and control the traffic and congestion (these functions may be used in appropriate combinations):

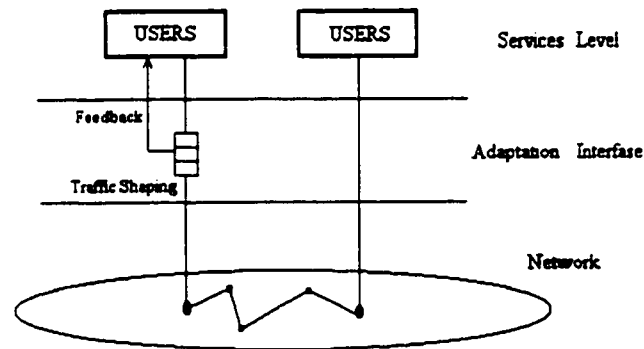


Figure 5.1. Traffic shaping function environment.

- **Network Resource Management (NMR)**, allocates network resources in order to separate traffic flows according to the service characteristics, by the use of Virtual Paths.
- **Connection Admission Control (CAC)**, determines whether a virtual channel/virtual path connection request can be accepted or should be rejected.
- **Feedback controls** regulate the traffic submitted on ATM connections according to the state of network elements.
- **Usage Parameter Control (UPC)**, protects network resources from malicious as well as unintentional misbehavior which can affect the Quality of Service (QOS) of other already established connections.
- **Priority Control**: the user may generate different priority traffic flows by using the Data loss Priority bit.
- **Traffic shaping**, achieves a desired modification on the traffic characteristics of the source.

- **Fast Resource Management**, based on re-routing and connection release.

Even though the proposed model does not require the implementation of all the functions mentioned above, it performs many of them while searching for an adequate resource management scheme.

The organization of the rest of the chapter is as follows: in section 5.2, the network model is described and discussed. Section 5.3 evaluates the feasibility of obtaining a clear and realistic analytical solution taking into account the set of objectives proposed. Simulation results are reported and discussed in section 5.4 and conclusions are drawn in section 5.5.

5.2. Model Description

Since ATM networks are connection oriented (a communication must be set before the data interchange occurs) it is proposed that **a new virtual path be established for every burst to be transmitted over the network**, emulating in that way a connection-less service over a connection oriented network. The model makes the contracted resources to be released after every burst transmission. This also implies that the burst and call level are equivalent. Once a burst is accepted, it will not lose cells, since the network operates according to the Universal Objective Performance.

The access protocol at the burst level, called **Go-and-Store**, is a modified version of the well-known **Tell-and-Go** protocol [Widjaja, 1995, and Woodruff *et al*, 1988]. The source transmits its burst as soon as it receives the message from its application level. A copy of the message is kept at the source until the source realizes that the burst has arrived to the destination successfully. When an intermediate node has to discard the burst due to congestion, it will send a **NACK** message to the source. Subsequently, the source re-transmits the same burst after a back-off time.

Figure 5.2. gives an example of a communication over a three-hop network, using the proposed protocol. It is observed that the burst transmission fails at the first attempt but succeeds at the second attempt.

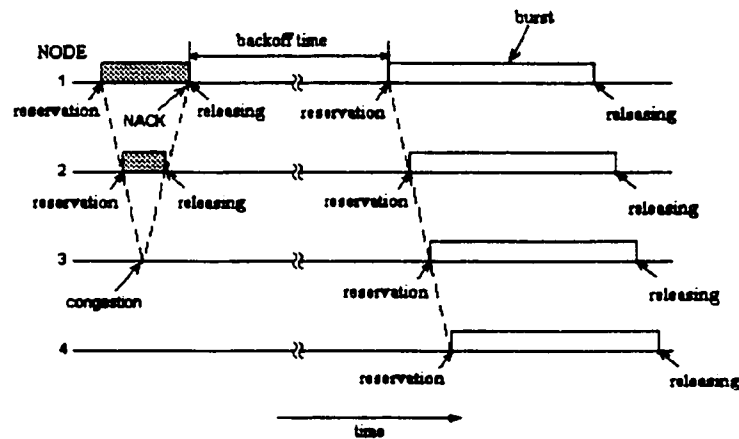


Figure 5.2. Communication with Go-and-Store protocol.

It is important to point out that the first virtual path may not be the same as the second one. This variation will depend on the routing algorithm currently used. In the scheme employed in this thesis, the routing is made up by a distributed asynchronous algorithm [Hong and Suda, 1991] where the network information is in the form of a bit vector. Each bit corresponds to a link and indicates whether or not the link is congested. Nodes independently develop network information vectors and periodically broadcast them to their neighbors.

ATM networks were intended to support many service types such as Constant Bit Rate (CBR) services. When the service is CBR, the access protocol operates based on the peak cell rate declaration as specified by the ATM Forum. A blocking experienced by a CBR source in that case, will correspond to a call blocking.

The general structure of a node is depicted in Figure 5.3. Every node has a certain number of local users (sources) entering traffic to the link, where the input traffic is divided into n classes (each one with k users). An Access Controller Module is connected to every existing traffic source. The output of each Access Controller is statistically multiplexed on the outgoing link by a scheduler, that divides the instantaneous available channel capacity j among the users. Each source connected to the node (by means of call request packets) will ask the local access controller for a bandwidth equivalent to its peak cell rate. The access controller decides whether to accept or to reject the new requirement based on the bandwidth availability that a Capacity

Allocation Controller module calculates. The Capacity Allocation Controller parameter calculation will depend on the resource allocation policy.

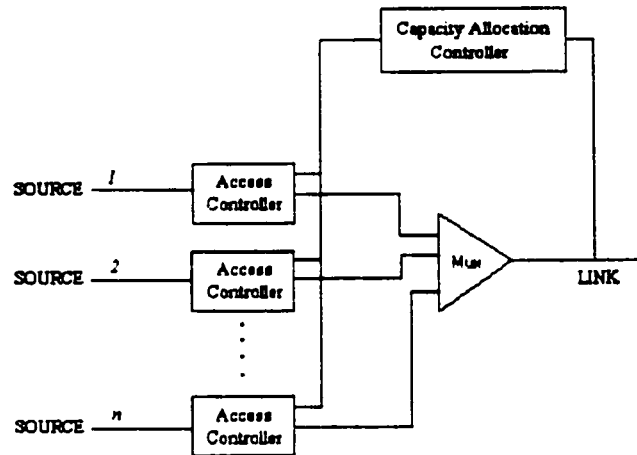


Figure 5.3. Block diagram of the proposed model.

