THESIS

to be presented at

Universidad de La República, UdelaR

in order to obtain the title of

MAGISTER EN INGENIERÍA MATEMÁTICA

for

Ing. Cecilia PARODI

Research Institute : LPE - IMERL
University Components :
UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA

Thesis title :

Integer Optimization Applied to the Design of Robust Minimum Cost Multi-Layer Networks

to be presented on april 2011 to the Comitee of Examiners

Dr. Franco	ROBLEDO	Academic Director
MSc. Carlos	TESTURI	Thesis Director
Dr. Ing. Pablo	BELZARENA	President
Dra. Paola	BERMOLEN	
Dr. Ing. Héctor	CANCELA	
Dr. Ing. Antonio	MAUTTONE	

Acknowledgements

I would like to thank everyone that supported me or helped me in any way during the process to achieve this thesis. I have received help from very different sides, from strictly regarding the work done to strictly emotional support. So in general I would like to thank everyone who collaborated with me in any of these ways.

I am not giving any special order to this, but there is one person that I am most grateful and is my boyfriend Andrés. I must say that he has been very supportive and he has the ability to turn his wisdom into great advise that has helped me and motivated me during the process towards the accomplishment of this work.

I would also like to thank my Academic Director Dr. Franco Robledo for his energy and motivation and to my Thesis Director MSc. Carlos Testuri for all his support and help to find solutions to every problem that stepped in the way.

To all the project team, specially to MSc. Andrés Corez for his friendship and always listening to me and for his help in the processing of the data and to MSc. François Despaux for the analysis and modelling of the entities of the problem and for his constant support in the study of the results. I also would like to acknowledge MSc. Claudio Risso for his help in the understanding of the problem.

To Dr. Maurice Queyranne from British Columbia University whose presence in the team was invaluable and helped us verify the correctness of the binary model.

I would also like to acknowledge Laura Saldanha from ANTEL for her help in the depuration of all the data that we needed.

I am greatly helpful to my friends and family for their support and for always listening when I needed.

General index

Summary

In this work we solve the problem of designing an MPLS data network, to be deployed over an existing SDH/DWDM transport infrastructure, which is itself a combination of technologies. The data and different work hypothesis are from a particular operator: the National Telecommunications Administration (ANTEL) of Uruguay where the problem was originated.

The target Data Network is an IP/MPLS Multi-Layer network over which different kinds of services are delivered and therefore with different quality requirements. During the process we will seek to minimize the economical resources incurred by the deployment over the Transport Infrastructure (Transport Network) of ANTEL.

The solution found should be of optimal (minimum) cost and must be able to send a known traffic meeting certain quality parameters, even considering simple failures in some section of the Transport Network.

The proposed problem is NP-Hard class in terms of computational complexity. Special cases of it are NP-complete problems. We model the problem algebraically as an Integer Mathematical Programming Problem and solve it approximately.

Resumen

En este trabajo se resuelve el problema de diseñar una red de datos MPLS, a ser desplegada sobre una infraestructura de transporte SDH/DWDM ya existente, que es a su vez una combinación de tecnologías. Los datos y las distintas hipótesis de trabajo son de un operador particular: la Agencia Nacional de Telecomunicaciones (ANTEL) de Uruguay, donde este problema fue originado.

La Red de Datos objetivo es una red multi-capa IP/MPLS sobre la cual se distribuyen diferentes tipos de servicios que tiene distintos requerimientos de calidad. Durante el proceso se buscará minimizar los recursos económicos que surgen del despliegue sobre la infraestructura de transporte (Red de Transporte) de ANTEL.

La solución encontrada debe ser de mínimo costo y debe poder enviar determinado tráfico cumpliendo con ciertos parámetros de calidad, incluso frente a fallas simples en la Red de Transporte.

El problema propuesto es de la clase NP-Hard en términos de complejidad computacional. Casos particulares del mismo son problemas NP-Completos. Modelamos el problema de forma algebraica como un problema de Programación Entera y lo resolvemos de forma aproximada.

Part I

INTRODUCTION

Chapter 1

Introduction

A telecommunication network is composed of terminals, links and nodes which are connected together in order to establish a communication between users of the terminals. The basic unit of information transmitted is called *bit* and the transmission rate[1] is called *bandwidth*. The latter is expressed in bits per second (bps).

The amount of information to be exchanged between users will be called *traffic* or *demand*. It is associated with the bandwidth sold between them.

Nowadays, telecommunication networks are designed with a multiple layer structure[2] (overlays), according to different technologies where each layer has its own technology. For example, Internet is designed over PTSN, P2P over Internet, ATM over SDH and VoIP applications such as Skype over Internet.

In the problem we are dealing with, there is a two-layer structure. One of these networks is called *Transport Network* while the other is called *Traffic* or *Data Network*. These two networks have different characteristics.

The former is a physical network, where the links are composed of fiber optic lines. The structure of this network does not change very quickly because the construction of new lines is very expensive, so its structure is more or less static. Besides, the traffic exchanged is completely static. When a demand is established between two points in the network, a path[3] in the network with enough capacity has to be found to route the demand. Once this is done, the bandwidth in these links is consumed completely whether it is effectively used or not.

In the latter one, the reality is quite distinct as the traffic is very dynamic. A stochastic analysis of this demand can be found in [29]. In that analysis, a function named $z_Q : \mathbb{R} \to \mathbb{R}$ is

[1] Amount of data that can be carried from one point to another in a given time period (usually a second).
[2] They are called Multi-Layer Networks.
[3] A path is a set of links.

searched, where given a specific bandwidth it returns the capacity to reserve in any link so that the probability that two users using those links detect congestion is no greater than Q.

When designing networks, the demands between nodes are usually called commodities. In this project, there are many demands between users, making this a multicommodity flow problem. Multicommodity flow problems have extensively been treated in the literature, and few examples are in [1, 7, 13, 24, 25].

One of the main concerns when designing telecommunication networks is to build robust networks. The survivability of a network is the ability that it has to recover from failures. It is an important task because in a multilayer network, a single failure in a lower layer can cause multiple failures at a higher layer. Survivability can be implemented using techniques such as protection and restoration where examples of this can be found at [27, 28, 32]. In the last two, the fault management is treated only in a one layer network, while in the first one a multilayer network is considered. Survivability has also been addressed in [1, 5, 6, 19, 35].

In this project, the design of a Data Network using an existing Transport network is addressed, in a robust way and at minimum cost. Network design involves decisions about the network topology, link capacities, and traffic routing.

A huge literature exists on optimal network design problems. Surveys can be found in [5, 18, 34, 36, 37]. The last one only considers the design of networks, whereas the others also take into account survivability.

One characteristic of our problem is that the flow is unsplittable. It has to be routed in an integral way. Most of the works consider the design of networks but they consider the flow to be splittable [1, 7, 11, 19, 23–26, 35]. In [1, 7, 19, 23, 24, 26] they work with a one-layer problem, while the others consider multiple layers. Unsplittable flow problems have been addressed in [2, 33], but they only consider a one-layer network.

In many articles there are several facilities that can be installed an integer number of times on each link [7, 9, 23, 24, 26, 31]. The works in [7, 23, 24, 26] consider one-layer networks, while the others work with multilayer networks.

The design of Multi-Layer networks has also been studied in [3–5].

Although a huge amount of works can be found regarding network design as pointed out before, after an extensive research it was not possible to find problems that were close enough to the one addressed in this thesis. They take into account some of the variables considered here, but not all of them. Those considered to be the most approximate can be found in [1, 18, 34, 38].

This thesis is part of a research group and is in the context of a project proposed by ANTEL[4] and was carried jointly by ANTEL and the Statistics and Probability Laboratory, LPE[5], a specific activity known as *"Optimización bajo diseño robusto en Redes Multi-Overlay"*.

[4] *Administración Nacional de Telecomunicaciones* of Uruguay.
[5] Engineering University of the Republic.

As a final remark, the problem being addressed in this project is **NP-Hard**, and different approaches have been taken to solve it, all in the context of this project. They can be found in [8, 10, 29].

1.1 Objectives

In this thesis, the problem to be solved is the design of a network that has a two-layer structure. The overlay network is a MPLS Data Network that physically exchanges traffic over a Transport Network infrastructure. The design consists of several things. It should be able to assign a capacity to each link in the Data Network and meet the traffic requirements between nodes in the Data Network between certain quality parameters. There are different kinds of traffic each one with different quality requirements.

During this design, the costs involved should be minimized, which are dependent on the distance between nodes in the Transport Network and the capacity assigned to each link in the Data Network.

Another important thing is that the solution found must be resilient to simple failures in the Transport Network, meaning that when one link of the Transport Network fails, the traffic still has to be routed.

This work needs to solve a particular reality that is the one of ANTEL. However, the abstract formulation made allows any kind of network to be tested.

To put all the ideas together, we can summarize the objectives of this work as:

- Determine in which stations should MPLS equipment be installed.
- Find the links that should be created in the Data Network and determine what capacity has to be assigned to them.
- For each data link established, give the best path in the Transport Network that this link should follow.
- Determine which technology should be used for each transport edge.
- Answer how to route traffic requirements in every scenario of transport link failure and in the working scenario.

In order to be able to solve this, the following information is needed as an input to the problem:

- Set of Network Stations (these are typically Telephone Stations).

- Stations where it is feasible to install MPLS switches.
- Potential links between switches that may be established.
- Capacities available for each data link.
- Transport technologies available for each bandwidth and its cost.
- Transport Network topology (nodes, links, distances between nodes).
- Traffic requirements given by a source, destination and demand of traffic.
- The statistical behavior of clients traffic.

1.2 Documentation organization

The chapter organization of this thesis is the following:

Chapter 1: *Introduction*, provides a general introduction to the thesis, giving a brief description of the state of art and to the objectives proposed.

Chapter 2: *Problem description*, defines some basic concepts about networks in order to give a better explanation of the problem being addressed.

Chapter 3: *Formalization*, gives a description of the representation of the networks involved presenting their interconnection, and other important variables.

Chapter 4: *Abstract model*, begins explaining a simplification made to the problem in order to make it more workable. After that, an abstract model is presented along with a description of every constraint and the objective function. The chapter ends with an analysis of the complexity of the problem.

Chapter 5: *Binary Integer Programming Model*, describes the Binary Integer Programming Model developed from the abstract model and its evolution into different versions. It describes all the binary variables and the objective function, giving also a detailed explanation to every constraint.

Chapter 6: *Decomposition Approach*, explains a different approach used to solve the problem. This approach involves an iterated way of obtaining a solution.

Chapter 7: *Test cases description*, presents all the test cases designed to evaluate the implementation.

Chapter 8: *Implementation and Results of the tests*, gives a brief explanation of the implementation made and also studies the results achieved comparing them to the results obtained by applying two different metaheuristics.

Chapter 9: *Conclusions*, it closes this work analyzing the achievements accomplished and possible extensions to this work.

Chapter 2

Problem description

In this chapter some preliminary concepts are defined and a more detailed description of the problem is exposed.

As mentioned in the previous chapter, the problem consists on designing a survivable IP/MPLS Data Network over ANTEL transport infrastructure. In order to clarify these concepts, some definitions are given.

2.1 MPLS Networks

MPLS[1] [20, 30] is a mechanism in high-performance telecommunications networks that directs and carries data from one network node to the next. Networks are divided into layers (according to the OSI Model[2] [20]) that communicate through protocols. It creates "virtual links" between distant nodes and can encapsulate packets of various network protocols.

This mechanism lies between Layer 2 (Link Layer) and Layer 3 (Network Layer) of the OSI Model.

Although there are potential increases in switching speeds, the current interest lies in the new traffic management capabilities that MPLS enables. For instance, it provides the ability to forward packets along routes that would not be possible using standard IP routing protocols. This may give the operator the flexibility to route traffic through different links in case of congestion or link failures for example.

[1] Multi-Protocol Label Switching
[2] Open Systems Interconnection

2.2 Overlay Networks

Nowadays, telecommunication networks are designed with a layered structure. An Overlay Network is a virtual network that is built on top of an existing (physical) network, where the nodes are connected by logical links.

Demands are usually given in the virtual layer, and they have to be routed in it, leading to the installation of "virtual capacities" (which are routers or other devices). Virtual capacities define demands for the transport layer, leading to the installation of capacities (for example optical fibers), in the physical layer.

Therefore, when a demand is routed through a path in the virtual layer (composed of many virtual edges), each link corresponds to a path in the underlying network, where the path may consist of many physical links.

In figure 2.1 there is a schematic view of an Overlay Network, that corresponds to the problem that we are trying to solve in this project. In it, the Data Network is built on top of the already existing Transport Network.

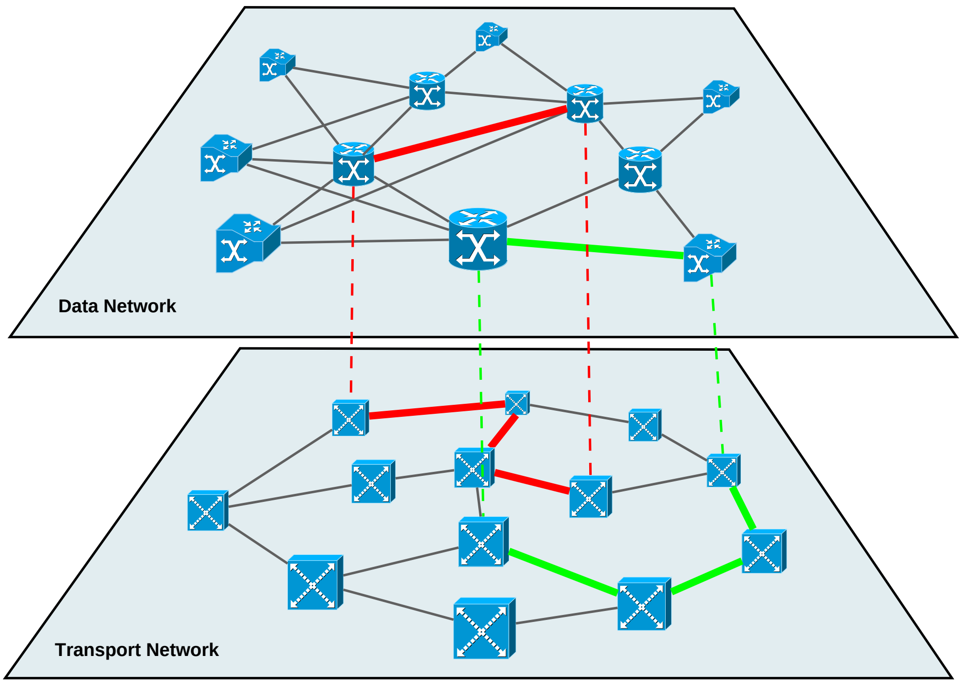


Figure 2.1: Overlay Network.

There are many examples of Overlay Networks, where we chose to mention the following:

Internet It is built on top of the Telephone Network (PSTN).

P2P P2P networking is a distributed application architecture for content distribution between peers. This content is exchanged directly over the underlying Internet network.

ATM The connection between nodes uses SDH links.

MPLS It is similar to ATM but it is supported by more transport technologies and cheaper interfaces, such as Ethernet.

2.3 The problem

Having defined those necessary concepts, we are now able to give a better explanation of the problem itself.

In this work the problem of designing an IP/MPLS Network will be solved. This Data Network is an Overlay Network displayed over an existing Transport Infrastructure that corresponds to ANTEL and it is already developed. Although general scenarios will be tested, the main data belongs to this operator. As explained before (2.2), the links in the Data Network are virtual links.

Regarding the Transport Network, different types of technologies must be considered, while in the Data Network, several types of services are delivered with different quality requirements. The network found as a solution should have minimum cost and must be able to satisfy all the traffic requirements. It also has to be robust to simple failures in the Transport Network.

In order to solve this problem, all the Transport Network along with the possible technologies to be used should be known. Also, about the Data Network to design, all the potential links between nodes and the traffic requirements must be given as an input.

The concept of designing the Data Network includes basically two things. Firstly, from the potential links that are given, those ones that make the cost optimal and let all the traffic be routed (in the scenario where no transport link fails and in every fault scenario) should be chosen. Secondly, the capacities to these selected links must be assigned, which correspond to the supported technologies of the Transport Network.

Part II

PROBLEM FORMALIZATION

Chapter 3

Formalization

In this chapter, all the basic entities along with their main characteristics and relations are explained.

As we stated in the previous chapter, we will be working with two different networks, the Data Network and the Transport Network. In the following sections, the representation of each of these networks is given including their properties, and at last the connections between them is clarified.

3.1 The Data Network

The Data Network will be represented as the graph $G_D = (V_D, E_D)$, where V_D is the set of nodes and E_D the set of potential virtual links. An assumption made is that this is an undirected graph.

Besides the property of being undirected, G_D also has other characteristics. The first one is that it need not be a planar graph, as it has a mesh topology and may be very dense in amount of links. The other property is that the traffic in this network is stochastic and very dynamic, both in the routes and the volume.

3.1.1 Data Nodes

The nodes in the Data Network are classified in two ways.

First classification. There are two types of nodes called *Access* nodes and *Edge* nodes.

The Access nodes are those ones that deliver traffic towards final clients. The traffic that goes through them into the network is from the set of clients connected to them. Typically, they are DSLAM xDSL commuters that are represented as in Fig-3.1.

The Edge nodes are more capable nodes. They are not only capable of routing traffic towards its destiny, but also they can manage the topology of the network in order to detect failures in links. When some link failure is detected, they have mechanisms to search for another route to send the traffic, in order not to congest the network. They are usually connected to many access nodes and other edge nodes. These nodes are represented as in Fig-3.1.

The set of Access and Edge nodes will be denoted by V_A and V_E respectively. As one node belongs only to the set of Access nodes or to the set of Edge nodes but not both, and that each node is either one of those two types, $\{V_A, V_E\}$ is a partition of V_D.

We introduce the subnet G_E of G_D induced by the nodes V_E. So, $G_E = (V_E, E_E)$ where $E_E = \{e_E = (v_i, v_j) \in E_D \ / \ v_i, v_j \in V_E\}$.

Second classification In this classification, the nodes are said to be *Fixed* or *Steiner* nodes.

The fixed nodes are those that must be present in the final solution to the problem. On the other hand, Steiner nodes are optional nodes and they may or may not be in the solution. Including Steiner nodes may cause some benefit, for instance, a decrease in the cost of the solution.

Calling V_F and V_S the sets of Fixed and Steiner nodes respectively, they also form a partition of V_D. It is worth noting that Access nodes should always be in the final solution, otherwise the final clients connected to them will be disconnected from the network. In terms of the definitions given, we have $V_A \subseteq V_F$, that relates both classifications. Edge nodes are generally Steiner nodes, but some of them may be Fixed nodes.

3.1.2 Data Links

As mentioned in the previous chapter, the links in the Data Network are virtual links. This means that a link $e = (v_i, v_j) \in E_D$ is a *potential* link to be established. These links may or may not be present in the final solution, but if they are in G_D then they will be evaluated when generating the solution to see if they are included or not in it.

The Data Links should be dimensioned with a capacity that is technologically supported by the Transport Infrastructure. These capacities are known to the problem and they are represented by the set $\hat{B} = \{\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_{\bar{B}}\}$. One of these capacities has to be chosen to each of the links that are present in the final solution. If a link does not become a part of the solution, then we assume that its capacity is 0 and represent it by $\hat{b}_0$. As a result of this, we only have to find the set $B = \{b_e \in \hat{B}, \forall e \in E_D\}$ as a part of the solution.

If the set B is known, then all the links included in the solution are also known (those with capacity different from $\hat{b}_0$). Besides, it can be determined whether a node v_i belongs to the solution, when $b_e \neq 0$ for some $e = (v_i, v_j) \in E_D$.

