# Optimisation of passive optical network design under demand uncertainty

**SP van Loggerenberg**

**20289278**

Thesis submitted for the degree *Philosophiae Doctor* in Computer and Electronic Engineering at the Potchefstroom Campus of the North-West University

Promoter: Dr M Ferreira

Co-promoter: Prof SE Terblanche

Assistant-promoter: Dr MJ Grobler

October 2016

# Optimisation of passive optical network design under demand uncertainty

---

**Thesis submitted in fulfilment of the requirements for the degree *Philosophiae Doctor* in Computer and Electronic Engineering at the Potchefstroom campus of the North-West University**

---


## S.P. van Loggerenberg

B.Sc Business Mathematics and Informatics / B.Eng Computer and Electronic Engineering / M.Eng Computer and Electronic Engineering

**20289278**


**Promoter:** **Dr. M. Ferreira**

**Co-promoters:** **Prof. S.E. Terblanche**

**Dr. M.J. Grobler**

**November 2015**

# Acknowledgements

First and foremost, I would like to thank my promoters Dr. Melvin Ferreira, Prof. Fanie Terblanche and Dr. Leenta Grobler for their support, guidance and time invested throughout this study. In particular, I sincerely appreciate the enthusiastic problem-solving discussions we had; it kept me going even in uncertain times.

Next, I would like to extend my gratitude to the following people and institutions for their critical support:

- Gys Booysen, Telkom SA SOC Ltd.
- Prof. Dr. Andreas Bley, University of Kassel, Germany
- atesio GmbH, Berlin, Germany
- Telkom Centre of Excellence, Telkom SA SOC Ltd.
- TeleNet research group, North-West University

To my family, Miemie and Cecile van Loggerenberg: thank you for your unfaltering belief in me and your support and encouragement during trying times.

My friends, Casper Coertze, Jean du Toit, Arno Meiring, Hansie & Christelle Swanepoel and Heinrich van Nieuwenhuizen, thank you for your friendship, support and motivation. In particular, Jean du Toit, for enduring and alleviating the brunt of my complaints and frustration.

*In memory of my father, Wim*

# Abstract

The Passive Optical Network (PON) is a point-to-multipoint, optical fibre telecommunication network used at the access level, in which a signal is distributed via a single fibre from the Central Office (CO) to a number of downstream Optical Network Units (ONUs) at customer premises. In addition to sharing a single fibre between a number of customers, these networks use passive components in the field, providing future-proof networks with no electricity requirements. All these benefits, together with high bandwidth potential, makes PONs and, in particular, the ITU-T G.984 Gigabit Passive Optical Network (GPON), the access network of choice for service providers.

Traditionally planned by hand, advanced methods have been developed to design PON deployments, including heuristics, meta-heuristics and exact mathematical models. Unfortunately, heuristic methods provide sub-optimal solutions, which, due to high deployment costs in general, result in high and unnecessary overhead. Conversely, exact mathematical models of the Passive Optical Network Design Problem (PONDP) can give optimal, minimum cost solutions, but are very demanding in terms of computational effort, limiting the size of networks that can be solved in an acceptable time period. Furthermore, since PONs are mostly deployed in a greenfield setting, customer demand is uncertain, complicating the design of an accurate model even more.

This thesis addresses two concerns in the exact mathematical modelling framework: model accuracy and computational tractability. To improve computational performance, a row- and column generation approach based on Benders decomposition is provided, strengthened by additional cut separation algorithms. This approach is found to be much more scalable and flexible than the classical arc flow approach when accounting for physical network constraints inherent in the PON specifications, due to the efficient handling of path length constraints. Furthermore, the framework presented contributes towards general hierarchical network connectivity problems with

path length constraints, which have not been studied extensively in literature, and its flexibility is demonstrated by means of a number of model refinements, including the addition of different splitter types, edge-disjoint survivability between the CO and splitters, and homo- and heterogeneous multi-level networks. To address demand uncertainty, two distinct approaches are followed, resulting in a two-stage recourse and a robust formulation. These both serve to lower cost through optical fibre and splitter dimensioning while ensuring a minimum level of connectivity. A revenue-based model is formulated in conjunction with the stochastic formulations to illustrate the impact of directly maximising return on investment.