5.2.1 Bit Rate Reduction Strategy

During the call establishment phase, a negotiation takes place between the source interface and the network. Since the proposed model establishes a new connection for every burst to be transmitted over the network, negotiation for resources takes place before every burst transmission.

In order to protect the network from congestion, the access controller shapes the bursts based on a set of traffic parameters that a user specifies during the negotiation. The appropriate set of traffic parameters used for the access controller is discussed later.

A burst shaping is illustrated in Figure 5.4., where B_p is the capacity requirement and T_p is the temporal requirement.

A burst requiring say B_p Mbps and T_p sec will be allocated to a B_s Mbps bandwidth, which will be transmitted over the network for T_s sec. The B_p/B_s , or T_p/T_s rate gives a measure of the applied enforcing. This measure will be used by the network to calculate a new bandwidth offering to the source for its forthcoming burst. The bandwidth will depend on the instantaneous value of

the network-load. The bandwidth offering can be either the sustainable (SCR) or peak cell rate (PCR) of the source.

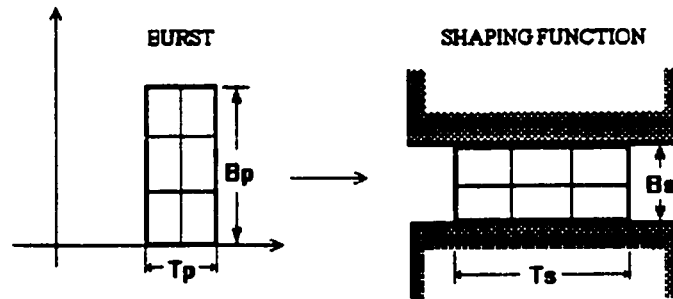


Figure 5.4. Bit Rate Reduction.

A multiplexing gain is expected as a consequence of shaping the input traffic and managing the available bandwidth through an adequate use of the network offering [Ríos and Cárdenas, 1995].

5.3. Analysis

ATM networks are expected to operate with a broad range of traffic types, each one with diverse characteristics. When a network is offered heterogeneous traffic (characterized by Poisson processes), where every source type has its own spatial requirement, the system state can be determined using multidimensional state equations or by use of the Kaufman-Roberts recursion [Kaufman, 1981].

In the Kaufman-Roberts recursion, each source (client) generates bursts (requirements) with Poisson distribution at a mean rate of λ_i (bursts/second). The mean burst duration is of μ_i (sec), requiring b_i (bps), which is the spatial requirement. In case there is not resource availability to accept a new requirement, the connection is lost (blocked).

The entire system can be analyzed as a birth-death process [Kleinrock, 1975], whose state can be simply described by the vector $n=(n_1, n_2, \dots, n_k)$ where n_i is a non-negative random variable denoting the number of type i connections (customers). The basic equation, which

describes the behavior of the system of queues in steady state, obeys the well-known product form solution for a network of queues [Hui, 1990].

$$P(n_1, \dots, n_k) = \begin{cases} \eta \prod_{i=1}^k \rho_i^{n_i} & \text{for } n \in S \\ 0 & \text{otherwise} \end{cases}$$

where η is a normalization factor, which takes into account the zero state probability, that is the probability of an empty system $P(0)$ and is given by:

$$\eta(S) = \sum_{n \in S} \left(\prod_{i=1}^k \rho_i^{n_i} \right)$$

S is the resource sharing policy domain.

Many solutions have been proposed to analyze the equations above [Ireland, 1978, Kaufman, 1981, Beshai, 1989, and Delbrouck, 1983] in order to find out the performance measure of primary interest, which is the probability P_{b_i} that a type i arrival (requiring b_i bandwidth units) is blocked.

The Kaufman-Roberts recursion gives a very useful tool to calculate those measures of interest with exact results.

On the other hand, when the input traffic comprises non-Poissonian streams, the convenience of simple and exact formulations is lost. However many of the methods used to treat non-Poissonian inputs are based on characterizing those input streams by means of series of Poissonian streams. Delbrouck presents the solution of the problem of heterogeneous traffic with different peakedness factors and channel requirements, and verifies the Kaufman-Roberts recursion (where the peakedness factor Z equals one), he also presents his solution to the generalized case where the peakedness factor $Z > 1$. The two main contributions of Delbrouck's solution are an approximation to the calculation of the individual blocking probability P_{b_i} and the study of the blocking probability as a function of the peakedness value. He shows by recursive studies that to great values of Z , and low bandwidth requirements, the blocking probability is close to that of streams with lesser peakedness factor and greater bandwidth requirements. This

study verifies the importance and hence, the difficulty of studying the blocking probability as a general performance measure, in a heterogeneous environment.

An exact method to calculating the overall blocking probability for any burst arriving at the network (no matter its associated service and bandwidth requirement), is given by Weinstein [Weinstein, 1978] in the context of TASI (Time Assigned Speech Interpolation), referred to as the TASI or zero buffer formula. The TASI formula takes into account a superposition of on-off input streams and the specification of the peak and mean bit rate is needed. Murase [Murase *et al.*, 1991] presents the extension of Weinstein's formula to the "multi-class" environment. Nevertheless, asking the user to specify its future mean bit transfer rate is quite unrealistic.

Given the individual blocking probability P_{bi} the overall blocking probability PB can be calculated by:

$$PB = \sum_{i=1}^k p_i P_{bi} \quad \text{where } p_i = \rho_i / \rho \text{ is the probability that an arriving burst is an } i \text{ burst, and}$$

$$\rho = \sum_{i=1}^k \rho_i \text{ is the total normalized arriving rate.}$$

The throughput can then be easily calculated by:

$$\gamma = (1 - PB) \rho$$

5.3.1 Resource sharing policy

The chosen resource sharing policy, is expected to fairly manage the network resources. In completely heterogeneous environments, it will be necessary to limit the accessibility of resources by traffic sources in order to avoid an unbalanced utilization of the resources.

This chapter studies three sharing policies. Two of them are based on the Complete Sharing policy (CS), where the requirements are the peak (PCR model) or the sustainable data-transfer rate (SCR model) respectively. The third one is based on the square root rule (called SQR model) [Irland, 1978].

The SQR model limits the resource accessibility to a fixed value, which completely depends on the instantaneous network load. The value is:

$$U = \frac{C_o - j}{\sqrt{i}}$$

where $C_o - j$ represents the instantaneous free bandwidth, C_o is the total link capacity, and i is the number of instantaneous traffic types over the link.

The SCR model shapes the input traffic. The main objective of reducing the bit rate by shaping the input traffic is to diminish the discrepancy between the required and the natural traffic flow, allocating at the same time a minimum virtual bandwidth for the data transfer in order to efficiently manage the network resources.