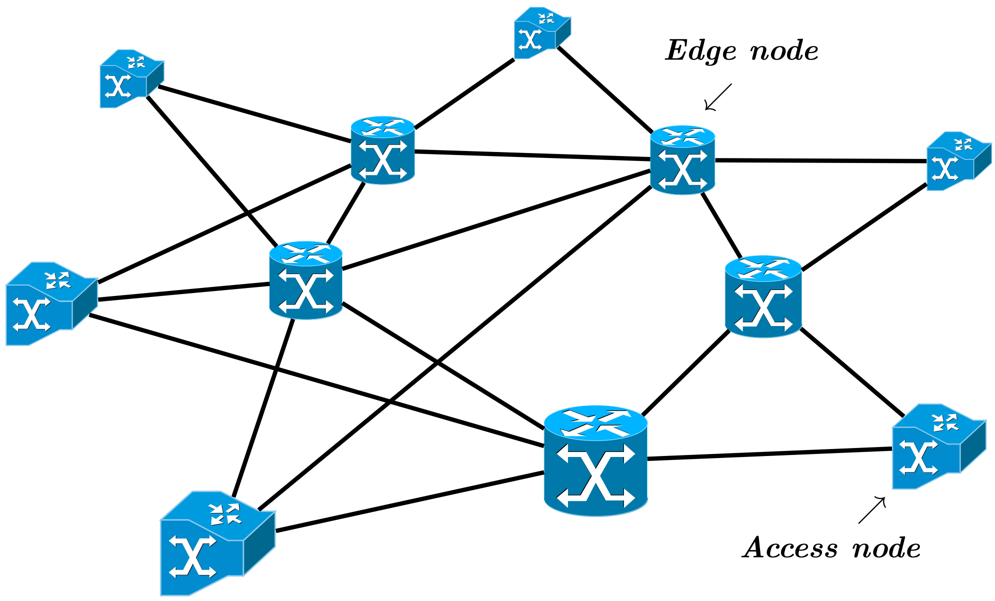


Figure 3.1: Example of a Data Network.

3.1.3 The Traffic in the Data Network

Part of the problem is to satisfy all the traffic requirements, and these are also divided into two kinds: *committed* traffic and *excess* traffic. In what follows we will give a brief explanation of each of these kinds.

Committed traffic. This traffic must be exchanged 100% of the time with minimum delay. It even should be satisfied in the presence of any simple transport link failure. It can be associated with multimedia applications traffic such as VoIP or VoD.

Excess traffic. This traffic is only available part of the time. The fraction of time that it is available is called quality factor and this factor need not be the same in the scenario where there are no faults as in scenarios where there is a link failure. It is typically the Internet traffic so it is treated as best effort. This is where the z_Q function described in the introduction and better explained in [29] plays a role. It assigns to each bandwidth sold the minimum required capacity of the links that carry that traffic.

The traffic (committed and excess) between each pair of nodes in the Data Network is supposed to be known as it represents the demands of the clients. It will be denoted by $\vec{m}_{ij} = (\dot{m}_{ij}, \ddot{m}_{ij})$, for $v_i, v_j \in V_D$. In this notation, $\dot{m}_{ij}$ corresponds to the committed traffic, whereas $\ddot{m}_{ij}$ to the excess traffic.

From this, we introduce a matrix that is called *traffic matrix* that contains the traffic vectors of every pair of nodes in the Data Network: $\bar{M} = ((\vec{m}_{ij}))_{1 \leq i,j \leq |V_D|}$.

Definition 3.1.1. *Let* $V_D = \{v_1, v_2, \ldots, v_h\}$*, there exists a traffic matrix* $\bar{M} = ((\vec{m}_{ij}))_{1 \le i,j \le h}$ *where the vectors* $\vec{m}_{ij} = (\dot{m}_{ij}, \ddot{m}_{ij}) \in \{\mathbb{R}_0^+ \times \mathbb{R}_0^+, \forall v_i, v_j \in V_D\}$*. Here,* $\dot{m}_{ij}$ *is the committed traffic between the nodes* v_i *and* v_j *while* $\ddot{m}_{ij}$ *is the excess traffic between them.*

Note: 3.1.2. $\bar{M}$ *is symmetric, i.e.* $\dot{m}_{ij} = \dot{m}_{ji}$ *and* $\ddot{m}_{ij} = \ddot{m}_{ji}$*.*

Lastly, as we stated before, the nodes that exchange traffic are fixed nodes, so if $\vec{m}_{ij}$ is non zero, then $v_i, v_j \in V_F$.

3.2 The Transport Network

As we did with the Data Network, the Transport Network will be represented by the graph $G_T = (V_T, E_T)$, where V_T is the set of nodes and E_T is the set of links. For this network it will also be assumed to be an undirected graph.

The main characteristics of this network that we can point out are that it is planar and 2-node connected. The planarity of this network can be assumed because of the way the fiber optic is stretched. It also has a ring topology as it can be seen that it consists of rings that share one or more links. The 2-node connectivity follows as a consequence of this ring topology.

There is also another property that distinguishes the Transport Network from the Data Network, and it is that the traffic here is completely static. When a path in this network is associated with a link in the Data Network, then this path is fixed and it is the same for every fault scenario.

3.2.1 Transport nodes

We will call *Network Station* to the place where the telecommunication structure is physically installed. In these stations there are transport nodes as well as data nodes.

It will be assumed that in each Network Station there is only one transmission node. If we take a closer look into these stations, they include different equipment because of the several technologies (SDH, DWDM, etc.), ports, bandwidth. The assumption made is that in these Network Stations there is an equipment of unlimited capacity that may use any of the available technologies. This, in the understanding that including the resolution of this problem is beyond the scope of the project.

Transport nodes are represented as shown in Fig-3.2.

3.2.2 Transport Links

The links in this network represent the physical canalizations that are made in order to connect two Network Stations. Unlike the Data Links, these ones are supposed to have unlimited capacity, in the sense that the necessary amount of equipment should be installed to satisfy the demand.

If a link in this network fails it affects every connection of the data network that uses that link.

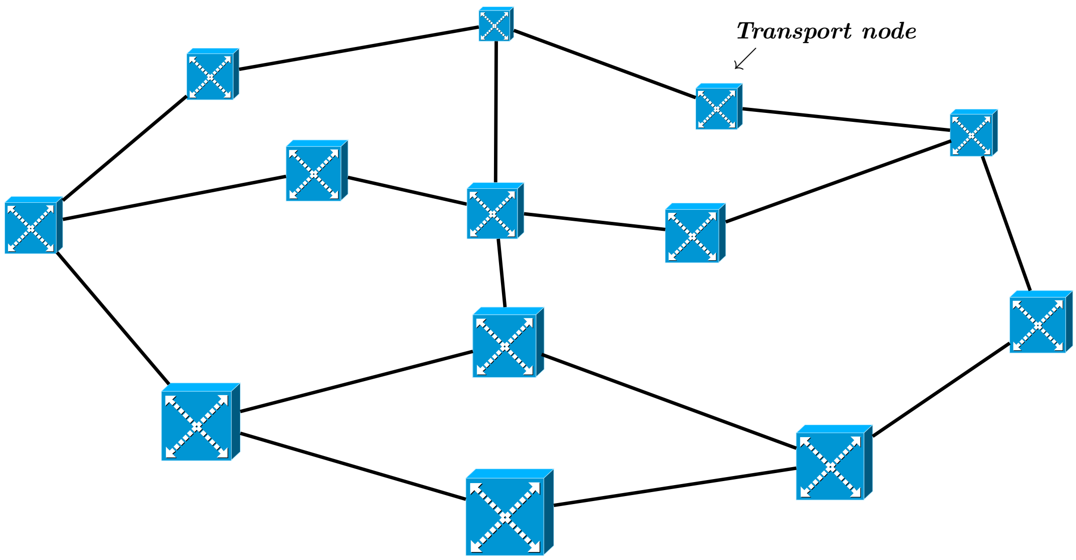


Figure 3.2: Example of a Transport Network.

3.3 Connection between Data Network and Transport Network

There are some things that need to be explained and are not yet defined. However, before doing so, a connection between both layers is explained.

As stated before, in each Network Station there is only one transport node. We mentioned that there were also data nodes, and we assume that there is at most one access node and at most one edge node.

With this assumption we can define a function that relates the nodes that are installed in the same Network Station. It is defined as follows:

$$ns : (V_D \cup V_T) \times (V_D \cup V_T) \to \{0, 1\},$$

where $ns(u, v) = 1 \Leftrightarrow$ both u and v belong to the same Network Station.

From this, given a data node we can determine which transport node belongs to the same Network Station as it does. In view of this, let the function $tns : V_D \to V_T$ such that $tns(v) = t$ when $ns(v, t) = 1$. This function returns the transport node t located in the same Network Station as v.

3.4 Traffic routing

In this section we will take care of the paths followed in the Transport Network by the links in the Data Network, as well as the paths that follow the traffic inside the Data Network in each failure scenario.

3.4.1 Routing in the Data Network

In this case, we should be able to answer how the traffic between data nodes v_i and v_j is exchanged, that is, which links in the Data Network the traffic should follow. In an MPLS network, the usual is that the traffic follows a unique path in the network, that is called *tunnel*.

We begin with some definitions. The graph $G_D = (V_D, E_D)$ will denote the Data Network.

Definition 3.4.1. *The set P_D will represent the set of all possible paths in the graph G_D. We call* routing scenario *to every subset $\rho_D \subset P_D$, and $g_D : V_D \times V_D \to 2^{P_D}$*[1] *is the* routing function *that given two nodes returns all possible paths between them in G_D.*

Definition 3.4.2. *Let $\rho_D \subset P_D$ a routing scenario and two data nodes $v_i, v_j \in V_D$. Define the* routes from v_i to v_j *in scenario ρ_D as the elements of the set $\rho_D^{ij} = g_D(v_i, v_j) \cap \rho_D$.*

This last definition gives all the paths from v_i to v_j that belong to the scenario (subset) ρ_D. Only one tunnel is wanted to go from v_i to v_j, but this condition should be satisfied in every routing scenario. If a link in the Transport Network fails, and the route from v_i to v_j is interrupted, then another one must be found.

The following example intends to clarify these concepts.

Example. Consider a Data Network as shown in Fig-3.3.

The set of nodes and edges are:
$V_D = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$.
$E_D = \{(v_1, v_2), (v_2, v_5), (v_5, v_7), (v_7, v_6), (v_6, v_3), (v_3, v_1), (v_3, v_4), (v_4, v_5), (v_6, v_4), (v_4, v_2), (v_3, v_7), (v_2, v_7)\}$.

[1]The set 2^A is the power set of A, that means the set composed by all subsets of A.

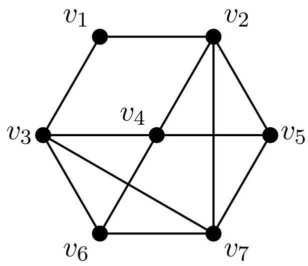


Figure 3.3: Routing in Data Network.

A possible routing scenario is:
$\rho_D = \{\{(v_1, v_2)\}, \{(v_1, v_2), (v_2, v_7)\}, \{(v_7, v_6)\}, \{(v_1, v_3), (v_3, v_6), (v_6, v_7)\}, \{(v_1, v_3), (v_3, v_4)\}, \{(v_3, v_4), (v_4, v_5)\}\}$.

In this example, the possible routes from the data node v_1 to v_7 in this routing scenario (ρ_D^{17}) are $\{(v_1, v_2), (v_2, v_7)\}$ and $\{(v_1, v_3), (v_3, v_6), (v_6, v_7)\}$. ♠

To close this section, we need to give one more definition. Let $G_T = (V_T, E_T)$ be the Transport Network.

Definition 3.4.3. *We call* routing scenarios function *to* $\Phi : (E_T \cup \emptyset) \to 2^{P_D}$ *such that each* $e_t \in E_T$*, as well as the empty set, are assigned a routing scenario* $\rho_D \subset P_D$*. In particular,* $\Phi(\emptyset)$ *is denoted* nominal routing scenario.

In the final solution we need to be able to route each demand in every simple fault scenario as well as in the nominal scenario. The nominal scenario corresponds to the case where there are no failures in the Transport Network. So, from all possible routing scenarios we need to find the one that achieves this and at minimum cost.

In order to know which data links were affected by a transport link failure, it is necessary to know the paths that the data links follow in the Transport Network. This routing is also a part of the solution and it is explained in the next subsection.

3.4.2 Routing in the Transport Network

In this subsection some concepts regarding the routing in the Transport Network are defined. They are similar to the ones defined in the previous subsection for the Data Network. The graph $G_T = (V_T, E_T)$ will represent the Transport Network and $G_D = (V_D, E_D)$ will denote the Data Network.

Definition 3.4.4. *We denote* P_T *to the set of all possible paths in the graph* G_T*. We name* flow configuration *to every subset* $\rho_T \subset P_T$*, and* $g_T : V_T \times V_T \to 2^{P_T}$ *is the* flows function *which determines all possible paths between a pair of transport nodes.*

Definition 3.4.5. *Given a flow configuration $\rho_T \subset P_T$ and a data link $e_d \in E_D$ such that $e_d = (v_i, v_j)$, $v_i, v_j \in V_D$, the* flow implementation for e_d *are the elements of the set* $\rho_T^{ij} = g_T(tns(v_i), tns(v_j)) \cap \rho_T$.

The set ρ_T^{ij} contains all the paths from $tns(v_i)$ to $tns(v_j)$ that belong to ρ_T.

Lets consider the following example to illustrate these concepts.

Example. Suppose the Data Network $G_D = (V_D, E_D)$ and the Transport Network $G_T = (V_T, E_T)$ are as shown in Fig-3.4:

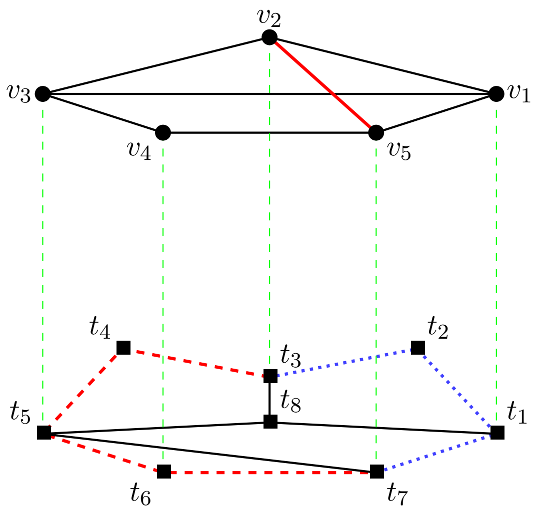


Figure 3.4: Routing in Transport Network.

The sets of nodes and edges are:
$V_D = \{v_1, v_2, v_3, v_4, v_5\}$.
$E_D = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_1), (v_2, v_5), (v_3, v_1)\}$.

$V_T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$.
$E_T = \{(t_1, t_2), (t_2, t_3), (t_3, t_4), (t_4, t_5), (t_5, t_6), (t_6, t_7), (t_7, t_1), (t_3, t_8), (t_5, t_8), (t_8, t_1), (t_5, t_7)\}$.

The nodes from the Data Network and the Transport Network that are brought together by a dashed line are nodes that belong to the same station. Using the tns function defined, this means that for example $tns(v_1) = t_1$ and so on.

A possible flow configuration is:
$\rho_T = \{\{(t_3, t_2), (t_2, t_1), (t_1, t_7)\}, \{(t_3, t_8), (t_8, t_5)\}, \{(t_3, t_4), (t_4, t_5), (t_5, t_6), (t_6, t_7)\}, \{(t_1, t_2)\}, \{(t_6, t_7), (t_7, t_5)\}\}$.

In this example, a flow implementation for (v_2, v_5) in this flow configuration (ρ_T^{25}), goes from the node $tns(v_2) = t_3$ to $tns(v_5) = t_7$ and are the sets $\{(t_3, t_2), (t_2, t_1), (t_1, t_7)\}$ and $\{(t_3, t_4), (t_4, t_5), (t_5, t_6), (t_6, t_7)\}$. ♠

In the final model, we will assume that each data edge has a unique transport path associated. As a consequence, with the definition given we will request that $|\rho_T^{ij}| = 1$.

The last definition of this section is the function that assignes to each link in the Data Network a flow configuration.

Definition 3.4.6. *We call* flows configuration function *to* $\Psi : E_D \to 2^{P_T}$ *such that each* $e_d \in E_D$ *is assigned a flow configuration* $\rho_T \subset P_T$.

If $e_d = (v_i, v_j) \in E_D$ and $\rho_T = \Psi(e_d)$, this means that ρ_T^{ij} is the implementation in the Transport Network of the data link e_d.

3.5 Transmission Costs

At this point, we have almost defined everything that is needed for the problem. In this section we explain how the cost is calculated. This is an important aspect because the main objective proposed is to minimize the cost of the solution found.

To find the global cost of the solution, the cost of each transport path chosen (one for each data link) must be calculated. This is highly dependent on the distance between transport nodes. We introduce the following definition.

Definition 3.5.1. *Let* $r : E_T \to \mathbb{R}_0^+$ *the* distance function *that assigns to each transport link its length. It is possible to extend the domain of the function to any flow implementation* ρ_T^{ij}, *considering the sum of distances of its links, i.e.* $r(\rho_T^{ij}) = \sum_{e \in \rho_T^{ij}} r(e)$. *It is assumed that* $r(\emptyset) = 0$.

To calculate the cost, besides considering the distance there are other considerations depending on the technology selected. Because of this, we divide the analysis between them.

TDM[2] (PDH/SDH)[3] technologies: The cost depends on the distance, but it is also proportional to the bandwidth selected, because TDM reserves containers for traffic of different sizes. So, a flow twice a capacity consumes twice the resources in every link of the path it follows.

Hence, for traditional technologies, the cost is calculated as:

$$cost(\rho_T^{ij}, b_e) = k \times r(\rho_T^{ij}) \times b_e,$$

[2]Time-Division Multiplexing

[3]Plesiochronous/Synchronous Digital Hierarchy

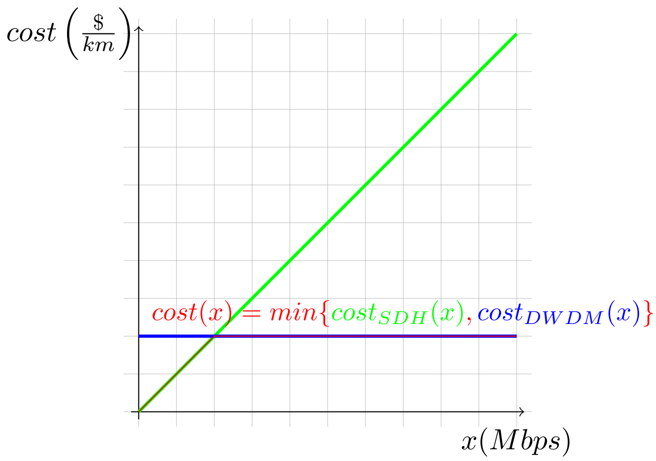


Figure 3.5: Cost function.

where k is a constant parameter, $r(\rho_T^{ij})$ is the distance of the path ρ_T^{ij} in the Transport Network and b_e is the capacity assigned to the link $e = (v_i, v_j)$ in the Data Network.

DWDM[4] technology: In this case, the cost depends only on the distance. This is because given a particular wavelength to the client, it is expected that he will make the best use of it, meaning the highest bandwith. The formula that arises is:

$$cost(\rho_T^{ij}, b_e) = k \times r(\rho_T^{ij}),$$

where r means the same as before.

The graphic shown in Fig-3.5 shows that depending on the capacity wanted, it is convenient to use one or the other technology. If the capacity required is small, then SDH is more convenient. Conversely, if a high capacity link is wanted, then DWDM is a better choice.

In order to make it easier to calculate the cost, we include both expressions into only one that makes use of a function $T : \hat{B} \rightarrow \mathbb{R}$, that associates a technology b with its cost per kilometre.

With this, the cost can be calculated as:

$$cost(\rho_T^{ij}, b_e) = r(\rho_T^{ij}) \times T(b_e).$$

[4]Dense Wavelength Division Multiplexing

Chapter 4

Abstract model

In this chapter, a translation of the problem in terms of mathematical equations will be presented. This abstract model could be a bit hard to understand when everything is being taken into consideration. However, a simplification to the problem was made and it results in a much lighter set of equations. As the problem that is being solved in this thesis includes the mentioned simplification, the reduced model will be presented. In order to view the complete model, one can refer to the Master Thesis of Claudio Risso [29].

In the following sections, an Access-Edge nodes simplification is explained and afterwards the abstract model is given including a detailed explanation of each equation.