Finally, the methods are verified and validated using cross-model verification, an external feasibility checker and face validation, before ensuring all network parameters conform to the G.984 specification, resulting in a practically feasible network design.

***Keywords:*** *Benders decomposition, Column generation, Integer Linear Program (ILP), Passive Optical Networks (PONs), Robust optimisation, Stochastic programming*

# Contents

# List of Figures

# List of Tables

# List of Acronyms

**AE** Active Ethernet

**AES** Advanced Encryption Standard

**ADSL** Asymmetric Digital Subscriber Line

**AON** Active Optical Network

**ARC** Adjustable Robust Counterpart

**ATM** Asynchronous Transfer Mode

**BCA** Branch Contracting Algorithm

**CAPEX** Capital Expenditure

**CIL** Channel Insertion Loss

**CLP** COIN-OR LP

**CO** Central Office

**ConFL** Connected Facility Location Problem

**DAG** Directed Acyclic Graph

**DARPA** Defense Advanced Research Projects Agency

**EA** Evolutionary Algorithm

**EC** Evolutionary Computation

**EFM** Ethernet in the First Mile

**EPON** Ethernet Passive Optical Network

**FBT** Fused Biconical Taper

**FEC** Forward Error Correction

**FFT** Fast Fourier Transform

**FSAN** Full Service Access Network

**FTTB** Fibre to the Building

**FTTC** Fibre to the Curb

**FTTH** Fibre to the Home

**FTTN** Fibre to the Node

**FTTx** Fibre to the x

**GA** Genetic Algorithm

**GEM** GPON Encapsulation Method

**GIS** Geographic Information System

**GMI** Gomory Mixed-Integer

**GPON** Gigabit Passive Optical Network

**GTC** GPON Transmission Convergence

**IEEE** Institute of Electrical and Electronics Engineers

**ILP** Integer Linear Program

**IP** Internet Protocol

**IPG** Interpacket Gap

**ISO** International Organisation for Standardisation

**ITU-T** International Telecommunication Union - Telecommunication Standardisation Sector

**LLID** Logical Link ID

**LP** Linear Program

**LR-PON** Long-Reach Passive Optical Network

**LWPF** Low Water Peak Fibre

**MILP** Mixed Integer Linear Program

**MIR** Mixed Integer Rounding

**MST** Minimum Spanning Tree

**NLP** Non-linear Program

**OLT** Optical Line Terminal

**ONT** Optical Network Terminal

**ONU** Optical Network Unit

**OSI** Open Systems Interconnection

**P2MP** Point-to-Multipoint

**P2P** Point-to-Point

**PLC** Planar Lightwave Circuit

**PMD** Physical Medium Dependent

**POF** Plastic Optical Fibre

**POI** Point of Interest

**PON** Passive Optical Network

**PONDP** Passive Optical Network Design Problem

**QCP** Quadratically Constrained Program

**QoS** Quality of Service

**QP** Quadratic Program

**RC** Robust Counterpart

**RINS** Relaxation Induced Neighbourhood Search

**SA** Simulated Annealing

**SDH** Synchronous Digital Hierarchy

**SLA** Service Level Agreement

**SREG** Shared Risk Equipment Group

**SRLG** Shared Risk Link Group

**TC** Transmission Convergence

**TDM** Time Division Multiplexing

**TDM-PON** Time Division Multiplexing PON

**TDMA** Time Division Multiple Access

**VDSL** Very-high-bit-rate Digital Subscriber Line

**VLSI** Very-large-scale Integration

**VoIP** Voice-over-IP

**WDM-PON** Wavelength Division Multiplexing PON

# Chapter 1

# Introduction

---

*In this chapter, an introduction is given, starting with a brief contextualisation of the research conducted to provide a theoretical and practical motivation. This is followed by the research goal and contributions, with reference to the literature. Finally, the research methodology, containing both modelling and validation and verification processes, is detailed before concluding with a chapter orientation.*