The ATM Forum defines a set of user traffic parameters, some of them required and some optional. The parameters are defined in an operational way based on the Generic Cell Rate Algorithm (GCRA). The Peak Cell Rate traffic parameter specifies an upper bound on the traffic that can be submitted onto an ATM connection. It is defined and related to the Transfer delay Variation Tolerance (CDV tolerance) by $GCRA(T, \tau)$. T is the inverse of the contracted PCR and represents the peak emission interval and τ is the CDV Tolerance parameter.

The Sustainable data-transfer rate traffic parameter specifies a greater and allowable emission interval. This sustainable emission interval (T_s) is specified by the user and indicates a measure of how a burst can be enlarged in time while still conforming to $GCRA(T_s, \tau_s)$ where τ_s is the Burst Tolerance. In relation to T , $T_s > T$.

A key parameter to shape any burst type is the minimum spacing between bursts (T_r) and the maximum burst size to its related service type. The shaper takes advantage of the minimum spacing between bursts to reduce the bit rate (spatial requirement).

5.3.2 Shaper Model

The shaper model allows the user to emit at most B cells with a minimum emission interval T (that is at its peak cell rate, PCR) and a minimum spacing between bursts of T_r . Cells could be sent through the network at the sustainable or peak cell rates (depending on the network load) specified by the user. Figure 5.5 shows the shaper model.

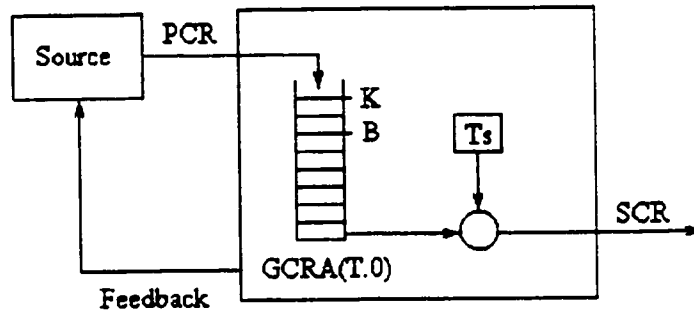


Figure 5.5. Shaper Model

The shaper will stop the source as soon as it detects a non-conforming cell according to GCRA(T,0) (peak cell rate emission) and may transmit the cells at the sustainable data-transfer rate. B is the maximum burst size for the source; those B cells are stored at a buffer before transmission, however, the minimum buffer size will be set according to a maximum tolerable delay of all services T_m , and may satisfy:

$$K = \frac{T_m \cdot C_o}{L}$$

where L is the cell size (in bits) and T_m is expressed in seconds [Saito, 1992].

The minimum set of traffic parameters that a user specifies at the call establishment phase can now be listed as:

- Peak cell arrival time, T , the inverse of PCR.
- Sustainable cell arrival time T_s , the inverse of SCR.
- Minimum time spacing between bursts, T_r .
- CDV Tolerance, τ .
- Burst Tolerance, τ_s .

It should be noted that the mean cell rate is not listed above. The set of traffic parameters and its correct specification guarantees the conformance with the traffic contract.

5.4. Simulation results and discussions

The proposed model has been tested by discrete event simulation. Two source types have been characterized and simulated: data transmission and packetized voice.

The cell stream of a single voice source is characterized by arrivals at fixed intervals of T seconds during talk-spurts and no arrivals during silences. It is assumed that successive talk-spurts and silence periods form a renewal process. Each talk-spurt has a random length of NT and each silence period is of random length X . An important assumption is that the number N of cells in a talk-spurt is geometrically distributed. Assuming that the voice packetization period is $T=6$ ms, the mean number of cells per talk-spurts is $E(N)=59$ so that the mean talk-spurt time is 352 ms. The silence periods are exponentially distributed with mean of 650 ms. [Brady, 1968]. In fact, any distribution of X could be used in our analysis [Sriram and Whitt, 1986]. There is no silence which duration is less than 200 ms.

As mentioned earlier, the sustainable data-transfer rate will depend on the minimum inter burst spacing as well as the size of the burst. The maximum burst size could be estimated by the Chebyshev's inequality, so that if it is assumed that the probability that the burst size will be in the closed interval $[\mu - 4\sigma, \mu + 4\sigma]$ is 0.9375, the maximum burst size results of 297 cells.

The biggest burst of 297 cells will require a transmission time (at peak cell rate) of 297×6 ms = 1.782 seconds. The total time will be of $1.782 + T_r = 1.982$ seconds, which divided by the number of cells of the biggest burst (297) gives us the sustainable data-transfer rate value of $T_s=6.67$ ms. ($T_s=1/SCR$).

Figure 5.6 shows the sustainable data-transfer rate concept graphically and its effect on the resource allocation.

Figure 5.6-a shows a typical cell stream at the emission level. In Figure 5.6-b, the burst longing effect in time can be seen (the burst transmission over the network will be in that shape) and Figure 5.6-c presents the burst shaped at the destination point. The peak bit rate for the packetized voice source type is of 64 Kbps.

The cell stream of a single data source, like the packetized voice service, can be characterized by arrivals of T seconds during transmission periods and no arrivals during silences.

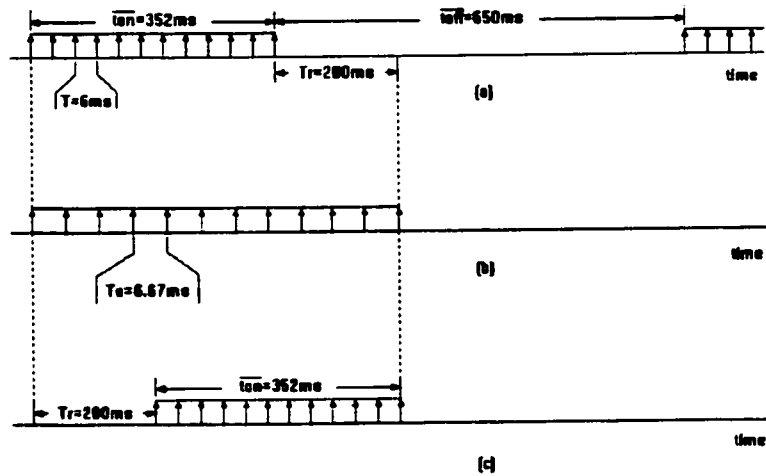


Figure 5.6. Voice transmission based on the sustainable data-transfer rate.