4.1 Access-Edge connections

In this section the connection between access and edge nodes will be explained. A solution to connect the access nodes to the edge nodes beforehand is proposed, so that in the problem only edge nodes are considered, resulting in a much more simple problem.

An access node is connected to only one data node, and it must be an edge node. This is because if two access nodes were connected, they would become isolated from the rest of the network, making it impossible for them to send traffic to the other nodes, as no other connections are allowed for them.

Although access nodes are connected to only one edge node, there is no limit in the number of connections through different transport paths they can have.

The connections between access and edge nodes may or may not use the Transport Network. We shall observe that whenever access and edge nodes are in the same Network Station, they are connected locally without need of the Transport Network. If that is not the case, then we can assume that those sites are non critical, making the assumption that most important sites do have access and edge nodes.

To continue with the analysis, the next question is whether the access-edge connections need to be protected against transport failures. In this case, at least two independent transport paths are needed to connect the access node to its corresponding edge node. By independent paths, we mean paths with no common edges.

As the Transport Network has a ring topology, in order to accomplish this, both sides of the ring should be used. This has a first disadvantage, that is that the capacity of the ring that contains the access and edge nodes is fully consumed. This makes the protection against transport failures very expensive. Another disadvantage is that there is an increase in the cost with respect to the case where both flows follow the same transport path (the shortest path). If there are independent paths, the alternative path may be several times larger than the shortest one.

Let us compare this with the model where the access-edge connections are not protected, and the path is chosen by the shortest path method. Before proceeding, we must remark that an assumption made is that there is a balanced utilization of the capacities, which means that each transport edge in a path carries a similar amount of flow.

We give an example to illustrate what happens in this case. In Table-4.1 and Fig-4.1 there is an example of 6 nodes A, B, C, D, E and F, where A and F are edge nodes. There, the second column represents the demand while the third one is the capacity needed to be reserved in the transport flows. In order to have a better insight in this, refer to [29]. With the shortest path method, the access nodes B and C get associated with edge node A, because it is closer in distance than F, and nodes D and E get connected to F. If we take a closer look at this, we notice that the link from A to B concentrates all the traffic from B and C, while link from C to D has no traffic at all. As a consequence, we get a unbalanced charge and may become a precision error in the calculation of the cost. Another important thing is that the path between two important stations should have the same capacity all along, because we are assuming a balanced charge between two edge nodes. If we consider nodes A and F as being important stations, then all 5 links in the path must be $40\ VC12$[1], adding a total of $200\ VC12$, while the real charge of the path is $25 + 5 + 0 + 20 + 40 = 90\ VC12$.

This unbalancing problem can be fixed as we will explain in the following. At the end, we only want to have edge nodes, so first we need to find a way of assigning the demand of access nodes to edge nodes. Afterwards, in order to balance the services, it is assumed that in each station there are two access nodes with half the real demand. Each one of them will be connected to the 2 closest edge nodes. We extend the previous example to clarify what is being done. In Table-4.2 and Fig-4.2 nodes B, C, D and E send half their demand to A and half to F, which reflects in a balanced utilization of every transport edge of $35\ VC12$.

[1] $VC12$ stands for Virtual Container.

Station	m(Mbps)	$\hat{\mathbf{b}} \in \hat{\mathbf{B}}$
A	879	441
B	34	20
C	8	5
D	28	20
E	38	20
F	874	441

Table 4.1: Example: Unbalanced stations and capacities.

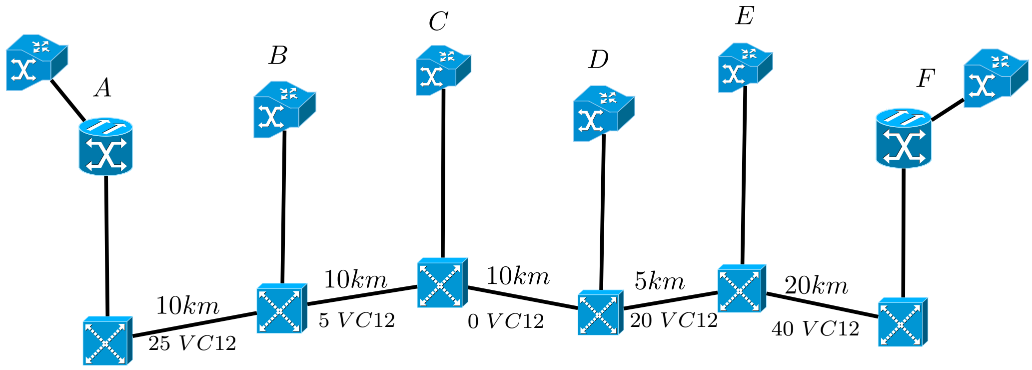


Figure 4.1: Unbalanced Transport Network

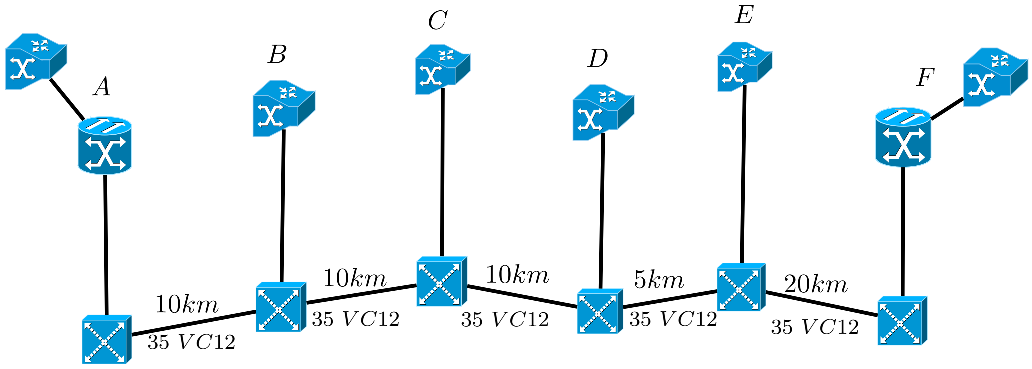


Figure 4.2: Balanced Transport Network

Having two access nodes in every station connected to two different edge nodes allows configurations of high availability in services, much better than those when using two independent paths in transport. It also makes it possible to have simpler algorithms and less expensive

Station	m(Mbps)	$\hat{\mathbf{b}} \in \hat{\mathbf{B}}$	**Station**	m(Mbps)	$\hat{\mathbf{b}} \in \hat{\mathbf{B}}$
B_1	17	10	B_2	17	10
C_1	4	5	C_2	4	5
D_1	14	10	D_2	14	10
E_1	19	10	E_2	19	10

Table 4.2: Balanced stations and capacities.

solutions.

As a final consideration, we change the formulation of the problem regarding the transport failure tolerance. In case a simple transport links fails, the solution must be able to route all the traffic in every link not affected by the failure, except the traffic of those access nodes that were affected. This means that the tolerance to failures cannot be extended to access-edge conections.

To conclude this section, we summarize the previous analysis in saying that the problem that we will work in has all the access-edge connections solved. We will assume that there are no access nodes which simplifies the problem. The edge nodes encapsulate all the traffic from the access nodes connected to them.

4.2 Mathematical Model

In this section, the abstract model is presented. As stated before, we will work with the simplification made in the previous section, so it is assumed that there are no access nodes.

The problem is going to be represented with the initials $MORN$ for Multi-Overlay Robust Network problem. As we included the simplification of deleting the access nodes, we will call it reduced-MORN or RMORN.

Before presenting it lets summarize the input to the problem:

Data Network:

- $\boldsymbol{G_D = (V_D, E_D)}$: Data Network graph, nodes and links.
- $\boldsymbol{\bar{M}}$: traffic matrix.
- $\boldsymbol{z_Q}$: excess traffic capacity function. Recall that it assigns to each bandwidth sold, $\ddot{m}_{ij}$, the minimum required capacity of the links that carry that traffic.

Transport Network:

- $\boldsymbol{G_T = (V_T, E_T)}$: Transport Network graph, nodes and links.
- $\boldsymbol{\hat{B}}$: available capacities.
- $\boldsymbol{T}$: technology cost function.
- $\boldsymbol{r}$: distance function.
- $\boldsymbol{ns}$: network station function.
- $\boldsymbol{tns}$: transport network station function.

Besides, the variables of the problem are:

- $\mathbf{B}$: assigned capacities for the data links.
- $\boldsymbol{\rho_D}$: data routes.
- $\boldsymbol{\rho_T}$: transport paths.

Recall from chapter 3 that G_E is the subnet of G_D induced by the edge nodes. As a consequence, we will only be working with G_E, which means that every node in the Data Network will be considered as an edge node.

In the following, the abstract model is presented along with an explanation of each of the equations.

$$
\begin{cases}
\min\limits_{B,\Phi,\Psi} \sum\limits_{\substack{\rho_T=\Psi(e) \\ e=(v_i,v_j)\in E_D}} r(\rho_T^{ij})T(b_e) & \\
|\rho_T^{ij}| = 1 & \forall e=(v_i,v_j)\in E_E,\ b_e\neq 0, \rho_T=\Psi(e). \\
|\rho_{D_t}^{ij}| = 1 & \forall v_i,v_j\in V_E,\ \forall t\in\{\emptyset\cup E_T\},\ \rho_{D_t}=\Phi(t), \vec{m}_{ij}\neq\vec{0}. \\
|\rho_{D_t}^{pq} \cap (\bigcup\limits_{\substack{e=(v_i,v_j)\in E_E \\ |\rho_T^{ij}\cap t|\neq 0}} e)| = 0 & \forall t\in E_T, \rho_{D_t}=\Phi(t),\ \forall v_p,v_q\in V_E. \\
b_{pq} \geq \sum\limits_{\forall v_i,v_j\in V_E} \left(\dot{m}_{ij}|\rho_{D_t}^{ij}\cap e|\right) + z_Q \left(\sum\limits_{\forall v_i,v_j\in V_E} \left(\ddot{m}_{ij}|\rho_{D_t}^{ij}\cap e|\right)\right) & \forall e=(v_p,v_q)\in E_E,\ \forall t\in\{\emptyset\cup E_T\},\ \rho_{D_t}=\Phi(t),\ \vec{m}_{ij}=(\dot{m}_{ij},\ddot{m}_{ij}).
\end{cases}
$$

Now, each constraint is explained.

- The first equation is the objective function that has to be optimized. It consists of a sum of costs that must be minimized. The tuple (B, Φ, Ψ) that achieves that minimum cost must be obtained.

  Each term of the sum corresponds to the transport cost of a data link $e \in E_D$ included in the solution ($b_e \neq 0$). So the total cost is obtained adding all those terms. As previously stated, the cost is expressed as the product of the cost of the path that the flow follows $r(\rho_T^{ij})$ and the cost per kilometer of the corresponding capacity selected for the link $T(b_e)$.

  The remaining equations are the constraints of the problem.

- The second equation means that each link $e \in E_E$, that is part of the solution ($b_e \neq 0$), must have a unique flow associated in G_T. This is what was pointed out in 3.4.2.

- The second constraint implies that in every possible scenario, either nominal or with a link failure ($t \in \{\emptyset \cup E_T\}$), the corresponding routing found ($\rho_{D_t} = \Phi(t)$), must contain a unique tunnel between any two edge nodes $v_i, v_j \in V_E$ with possitive effective demand of traffic between them for that particular scenario ($\vec{m'}_{ij}^{t} \neq \vec{0}$).

- The third constraint means that if a simple transport link fails ($t \in E_T$), the proposed routing for each pair of data nodes ($\rho_{D_t}^{pq}$), must not use any edge that uses that link. The term $\left(\bigcup_{\substack{e=(v_i,v_j)\in E_E \\ |\rho_T^{ij}\cap t|\neq 0}} e \right)$ in the equation contains all the edges in the Data Network that use link t, so none of those edges can be used to route the demand between v_p and v_q.

  In this equation, we are assuming the two previous constraints in order to simplify the notation. With this, we can suppose that ρ_T^{ij} and $\rho_{D_t}^{pq}$ are both composed by edges (in reality they are composed by paths, but this assumption is because in those sets there is only one path).

- This is the last constraint. A link between edge nodes carries not only traffic between them, but also traffic exchanged between other nodes for the scenario $t \in \{\emptyset \cup E_T\}$.

  What happens is that for every tunnel for this scenario ($\rho_{D_t}^{ij}$) that goes through the link e the corresponding traffic demand is added to e.

  This constraint establishes that all these aggregated demands must not exceed the capacity assigned to the link e.

  It is not necessary to require that the link $e = (v_p, v_q)$ is actually used ($b_e \neq 0$). As the problem wants to minimize the cost, if this particular link is not used, then its capacity eventually will be zero (so that no additional cost is added to the objective function).

4.3 Computational Complexity

In this section we prove that the RMORN problem is a **NP-Hard** problem in terms of computational complexity. We first give some relevant definitions in order to understand what this means. Basically, there are problems for which there exist an efficient algorithm to solve them, while there are others that are more complex and there are not efficient algorithms known that solve them. We prove that RMORN is one of those complex problems.

4.3.1 NP-completeness theory

The algorithms are measured by the time and memory they take to solve the problem they are designed for. The time complexity function for an algorithm gives its time requirements by giving, for each possible input length, the largest amount of time needed to solve any instance of that size. The time requirements are expressed in terms of the size of a problem instance, which intends to reflect the amount of input data needed to describe the instance. By an instance of a problem it is meant a given specification of its parameters.

The **NP**-completeness theory is applied only to decision problems which have two possible answers: *YES* or *NO*. A decision problem π consists of a set D_π of instances and a subset $Y_\pi \subset D_\pi$ of instances that have *YES* as an answer.

An optimization problem, $\tau := \min f(x)$, can be transformed into an equivalent decision problem by considering an objective bound L such as $\pi := (\min f(x) \leq L)$.

4.3.2 P and NP

Prior to defining the sets **P** and **NP** some concepts are needed.

Let us start by saying that a function $f(n)$ is $O(g(n))$ if there exists a constant c such that $|f(n)| \leq c \cdot |g(n)|$ for all $n \geq 0$.

A *polynomial time algorithm* is defined to be one whose time complexity function is $O(p(n))$ where p is a polynomial and n denotes the input length. This means that they are algorithms whose time complexity function is bounded by a polynomial. Those algorithms whose time complexity function cannot be so bounded are called *exponential time algorithms*. In general, polynomial time algorithms are preferred to those exponential ones, especially when regarding large problem instances.

Definition 4.3.1. *A problem $\pi \in$ **P** if there exists an algorithm of polynomial time complexity that solves it.*

In order to go deeper in this formalization, one can refer to [12]. There, a model for computation is fixed, which is the *Deterministic one-tape Turing Machine* or DTM. With this concept, a problem is said to be in **P** if there exists a DTM of polynomial complexity that solves it. For the purposes of this thesis, it suffices to consider the given definition.

Definition 4.3.2. *A problem $\pi \in$ **NP** if it is possible to verify that an instance of Y_π is correct in polynomial time.*

For the problems in **P** there are efficient algorithms that solve them. However, in the case of **NP** there are not known efficient algorithms that solve them (polynomial time algorithms). From the definitions, one can conclude that **P** $\subset$ **NP**, because if a solution can be found in polynomial time, it sure can be verified in polynomial time. There is a non proved conjecture that is whether **P** $=$ **NP**. Most researchers believe that this equality does not hold, but a proof has not been found yet. While this is not known, the problems are being classified in order of complexity. In the following subsection one of these clasiffications is presented.

4.3.3 NP-completeness

In this subsection we introduce the **NP**-complete problems. To do so, the concept of polynomial transformation is central.

Definition 4.3.3. *A polynomial reduction from decision problem π' to decision problem π, is a function $f : D_{\pi'} \to D_\pi$ that can be computed in polynomial time and for every $d \in D_{\pi'}$, $d \in Y_{\pi'} \iff f(d) \in Y_\pi$. We note $\pi' \preccurlyeq \pi$ if there exists a polynomial transformation from π' to π.*

The significance of polynomial reduction comes from the following lemma:

Lemma 4.3.4. *If $\pi' \preccurlyeq \pi$, then $\pi \in$ **P** implies $\pi' \in$ **P**. Equivalently $\pi' \notin$ **P** implies $\pi \notin$ **P**.*

Considering this, one can determine the difficulty of any given problem, just by finding another equivalent complexity-known problem and establishing a polynomial reduction to it.

Another thing to observe is that if $\pi'' \preccurlyeq \pi'$ and $\pi' \preccurlyeq \pi$ then $\pi'' \preccurlyeq \pi$. This is because composing two polynomial reductions results also in a polynomial reduction.

The **NP-Complete** problems are those problems in **NP** for which any other problem in **NP** can be reduced by a polynomial time bounded transformation into them.

Definition 4.3.5. *A problem π is **NP-Complete** ($\pi \in$ **NP-c**) if:*

1. *$\pi \in$ **NP**.*
2. *For every $\pi' \in$ **NP**, $\pi' \preccurlyeq \pi$.*

Occasionally, it cannot be shown that a problem belongs to **NP**, but verifies condition 2. So the problem is said to be **NP-Hard**, meaning that is at least as difficult as all the problems in **NP**.

If there is a problem $\pi \in$ **NP-c** $\cap$ **P**, then **P** $=$ **NP**. This is because if $\pi \in$ **NP-c** $\cap$ **P** there exists a polynomial time algorithm that solves it, and as π is **NP-Complete** for all $\pi' \in$ **NP**, $\pi' \preccurlyeq \pi$. From this, the polynomial reduction from π' to π is first applied and then the polynomial algorithm that solves π, resulting in an overall polynomial algorithm that solves π'.

Otherwise, if there is a problem $\pi \in$ **NP** $\setminus$ **P**, then **P** $\neq$ **NP**. For the moment, it is not known any problem that verifies either one of those properties.

4.4 MORN complexity

In this section we prove that MORN is a **NP-Hard** problem. The idea is to work with the reduced model and prove that it is **NP-Hard**. If RMORN is a **NP-Hard** problem, then it is clear that the MORN is also as it is at least as difficult at RMORN.

Notation 4.4.1. *We will denote by RRMORN to the RMORN problem where the constraint to resist simple failures of the Transport Network is relaxed.*

We will prove that this relaxation is **NP-Hard**.

Definition 4.4.2. MPPC: Minimum Point-to-Point Connection.
Let $G = (V, E)$ be a graph. Given a non-negative weight function $C : E \to \mathbb{N}$ associated to its edges, a subset $S = \{s_1, s_2, \dots, s_p\} \subseteq V$ of sources and a subset $D = \{d_1, d_2, \dots, d_p\} \subseteq V$ of destination nodes, the Minimum Point-to-Point Connection problem, $MPPC(V, E, S, D)$, consists of finding a subset $E' \subseteq E$ such that each source-destination pair is connected using only arcs of E' [21]. The cost to minimize is given by $\sum_{e \in E'} C(e)$.

Proposition 4.4.3. *MPPC is a **NP-Complete** problem [21], [15], [14], [22].*

Theorem 4.4.4. *Given an instance of the MPPC problem, there exists an instance of the RRMORN problem that solves it.*

Proof.

Lets consider the $MPPC(V, E, S, D)$ instance, where $S = \{s_1, s_2, \dots, s_p\} \subseteq V$ and $D = \{d_1, d_2, \dots, d_p\} \subseteq V$. Form this we define the following instance of the RRMORN problem:

1. G_T and G_D are topological copies of $G = (V, E)$.
2. The traffic matrix $\bar{M} = ((\vec{m}_{ij}))_{1 \le i,j \le |V_D|}$ is such that:

$$\begin{cases} \dot{m}_{s_i d_j} = 1 & \forall i, j \in \{1, 2, \dots, p\}. \\ \dot{m}_{uv} = 0 & \text{otherwise.} \\ \ddot{m}_{uv} = 0 & \forall u, v \in V_D. \end{cases}$$

3. $\hat{B} = \{b_0, b_1\}$ with $b_0 = 0$ and $b_1 \gg \sum_{\dot{m}_{ij} \neq 0} 2\dot{m}_{ij}$.
4. The lengths of the transport links are given by the cost function of the MPPC problem. This way, we define $r(l) = C(l),\ \forall l \in E_T$. These are the only costs involved in the objective function of RRMORN.
5. For $e = (v_i, v_j) \in E_D$ define T(·) as follows: $T(b_e) = \begin{cases} 1 & \text{if } b_e = b_1 \\ 0 & \text{otherwise.} \end{cases}$

   The objective is to minimize $\sum_{\substack{\rho_T = \Psi(e) \\ e=(v_i,v_j)\in E_D}} r(\rho_T^{ij}) \times T(b_e)$, so with this definition of T the product involved in the sum becomes: $r(\rho_T^{ij}) \times T(b_e) = \begin{cases} r(\rho_T^{ij}) & \text{if } b_e = b_1 \\ 0 & \text{otherwise.} \end{cases}$

   This means that the cost of the solution is not influenced by the technologies and it is only influenced by the lengths of the transport links, which means that the objective function is only affected by the cost function of the MPPC problem. From this we can observe that this objective function is the same as the one in the MPPC problem.