---

## 1.1 Contextualisation

Since the advent of the internet, telecommunication networks have been steadily improving, pushing for ever increasing bandwidth and connectivity. Recently, this has increased at an exponential rate due to rising popularity of high-bandwidth services such as video streaming. In particular, according to [5], global bandwidth demand has increased by 44 % during 2014 to more than 211 Tbps, up from 39 % in the previous year [6]. Additionally, bandwidth demand in Africa is expected to exceed this figure, with a 51 % growth expected annually up to 2019 [7]. It is also expected that up to 80 % of all internet traffic in 2019 [8] will consist of Internet Protocol (IP) video. These fig-

ures are quite daunting for telecommunication service providers, as infrastructure has to be expanded and upgraded to keep up with the insatiable demand for broadband internet. Furthermore, since new infrastructure is expensive to install, it has to be able to handle the bandwidth demand of the future, driving service providers to invest in solutions that can provide bandwidth far exceeding the current demand.

Therefore, service providers are investing into optical networks, which provide the highest sheer bandwidth potential of any network technology. Since bandwidth demand per consumer has increased to a point where legacy technologies can not possibly compete, many are opting to install optical fibre right up to the customer premises at the edge of the network. This penetration of fibre from the core to the access network is known as either Fibre to the Home (FTTH) or Fibre to the Building (FTTB), depending on the customer demarcation point. While core optical fibre networks have been studied extensively in literature since the inception of optical Synchronous Digital Hierarchy (SDH) networks in the 1980s, it is only recently that research has focussed on access level optical fibre networks, with standards being ratified as late as 2004.

Deploying individual optical fibres to each customer, as is the case in FTTH and FTTB, can be expensive, which is why a number of Point-to-Multipoint (P2MP) technologies have become popular. These networks aggregate data traffic of a number of customers on a single optical fibre, reducing the overall deployment cost by up to 50 % [9]. As different P2MP networks have emerged, the Passive Optical Network (PON) has become one of the primary contenders, using passive optical splitters to do the aggregation. Since these splitters are simple, robust and do not require any power, the resulting network is less expensive, greener and more future proof.

In terms of physical construction, a PON consists of an Optical Line Terminal (OLT), located at the Central Office (CO), connected to a number of splitters via a single optical fibre each. The splitter splits the incoming optical signal into a number of identical signals, one for each of its output ports, which in turn are transmitted via optical fibre to Optical Network Units (ONUs) at the customer premises. This results in a tree topology, where a single optical fibre from the CO can serve tens or even hundreds of

ONUs, reducing the overall network deployment cost [9].

### 1.1.1 Mathematical PON design

When designing a PON, we are presented with a large number of parameters, all of which have to be incorporated to produce a viable network. These include topological inputs, e.g. where optical fibres may be installed, the locations of the central office, customer premises and potential splitter sites, as well as a number of technology constraints, including attenuation considerations, equipment capabilities and the physical topology. Topological inputs are usually provided by the service provider while the technology constraints are standardised, ratified in either the Institute of Electrical and Electronics Engineers (IEEE) 802.3ah Ethernet Passive Optical Network (EPON) standard [10] or the International Telecommunication Union - Telecommunication Standardisation Sector (ITU-T) G.984 Gigabit Passive Optical Network (GPON) series of recommendations [11–14]. Only by combining all these factors can a network design be produced that will satisfy the service provider requirements.

One of the most important aspects to consider when designing a PON is attenuation, as the introduction of passive splitters results in a substantial reduction in signal strength. The sum of all optical fibre, connector, splicing and splitter losses may not exceed a given power budget, i.e. the difference between transmission power and receiver sensitivity, as this could result in a non-functioning network. Although attenuation is generally dismissed by practitioners in the literature (as we will see in chapter 3), often due to the difficulty of integration, it will be covered in great detail in this thesis.