Each transmission period has a random length of NT and each silence period is of random length X . It is assumed that the number N of cells in a t_{on} period is geometrically distributed. For a data service facility of 10 Mbps of peak rate and 2 Mbps of mean rate, the packetization period is $T=38.4 \mu\text{s}$, the mean number of cells per t_{on} time is $E(N)=52.08$ Kcells so that the mean transmission time is 2 seconds. The silence periods are exponentially distributed with mean of 8 seconds. Here again, any distribution of X could be used in our analysis. The minimum spacing between bursts was assumed to be 3 seconds.

Like the packetized voice service, the sustainable data-transfer rate parameter can be calculated by the Chebyshev's inequality. If a maximum burst size of 278.88 Kcells is considered, the sustainable data-transfer rate obtained will be $T_s=49.1 \mu\text{s}$.

During simulation, it has been assumed that data sources make transmission retries once they perceive a blocking. Every re-transmission will be done after a certain randomized time (back-off time) emulating in that way a connection-less service over a connection oriented network which is ATM by definition. In LAN networks the back-off time varies between one and 1024 time slots, depending on the number of detected collisions. In ATM networks, where a highly heterogeneous environment is expected, the back-off time parameter should take into account the actual traffic aggregate (recall that an Integrated services network will manage continuous bit rate services among others). A blocking probability graph in function of the

maximum back-off time is shown in figure 5.7. [100% of network load and using peak cell rate allocation].

In the simulation model, the back-off time is normally distributed, and varies between 28.3 μ s (equivalent to 10 time slots in ATM) and 100 ms.

As a comment, it is believed that if the t_{on} time for the data service would be lesser than 2 seconds, the Figure 5.7 graphic would be quite different in shape, that is, it would be sharply decreasing. The back-off time effect is worth to be further studied. Figure 5.8 presents a comparison of the three models studied.

Define trunk group utilization ρ of a type of terminal as the number of calls established times the average bit rate, divided by the total capacity. Thus, a fully utilized ($\rho=1$) trunk group of 100 Mbps bandwidth can support 50 terminals of data type (peak rate of 10 Mbps and mean rate of 2 Mbps).

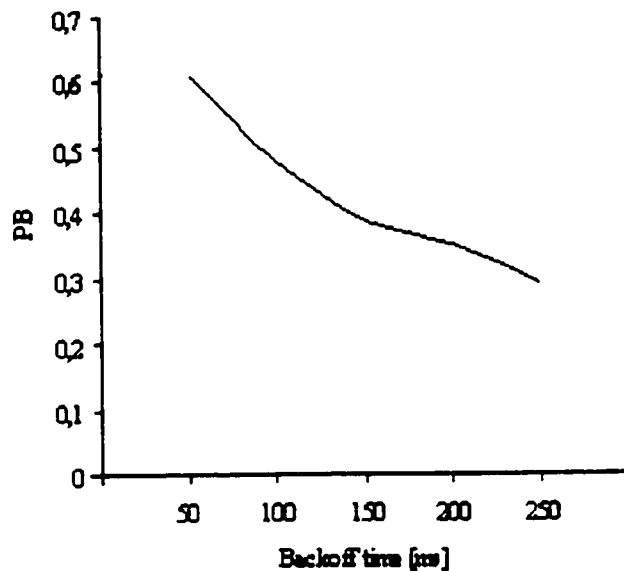


Figure 5.7. Back-off time Vs. Blocking probability.

In the performance graphs the traffic load for the packetized voice service type is kept constant at $\rho_1=0.3$ while the traffic load for the data service type (ρ_2) is varied.

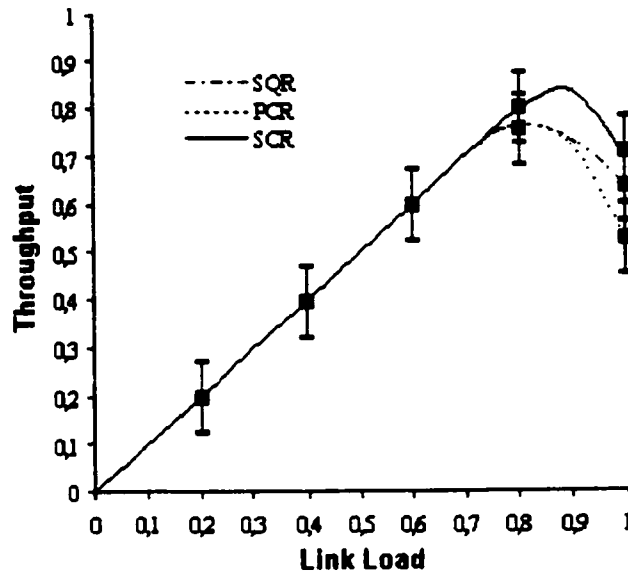


Figure 5.8. Comparison of the three schemes.

As expected, to light loads (under 50% of the link load) the three schemes present the same results. The study of the model at the “normal operating range” (load between 60% and 100% of link capacity) is of special interest.

SCR model gives the expected gain. In this model, every burst is shaped before being transmitted over the network based on the traffic parameters previously declared by the source, regardless of the link load conditions.

PCR model’s performance seems to present a slight improvement if compared to other homogeneous approaches [Huy, 1988]. This is because in the model here simulated, the data sources make re-transmission retries after experiencing a blocking. *It may increase the throughput of the network.*

SQR model works according to the following set of heuristics. When a burst, no matter its associated service tie, arrives at the CAC module, it asks for a spatial requirement equal to its peak rate. If the peak requirement value is less than the of the network bandwidth offering, the requirement is accepted and the transmission is set at the peak rate.

On the other hand, if the peak requirement is greater than the offering of the network, the source asks for a spatial requirement equivalent to its sustainable rate. If the sustainable

requirements value is less than the networks offering, the requirement is accepted and the transmission is performed.

If neither the peak requirement nor the sustainable requirement is accepted, the call is blocked. In the case of the packetized voice service type, if a burst is blocked, it is lost.

An important performance parameter is the blocking probability experienced by a determined source type. Figure 5.9 presents data type blocking probability. It can be seen that to higher loads SCR and SQR models perform a better behavior than PCR model, however, even though high, it should be taken into account that data terminals perform transmission retries when blocked, therefore, assuming binomial distribution (number of succeeds in n Bernoulli trials), and adopting the LAN's model which limits the number of retries to 15, the lost probability corresponding to a blocking probability of 0.3 will be 0.3^{15} , that is the order of 14×10^{-9} . This means that practically there is no burst losing.