Considering these parameters and definition of $T(\cdot)$, the $MPPC(V, E, S, D)$ problem is equivalent to solving the defined instance of the RRMORN problem. As a component of the solution all the paths in the Data Network to satisfy all the demands should be found. Selecting all the data links from each of the paths we construct a subset $E' \subseteq E_D$ where all the source-destination pairs are connected by a path using only links of E'. We can see from this that the defined instance of the RRMORN problem solves the given instance of the MPPC problem.

Another thing that can be observed from the definition of the instance is that it can be achieved in polynomial time. The transformation from one instance to the other is very straight.

To conclude, we can say that RRMORN is at least as difficult as the MPPC problem because given instances of the MPPC problem can be reduced in polynomial time to instances of the RRMORN problem. Finally, as the MPPC is a **NP-Complete** problem, then the RRMORN problem is at least **NP-Hard**.

□

Part III

MIXED INTEGER PROGRAMMING MODEL

Chapter 5

Binary Integer Programming Model

In this chapter we present the Binary Integer Programming Models (BIP) designed to solve the problem.

Throughout the project development the programming model has evolved and changed many times. The purpose of this chapter is to present the main models that have emerged and to explain the reasons of their existence.

In order to be consistent with the assumptions made in the previous chapter, the models presented here will have the Access-Edge connections also solved (4.1) in a previous stage. This result in much more simple models as it did with the abstract model (4.2).

We will basically use the same notation as before for the input to the problem:

- $G_D = (V_E, E_D)$ and $G_T = (V_T, E_T)$ for the Data and Transport Network respectively. The nodes of G_D are all Edge nodes, that is why the set is noted by V_E.
- $\bar{M} = ((\vec{m}_{ij}))_{1 \leq i,j \leq |V_E|}$ for the traffic matrix. The vectors $\vec{m}_{ij} = (\dot{m}_{ij}, \ddot{m}_{ij}) \in \{\mathbb{R}_0^+ \times \mathbb{R}_0^+, \forall v_i, v_j \in V_E\}$. Here, $\dot{m}_{ij}$ is the committed traffic between the nodes v_i and v_j while $\ddot{m}_{ij}$ is the excess traffic between them.
- z_Q for the excess traffic capacity function. It assigns to each bandwidth sold, $\ddot{m}_{ij}$, the minimum required capacity of the links that carry that traffic.
- $\hat{B} = \{\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_{|\hat{B}|}\}$ for the available capacities, where $\hat{b_0}$ stands for the zero capacity and it means that the data links dimensioned with that capacity are not included in the solution.
- $T : \hat{B} \rightarrow \mathbb{R}_0^+$ for the technology cost function. It is assumed that $T(\hat{b_0}) = 0$.
- $r : E_T \rightarrow \mathbb{R}_0^+$ for the distance function.

- ns for the network station function.
- tns for the transport network station function.

One of the objectives is to find the capacities chosen to dimension the data links. This can be translated in finding a function $b : E_D \rightarrow \hat{B}$. We will denote $b_{ij} = b((i, j))$ to simplify the notation.

As a final comment before introducing the models, in the following, instead of representing the nodes by v_i, they will be represented by i. This is only for the purpose of being clearer.

5.1 Non-linear BIP Model

In this section we present the model achieved after solving the Access-Edge connections. As the section title suggests, this model is non linear. In the following section the solution to this non-linearity problem is presented.

5.1.1 Variables of the Problem

Before introducing the model, the definitions of the variables involved are explained. We chose for this model a node-link representation, meaning that the sets where the indexes belong are nodes and they represent links. The main binary variables are:

- $x_{ij}^{kl} = \begin{cases} 1, & \text{If the link } (i,j) \text{ of the Data Network is} \\ & \text{used to send traffic from } k \text{ to } l;\ k,l \in V_E; \\ 0, & \text{Otherwise.} \end{cases}$
- $x_{ij}^{kle} = \begin{cases} 1, & \text{If the link } (i,j) \text{ of the Data Network is} \\ & \text{used to send traffic from } k \text{ to } l,\ k,l \in V_E, \text{ in the fault scenario } e \in E_T; \\ 0, & \text{Otherwise.} \end{cases}$
- $y_{ij}^{kl} = \begin{cases} 1, & \text{If the link } (i,j) \text{ of the Transport Network is} \\ & \text{used to send data flow from } k \text{ to } l,\ k,l \in V_E; \\ 0, & \text{Otherwise.} \end{cases}$
- $w_{ij}^{t} = \begin{cases} 1, & \text{If } (i,j) \in E_D \text{ uses capacity } \hat{b}_t \in \hat{B}; \\ 0, & \text{Otherwise.} \end{cases}$

The reason why these are the main variables is that by knowing their values, all the information required to solve the problem is also known. This means that on the one hand, the structure of the Data Network is known along with the link capacities, and on the other hand, all the routing is also established.

An important remark at this point is that one should know all the values of those variables, where they have three, four or five indexes each. Intuitively, one can imagine that if the instance of the problem is big (many nodes or links or traffic requirements) then there will be a huge amount of variables.

There is other variable that appears in the model, but it is auxiliary.

- $\chi_{ij} = \begin{cases} 1, & \text{If there is a data link between the nodes } (i,j) \in E_D; \\ 0, & \text{Otherwise.} \end{cases}$

This variable serves as an indicator to know whether a link is being considered in the solution or not.

5.1.2 Complete Binary Integer Nonlinear Programming Model

Now we are in conditions to introduce the mathematical binary integer model. A solution to it results in a solution of our problem. After the model, an explanation of the equations is given.

$$\min \sum_{i,j \in V_E} \left(\sum_{(u,v) \in E_T} y_{uv}^{ij} \cdot r_T(u,v) \right) \cdot T\Big(\sum_{\hat{b}_t \in \hat{B}} w_{ij}^t \cdot \hat{b}_t\Big) \tag{5.1}$$

s.t. :

$$\sum_{\hat{b}_t \in \hat{B}} w_{ij}^t \leq 1 \qquad \forall (i,j) \in E_D. \tag{5.2}$$

$$\chi_{ij} = \sum_{\hat{b}_t \in \hat{B}} w_{ij}^t \qquad \forall (i,j) \in E_D. \tag{5.3}$$

$$\sum_{(tns(i),k) \in E_T} y_{tns(i),k}^{ij} \geq \chi_{ij} \qquad \forall i,j \in V_E. \tag{5.4}$$

$$\sum_{(k,tns(j)) \in E_T} y_{k,tns(j)}^{ij} \geq \chi_{ij} \qquad \forall i,j \in V_E. \tag{5.5}$$

$$\sum_{(p,k) \in E_T} y_{pk}^{ij} - \sum_{(k,p) \in E_T} y_{kp}^{ij} \geq \chi_{ij} - 1 \qquad \forall i,j \in V_E \ \forall p \in V_T \backslash \{tns(i), tns(j)\}. \tag{5.6}$$

$$\sum_{(p,k)\in E_T} y_{pk}^{ij} - \sum_{(k,p)\in E_T} y_{kp}^{ij} \leq 1 - \chi_{ij} \qquad \forall i,j \in V_E,\ \forall p \in V_T \setminus \{tns(i), tns(j)\}. \tag{5.7}$$

$$y_{kl}^{ij} \leq \chi_{ij} \qquad \forall i,j \in V_E,\ \forall k,l \in V_T. \tag{5.8}$$

$$\sum_{u,v\in V_E} x_{ij}^{uve} \cdot \dot{m}_{uv} + z_Q\Big(\sum_{u,v\in V_E} x_{ij}^{uve} \cdot \ddot{m}_{uv}\Big) \leq \sum_{\hat{b}_t\in\hat{B}} w_{ij}^{t} \cdot \hat{b}_t \qquad \forall e \in E_T,\ \forall (i,j) \in E_D. \tag{5.9}$$

$$\sum_{(i,k)\in E_D} x_{ik}^{ije} - 1 \geq 0 \qquad \forall e \in E_T, \forall i,j \in V_E,\ \dot{m}_{ij} \geq 0. \tag{5.10}$$

$$\sum_{(k,j)\in E_D} x_{kj}^{ije} - 1 \geq 0 \qquad \forall e \in E_T, \forall i,j \in V_E,\ \dot{m}_{ij} \geq 0. \tag{5.11}$$

$$\sum_{(p,k)\in E_D} x_{pk}^{ije} - \sum_{(k,p)\in E_D} x_{kp}^{ije} = 0 \qquad \forall e \in E_T,\ \forall i,j \in V_E\ \forall p \in (V_D \setminus \{i,j\}),\ \dot{m}_{ij} \geq 0. \tag{5.12}$$

$$x_{ij}^{uve} \leq 1 - y_{e}^{ij} \qquad \forall e \in E_T, \forall u,v \in V_E,\ \forall (i,j) \in E_D. \tag{5.13}$$

$$\sum_{k,l\in V_E} x_{ij}^{kl} \cdot \dot{m}_{kl} + z_Q\Big(\sum_{k,l\in V_E} x_{ij}^{kl} \cdot \ddot{m}_{kl}\Big) \leq \sum_{\hat{b}_t\in\hat{B}} w_{ij}^{t} \cdot \hat{b}_t \qquad \forall (i,j) \in E_D. \tag{5.14}$$

$$\sum_{(i,k)\in E_D} x_{ik}^{ij} - 1 \geq 0 \qquad \forall i,j \in V_E,\ \dot{m}_{ij} \geq 0. \tag{5.15}$$

$$\sum_{(k,j)\in E_D} x_{kj}^{ij} - 1 \geq 0 \qquad \forall i,j \in V_E,\ \dot{m}_{ij} \geq 0 \tag{5.16}$$

$$\sum_{(p,k)\in E_D} x_{pk}^{ij} - \sum_{(k,p)\in E_D} x_{kp}^{ij} = 0 \qquad \forall i,j \in V_E,\ \forall p \in (V_E \setminus \{i,j\}),\ \dot{m}_{ij} \geq 0 \tag{5.17}$$

$$w_{ij}^{t},\ x_{ij}^{kle},\ x_{ij}^{kl},\ y_{ij}^{kl} \in \{0,1\} \tag{5.18}$$

5.1.3 Description of Constraints

In this subsection an explanation of every constraint is given. Hopefully, this will help to understand a little bit better the model proposed.

(5.1)**:** The first equation is the objective function that has to be minimized. That equation represents the total cost of the network.

For every pair of data nodes, if there is a link between them, then it has a non zero cost per length associated to it. In the calculation of that cost, only the lengths of the transport links that are used to send data flow are considered. This is represented by the term $\sum_{(u,v)\in E_T} y_{uv}^{ij} \cdot r_T(u,v)$, where (i,j) is the link considered.

The other term, $T(\sum_{\hat{b}_t\in\hat{B}} w_{ij}^t \cdot \hat{b}_t)$ is the cost of the technology selected for the link (i,j).

The global cost is obtained adding all these individual costs.

At this point it is worth mentioning again the function $b : E_D \rightarrow \hat{B}$ introduced at the begining of this chapter. The value b_{ij} that is the capacity assigned to the link (i,j) now has a clear expression: $b_{ij} = \sum_{\hat{b}_t\in\hat{B}} w_{ij}^t \cdot \hat{b}_t$.

Besides, it is also possible to express the objective function as in the MORN model (4.2):

$$\sum_{i,j\in V_E} r_T(\rho_T^{ij})T(b_{ij})$$

because the binary variables present in the objective function indicate the transport path of a data link.

As a last observation regarding the objective function, we remark that it is a non-linear expression.

(5.2): The first constraint means that for a link $(i,j) \in E_D$ at most one technology can be chosen to dimension it.

(5.3): The second constraint defines the variable χ defined before. If a technology was chosen for the link $(i,j) \in E_D$ its value will be 1, otherwise it will be zero as every w_{ij}^t is zero.

(5.4), (5.5), (5.6), (5.7): These constraints construct the paths in the Transport Network associated to each data link.

The equations (5.4), (5.5) imply that if a non-zero capacity was chosen to dimension data link $(i,j) \in E_D$, then at least one transport link must start from $tns(i)$ and at least one transport link has to arrive at $tns(j)$. They create the extremities of the path.

The equations (5.6), (5.7) refer to the middle of the path. That is why they consider only nodes that differ from $tns(i)$ and $tns(j)$. The traffic from $tns(i)$ to $tns(j)$ that passes through an intermediate transport node must leave it.

(5.8): This is an integrity constraint. If no capacity was assigned to data link $(i,j) \in E_D$ then no transport path should be found.

(5.9): The capacity of a data link must be able to support the whole committed traffic carried by it, plus a certain percentage of the excess or peak traffic (given by the function z_Q), in every fault scenario. It is worth mentioning that the z_Q function is completely established in [29] and it is piecewise linear and it was embedded in the data, so this constraint results linear.

(5.10), (5.11), (5.12): These constraints correspond to the routing in the Data Network in every fault scenario $e \in E_T$. They mean that $\forall i, j \in V_E$ such that $\dot{m}_{ij} > 0$, there must be at least one path between i and j that connects them:

$$\sum_{(i,k) \in E_D} x_{ik}^{ije} \geq 1,$$
$$\sum_{(k,j) \in E_D} x_{kj}^{ije} \geq 1,$$
$$\sum_{(p,k) \in E_D} x_{pk}^{ije} - \sum_{(k,p) \in E_D} x_{kp}^{ije} = 0, \quad \forall p \in (V_D \setminus \{i, j\}).$$

(5.13): This constraint establishes the relation between data links and transport links. It means that if a transport link $e \in E_T$ fails, then all the data links whose transport flow is routed through e cannot be used in fault scenario e.

(5.14): The capacity of a data link must be able to support the whole committed traffic carried by it, plus a certain percentage of the excess or peak traffic, in the nominal scenario (the one where no transport link fails). As in constraint (5.9), this one is also linear.

(5.15), (5.16), (5.17): These constraints correspond to the routing in the Data Network in the nominal scenario. For every $i, j \in V_E$ such that $\dot{m}_{ij} > 0$, there must be at least one path between i and j that connects them:

$$\sum_{(i,k) \in E_D} x_{ik}^{ij} \geq 1,$$
$$\sum_{(k,j) \in E_D} x_{kj}^{ij} \geq 1,$$
$$\sum_{(p,k) \in E_D} x_{pk}^{ij} - \sum_{(k,p) \in E_D} x_{kp}^{ij} = 0, \quad \forall p \in (V_D \setminus \{i, j\}).$$

(5.18): The last constraint remarks the binary characteristic of the variables.

All the constraints have been explained and now we are in conditions to take a global view of the model.

Firstly, the flow implementation can be viewed as follows:

Definition 5.1.1. *Let* $i, j \in V_E$, $\rho_T^{ij} = \{(u, v) \in E_T \ / \ y_{uv}^{ij} = 1\}$.

Secondly, if we take a look at the objective function, we rapidly see that there is a product of binary variables involved. This makes the model non-linear. Many approaches have been undertaken and in the next section we present one of the models achieved.

It is worth remarking that solving this problem in an exact way is not an easy task. Moreover, when testing it with big instances it is not possible. All the other parallel works have taken a metaheuristic approach [29], [8], [10]. Our approach will be explained in the next chapter.

5.2 Linear BIP Model

In this section we present a solution to the non-linearity problem of the objective function found in the previous section. Despite the fact that there is a whole new model to be introduced in this section, it is very alike to the one that we just showed in Section-5.1.

As we did before, we begin with the introduction of variables. The first group has the same meaning that the main variables of the previous section. The reason why we explain them again is that there was an initial change from the non-linear model that is the representation of the variables. The ones explained in the next subsection differ from the ones already defined in that their indexes represent links and not nodes. This makes the model easier to understand and besides it has a computational effect as the number of total variables is considerably reduced.

In this section we will assume that the demands are given in a graph $G_M = (V_E, E_M)$, where the set of nodes is the same as in the Data Network and the links represent the nodes among which there is a demand.

5.2.1 Variables of the Problem

For this model, we chose a link representation so that the indexes of the variables represent links.

- $x^{de} = \begin{cases} 1, & \text{If the link } e \text{ of the Data Network is used to send traffic} \\ & \text{between the nodes of } d \in E_M \text{ in the nominal scenario;} \\ 0, & \text{Otherwise.} \end{cases}$

- $x_f^{de} = \begin{cases} 1, & \text{If the link } e \text{ of the Data Network is used to send traffic} \\ & \text{between the nodes of } d \in E_M\text{, in the fault scenario } f \in E_T\text{;} \\ 0, & \text{Otherwise.} \end{cases}$

- $y_f^e = \begin{cases} 1, & \text{If the link } f \text{ of the Transport Network is} \\ & \text{used to send data for link } e \in E_D \text{ of the Data Network;} \\ 0, & \text{Otherwise.} \end{cases}$

- $w_t^e = \begin{cases} 1, & \text{If link } e \in E_D \text{ uses capacity } \hat{b}_t \in \hat{B}\text{;} \\ 0, & \text{Otherwise.} \end{cases}$

The main change in these variables is that in the ones defined here the links and demands are represented by one index instead of two as before. What is indexed here are the links and demands and not the nodes. This results in a clearer reading of them aside from what was mentioned at the begining of this section.

Inspite that change, the main difference between models is the introduction of a new set of variables that will allow us to linearize the objective function, that was a difficulty to circumvent in the other model. Its definition is the following:

Definition 5.2.1. $\eta_{ft}^{e} := y_f^e \cdot w_t^e$

These variables allow a change in the definition of the objective function (as that product was directly involved in it).

There are other variables that appear in the model, but they are auxiliary. One that we can highlight is:

- $\chi^e = \begin{cases} 1, & \text{If data link } e \in E_D \text{ has a technology assigned;} \\ 0, & \text{Otherwise.} \end{cases}$

This variable serves as an indicator to know whether a link is being considered in the solution or not.

The last group of variables that is completely auxiliary is the following:

- $\gamma^{ep} = \begin{cases} 1, & \text{If node } p \in V_E \text{ of the Data Network is used in the routing} \\ & \text{of data link } e \in E_D \text{ in the nominal scenario;} \\ 0, & \text{Otherwise.} \end{cases}$

- $\gamma_f^{ep} = \begin{cases} 1, & \text{If node } p \in V_D \text{ of the Data Network used in the routing} \\ & \text{of data link } e \in E_D \text{ in fault scenario } f \in E_T \text{ of the Transport Network;} \\ 0, & \text{Otherwise.} \end{cases}$

- $\delta_q^e = \begin{cases} 1, & \text{If node } q \in V_T \text{ of the Transport Network is} \\ & \text{used in the routing of } e \in E_D \text{ of the Data Network;} \\ 0, & \text{Otherwise.} \end{cases}$

5.2.2 Complete Binary Integer Programming Model

Now we introduce the linearized mathematical binary integer model. A solution to it results in a solution of our problem. After the model, an explanation of the equations is given.

Note: 5.2.2.