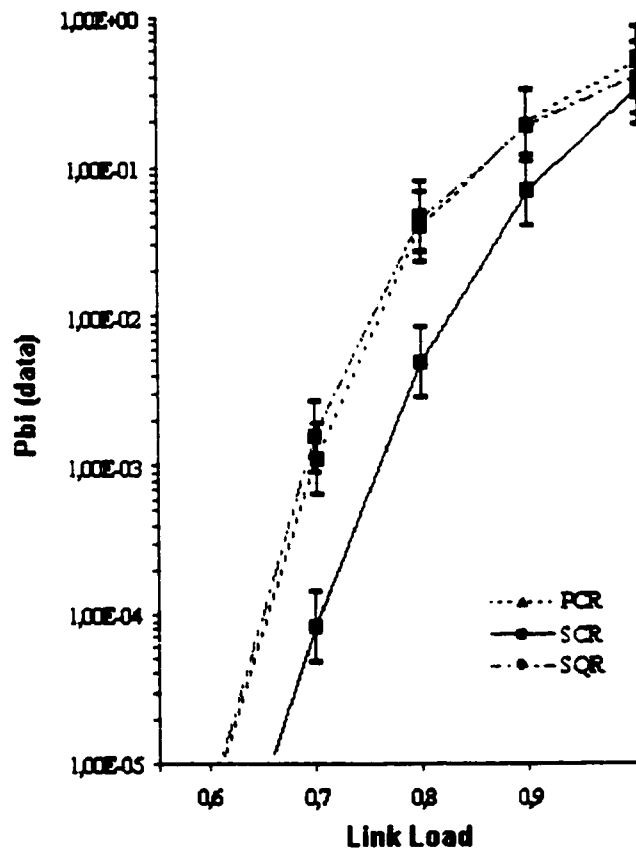


Figure 5.9. Data blocking probability.

Figure 5.10 shows the blocking probability experienced by voice terminals. There has not been found blocking to SQR model. The blocking shown in Figure 5.10 is presented between 1 and 1.6 of link load.

The reason why it is clearly lesser than the blocking experienced by data terminals is because packetized voice requires a really small bandwidth if compared to data service type. It should be reminded that in this particular model, when a burst (voice burst) is blocked, it is lost. Losing a burst under this condition means that an average of 352 ms of conversation is lost.

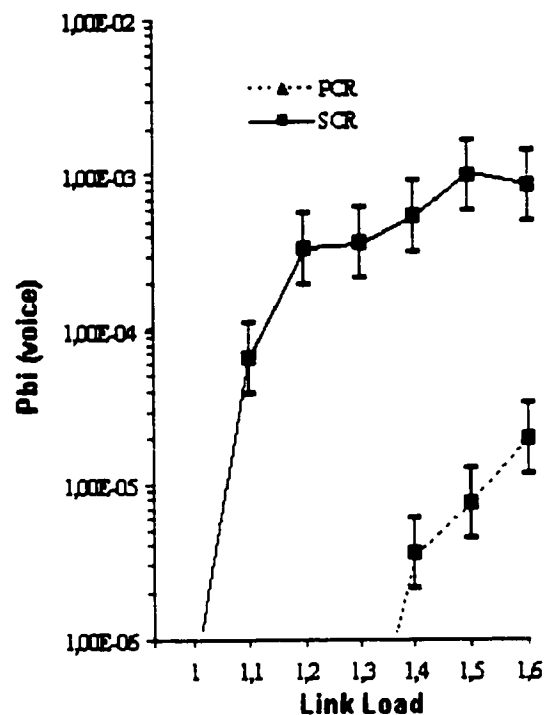


Figure 5.10 Packetized voice blocking probability.

5.5. Conclusions

This chapter has focused on the congestion problem in broadband networks. An admission control and a traffic shaping method for BISDN/ATM networks has been proposed. The method operates at the burst connection level and is based on the Sustainable data-transfer rate concept. The model complies with the Universal Performance Objective. A new virtual path is established

for every burst to be transmitted, causing the allocated resources to be released after the burst transmission, resulting in a higher network performance.

The shaping or policing function is based on a set of traffic descriptors that the user specifies at the call establishment phase. The appropriate minimum set of traffic descriptors was proposed, studied and discussed.

Obtained results indicate that on the “normal operating range”, Sustainable data-transfer rate assignment is preferable because of the gain obtained. Besides, since SCR model shapes the input traffic based on user declared parameters, it may not affect the QOS objectives.

The model presented is flexible, since it is designed to support a wide range of services, with total control. Also the model is scalable, by allowing the set of services to grow or to add new types of services. It is also simple and efficient, by avoiding multiple operations over each cell, increasing the quality of service. Once the connection is accepted, cells will not be lost.

One important aspect is that the traffic parameters to be specified by the user, do not include the average transfer rate since the user does not know its future communication behavior.

CHAPTER 6. CONCLUSIONS

In this thesis, two of the many problems involved in telecommunication networks design have been addressed. The first is the **connection admission control (CAC)**, which refers to the set of actions, taken at connection time, to decide whether to accept or not the request for a new call/connection. The decision is based on the connection's traffic characteristics, the required QoS and the network load. If the request is accepted, the necessary resources are allocated to the connection. The second problem is the **network-wide link sizing problem**, which selects the topology of the (ideally) optimal network interconnecting the nodes, based on traffic characterization, nodal layout, routing and admission control procedures.

In chapter 2, two models have been formulated for the design of telecommunications networks, whose novel characteristic is the explicit treatment of the Cell-Loss Probability and the Transfer delay Variation. This is done by keeping a short buffer length at each switch, so that the cell-delay variation is not too high. As the buffer length is shorter, the higher the cell-loss probability. Thus, the models choose locations for the switches in such a way that the cell-loss probability is kept constrained. Computational experience is offered with one of the models, that shows that, as the cell-loss probability is decreased, the locations of the switches do not change importantly, but the allocations of users to switches do change.

In chapter 3, four mathematical formulations have been presented for the analysis of the Fault Tolerant Network Design Problem. Three of them are new. Furthermore a Lagrangean Relaxation and Heuristics methods have been presented to solve the problem for larger instances. The computational results carried out over randomly generated networks show that the relaxation procedure is effective in producing low-cost solutions in reasonable time even for fairly large instances. The relaxation method proved to be superior to the comparison heuristics considered, always producing solutions with lower costs. Further work could be directed towards improving the lower bounds and developing relaxations that include reservation.

In chapter 4, a novel model for the design of telecommunications networks has been presented and solved with the help of genetic algorithms. The model considers delay, routing, discrete capacity assignment and bi-connectivity.

The GATSP heuristic provided results that compare favorably with those obtained through the use of the Branch Exchange heuristic, and also with the GARANDOM heuristic. In this sense, the use of initial solutions based in TSP solutions, proved to perform better, both in cost and in convergence time than the use of random initial solutions.

The use of genetic algorithm to find fast and good solutions to the TSP problem was a key issue in making this method to work well.

In chapter 5, an admission control and a traffic shaping method for BISDN/ATM networks has been proposed. The method operates at the burst connection level and is based on the Sustainable data-transfer rate concept. The model complies with the Universal Performance Objective. A new virtual path is established for every burst to be transmitted, causing the allocated resources to be released after the burst transmission, resulting in a higher network performance.

The shaping or policing function is based on a set of traffic descriptors that the user specifies at the call establishment phase. The appropriate minimum set of traffic descriptors was proposed, studied and discussed.

Obtained results indicate that on the “normal operating range”, Sustainable data-transfer rate assignment is preferable because of the gain obtained. Besides, since SCR model shapes the input traffic based on user declared parameters, it may not affect the QOS objectives.

The model presented is flexible, since it is designed to support a wide range of services, with total control. Also the model is scalable, by allowing the set of services to grow or to add new types of services. It is also simple and efficient, by avoiding multiple operations over each cell, increasing the quality of service. Once the connection is accepted, cells will not be lost.

We do believe that our results will be found very useful in the telecommunications industry for dimensioning large high-speed networks.

REFERENCES

- Ahuja, R., Magnanti, T., Orlin J., (1993) *Network Flows*. Prentice-Hall, New Jersey, 605-620.
- Alcalde, J., (1993), *Introducción a la Teleinformática*, McGraw Hill.
- Alevras, D., Grötschel, M., Jonas, P., Paul, U., Wessály, R., (1998). "Survivable Mobile Phone Network Architectures: Models and Solution Methods". *IEEE Communications Magazine*, 36(3), 88-93.
- Alvarez, M.; Ríos, M.; Marianov, M., (1996) "Dimensionamiento de Enlaces en Redes ATM mediante Algoritmos Genéticos". *Proceedings VII Seminario Nacional y I Congreso Internacional en Telecomunicaciones*, Valdivia, Chile, 120-125.
- Alvarez, M.; Ríos, M., (1997) "Genetic Algorithm for the Solution of the Minimum Cost Link Sizing Problem on an ATM Network". *Proceedings of the 1997 Summer Computer Simulation Conference*, Arlington, Virginia, 709-714.
- Amiri, A., Pirkul, H., (1997) "New Formulation and Relaxation to solve a Concave-Cost Network Flow Problem". *Journal of the Operational research Society*, 48, 278-287.
- ATM-Forum (1993), *ATM User Network Interface Specification: Version 3.0*, Prentice Hall, New Jersey.
- Back T., Hoffmelster H., (1991), "Extended Selection Mechanisms in Genetic Algorithms", *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Mateo, California, USA, Morgan Kaufmann Publishers, 92-99.
- Ball, M., Magnanti, T., Monma, C., Nemhauser, G., (1995), *Network models*, Elsevier, Amsterdam.
- Barahona, F., (1996), "Network Design Using Cut Inequalities". *SIAM Journal on Optimization*, 6(3), 823-837.
- Batta R, (1988): "Single Server Queueing - Location Models with Rejection", *Transportation Science*, 22, 209 - 216.
- Batta R, Larson R, Odoni A, (1988): "A Single - Server Priority Queueing - Location Model", *Networks*, 8, 87 - 103.
- Batta R, (1989): "A Queueing - Location Model with Expected Service Time dependent Queueing Disciplines", *European Journal of Operational Research*, 39, 192 - 205.
- Batta R, Dolan J, Krishnamurthy N, (1989): "The Maximal Expected Covering Location Problem: Revisited", *Transportation Science*, 23(4), 277 - 287.
- Berman O, Larson R, Chiu S, (1985): "Optimal Server Location on a Network Operating as a M/G/1 Queue", *Operations Research*, 12(4), 746 - 771.
- Berman O, Larson R, Parkan C, (1987): "The Stochastic Queue p - Median Location Problem", *Transportation Science*, 21, 207 - 216.
- Berman O, Mandowsky, R, (1986): "Location - Allocation on Congested Networks", *European Journal of Operational Research*, 26, 238 - 250.
- Bertsekas, D. Gallager R., (1992), *Data Networks*, Prentice Hall.
- Beshai, M., (1989), "The Poissonian-Spectrum Method for Treating a Loss System Serving Non-Poissonian Multi-Bit-Rate Traffic", *IEEE Infocom 89*, Ottawa, Ont., Canada.
- Bienstock, D., Gunluck, O., (1995), "Computational Experience with a Difficult Mixed-Integer Multicommodity Flow Problem". *Mathematical Programming*, 68, 213-237.

- Blickle, T., Thiele, L., (1995), "A Comparison of Selection Schemes used in Genetic Algorithms", TIK Report N° 11.
<http://www.tik.ee.ethz.ch/tik/research/publications/Publications.html>.
- Bolla, R., Davoli, F., Marchese, M., (1994), "A Distributed Routing and Access Control Scheme for ATM Networks", *IEEE Supercomm/ICC 94*, New Orleans.
- Brady, P. (1968), "A Statistical Analysis of On-Off Patterns in 16 Conversations", *The Bell System Technical Journal*, **47**, No. 1, 73-91.
- Church R, ReVelle C, (1974): "The Maximal Covering Location Problem", *Papers of the Regional science Association*, **32**, 101 - 118.
- Clarke, Ll., Anandalingam, G., (1995) "A Bootstrap Heuristic for Designing Minimum Cost Survivable Networks". *Computers Operations Research*, **22**, No 9, 921-934.
- Cormen, T. , (1990), *Introduction to Algorithms*, The MIT Press, Cambridge.
- Dahl, G., Stoer, M., (1994), "A Polyhedral Approach to Multicommodity Survivable Network Design". *Numerische Mathematik*, **68**, 149-167.
- Dahl, G., Stoer, M., (1998) "A Cutting Plane Algorithm for Multicommodity Survivable Network Design Problems". *INFORMS Journal on Computing*, **10**, No. 1, Winter 1998.
- Davis, L., (1987), *Genetic Algorithms and Simulated Annealing*, London Pitman.
- Delbrouck, L., (1983), "On the Steady-State Distribution in a Service Facility Carrying Mixtures of Traffic with Different Peakedness Factors and Capacity Requirements". *IEEE Transactions on Communications*, **31**, No. 11, 1209-1211.
- DeNardo E., (1982), *Dynamic Programming*, Prentice Hall, Englewood Cliffs.
- Dorigo, M., Bersini, H., (1994) "A Comparison of Q-Learning and Classifier Systems", *Proceedings of: From Animals to Animats, Third International Conference on SIMULATION of Adaptive Behavior (SAB94)*, Brighton, UK, MIT Press, 248-255.
- Eckberg, A. E. (1992), "B-ISDN/ATM Traffic and Congestion Control", *IEEE Network Magazine*, **6**, No. 5, 28-37.
- Elbaum, R., Sidi, M., (1995), "Topological Design of Local Area Networks Using Genetic Algorithms", *Proceedings IEEE Infocom '95*, 64-71.
- Fischer, W., Wallmeier, E., Worster, T., Davis, S. P., Hayter, A. (1994), "Data Communications Using ATM: Architectures, Protocols and Resource Management", *IEEE Communications Magazine*, **32**, No. 8, 24-33.
- Fogel, D., (1995), *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ.
- Garey, M., Johnson, D. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, California.
- Gallasi, G., Rigolio, G., Fratta, L. (1989), "ATM: Bandwidth Assignment and Bandwidth Enforcement Policies", *IEEE Globecom 89*, Dallas, Texas.
- Gavish, B., Trudeau, P., Dror, M., Gendreau, M., Mason, L., (1989) "Fiberoptic Circuit Network Design under Reliability Constraints". *IEEE Journal on Selected Areas in Communications*, **7**, 1181-1187.
- Gavish B, (1990): "Topological Design of Computer Communications networks - The overall design problem", *European Journal of Operational Research*, **58** 149 - 172.
- Gavish B, (1991): "Topological Design of Telecommunications networks - Local access design methods", *Annals of Operations Research*, **33**, 17 - 71.