- $\sim$ *means "incident to";*
- e_1 *is the source of* e *and* e_2 *the target of* e*.*

$$\min \sum_{e \in E_D} \sum_{f \in E_T} \sum_{\hat{b}_t \in \hat{B}} r_T(f) \cdot T(\hat{b}_t) \cdot \eta^e_{ft} \tag{5.19}$$

s.t. :

$$y^e_f + w^e_t - 1 \leq \eta^e_{ft} \qquad \forall e \in E_D, \forall f \in E_T, \forall \hat{b}_t \in \hat{B}. \tag{5.20}$$

$$\eta^e_{ft} \leq y^e_f \qquad \forall e \in E_D, \forall f \in E_T, \forall \hat{b}_t \in \hat{B}. \tag{5.21}$$

$$\eta^e_{ft} \leq w^e_t \qquad \forall e \in E_D, \forall f \in E_T, \forall \hat{b}_t \in \hat{B}. \tag{5.22}$$

$$\sum_{\hat{b}_t \in \hat{B}} w^e_t \leq 1 \qquad \forall e \in E_D. \tag{5.23}$$

$$\chi^e = \sum_{\hat{b}_t \in \hat{B}} w^e_t \qquad \forall e \in E_D. \tag{5.24}$$

$$\sum_{f \in E_T : f \sim tns(e_1)} y^e_f = \chi^e \qquad \forall e \in E_D. \tag{5.25}$$

$$\sum_{f \in E_T : f \sim tns(e_2)} y^e_f = \chi^e \qquad \forall e \in E_D. \tag{5.26}$$

$$\sum_{f \in E_T : f \sim q} y^e_f = 2\delta^e_q \qquad \forall e \in E_D \, \forall q \in V_T \setminus \{tns(e_1), tns(e_2)\}. \tag{5.27}$$

$$y^e_f \leq \chi^e \qquad \forall e \in E_D, \forall f \in E_T. \tag{5.28}$$

$$\sum_{d \in E_M} x^{de}_f \cdot \dot{m}_d + z_Q\Big(\sum_{d \in E_M} x^{de}_f \cdot \ddot{m}_d\Big) \leq \sum_{\hat{b}_t \in \hat{B}} w^e_t \cdot \hat{b}_t \qquad \forall f \in E_T, \forall e \in E_D. \tag{5.29}$$

$$\sum_{e \in E_D : e \sim d_1} x_f^{de} = 1 \qquad \forall f \in E_T,\ \forall d \in E_M. \tag{5.30}$$

$$\sum_{e \in E_D : e \sim d_2} x_f^{de} = 1 \qquad \forall f \in E_T,\ \forall d \in E_M. \tag{5.31}$$

$$\sum_{e \in E_D : e \sim p} x_f^{de} = 2\gamma_f^{ep} \qquad f \in E_T,\ \forall d \in E_M\ \forall p \in (V_E \backslash \{d_1, d_2\}). \tag{5.32}$$

$$x_f^{de} \leq 1 - y_f^e \qquad \forall f \in E_T, \forall e \in E_D,\ \forall d \in E_M. \tag{5.33}$$

$$\sum_{d \in E_M} x^{de} \cdot \dot{m}_d + z_Q\Big(\sum_{d \in E_M} x^{de} \cdot \ddot{m}_d\Big) \leq \sum_{\hat{b}_t \in \hat{B}} w_t^e \cdot \hat{b}_t \qquad \forall e \in E_D. \tag{5.34}$$

$$\sum_{e \in E_D : e \sim d_1} x^{de} = 1 \qquad \forall d \in E_M. \tag{5.35}$$

$$\sum_{e \in E_D : e \sim d_2} x^{de} = 1 \qquad \forall d \in E_M. \tag{5.36}$$

$$\sum_{e \in E_D : e \sim p} x^{de} = 2\gamma^{ep} \qquad \forall d \in E_M,\ \forall p \in (V_E \backslash \{d_1, d_2\}). \tag{5.37}$$

$$x_f^{de},\ x^{de},\ y_f^e,\ w_t^e,\ \chi^e,\ \gamma^{ep},\ \gamma_f^{ep},\ \delta_q^e,\ \eta_{ft}^e, \in \{0, 1\} \tag{5.38}$$

5.2.3 Description of Constraints

In this subsection a brief explanation of every constraint is given as they are basically the same equations as before.

(5.19): The first equation is the objective function that has to be minimized. That equation represents the total cost of the network.

This expression is exactly the same as the one in the previous model. The difference is that the product of the variables y and w in the old model is replaced by the new variable η in the new model.

(5.20), (5.21), (5.22): This group of constraints is the principal difference from the constraints of the previous model. They are key to the model because they make the both models equivalent. They represent the linearization of the objective function and make the variable η behave as the product of the variables y and w. This is explained at the end of the description of the constraints.

(5.23): This constraint means that for a link $e \in E_D$ at most one technology can be chosen to dimension it.

(5.24): It defines the variable χ. If a technology was chosen for the link $e \in E_D$ its value will be 1, otherwise it will be zero.

(5.25), (5.26), (6.4): These constraints construct the paths in the Transport Network associated to each data link.

The equations (5.25) and (5.26) imply that if a non-zero capacity was chosen to dimension data link $e \in E_D$, then one transport link must start from $tns(e_1)$ and one transport link has to arrive at $tns(e_2)$. They create the extremities of the path.

The equation (6.4) refers to the middle of the path. The degree of an intermediate node in the path from $tns(e_1)$ to $tns(e_2)$ must be zero (if the node is not used) or 2 (if the node is used). The traffic that enters that node must leave it.

(5.28): If no capacity was assigned to data link $e \in E_D$ then no transport path should be found.

(5.29): The capacity of a data link must be able to support the whole committed traffic carried by it, plus a certain percentage of the excess or peak traffic (given by the function z_Q), in every fault scenario. For the explanation of the linearity of this constraint refer to the explanation of constraint (5.9).

(5.30), (5.31), (5.32): These constraints correspond to the routing in the Data Network in every fault scenario $f \in E_T$.

Equations (5.30) and (5.31) imply that for every demand $d \in E_M$, one data link must start from d_1 and one data link has to arrive at d_2. They create the extremities of the path.

Equation (5.32) says that the degree of an intermediate node in the path from d_1 to d_2 must be zero (if the node is not used) or 2 (if the node is used). The traffic that enters that node must leave it.

(5.33): This constraint establishes the relation between data links and transport links. It means that if a transport link $f \in E_T$ fails, then all the data links whose transport flow is routed through f cannot be used in fault scenario f.

(5.34): Same as (5.29) but in the nominal scenario (the one where no transport link fails).

(5.35), (5.36), (5.37): Same as (5.30), (5.31), (5.32) but in the nominal scenario.

(5.38): The last constraint remarks the binary characteristic of the variables.

This is the end of the description of the constraints. Now we explain why η behaves as the product of the variables y and w. Lets see this in a general way. Suppose that we have a product of binary variables x_1 and x_2 and we want to replace it by $x = x_1 \cdot x_2$ without changing their meaning, that is what we have here with η and the others. The new variable takes value one if both x_1 and x_2 are also one, or zero in any other case. After replacing the product with x these constraints were added:

$$\begin{cases} x_1 + x_2 - 1 \leq x \\ x \leq x_1 \\ x \leq x_2 \end{cases}$$

If any of x_1 or x_2 is zero, then by the last two equations so will be x. Otherwise, if both are one, then the first equation states that $1 + 1 - 1 \leq x$, meaning that $x \geq 1$. But x is binary, so it should be one. And this is what we wanted.

What is interesting about this is that it can be easily generalized to a product of n binary variables in the following way. Suppose that now we have a product of n binary variables $x_1, x_2, \dots x_n$ and we want to replace it by $x = x_1 \cdot x_2 \dots x_n$ without changing the meaning. The new variable takes value one if all $x_1, x_2, \dots x_n$ are also one, or zero in any other case. So, after replacing the product with x, $n + 1$ constraints should be added:

$$\begin{cases} x_1 + x_2 + \dots x_n - (n-1) \leq x \\ x \leq x_1 \\ x \leq x_2 \\ \vdots \\ x \leq x_n \end{cases}$$

In this case, if all $x_1, x_2, \dots x_n$ are one, then the first equation gives us the inequality

$$\underbrace{1 + 1 + \dots + 1}_{n\ times} - (n-1) \leq x,$$

that means $x \geq 1$. And, as x is binary, it must be one. With this procedure, we can have any product of n binary variables linearized.

As final comments on this section, we can see that the flow implementation can be viewed using the variables y as in the previous section.

With this model, the objective function and all the constraints are linear, and linear problems are easier to solve than non-linear problems.

Despite the fact that the model is linear, it continues to be very big and the amount of variables becomes huge as the number of nodes, links and demands increase.

5.3 Linear BIP Model with Flow Implementation

The model presented in this section is mainly the same as the lastly introduced with the difference that we use a flow implementation in the Transport Network. Therefore, the only constraints that change with respect to the previous model are those regarding the paths in the Transport Network. The aim of doing this was to test if there was a significant improvement in the performance of the model, since network flow problems are easy to solve.

In this section we will also assume that the demands are given in a graph $G_M = (V_E, E_M)$, where the set of nodes is the same as in the Data Network and the links represent the nodes among which there is a demand.

5.3.1 Variables of the Problem

These variables are the same as before:

- $x^{de} = \begin{cases} 1, & \text{If the link } e \text{ of the Data Network is used to send traffic} \\ & \text{between the nodes of } d \in E_M \text{ in the nominal scenario;} \\ 0, & \text{Otherwise.} \end{cases}$

- $x_f^{de} = \begin{cases} 1, & \text{If the link } e \text{ of the Data Network is used to send traffic} \\ & \text{between the nodes of } d \in E_M\text{, in the fault scenario } f \in E_T\text{;} \\ 0, & \text{Otherwise.} \end{cases}$

- $w_t^e = \begin{cases} 1, & \text{If link } e \in E_D \text{ uses capacity } \hat{b}_t \in \hat{B}\text{;} \\ 0, & \text{Otherwise.} \end{cases}$

Before continuing, we use the same notation as before:

Note: 5.3.1.
- *$e \sim v$ means "v is incident to link e";*
- *e_1 is the source of e and e_2 the target of e.*

We introduce a new set that is explained immediately afterwards:

Definition 5.3.2. $E_T^P = \{(i,j) \ / \ (i,j) \in E_T \text{ or } (j,i) \in E_T,\ \forall i,j \in V_T\}$.

In E_T every arc is undirected. However, in order to make a flow implementation, we need to have directed arcs. This is the reason for introducing this new set E_T^P that for every undirected arc $(i,j) \in E_T$ considers two directed arcs (i,j) and (j,i). Therefore, the amount of links in E_T^P exactly doubles the amount of links in E_T.

One variable that has changed from the previous model to this model is y. This is now a flow variable and the direction of the links is necessary and needs to be represented. In order to do so, instead of indexing the transport links we now need to represent them by their extremities. Also, it is now a real variable and not binary:

- $y^e_{f_1 f_2} \in [0,1], \forall (f_1, f_2) \in E^P_T$.

The interpretation that we give to this variable is the following:

- $y^e_{f_1 f_2} = \begin{cases} 1, & \text{If the link } (f_1, f_2) \in E^P_T \text{ of the Transport Network is} \\ & \text{used to send data for link } e \in E_D \text{ of the Data Network;} \\ 0, & \text{Otherwise.} \end{cases}$

With this modification, the variable η also needs an actualization.

- $\eta^e_{ft} := (y^e_{f_1 f_2} + y^e_{f_2 f_1}) \cdot w^e_t$

All the auxiliary variables (5.2.1) are exactly the same, except for δ that does not appear in this model.

5.3.2 Complete Binary Integer Programming Model

In this subsection the model is introduced. As stated before, the only big change was in the variable y. The whole model is presented only with the purposes of having it complete.

$$\min \sum_{e \in E_D} \sum_{f \in E_T} \sum_{\hat{b}_t \in \hat{B}} r_T(f) \cdot T(\hat{b}_t) \cdot \eta^e_{ft} \tag{5.39}$$

s.t. :

$$y^e_{f_1 f_2} + y^e_{f_2 f_1} + w^e_t - 1 \leq \eta^e_{ft} \qquad \begin{array}{l} \forall e \in E_D, \forall \hat{b}_t \in \hat{B}, \\ \forall f = (f_1, f_2) \in E_T. \end{array} \tag{5.40}$$

$$\eta^e_{ft} \leq y^e_{f_1 f_2} + y^e_{f_2 f_1} \qquad \begin{array}{l} \forall e \in E_D, \forall \hat{b}_t \in \hat{B}, \\ \forall f = (f_1, f_2) \in E_T. \end{array} \tag{5.41}$$

$$\eta^e_{ft} \leq w^e_t \qquad \begin{array}{l} \forall e \in E_D, \forall f \in E_T, \\ \forall \hat{b}_t \in \hat{B}. \end{array} \tag{5.42}$$

$$\sum_{\hat{b}_t \in \hat{B}} w^e_t \leq 1 \qquad \forall e \in E_D. \tag{5.43}$$

$$\chi^e = \sum_{\hat{b}_t \in \hat{B}} w^e_t \qquad \forall e \in E_D. \tag{5.44}$$

$$\sum_{\substack{(f_1,f_2)\in E_T^P: \\ f_1=tns(e_1)}} y^e_{f_1 f_2} \geq \chi^e \qquad \forall e{\in}E_D. \tag{5.45}$$

$$\sum_{\substack{(f_1,f_2)\in E_T^P: \\ f_2=tns(e_2)}} y^e_{f_1 f_2} \geq \chi^e \qquad \forall e{\in}E_D. \tag{5.46}$$

$$\sum_{f_2:(f_1,f_2)\in E_T^P} y^e_{f_1 f_2} = \sum_{f_2:(f_2,f_1)\in E_T^P} y^e_{f_2 f_1} \qquad \begin{array}{l}\forall e{=}(e_1,e_2){\in}E_D \\ \forall f_1{\in}V_T\backslash \\ \{tns(e_1),tns(e_2)\}.\end{array} \tag{5.47}$$

$$y^e_{f_1 f_2} + y^e_{f_2 f_1} \leq \chi^e \qquad \begin{array}{l}\forall e{\in}E_D, \\ \forall f{=}(f_1,f_2){\in}E_T.\end{array} \tag{5.48}$$

$$\sum_{d\in E_M} x^{de}_f \cdot \dot{m}_d + z_Q(\sum_{d\in E_M} x^{de}_f \cdot \ddot{m}_d) \leq \sum_{\hat{b}_t\in\hat{B}} w^e_t \cdot \hat{b}_t \qquad \forall f{\in}E_T,\ \forall e{\in}E_D. \tag{5.49}$$

$$\sum_{e\in E_D: e\sim d_1} x^{de}_f = 1 \qquad \begin{array}{l}\forall f{\in}E_T, \\ \forall d{=}(d_1,d_2){\in}E_M.\end{array} \tag{5.50}$$

$$\sum_{e\in E_D: e\sim d_2} x^{de}_f = 1 \qquad \begin{array}{l}\forall f{\in}E_T, \\ \forall d{=}(d_1,d_2){\in}E_M.\end{array} \tag{5.51}$$

$$\sum_{e\in E_D: e\sim p} x^{de}_f = 2\gamma^{ep}_f \qquad \begin{array}{l}f{\in}E_T, \\ \forall d{=}(d_1,d_2){\in}E_M \\ \forall p{\in}(V_E\backslash\{d_1,d_2\}).\end{array} \tag{5.52}$$

$$x^{de}_f \leq 1 - (y^e_{f_1 f_2} + y^e_{f_2 f_1}) \qquad \begin{array}{l}\forall d{\in}E_M,\forall e{\in}E_D, \\ \forall f{=}(f_1,f_2){\in}E_T.\end{array} \tag{5.53}$$

$$\sum_{d\in E_M} x^{de} \cdot \dot{m}_d + z_Q(\sum_{d\in E_M} x^{de} \cdot \ddot{m}_d) \leq \sum_{\hat{b}_t\in\hat{B}} w^e_t \cdot \hat{b}_t \qquad \forall e{\in}E_D. \tag{5.54}$$

$$\sum_{e\in E_D: e\sim d_1} x^{de} = 1 \qquad \forall d{=}(d_1,d_2){\in}E_M. \tag{5.55}$$

$$\sum_{e\in E_D: e\sim d_2} x^{de} = 1 \qquad \forall d{=}(d_1,d_2){\in}E_M. \tag{5.56}$$

$$\sum_{e\in E_D: e\sim p} x^{de} = 2\gamma^{ep} \qquad \begin{array}{l}\forall d{=}(d_1,d_2){\in}E_M, \\ \forall p{\in}(V_E\backslash\{d_1,d_2\}).\end{array} \tag{5.57}$$

$$x^{de}_f,\ x^{de},\ y^e_f,\ w^e_t,\ \chi^e,\ \gamma^{ep},\ \gamma^{ep}_f,\ \eta^e_{ft},\ \in \{0,1\} \tag{5.58}$$

5.3.3 Description of Constraints

It is expected that the similarities between this model and the one in the previous section (5.2) are clear. There is not much to say about the constraints that has not been already said.

The objective function (5.39) and constraints (5.42), (5.43), (5.44), (5.49), (5.50), (5.51), (5.52), (5.54), (5.55), (5.56), (5.57) and (5.58) are without doubt exactly the same defined in the previous section (5.2).

With respect to constraints (5.40), (5.41), (5.48) and (5.53) they are also the same. Instead of having y_f^e in the equations it appears $y_{f_1f_2}^e + y_{f_2f_1}^e$. The question is why this is the same. One hypothesis is that transport links are bidirectional (G_T is an undirected graph). So it is enough that only one of $y_{f_1f_2}^e$ or $y_{f_2f_1}^e$ is one to y_f^e be also one. And if both of them are zero, so will be y_f^e.

If we refer to the section title, it says that there is going to be a flow implementation. This is the main difference between both models and it is represented in constraints (5.45), (5.46) and (5.47).

They construct the paths in the Transport Network associated to each data link as they did before, but in a different way. Equations (5.45) and (5.46) imply that if a non-zero capacity was chosen to dimension data link $e \in E_D$, then one transport link must start from $tns(e_1)$ and one transport link has to arrive at $tns(e_2)$. They create the extremities of the path. This is the same idea as before.

With respect to the capacity constraints, they are linear as explained in constraint (5.9).

Constraint (5.47) is the one that differs the most. It also refers to the middle of the path, but it is implemented as a balance equation. The traffic that enters that node must leave it.

Chapter 6

Decomposition Approach

In this chapter we present a different approach to solve the problem.

As the problem we are dealing with is **NP-Hard** it is expected that large instances of it will not be solved with the exact approach. This is the reason of this chapter, where an approximation to the solution will be searched.

To be consistent with the previous chapters, the same assumptions are made in the present chapter, as well as the notations used. The starting point for the model presented below is the second model introduced in the previous chapter, in 5.2. Because of this, the variables of this model are exactly the same as the ones in that model. However, only with the purpose of having them present, they are summarized in the following section.

6.1 Variables of the Problem

The most relevant variables of the model are shown together in this section.

- $x^{de} = \begin{cases} 1, & \text{If the link } e \text{ of the Data Network is used to send traffic} \\ & \text{between the nodes of } d \in E_M \text{ in the nominal scenario;} \\ 0, & \text{Otherwise.} \end{cases}$

- $x_f^{de} = \begin{cases} 1, & \text{If the link } e \text{ of the Data Network is used to send traffic} \\ & \text{between the nodes of } d \in E_M\text{, in the fault scenario } f \in E_T; \\ 0, & \text{Otherwise.} \end{cases}$

- $w_t^e = \begin{cases} 1, & \text{If link } e \in E_D \text{ uses capacity } \hat{b}_t \in \hat{B}; \\ 0, & \text{Otherwise.} \end{cases}$

- $y_f^e = \begin{cases} 1, & \text{If the link } f \text{ of the Transport Network is} \\ & \text{used to send data for link } e \in E_D \text{ of the Data Network;} \\ 0, & \text{Otherwise.} \end{cases}$

6.2 Description of the new model

This section explains the approximation approach that was followed.

This approach consists on solving the problem instances in two steps. In chapter (3) we described how the traffic is routed in the Transport Network (3.4.2). There, we highlighted the fact that for each edge in the Data Network, its path in the Transport Network is unique. We take advantage of this and solve these paths for every data link in a first step.

After all these paths are chosen, the dimensioning of the data links and routing in the Data Network is solved in a second step.

An important remark is that although the base model is the one already mentioned, we will see that with this approach the objective functions are linear and there is no need to introduce the auxiliary variables η. Because of this, they will not appear in these models.

Taking into account all this, the model for each of the steps is described in the following subsections.