- Gerla, M., (1973), "Deterministic and adaptive routing policies in packet switched computer networks", *Third Data Communications Symposium*, Tampa, 13-15.
- Gerla M., Kleinrock L., (1977) "On the topological Design of Distributed Computer Networks", *IEEE Transactions on Communications*, 48-60.
- Girard, M., (1990), *Routing and Dimensioning in Circuit-Switched Networks*. Addison Wesley, Reading.
- Goldberg D. , (1989), *Genetic Algorithms in Search Optimization & Machine Learning*. Addison Wesley, Reading.
- Goldstein, M., (1983) "Design of Long Distance Telecommunications Networks: The TELPAK Problem". *IEEE Transactions on Circuit Theory*, May 1983, 186-192.
- Grötschel, M., Monma, C., Stoer, M., (1992) "Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constarins". *Operations Research*, 40, No 2.
- Grötschel, M., Monma, C., Stoer, M., (1995), "Design of Survivable Networks". In *Network Models of Handbooks in Operations Research and Management Science*, Chapter 10, 617-672.
- Hakimi S. L., (1964): "Optimal Locations of Switching Centers and the absolute Centers and Medians of a Graph", *Operations Research*, 12, 450 - 459.
- Hines, W., Montgomery, D (1990), *Probability and Statistics in Engineering and Management Science*, Wiley-Interscience, New York.
- Holmberg K, Hellstrand J, (1994): "Solving the Uncapacitated Network Design Problem by a Lagrangean Heuristic and Branch and Bound", *Working Paper LiTH-MATH-R-1994-11*, Department of Mathematics, Linköping University - Sweden.
- Holland J., (1992), *Adaptation in Natural and Artificial Systems*, U. of Michigan Press, Ann Arbor; reprinted by The MIT Press, Cambridge.
- Hong, D., Suda, T. (1991), "Congestion Control and Prevention in ATM Networks", *IEEE Network Magazine*, Vol 5, No. 4, 10-16.
- Hui, J. Y., (1988), "Resource Allocation for Broadband Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, 1598-1608.
- Hui, J. Y., (1990), *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Publishers, Boston.
- Irland, M. Y., (1978), "Buffer Management in a Packet Switch", *IEEE Transactions on Communications*, Vol. 26, No. 3, 328-337.
- ITU-T/I.371 (1993), "Traffic Control and Congestion Control in B-ISDN", *International Telecommunication Union*, Telecommunication Standarization Sector.
- Johnson, D., Lenstra, J., Rinnooy, G.,(1978), "The complexity of the network design problem". *Networks*, 8, 279-285.
- Kaufman, J. S. (1981), "Blocking in a Shared Resource Environment", *IEEE Transactions on Communications*, Vol. 29, No. 10, 1474-1481.
- Kershenbaum A. , (1993), *Telecommunications Network Design Algorithms*, McGraw-Hill, New York.
- Kleinrock L., (1964), *Communications Nets: Stochastic Message Flow and Delay*, McGraw Hill, N.Y.
- Kleinrock, L. (1975), *Queueing Systems Vol. 1:Theory*, Wiley-Interscience, New York.

- Ko, K., Tang, K., Chan, C., Man, K., Kwong, S., (1997), "Using Genetic Algorithms to Design Mesh Networks", *IEEE Computer*, 56-61.
- Law, A., Kelton, D. (1982), *Simulation, Modeling and Analysis*, McGraw-Hill, New York.
- Lea, C. (1992), "What Should Be the Goal for ATM", *IEEE Network Magazine*, Vol. 6, No. 5, 60-66.
- Lee, T., Lai, K., Duann, S. (1994), "Real Time Call Admission Control for ATM Networks with Heterogeneous Bursty Traffic", *IEEE Supercomm/ICC 94*, May 1994, New Orleans.
- Lee, K., Park, K., Spark, S.,(1995) "Spare Channel Assignment for DCS Mesh-Restorable Networks". *Proceedings of the 3rd International Conference on Telecommunication Systems*, Nashville, 296-307.
- Lawler, E., (1985), *The Traveling Salesman Problem : a Guided Tour of Combinatorial Optimization*, Chichester, Wiley.
- Magnanti, T., Mirchandani, P., Vachani, R., (1995) "Modeling and Solving the Two-facility Capacitated Network Loading Problem". *Operations Research*. **43**(1), 142-156.
- Marianov V., ReVelle C., (1994): "The Queueing Probabilistic Location Set Covering Problem and Some Extensions", *Socio-Economic Planning Sciences*, **28**(3), 167 - 178.
- Marianov V., ReVelle C., (1996): "The Queueing Maximal Availability Location Problem: A Model for the Siting of Emergency Vehicles", *The European Journal of Operational Research*, **93**(1), 110 - 120.
- Marianov, M.; Ríos, M.; ReVelle, C., (1990) "Modelos Matemáticos para la Ubicación Optima de Recursos en Redes de Comunicaciones", *Apuntes de Ingeniería*, **37**, 99-111.
- Marianov, V.; Pérez, M.; Ríos, M., (1995) "Location of Nodes on Communications Networks Considering a User Pre-Specified Minimum Reliability under Failures". *Proceedings III International Conference on Telecommunication Systems*, Nashville, Tennessee, 494-500.
- Marianov, V.; Ríos, M., (1996) "Queueing Plant Location and p-Median Models for the Design of Circuit Switched Telecommunication Networks with Constrained Buffer Lengths at Switches". *Proceedings ALIO-EURO Conference*, Valparaiso, Chile, 115-127.
- Minoli, D. (1993), *Broadband Network Analysis & Design*, Artech House, Boston.
- Minoux, M., (1981) "Optimum Synthesis of a Network with Non-Simultaneous Multicommodity Flow Requirements". *Studies on Graphs and Discrete Programming*, P. Hansen, North-Holland, Amsterdam, 269-277.
- Monma, C., Shalcross, D., (1989) "Methods for Designing Communications Networks with Certain Two-Connected Survivability Constraints". *Operations Research*, **37**, No. 4, July-August 1989.
- Monma, C., Munson, B., Pulleyblank, W., (1990) "Minimum-weight Two-connected Spanning Networks". *Mathematical Programming*, **46**, 153-171.
- Murase, T., Suzuki, H., Sato, S., Takeuchi, T. (1991), "A Call Admission Control Scheme for ATM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 9, 1461-1470.
- Orvosh, D., Davis, L., (1993), "Shall we repair? Genetic algorithms combinatorial optimization, and feasibility constraints", *Proceedings of the Fifth International Conference on Genetic Algorithms*, University of Illinois at Urbana - Champaign, pp. 650.
- Papadimitriou, C., Steiglitz, K., (1982), *Combinatorial Optimization*, Prentice Hall, Englewood Cliffs.