6.2.1 First step of the approach

This model will find the best transport path for each data link. In order to do that, the length of those paths is minimized, which will result in a minimization of the total cost of the final solution.

$$\min \sum_{e \in E_D} \sum_{f \in E_T} r_T(f) \cdot y_f^e \tag{6.1}$$

s.t. :

$$\sum_{f \in E_T : f \sim tns(e_1)} y_f^e = 1 \quad \forall e {\in} E_D. \tag{6.2}$$

$$\sum_{f \in E_T : f \sim tns(e_2)} y_f^e = 1 \quad \forall e {\in} E_D. \tag{6.3}$$

$$\sum_{f \in E_T : f \sim q} y_f^e = 2\delta_q^e \quad \begin{array}{l} \forall e {\in} E_D \; \forall q {\in} V_T \setminus \\ \{tns(e_1), tns(e_2)\}. \end{array} \tag{6.4}$$

$$y_f^e,\ \delta_q^e,\ \in \{0, 1\} \tag{6.5}$$

6.2.2 Second step of the approach

The solution that is returned from the first step serves as an input to the second part. In this model, we will abuse notation and y_f^e will not represent a variable, but a parameter whose value is the result of the first part. This is the reason why the objective function is linear; the only variables that appear are w_t^e.

Two things remain to have a complete solution to the problem, that are the capacity of each data link used and the paths on the Data Network to satisfy each demand. These paths should be found for every fault scenario in addition to the paths in the nominal scenario.

$$\min \sum_{e\in E_D} \sum_{f\in E_T} \sum_{\hat{b}_t\in\hat{B}} r_T(f)\cdot T(\hat{b}_t)\cdot y_f^e \cdot w_t^e \tag{6.6}$$

s.t. :

$$\sum_{\hat{b}_t\in\hat{B}} w_t^e \le 1 \qquad \forall e\in E_D. \tag{6.7}$$

$$\chi^e = \sum_{\hat{b}_t\in\hat{B}} w_t^e \qquad \forall e\in E_D. \tag{6.8}$$

$$\sum_{d\in E_M} x_f^{de}\cdot \dot{m}_d + z_Q(\sum_{d\in E_M} x_f^{de}\cdot \ddot{m}_d) \le \sum_{\hat{b}_t\in\hat{B}} w_t^e\cdot \hat{b}_t \qquad \forall f\in E_T,\ \forall e\in E_D. \tag{6.9}$$

$$\sum_{e\in E_D:e\sim d_1} x_f^{de} = 1 \qquad \forall f\in E_T,\ \forall d\in E_M. \tag{6.10}$$

$$\sum_{e\in E_D:e\sim d_2} x_f^{de} = 1 \qquad \forall f\in E_T,\ \forall d\in E_M. \tag{6.11}$$

$$\sum_{e\in E_D:e\sim p} x_f^{de} = 2\gamma_f^{ep} \qquad f\in E_T,\ \forall d\in E_M\ \forall p\in(V_E\setminus\{d_1,d_2\}). \tag{6.12}$$

$$x_f^{de} \le 1 - y_f^e \qquad \forall f\in E_T, \forall e\in E_D,\ \forall d\in E_M. \tag{6.13}$$

$$\sum_{d\in E_M} x^{de}\cdot \dot{m}_d + z_Q(\sum_{d\in E_M} x^{de}\cdot \ddot{m}_d) \le \sum_{\hat{b}_t\in\hat{B}} w_t^e\cdot \hat{b}_t \qquad \forall e\in E_D. \tag{6.14}$$

$$\sum_{e\in E_D:e\sim d_1} x^{de} = 1 \qquad \forall d\in E_M. \tag{6.15}$$

$$\sum_{e \in E_D : e \sim d_2} x^{de} = 1 \qquad \forall d \in E_M. \tag{6.16}$$

$$\sum_{e \in E_D : e \sim p} x^{de} = 2\gamma^{ep} \qquad \forall d \in E_M, \forall p \in (V_E \setminus \{d_1, d_2\}). \tag{6.17}$$

$$x_f^{de},\, x^{de},\, w_t^e,\, \chi^e,\, \gamma^{ep},\, \gamma_f^{ep},\, \in \{0,1\} \tag{6.18}$$

In order not to do this heavier than it already is, the constraints are not explained because they are exactly the same as in (5.3.3).

6.2.3 Final solution

The model for the second step is executed independently for every fault scenario and also for the nominal scenario. This brings us a last aspect that needs to be taken care of that is how the final solution is built from the solution of all scenarios.

The easiest part are the paths in the Data Network for each fault scenario as well as the nominal one, because they are exactly the same returned in each execution of the model.

Lets see now how the capacity for each data link is selected. Suppose that $G_D = (V_E, E_D)$ and $G_T = (V_T, E_T)$ are the Data Network and the Transport Network respectively. If $f \in E_T$ is the transport link that has failed, lets denote by b_e^f the capacity chosen for data link $e \in E_D$ in the fault scenario f. Also, $b_e^{f_0}$ will denote the capacity in the nominal scenario.

The capacity that we select for a data link $e \in E_D$ should be able to support the flow in every scenario, having all these capacities the data link e will be dimensioned with the biggest one of them:

$$b_e = \max_{f \in E_T \cup f_0} \{b_e^f\}.$$

With this procedure, a capacity is assigned to each data link and it is enough to route every demand. Also, all the paths for each data link used is given, as well as the paths in the Data Network to route each demand. Consequently, the solution returned with this iterated approach is a feasible solution for the original problem.

In the following chapter a description of the test cases is given in order to test and compare the complete model with this decomposition approach.

Part IV

RESULTS

Chapter 7

Test cases description

The aim of this chapter is to explain the tests cases generated. These test cases were created by the project team and are grouped in two categories. The first group intends to describe different situations and realities of ANTEL, and they take into consideration several parameters that are under their control [29]. The second group are tests created in order to evaluate correctness and performance of the algorithms.

The data provided by ANTEL as an input to the problem is also detailed.

7.1 Control Variables for the First Group of Test Cases

7.1.1 Demand

The Network structure is mainly defined by Residential Internet traffic. Other kinds of traffic such as 'Enterprise traffic' or 'Added Value traffic' (IPTV, VoIP, Cardales) have shown to be minor. Residential Internet traffic is dominant in the way that the presence or absence of the latter types of traffic in prototype problems only affected G_E in one of five times, and when G_E did change, it was in a minimal way [29].

The reasons for such a low impact are the relative low number of clients or low bandwidth demanded in the case of Enterprise services. However, when it comes to Added Value services, the multicast nature of the majority of this traffic appear to spread efficiently in the backbone.

Consequently, the important parameters identified are clients and bandwidth, more explicitly the amount of sold services and the associated bandwidth consumption of each service.

The demand is the total bandwidth sold, and we considered two different kinds of demands that are called `demL` and `demH`. They represent a low demand scenario and a high demand scenario respectively. They are the extreme realistic scenarios of ANTEL.

7.1.2 Requirements

The requirements over the network not only depend on the demand requirements, but also on the consumption of products offered by the telecommunication enterprise. In particular, on the variety of free and flexible products.

This concept was modelled as $z_Q(bw)$ [29] and it will be the second variable considered in the analysis.

We will name `zQL` and `zQH` the scenarios for a low z_Q design and a high z_Q design respectively. The explicit data is introduced in the next section.

7.1.3 Contents

The cost associated to the utilization of international links plays an important role because if the location of the content that clients demand is abroad then ANTEL has to pay foreign companies to access it. On the contrary, if the content required by clients is national, then its cost is lower.

As a consequence, a strategy followed in order to reduce the overcost of hiring international connections is the use of "caches" or other forms of national Internet content. With this, contents that were located abroad are stored locally and the cost of accessing them is lower.

The scenarios created acknowledge the percentage of national content demanded. The two alternatives are denoted `intL` and `intH`. The former represents a 25% of Internet content ending nationally, while the latter represents the totality of Internet traffic being international.

7.1.4 Architecture

The last variable considers different network architectures. Nowadays, the ATM and IP/MPLS Networks have the primary objective of concentrating the HSI traffic towards another network, known as "Public IP Network". This network is the part of Internet that is inside Uruguay, in particular, it is physically located in four AMM stations called Aguada, Centro, Cordón and Unión, besides the Americas NAP settled in Miami. The Public IP Network concentrates all international links and it is responsible of routing Internet traffic.

Previously, with only ATM Network there were no other options about this. However, the new MPLS Network gives us another option as it has the potential feature of concentrating the HSI traffic and routing the Internet traffic, this is, collapsing ATM and Public IP Networks in only one MPLS Network. Due to this, it is interesting to evaluate if there is any economical benefit by merging the networks.

We will call `napL` the scheme where the present architecture is maintained or, and `napH` the scheme where the IP/MPLS Network is extended to the international points of traffic exchange, performing both functions: gathering and routing of traffic.

7.2 Problem Data

This section describes the data used as an input to the problem. All the data needed was provided by ANTEL:

- The Transport Network G_T containing the nodes, edges and their lengths.
- The available capacities $\hat{B}$ and their cost per kilometer $T : \hat{B} \to \mathbb{R}_0^+$.
- The Data Network has been mostly established by ANTEL and modified to be in accordance with some abstractions of the model and scenarios.

The last piece of data that is needed is the function z_Q. This was derived by some members of the project team and is thoroughly explained in [29].

In the following the data is presented. The topology of the Transport Network used as an input to solve the problem is provided in Fig-7.1.

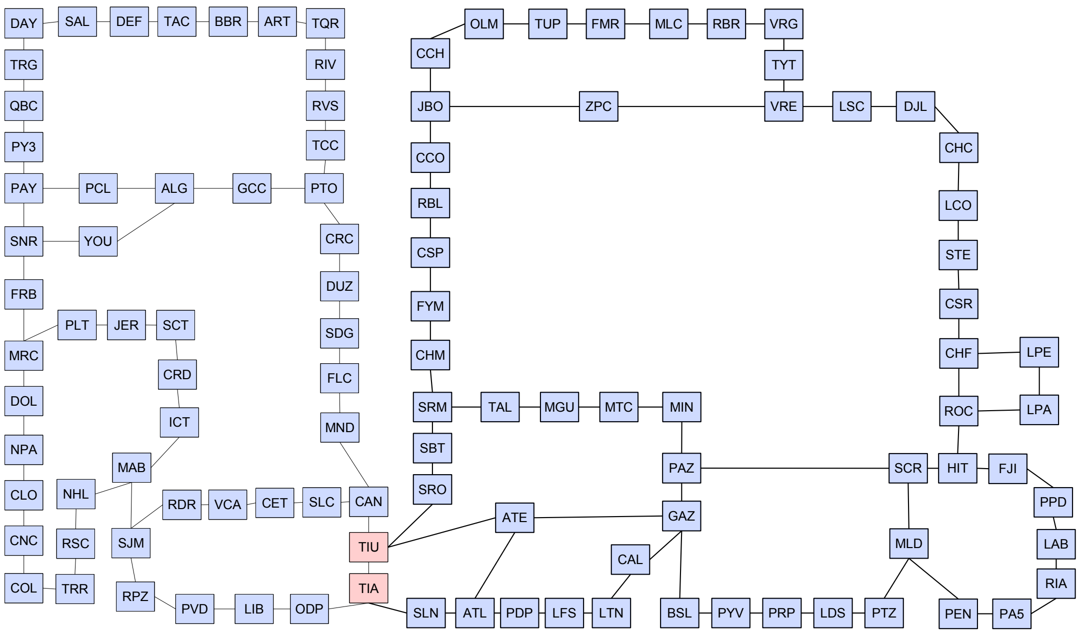


Figure 7.1: ANTEL Transport Network - Uruguay

Regarding the Data Network, ANTEL established access nodes in almost every Network Station where ADSL service is offered over ATM structure. With respect to the the edge nodes, different situations were handled. Fixed nodes were added to represent AMM and Internet, which is the node where international traffic ends. All the other nodes are optional. These nodes include departamental capitals, important cities, nodes whose station has degree three or more in the Transport Network.

In Table-7.1 the set $\hat{B}$ of available speeds and their costs are shown. In the algorithm executions b_8 and b_9 have been omitted due to the inconsistency in the assumption of a $T : \hat{B} \to \mathbb{R}_0^+$ strictly increasing function.

B̂	**Speed** (Mbps)	**VCAT/DWDM** Mapping	**Useful Capacity** (Mbps)	**E1 Equivalent** (E1)	**Cost** (US$/km)
b_0	0	-	0	0	0.00
b_1	10	5 VC12	9.5	5	3.46
b_2	20	10 VC12	19	10	6.92
b_3	40	20 VC12	38	20	13.84
b_4	50	1 VC3	42	21	14.54
b_5	100	2 VC3	84	42	29.10
b_6	140	1 VC4	132	63	43.60
b_7	280	2 VC4	264	128	88.60
b_8	560	4 VC4	528	256	177.20
b_9	1,000	7 VC4	924	441	305.25
b_{10}	10,000	1 λ	10,000	5,263	104.00

Table 7.1: Transport Network Capacities and Costs.

Lastly, in table Table-7.2 we present the statistical multiplexing factor function that we called z_Q.

The `demL` and `demH` scenarios have the lowest demand projections of 1,084,860 Mbps sold and the highest demand projections of 1,896,300 Mbps sold respectively. These values are explained also in [29] when the analysis of the z_Q function.

We have already explained `intL` and `intH`, and also `napL` and `napH` scenarios. Finally, combining `dem`, `zQ`, `int`, and `nap` with values low or high in each, 16 different test cases were constructed.

In Fig-7.2 and Fig-7.3 we illustrate the topology of the Transport Network, the data nodes and the potential data edges to include in the solution for test cases 01 and 02 respectively.

7.3 Second Group of Tests Cases

The first group of test cases are based on ANTEL data. However, they are extremely big to execute and this is the reason of the existence of this second group of test cases that will be explained in this section.

BW (Mbps)	**zQL** (Mbps)	**BW** (Mbps)	**zQH** (Mbps)
0	0	0	0
222	9.5	75	9.5
608	19.0	249	19.0
1547	41.2	1547	83.4
3095	76.1	1580	84.0
3450	84.0	2521	127.5
5390	127.5	3095	153.2
8670	200.0	4142	200.0
22900	510.0	11197	510.0
45588	1000.0	22587	1000.0
110792	2400.0	55548	2400.0
1161000	24777.0	1161000	48560.3
1896300	40411.9	1896300	79159.3

Table 7.2: z_Q function for zQL and zQH scenarios.

Although this set of test cases is new, it is based on the previous group. What is meant by this is that the Transport Network of ANTEL was divided in two regions: the East Region and the West Region. Thus the ring topology of the network is still maintained generating two smaller networks. This is illustrated in Fig-7.4 and as can be seen the nodes TIU and TIA appear on both networks. The data nodes corresponding to each region were preserved along with the induced data links.

These regions will be distinguish by the names `east_copy` and `west_copy`.

Besides, the traffic demands that were maintained were the ones between nodes from the east part in the `east_copy` case and the ones between west nodes in the `west_copy`. All the demands between east and west nodes were removed from both regions.

Regarding the other data some modifications were introduced. For instance, different bandwidths were created that go from 1000 Mbps to 10000 Mbps with a step of 1000 Mbps and their costs are proportional to the bandwidth. This is shown in table Table-7.3. The reason for introducing this variation was to test the performance of the metaheuristics [8, 10]. That is why the steps between available capacities were the same everytime. With respect to the upper capacity, the value of 10000 Mbps is the maximum capacity available for the first group of scenarios described, so the value was preserved. In this thesis we are using the values in order to compare the results with the ones obtained by the metaheuristics. The tests cases that introduce this change in possible bandwidths are recognized by the word `cap` in their name.

Moreover, there are new test cases where the Data Network was completely modified and

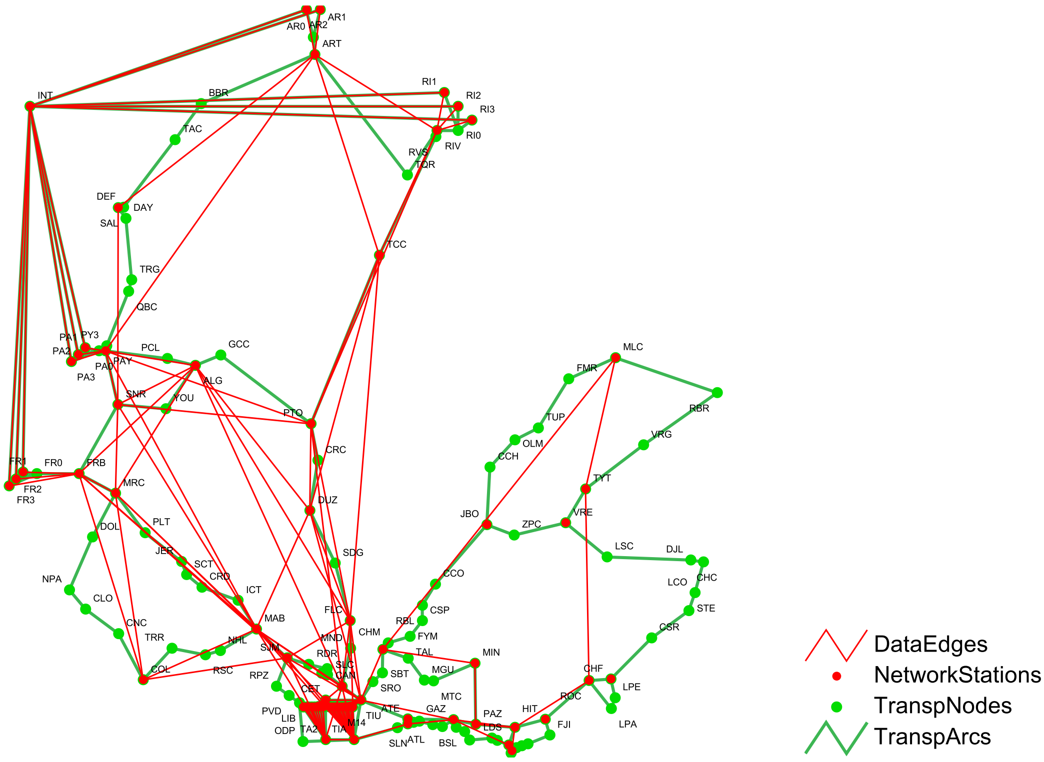


Figure 7.2: ANTEL Test Case 01.

they consist of a full-mesh topology or "half-full-mesh" topology. The test cases that include this modification are identified by `fm` or `hfm` respectively in their names. In the latter case, half of the totality of the data links from the full-mesh topology were removed.

Lastly, for some of the new test cases, the traffic demand was changed. For these cases, the traffic weight of the network was increased. In order to generate these traffic demands, for some couples of nodes their demands were chosen randomly and uniformly distributed from 0 to 30000 Mbps. The number of nodes with traffic demand is half the number of data edges. For those test cases their name will be added the word `charge`, giving the notion that in this networks almost all edges will be designed and charged, if feasible, near to its maximum capacity.