- Perros, H., Elsayed, K., (1996) "Call Admission Control Schemes: A Review". *IEEE Communications Magazine*, 34(11), 82-91.
- Pirkul H, Nagarajan V., (1992): "Locating concentrators in centralized computer networks", *Annals of Operations Research*, 36, 247 - 262.
- Pirkul H, Narasimhan S. and De P, (1988): "Locating Concentrators for Primary and Secondary Coverage in a Computer Communications Network", *IEEE Transactions on Communications*, 36 N° 4, 450 - 458.
- Pohlheim, H., (1996), *GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with MATLAB Version 1.83*, pohlheim@systemtechnik.tu-ilmenau.de
- Reinelt, G., (1995), *Library of sample instances for the TSP*, <ftp://elib.zib-berlin.de/pub/mp-testdata/tsp/index.html>
- Rigolio, G., Verri, L., Fratta, L. (1991), "Source Control and Shaping in ATM Networks", *IEEE Globecom 91*, Phoenix, Arizona.
- Ríos, M., Marianov, V., (1999) "Genetic Algorithms for Finding Good Solutions of the Traveling Salesman Problem", *Proceedings of the 1999 Summer Computer Simulation Conference*, Chicago, Illinois, 127-130.
- Ríos M., Marianov, V., (1999) "Location Models for the Design of Telecommunications Networks with Constrained Buffer Length at Switches", *sent to the Annals of Operational Research*
- Ríos, M., Gutierrez, M., Marianov, V., (1999) "Survivable Capacitated Network Design Problem: New Formulation and Lagrangean Relaxation", *sent to the Journal of the Operational Research Society*.
- Ríos, M.; Cárdenas, R., (1995) "Control de Acceso y Conformación de Tráfico en Redes de Comunicaciones de Banda Ancha", *Apuntes de Ingeniería*, 18, No.4, 25-46.
- Ríos, M.; Marianov, V.; Cárdenas, R., (1996) "A Connection Admission Control Method for B-ISDN Networks Based on Shaping the Input Traffic and the Sustainable Cell Rate Concept". *Proceedings 14th IASTED International Conference on Applied Informatics*, Innsbruck, Austria, 283-286.
- Saito, H. (1991), "Dynamic Call Admission Control in ATM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, 982-989.
- Saito, H. (1992), "Call Admission Control in an ATM Network Using Upper Bound of Cell Loss Probability", *IEEE Transactions on Communications*, Vol. 40, No. 9, 1512-1521.
- Saito, H. (1994), *Teletraffic Technologies in ATM Networks*, Artech House, Boston.
- Schwefel, H., (1987), "Collective Phenomena in Evolutionary Systems", *31st Annual Meeting Inter. Soc. for General System Research*, Budapest, 1025-1033,
- Sedgewick, R., (1992), *Algorithms in C++*. Addison-Wesley.
- Sriram, K., Whitt, W. (1986), "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data", *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, 833-846.
- Stoer, M., (1992) *Design of Survivable Networks*. Lecture Notes in Mathematics, Springer.
- Toregas C, Swain R, ReVelle C, Bergman L, (1974): "The Location of Emergency Services Facilities", *Operations Research*, 19, 1363 - 1373.
- ReVelle C. and Swain S, (1970) "Central Facilities Location", *Geographical Analysis*, 2, 30 - 42.
- Weinstein, C. J. (1978), "Fractional Speech Loss and Talker Activity Model for TASI and for Packed-Switched Speech", *IEEE Transactions on Communications*, Vol. 26, No. 8.

- Widjaja, I. (1994), "Random Access for ATM LANs and WANs", *IEEE Supercomm/ICC 94*, May 1994, New Orleans.
- Widjaja, I. (1995), "Performance Analysis of Burst Admission Control Protocols", Approved by Proceedings IEE, to be published during 1995; available from author upon request.
- Whitley D., (1989), "The GENITOR Algorithm and Selection Pressure: 41y Rank-Based allocation of Reproductive Trials is Best", *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, California, USA: Morgan Kaufmann Publishers, 116-121.
- Wolff R., (1989): *Stochastic Modeling and the Theory of Queues*, Prentice-Hall, Englewood Cliffs, NJ.
- Woodruff, G., Rogers, R., Richards, P., (1988), "A Congestion Control Framework for High-Speed Integrated Packetized Transport", *Proceedings IEEE Globecom 88*, Hollywood, Florida.
- Wu, X., Lambadaris I., Lee H., Kaye, A., (1994), "A Comparative Study of Some Leaky Bucket Network Access Schemes", *IEEE Supercomm/ICC 94*, May 1994, New Orleans.
- Zheng, B., Atiquzzaman, M., (1999) "Traffic Management of Multimedia over ATM Networks". *IEEE Communications Magazine*. 37(1), 33-38.
- Zorpette, G., (1989) "Keeping the Phones Lines Open". *IEEE Spectrum*, June 1989, 32-36.