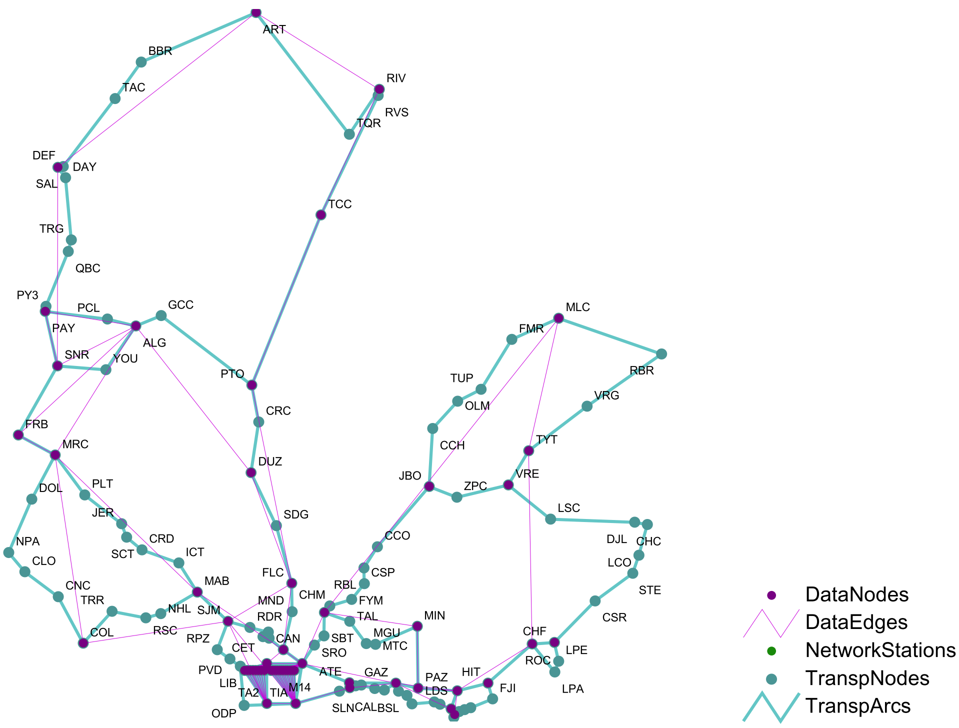
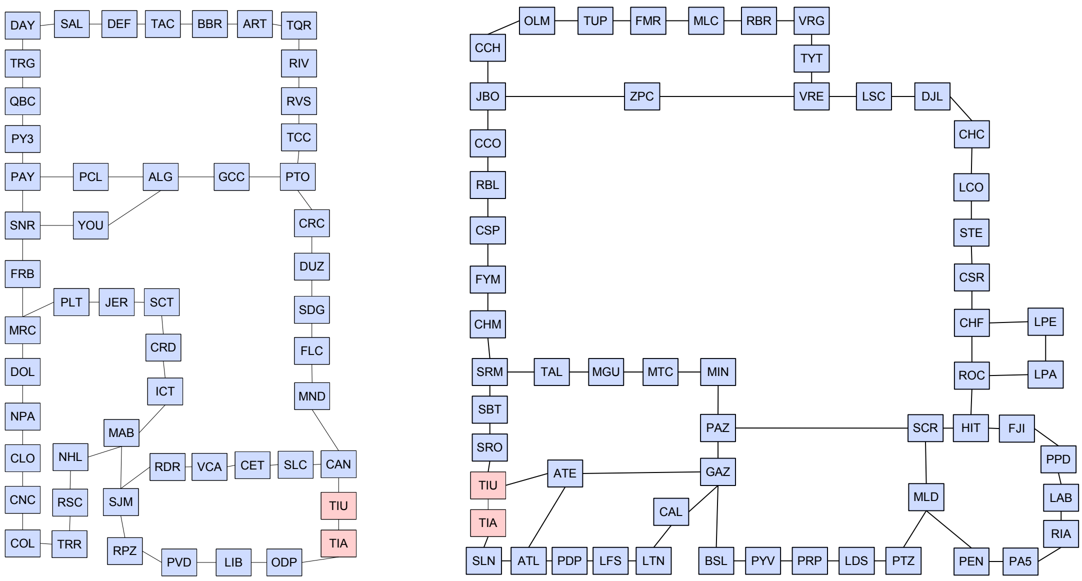


Figure 7.3: ANTEL Test Case 02.

$\hat{\text{B}}$	**Speed** (Mbps)	**Cost** (US$/km)
b_0	0	0
b_1	1000	10
b_2	2000	20
b_3	3000	30
b_4	4000	40
b_5	5000	50
b_6	6000	60
b_7	7000	70
b_8	8000	80
b_9	9000	90
b_{10}	10000	100

Table 7.3: Modified Capacities and Costs.



(a) West Region. (b) East Region.

Figure 7.4: Division of ANTEL Transport Network.

Chapter 8

Implementation and Results of the tests

This chapter intends to briefly describe the implementation of the models and to detail the results obtained after the execution of the tests.

8.1 Design and Implementation

In this section we give a brief explanation of the implementation made.

8.1.1 Domain Model

As a first step prior to the implementation, an analysis and design of the problem was made leading to a conceptual model. This model was created in order to identify the objects involved in the problem as well as the relationships between them. We designed the model shown in Fig-8.1.

The object called *serializer* has as a purpose to read the data of the problem instance and create all the necessary structures to proceed with the resolution of the problem.

8.1.2 Implementation

A first point to highlight is the programming language used for the implementation. This is an important aspect because all the basic structures were implemented by the majority of the members of the project group and it needed to be an efficient implementation as it was used by all the metaheuristics. One has to consider that large amounts of data are being manipulated and they should be managed as good as it is possible in order to take the best profit out of the algorithms.

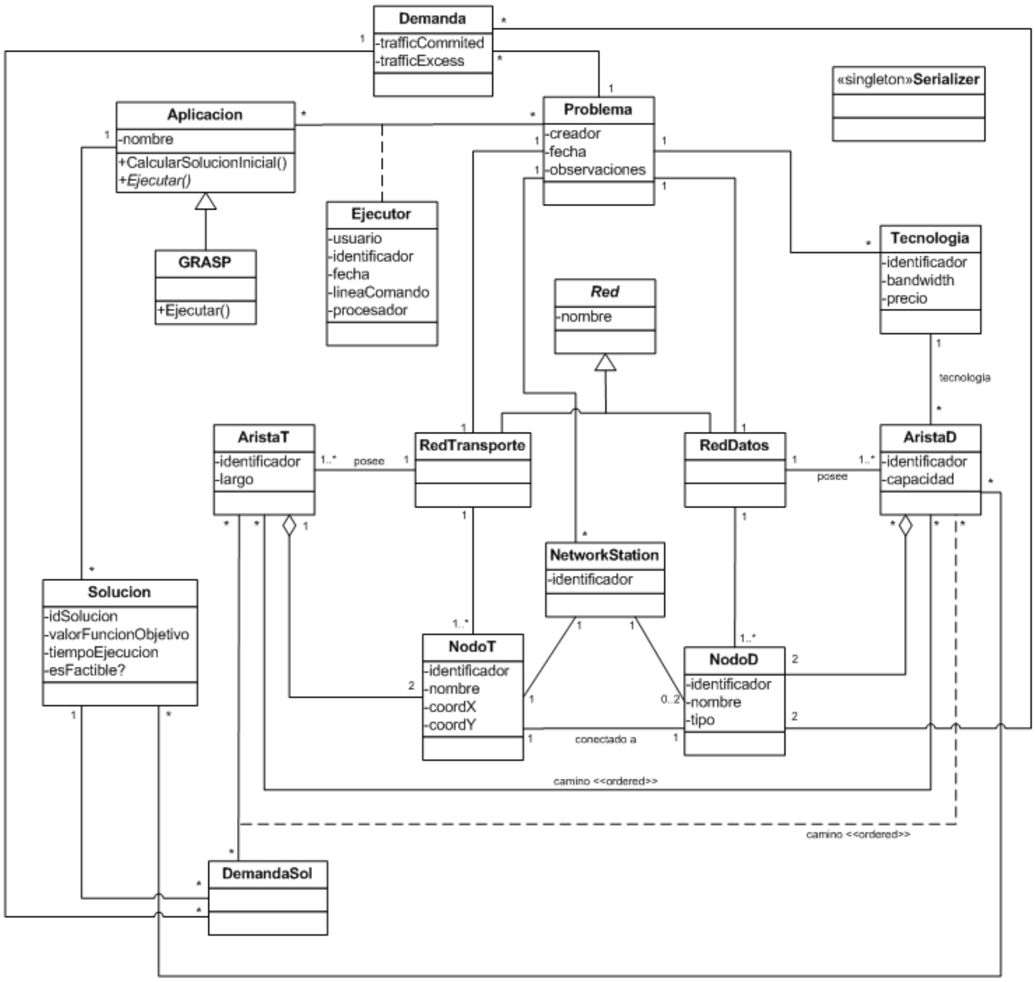


Figure 8.1: Conceptual Model.

The choice made was to use the object oriented programming language $C{+}{+}$ to implement all the basic structures and the algorithms. This thesis in particular used the $ILOG\ CPLEX$ [16] library to optimize the mathematical models.

As a way of maintaining the current and historical versions of the files corresponding to source code and documents in general, we used the software versioning system called Subversion often abbreviated as *SVN*.

For the implementation, an object oriented programming was carried out, where a class was created for every concept that appeared while formalizing the problem: data network, transport network, data edges and data nodes, transport edges and transport nodes, network stations, solutions and so on. As we stated before, the class corresponding to the *Serializer* reads all the data of the problem and is responsible of creating all the structures. In order to do this, a common format for saving the relevant data was needed and an example of this is shown in Appendix-A.

With that way of saving the data of each problem instance, an ordered lecture of it can be made.

With respect to the particular implementation of this thesis and not the basic structures, a study of CPLEX library had to be done.

Lets start by commenting some key aspects of the implementation, coded with CPLEX library, made for the complete model 5.2.1. The piece of code corresponding to the creation of the objective function is:

```
IloModel model(env);
IloExpr obj(env);
for(i = 0; i < nbEdgesT; i++)
    for(j = 0; j < nbEdgesD; j++)
        for(k = 1; k < nbCapacities; k++)
            obj += r[i] * T[k] * eta[j][k][i];
model.add(IloMinimize(env, obj));
obj.end();
```

In it, *nbEdgesT*, *nbEdgesD* and *nbCapacities* refer to the number of transport edges, data edges and available capacities respectively. Besides, *r*, *T* and *eta* are self explanatory if we go back to the mathematical model.

Regarding the constraints, they have to be added to the model. The following piece of code serves as an example and is a pseudocode for adding constraints (5.20) to (5.23).

```
for(i = 0; i < nbEdgesD; i++)
{
    for(h = 0; h < nbCapacities; h++)
    {
        for(j = 0; j < nbEdgesT; j++)
        {
            model.add(y[i][j] + w[i][h] - 1 <= eta[i][h][j]);
            model.add(eta[i][h][j] <= y[i][j]);
            model.add(eta[i][h][j] <= w[i][h]);
        }
    }
    model.add(IloSum(w[i]) == 1);
}
```

After adding all the constraints to the model, along with the objective function, the instruction to solve the model is:

```
IloCplex cplex(model);
cplex.solve();
```

Overall, the logic of the algorithm is not very complex and it does not get much more difficult when implementing the decomposition approach of chapter (6). This is because, as can be seen in the code shown, the translation from the mathematical programming model to the programming language is almost direct.

8.2 Results of the tests

In this section we detail the results obtained. It is worth mentioning that they were obtained from the following two GNU/Linux servers:

- Intel Core i7-975, 16GiB (3.33GHz, 8MiB Cache, 6.40GT/s).
- Intel Core 2 Quad Q9550, 4GiB (2.83GHz, 12MiB Cache, 1333MHz FSB).

Another remark is that the results for the complete version were obtained with the second model, presented in section 5.2. The model that used a flow implementation (5.3) to find the paths in the Transport Network was implemented but could not be tested for time reasons. So it is presented in a theoretical way.

8.2.1 Results of the first group of tests

Considering the NP-hardness of the problem, these tests proved to be very complex and could not be optimized using either the exact tool or the decomposition approach.

In order to find a solution to the problem a reduced version of the tests was carried out, that is shown in Fig-8.2. The orange nodes are those that were kept and correspond either to the most important stations or to transport nodes with degree three or more. The blue nodes were removed from the network but their distances were aggregated in order to keep the real distance between orange nodes.

With this reduction the exact version for test case 2 of ANTEL went from 309786 binary variables to 150407 binary variables. However, it was running for around 36 days and it also was not able to find any solution. With the decomposition approach a solution was found after a month but the cost was a 27.19% higher than the one found by the GRASP algorithm presented in [29].

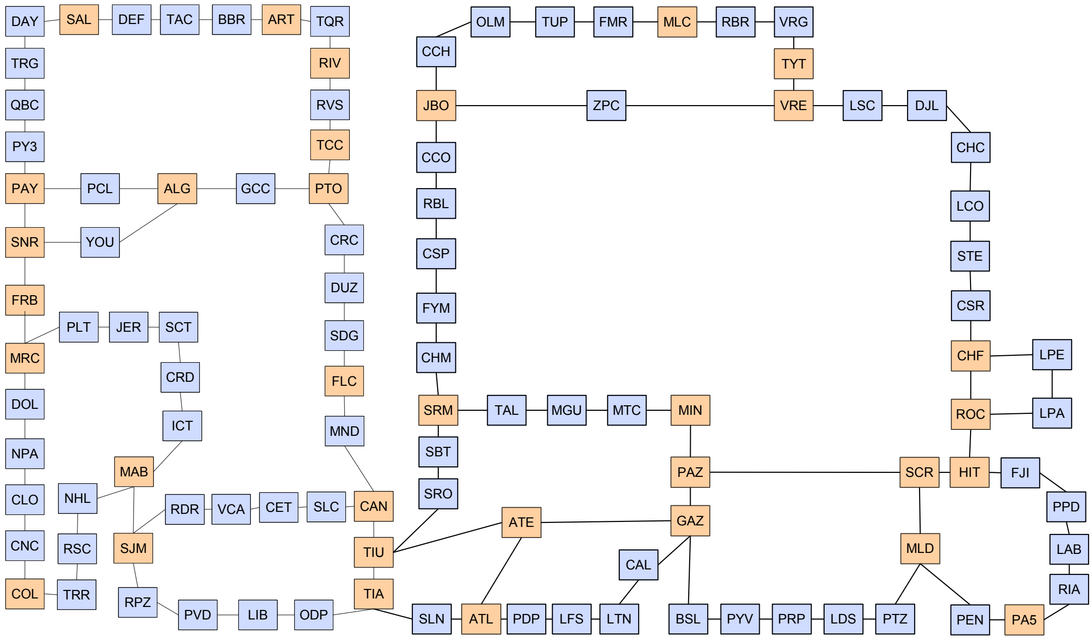


Figure 8.2: Reduced Transport Network

This group of tests is the one formulated for ANTEL. In [29] there is an exhaustive analysis of each of the tests and it is presented from an economical point of view, studying different strategies that ANTEL could follow including their benefits.

The other test cases set is not as heavy as this one and we will be able to make more comparisons. The aim of introducing this set first is that it helps to understand the essence of the second group as they served as a starting point for them.

8.2.2 Results of the second group of tests

Before introducing the results obtained, we explicit the size and amount of binary variables and constraints of the tests made. These values are shown in tables Table-8.1 and Table-8.2. In the former table the most relevant data is explicited such as the number of data links, data nodes, traffic requirements and available capacities. In the latter table the size of each test is given by showing the number of binary variables and constraints that they have. In the case of the decomposed version, the number of variables and constraints shown are for each failure scenario.

From now on we will refer to the version that is not decomposed as *Complete Version*.

We now present the results obtained from the metaheuristics in table Table-8.3. The column *VNS* corresponds to the best cost obtained by solving the test case using VNS (Variable

Test Case	Data Links	Data Nodes	Demands	Capacities
`east_copy`	15	18	11	2
`east_copy_cap`	15	18	11	10
`east_copy_charge_cap`	15	18	7	10
`east_fm`	153	18	11	2
`east_fm_cap`	153	18	11	10
`east_fm_charge`	153	18	75	2
`east_fm_charge_cap`	153	18	75	10
`east_hfm`	79	18	11	2
`west_copy`	47	18	26	2
`west_copy_cap`	47	18	26	10
`west_copy_charge_cap`	47	18	26	10
`west_fm`	47	18	16	2

Table 8.1: Most relevant data of test cases.

Test case	Complete Version		Decomposed Version	
	Binary Variables	Constraints	Binary Variables	Constraints
`east_copy`	11895	29145	145	866
`east_copy_cap`	11850	29145	265	866
`east_copy_charge_cap`	8055	20085	255	562
`east_fm`	46053	60733	1989	7214
`east_fm_cap`			3213	7214
`east_fm_charge`			11781	47406
`east_fm_charge_cap`			13005	47406
`east_hfm`	23657	34019		
`west_copy`	29281	43168	1316	5398
`west_copy_cap`	29657	43544	1692	5398
`west_copy_charge_cap`	29657	43544	1692	5398
`west_fm`	60129	75292	2754	10354

Table 8.2: Size of test cases.

Neighborhood Search), [8], and the column T_{VNS} represents the time it took. Similarly, column *TS* corresponds to the best cost obtained applying Tabu Search [10], and the column T_{TS} represents the time it took. Besides, in those columns where it says “NF” means that their results were that the problem instance was not feasible.

The results for the second group of tests are obtained from the execution of the second model presented in 5.2. They are presented in table Table-8.4. Columns *VNS* and *TS* mean the same as in the previous table.

Now some remarks that need to be made with respect to the results. Lets start with the

Test case	**VNS**	**T$_{VNS}$ (s)**	**TS**	**T$_{TS}$ (s)**
`east_copy`	112206	5	112207	90
`east_copy_cap`	107891	7	107891	77
`east_copy_charge_cap`	111247	240	95185	140
`east_fm`	128384	110	114536	51
`east_fm_cap`	106854	136	112207	49
`east_fm_charge`	1715966	174	1662190	152
`east_fm_charge_cap`	1375590	536	1598260	468
`east_hfm`	NF	-	NF	-
`west_copy`	977649	45	977650	78
`west_copy_cap`	816818	33	940048	112
`west_copy_charge_cap`	774090	34	940048	40
`west_fm`	354282	823	460764	67

Table 8.3: Results of the metaheuristics.

Test case	**Complete Version**			**Decomp. Version**		**VNS**	**TS**
	Cost	MIP **Gap (%)**	**Time (s)**	**Cost**	**Time (s)**		
`east_copy`	89175	0	2.19	89175	1.34	112206	112207
`east_copy_cap`	15114.4	0	28.13	15114.4		107891	107891
`east_copy_charge_cap`	28702.7	0	43.03	28702	0.55	111247	95185
`east_fm`	112418	32	1051980	171811.54	852.96	128384	114536
`east_fm_cap`				26303.5	13.22	106854	112207
`east_fm_charge`				512385.28	99112.97	1715966	1662190
`east_fm_charge_cap`				463313.75	899.09	1375590	1598260
`east_hfm`	413000	69.6	90240.34			NF	NF
`west_copy`	882468	58	11371	605957.01	70471	977649	977650
`west_copy_cap`	513074	95.5	48407	188020.2	1698.74	816818	940048
`west_copy_charge_cap`	-	-	806580	277386.9	16527.52	774090	940048
`west_fm`	413508.58	42.96	501060	335044.2	1080.69	354282	460764

Table 8.4: Results for second group of test cases.

complete version.

- Test cases with prefix `east_copy` clearly improved the results obtained from the execution of the metaheuristics.

- Test case `east_fm` was running for a long time (more than 12 days) and was able to find a better solution than the metaheuristics, but it is also worth noticing that it did not optimize completely but up to a MIP gap of 32%. CPLEX MIP gap is computed as:

$$\frac{best\_integer - best\_node}{best\_integer}.$$

The term *best_integer* refers to the best integer objective value found at the moment and the term *best_node* refers to the objective of the best node remaining.

- The rest of test cases with prefix `east_fm` were not able to find an integer solution in short time due to their complexity. These test cases have 18 data nodes and 153 data links.

- Test case `east_hfm` returned a solution although the results from the metaheuristics returned that it was not feasible. As a first comment, this test case was stopped when finding the first integer solution that took already a lot of time (a little bit more than one day). As a final comment, the algorithm to test whether the solutions are feasible or not in the metaheuristics is not exhaustive and that may be the reason for their result.

- Test cases `west_copy` and `west_copy_cap` improved the results obtained from the VNS and TS and they were executed up to gaps of 58% and 95.5% respectively.

- Test case `west_copy_charge_cap` run for approximately 9 days and 8 hours and it could not find any integer solution.

- Finally, test case `west_fm` was running for almost 6 days and it improved the TS result. It is worth remarking that it run up to a gap of 42.96% so that if it continues to optimize it will also reach a better solution than the VNS.

Lets make now similar remarks for the decomposed version. If it is not explicited otherwise, for each failure scenario the optimal solution was found. In those cases were this could not be done, the MIP GAP is clarified and it is the same for every failure scenario in that test case.

- Test cases with prefix `east_copy` gave the same solution as the exact version and improved the results obtained from the execution of the metaheuristics.

- Test case `east_fm` was able to find a solution not as good as the metaheuristics, but in comparison to the exact method it executed very quickly.

- Test case `east_fm_cap` run really fast and found a solution much better than the ones found with VNS and TS.

- Test cases `east_fm_charge` and `east_fm_charge_cap` improved significantly the results obtained from the VNS and TS and they were executed up to gaps of 20% and 10% respectively in each failure scenario.

- Finally, test case `west_copy_cap` improved not only the results obtained from the metaheuristics but also the one obtained with the complete version (that was not optimal) and it is worth noticing the speed with which it found that solution.

It should be clarified that the test cases that are missing in the table or the ones that do not have any results could not be executed because of lack of time. Considering the amount of data nodes and links and different charges, these remaining cases should be comparable in time to the most complex ones, needing probably near a month to solve each. I must remark that there were only two servers available for their execution.

To close this chapter we show two tables Table-8.5 and Table-8.6 that show the percentage of improvement of the exact version over the decomposed version and of the decomposed version over the metaheuristics.

For table Table-8.5 we calculate $\left(1 - \frac{cost\_of\_exact}{cost\_of\_decomposed}\right) \cdot 100$.

Test case	**%D-E**
`east_copy`	0.0
`east_copy_cap`	0.0
`east_copy_charge_cap`	$-2.4e^{-3}$
`east_fm`	34.6
`west_copy`	-45.6
`west_copy_cap`	-172.9
`west_fm`	-23.4

Table 8.5: Improvement of the exact version over the decomposed version.

For table Table-8.6 we calculate $\left(1 - \frac{cost\_of\_decomposed}{cost\_of\_metaheuristic}\right) \cdot 100$.

Test case	**%D-VNS**	**%D-TS**
`east_copy`	20.5	20.5
`east_copy_cap`	86.0	86.0
`east_copy_charge_cap`	74.2	69.8
`east_fm`	-33.8	-50.0
`east_fm_cap`	75.3	76.6
`east_fm_charge`	70.1	69.2
`east_fm_charge_cap`	66.3	71.0
`west_copy`	38.0	38.0
`west_copy_cap`	77.0	80.0
`west_copy_charge_cap`	64.2	70.5
`west_fm`	5.4	27.3

Table 8.6: Improvement of the decomposed version over the metaheuristics.

Chapter 9

Conclusions

In this final chapter we present some conclusions that are obtained from the project. An evaluation of positive and negative aspects of the development of this work as well as the results obtained is made.

From the point of view of the development of this project, the modelling of a very complex problem such as the design of a multi-overlay robust network was carried out successfully. As a way to diminish the complexity of the problem, in a second stage a simplification to the model was made resulting in an decomposed version of the original. Although this models the same problem, the results obtained are an approximation of the original.

Also, a research of the existing literature regarding problems of the same nature was done. As a result, we obtained that to the best of our knowledge there are not existing problems that approach the one addressed here in a way that we can consider them similar. All of them include a simplification of the one solved here in one way or another.

As a final stage of the process an implementation of the models was done successfully and in parallel with the study of the CPLEX library that was crucial for the solving of the problem instances.

We can say that even though the complexity of the problem, the proposed solution meets the objectives set at the begining of this project in a satisfactory way. And this is accomplished with both the complete version or the decomposed version as the solutions that we obtain have all the information needed. With the decomposition approach not optimal solutions are obtained but very good ones.

This tool not only is useful to solve problems with the characteristics of ANTEL networks, but it also can be used with any network structure, traffic and technologies. So this can be useful outside the context of ANTEL. One constraint to obtain a feasible solution is that both networks are two-edge-connected because otherwise it cannot be resilient to simple faults.

Another use of this tool is to give the metaheuristics approach a lower bound in their costs in order to study their results.

One weak aspect of this solution is that it is very time consuming and with large instances it could take months to achieve one feasible solution (and of course not optimal). So the key question is whether this tool has any advantages over the other tools implemented in this project that used metaheuristics. In those works, solutions were obtained that are suboptimal but they executed in a speed that is not comparable to this one.

A tool with this capacity, that solves a real difficult problem can be used to improve the decisions taken by the National Telephone company. It helps in the study of different strategies and in the decision of which fits best the business economically. Therefore, it could help to justify investments or even evaluate structural problems in the network. Besides, it could be used to evaluate the decision of making an infrastructure modification such as the adding of a new station in case ANTEL wants to expand its services to other points of the country.

Going back to the question raised, I would say that this tool is more accurate and can be utilized in the study of local investments rather than the study of global ones. In the case of local investments, the parts of the network that are not relevant can be collapsed into one or few nodes (in an intelligent way that does not affect the final result) and obtain a much smaller network that can be solved in an exact way and in relatively short time.

Another aspect to analyze is whether there are plausible future extensions to this work. A possible extension is to expand the tool in order to make it user friendly. The results obtained are printed in a file as text and they are not very readable. The functions implemented were not thought to be used by an enterprise employee in this stage but they were planned for academic purpose to be handled by an expert user. However, this can be improved making a user interface and showing the solutions in a graphical way (showing maps, routes and capacities) along with a user manual that explains how the tool works. With this enhancement the tool could be extended to the planification engineers of the enterprise. This expansion could improve significantly the potential of the tool.

In order to obtain more results, besides executing the tests cases for which there are no results, another path that can be followed is the testing of the model with flow implementation. This way, more comparisons can be made.

Another extension to this work may be the study of possible relaxations to the model in order to obtain bounds to the cost of the optimal solution. This could be a good approach in the sense of comparing it with the decomposition approach and to see if it improves the results obtained.

A different extension could be to model the problem with an arc-path entity basis in opposition to the arc-node representation used, as similar as was done on the GRASP and metaheuristics approaches. Following a decomposition approach, a path may be generated when it is needed. Furthermore, the implementation could be adjusted taking advantage of the problem instances.

Appendix A

Input and Output Formats

In this appendix we show the common format defined for saving the data as well as the format used to save the solutions returned.

A.1 Input format

In the following example this format is shown.

Example.

```
# Number of Data nodes
name nDataNodes := 9;
# Data Nodes
name DataNodes (DataNodeId, DataNodeName, IsReal) :=
0 EDGENDALG 1
1 EDGENDART 1
2 EDGENDCAN 1
3 EDGENDCOL 1
4 EDGENDDUZ 1
5 EDGENDFLC 1
6 EDGENDFRB 1
7 EDGENDMAB 1
8 EDGENDMRC 1
;

# Data Edge Nodes
name DataEdgeNodes (DataNodeId) :=
0
1
2
3
4
5
6
```

```
7
8
;

# Data Access Nodes
name DataAccessNodes (DataNodeId) :=
;

# Number of Data Edges
name nDataArcs := 13;
# Data Edges
name DataEdges (DataEdgeId, OrgDataNodeId, DestDataNodeId) :=
0 0 1
1 0 3
2 1 2
3 1 4
4 2 3
5 2 5
6 3 6
7 4 5
8 4 7
9 5 6
10 5 7
11 6 8
12 7 8
;

# Number of Traffic Requests
name nTraffic := 4;
# Traffic
name Traffic (TrafficId, OrgDataNodeId, DestDataNodeId, Committed, Excess) :=
0 0 7 10 6
1 1 6 12 8
2 2 8 8 4
3 3 4 16 2
;

# Number of Transport Nodes
name nTranspNodes := 9;
# Transportation Nodes
name TranspNodes (TranspNodeId, TranspNodeName, CoordX, CoordY) :=
0 ALG 349.93 6412.4
1 ART 436.18 6637.4
2 CAN 455.85 6179.94
3 COL 312.05 6184.55
4 DUZ 432.7 6307.15
5 FLC 461.85 6227.5
6 FRB 265.53 6333.98
7 MAB 394.1 6221.4
8 MRC 292.03 6319.61
;

# Number of Transport Edges
name nTranspArcs := 13;
```

```
# Transportation Edges
name TranspArcs (TranspArcId, OrgTranspNodeId, DestTranspNodeId, Length) :=
0 0 1 10
1 0 3 20
2 1 2 12
3 1 4 15
4 2 3 18
5 2 5 14
6 3 6 17
7 4 5 8
8 4 7 16
9 5 6 13
10 5 7 11
11 6 8 6
12 7 8 9
;

# Number of Technologies
name nTechnology := 7;
# Technologies
name Techno (TechnoId, Bandwidth, Price)
0 1 100
1 2 180
2 5 450
3 10 800
4 20 1600
5 40 3000
6 60 3500
;

# Number of NetworkStations
name nNetworkStation := 9;
# NetworkStations
name NetworkStations (NetworkStationId, DataNodeId, TranspNodeId) :=
0 0 0
1 1 1
2 2 2
3 3 3
4 4 4
5 5 5
6 6 6
7 7 7
8 8 8
;

# Zeta function
name ZetaQ (BW, ZQ) :=
0 0
75 9.5
249 19.0
1547 83.4
1580 84.0
2521 127.5
3095 153.2
```

```
4142 200.0
11197 510.0
22587 1000.0
55548 2400.0
1161000 48560.3
1896300 79159.3
;
```

♠

A.2 Output format

Example.

```
Cost of the solution: 309200

# Technology assigned to each data link. The first index corresponds to the link
# and the second one to the number of the technology. Between parenthesis the
# capacity is explicited.
w[0][6]  (40)
w[1][5]  (20)
w[2][5]  (20)
w[3][6]  (40)
w[4][5]  (20)
w[5][4]  (10)
w[6][6]  (40)
w[7][5]  (20)
w[8][5]  (20)
w[9][5]  (20)
w[10][4]  (10)
w[11][5]  (20)
w[12][5]  (20)

# For each data link the path in the transport network is given. The first index
# is the data link while the second one is the transport link.
y[0][0]
y[1][1]
y[2][2]
y[3][3]
y[4][4]
y[5][5]
y[6][6]
y[7][7]
y[8][8]
y[9][9]
y[10][10]
y[11][11]
y[12][12]

# For every failure scenario the paths in the Data Network are given. The first
# index is the failure scenario, the second one is the number of traffic
# requirement and the latter one is the data link used to satisfy the requirement.
```

```
x[0][0][1]
x[0][0][6]
x[0][0][11]
x[0][0][12]

x[0][1][3]
x[0][1][7]
x[0][1][9]

x[0][2][5]
x[0][2][10]
x[0][2][12]

x[0][3][2]
x[0][3][3]
x[0][3][4]
```

```
x[1][0][0]
x[1][0][3]
x[1][0][8]

x[1][1][3]
x[1][1][7]
x[1][1][9]

x[1][2][5]
x[1][2][10]
x[1][2][12]

x[1][3][2]
x[1][3][3]
x[1][3][4]
```

```
x[2][0][0]
x[2][0][3]
x[2][0][8]

x[2][1][3]
x[2][1][7]
x[2][1][9]

x[2][2][4]
x[2][2][6]
x[2][2][11]

x[2][3][0]
x[2][3][1]
x[2][3][3]
```

```
x[3][0][1]
x[3][0][6]
x[3][0][11]
x[3][0][12]

x[3][1][2]
x[3][1][4]
x[3][1][6]

x[3][2][5]
x[3][2][10]
x[3][2][12]

x[3][3][6]
x[3][3][7]
x[3][3][9]
```

```
x[4][0][0]
x[4][0][3]
x[4][0][8]

x[4][1][3]
x[4][1][7]
x[4][1][9]

x[4][2][5]
x[4][2][10]
x[4][2][12]

x[4][3][0]
x[4][3][1]
x[4][3][3]
```

```
x[5][0][0]
x[5][0][3]
x[5][0][8]

x[5][1][3]
x[5][1][7]
x[5][1][9]

x[5][2][4]
x[5][2][6]
x[5][2][11]

x[5][3][0]
x[5][3][1]
x[5][3][3]
```

```
x[6][0][0]
x[6][0][3]
x[6][0][8]

x[6][1][3]
x[6][1][7]
x[6][1][9]

x[6][2][5]
x[6][2][10]
x[6][2][12]

x[6][3][0]
x[6][3][1]
x[6][3][3]
```

```
x[7][0][0]
x[7][0][3]
x[7][0][8]

x[7][1][2]
x[7][1][4]
x[7][1][6]

x[7][2][5]
x[7][2][10]
x[7][2][12]

x[7][3][0]
x[7][3][1]
x[7][3][3]
```

```
x[8][0][0]
x[8][0][2]
x[8][0][4]
x[8][0][6]
x[8][0][11]
x[8][0][12]

x[8][1][3]
x[8][1][7]
x[8][1][9]

x[8][2][4]
x[8][2][6]
x[8][2][11]

x[8][3][0]
x[8][3][1]
x[8][3][3]
```

```
x[9][0][0]
x[9][0][3]
x[9][0][8]

x[9][1][2]
x[9][1][4]
x[9][1][6]

x[9][2][5]
x[9][2][10]
x[9][2][12]

x[9][3][0]
x[9][3][1]
x[9][3][3]
```

```
x[10][0][0]
x[10][0][3]
x[10][0][8]

x[10][1][3]
x[10][1][7]
x[10][1][9]

x[10][2][4]
x[10][2][6]
x[10][2][11]

x[10][3][0]
x[10][3][1]
x[10][3][3]
```

```
x[11][0][0]
x[11][0][3]
x[11][0][8]

x[11][1][0]
x[11][1][1]
x[11][1][6]

x[11][2][5]
x[11][2][7]
x[11][2][8]
x[11][2][12]

x[11][3][2]
x[11][3][3]
x[11][3][4]
```

```
x[12][0][0]
x[12][0][3]
x[12][0][8]

x[12][1][3]
x[12][1][7]
x[12][1][9]

x[12][2][4]
x[12][2][6]
x[12][2][11]

x[12][3][0]
x[12][3][1]
x[12][3][3]
```

```
x[13][0][0]
x[13][0][3]
x[13][0][8]

x[13][1][2]
x[13][1][4]
x[13][1][6]

x[13][2][5]
x[13][2][7]
x[13][2][8]
x[13][2][12]

x[13][3][0]
x[13][3][1]
x[13][3][3]
```

♠

Bibliography

[1] D. Alevras, M. Grötschel, and R. Wessäly. A network dimensioning tool. In *Preprint SC 96-49, Konrad-Zuse-Zentrum fur Informationstechnik*, 1996.

[2] Y. Azar and O. Regev. Combinatorial algorithms for the unsplittable flow problem. *Algorithmica*, 44:49–66, 2006.

[3] A. Balakrishnan, T. Magnanti, and P. Mirchandani. A dual-based algorithm for multi-level network design. *Management Science*, 40:567–581, 1994.

[4] A. Balakrishnan, T. Magnanti, and P. Mirchandani. Modeling and heuristic worst-case performance analysis of the two-level network design problem. *Management Science*, 40:846–867, 1994.

[5] A. Balakrishnan, T. Magnanti, and P. Mirchandani. Designing hierarchical survivable networks. *Operations Research*, 46:116–136, 1998.

[6] A. Balakrishnan, T. Magnanti, and P. Mirchandani. Connectivity-splitting models for survivable network design. *Networks*, 43:10–27, 2004.

[7] D. Bienstock, S. Chopra, O. Günlük, and C. Tsai. Minimum cost capacity installation for multicommodity network flows. *MATHEMATICAL PROGRAMMING*, 81:177–199, 1998.

[8] A. Corez. Multi-overlay network planning by applying a variable neighbourhood search approach. Master's thesis, Facultad de Ingeniería, UdelaR, http://premat.fing.edu.uy/IngenieriaMatematica/archivos/tesis_andres_corez.pdf, 2010.

[9] G. Dahl, A. Martin, and M. Stoer. Routing through virtual paths in layered telecommunication networks. *Operations Research*, 47(5):pp. 693–702, 1999.

[10] F. Despaux. Optimización de una red de datos ip/mpls sobre sdh/dwdm usando tabú search. caso de estudio: Red de datos de un operador de telefonía nacional. Master's thesis, Facultad de Ingeniería, UdelaR, http://premat.fing.edu.uy/IngenieriaMatematica/archivos/tesis_francois_despaux.pdf, 2011.

[11] B. Fortz and M. Poss. An improved benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, 37(5):359 – 364, 2009.

[12] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[13] A. Gersht and A. Shulman. A new algorithm for the solution of the minimum cost multicommodity flow problem. In *Decision and Control, 1987. 26th IEEE Conference on*, volume 26, pages 748 –758, 1987.

[14] M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. In *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, SODA '92, pages 307–316, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics.

[15] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.

[16] IBM. *IBM ILOG CPLEX V12.1: User's Manual for CPLEX*. International Business Machines Corp., Armonk, New York, USA, 2009.

[17] R. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[18] H. Kerivin and A. Mahjoub. Design of survivable networks: A survey. In *In Networks*, pages 1–21, 2005.

[19] H. Kerivin, D. Nace, and T. Pham. Design of capacitated survivable networks with a single facility. *Networking, IEEE/ACM Transactions on*, 13(2):248 – 261, 2005.

[20] J. Kurose and K. Ross. *Computer Networking, A Top-Down Approach*. Addison-Wesley, http://www.aw.com/kurose_ross, fifth edition, 2010.

[21] J. Leung, T. Magnanti, and V. Singhal. Routing in point-to-point delivery systems. Technical report, Massachusetts Institute of Technology, Operations Research Center, 1988-01.

[22] C. Li, S. McCormick, and D. Simchi-Levi. The point-to-point delivery and connection problems: complexity and algorithms. *Discrete Applied Mathematics*, 36(3):267 – 292, 1992.

[23] T. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the capacitated network loading problem. *Operations Research Center Working Paper*, 239(91), 1991.

[24] T. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the two-facility capacitated network loading problem. *Operations Research*, 43(1):pp. 142–157, 1995.

[25] S. Orlowski, A. Koster, C. Raack, and R. Wessäly. Two-layer network design by branch-and-cut featuring mip-based heuristics. In *Proceedings of the Third International Network Optimization Conference (INOC 2007), Spa, Belgium*, 2007.

[26] C. Raack, A. Koster, S. Orlowski, and R. Wessäly. On cut-based inequalities for capacitated network design polyhedra. *Networks*, 2010.

[27] S. Ramamurthy and B. Mukherjee. Survivable wdm mesh networks. ii. restoration. In *Communications, 1999. ICC '99. 1999 IEEE International Conference on*, volume 3, pages 2023 –2030 vol.3, 1999.

[28] S. Ramamurthy and B. Mukherjee. Survivable wdm mesh networks. part i-protection. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 744 –751 vol.2, 1999.

[29] C. Risso. Optimización de costos en redes multicapa robustas. Master's thesis, Facultad de Ingeniería, UdelaR, http://premat.fing.edu.uy/IngenieriaMatematica/archivos/tesis_claudio_risso.pdf, 2010.

[30] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. RFC 3031, Internet Engineering Task Force, www.ietf.org/rfc/rfc3031.txt, 2001.

[31] R. Sabella, P. Iovanna, G. Oriolo, and L. Sanità. Fine protection of data-paths in multilayer networks based on the gmpls paradigm. *Optical Switching and Networking*, 5(2-3):159 – 169, 2008. Advances in IP-Optical Networking for IP Quad-play Traffic and Services.

[32] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee. Fault management in ip-over-wdm networks: Wdm protection versus ip restoration. *Selected Areas in Communications, IEEE Journal on*, 20(1):21 –33, 2002.

[33] M. Skutella. Approximating the single source unsplittable min-cost flow problem. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 136 –145, 2000.

[34] M. Stoer. *Design Of Survivable Networks (lecture Notes In Mathematics, 1531)*. Springer-verlag, first edition, 1993.

[35] K. Sunggy, G. Sahin, and S. Subramaniam. Cost efficient lsp protection in ip/mpls-over-wdm overlay networks. In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 2, pages 1278 – 1282 vol.2, 2003.

[36] X. Wang, L. Guo, F. Yang, T. Wu, and W. Ji. Multi-layer survivable routing mechanism in gmpls based optical networks. *Journal of Systems and Software*, 81(11):2014 – 2023, 2008.

[37] R. Wong. A survey of network design problems. *Operations Research Center Working Paper*, 080(78), 1978.

[38] M. Zheng, C. Jiang, R. Yang, and K. Arvind. Optimal capacity sharing of networks with multiple overlays. In *IWQoS*, pages 72–81, 2006.

List of figures