Once we have all the required parameters, we can proceed in a number of ways. First, we could design a network deployment manually using an iterative process to refine a *best guess* solution. While this may yield a decent design, it will likely be more expensive than required as the best design may not be immediately obvious. Additionally, due to the enormous number of possible configurations, the probability of choosing the best one is slim for anything but the smallest of networks. For large networks, this

process may also entail a significant amount of time and effort. Secondly, we could employ a heuristic algorithm to essentially automate this process, following a number of rules or generating bounds in such a way as to quickly find a design that satisfies the requirements. This algorithm can be very fast, taking only a few minutes to find a good solution, but as with the manual approach there is no guarantee of the resulting solution quality. Finally, an exact approach could be followed. This entails modelling the problem as a mathematical program and solving it to optimality using a combination of Linear Program (LP) solvers and branching and separation algorithms. Such an approach trades fast computation times for the benefit of producing the global optimal solution. Additionally, the exact approach provides a measure of solution quality during the entire computation process, so that when the process is stopped halfway, a quantitative indication of how far the design deviates from the optimal is available.

Both heuristic and exact approaches to designing PONs have been studied in literature, as detailed in chapter 3, although it is envisioned that both approaches can be employed in a real-world network deployment design. In the preliminary design stages, a heuristic can provide quick solutions to get an estimate on the return on investment for a large number of potential sites, and can ensure that the input parameters result in a practically feasible network design. Then, in the final stages before deployment when higher solution quality, i.e. lower deployment cost, offsets the additional computation time required, an exact approach may be used to produce the final design. In this thesis, we will focus on the latter part of the design process, where solution quality is of the utmost importance.

### 1.1.2 Modelling accuracy vs. computational tractability

When modelling any real-world phenomena, the practitioner has to decide on a certain level of abstraction. For example, a simple set of gears may be modelled to include only the relative speed between them, or it may include minutiae such as the forces exerted on each gear tooth and how the material deforms under speed and load. Even though complexity may be added ad infinitum, the model will likely become impossi-

ble to solve, forcing the practitioner to compromise. Especially in the exact framework, where computation times may be substantial, the model complexity must be kept low enough to be solved in a feasible amount of time, which, depending on the application, may be minutes, hours or even weeks. With a fixed amount of time available to solve the model, an increase in computational performance allows the practitioner to add more complexity, resulting in a more practically relevant model. Additionally, with improved computational performance comes the ability to solve larger networks and it becomes possible to consider more complex variants of the PON topology, including dimensioning splitters, designing cascaded splitter arrangements, designing with redundancy in mind or explicitly accommodating uncertain parameters in the model.

### 1.1.3 Designing under demand uncertainty

During the early network design stages, it is often the case that some of the topological parameters are uncertain, in particular the customer demand, as it is difficult to determine before a network is deployed. While some service providers may follow an approach of deploying the network and hoping customers will utilise it, it is more likely that at least *some* information concerning the expected demand is available. Examples include estimations based on population data, where demand is based on household income, or estimations using current demand for legacy networks. This can be leveraged to maximise return on investment by designing the network to connect all relevant customers while avoiding over-dimensioning equipment and increasing cost. Therefore, we require a robust design which produces a viable network irrespective of how the demand realises. Two of the most commonly used techniques in literature to accomplish this include two-stage stochastic programming and robust optimisation, both of which will be investigated for application in the PON design process in this thesis.

### 1.1.4 State of FTTH and FTTB

The practical relevance of the research is dependent on the current and future state of FTTH and FTTB, both worldwide and in South Africa in particular, since the need for designing PONs presupposes a demand for these networks.

#### Worldwide

Worldwide, FTTH and FTTB is gaining substantial traction, especially in Asia, where almost 116 million subscribers have been connected according to the FTTH Global Alliance [15]. Europe comes in second with 14.8 million connected homes, while North America trails at 14.1 million. In terms of optical fibre connections per capita, the United Arab Emirates leads with close to 75 % penetration, with South Korea, Hong Kong and Japan all surpassing 50 % [16]. The most connected country in Europe is Lithuania, managing almost 35 % penetration while the United States trails at just over 10 %. Even though an enormous number of FTTH connections exist, indicating its popularity, penetration rates for a number of countries are still low, with only 35 countries having more than 1 % coverage. This indicates adoption rates are going to increase dramatically in the foreseeable future, suggesting that even small cost improvements in deploying these networks can amount to large overall savings for countries and service providers investing in these networks.

#### Sub-Saharan Africa and South Africa

In Sub-Saharan Africa, FTTH deployment has only just started, reaching 125,000 total subscribers in February of 2015 [15]. This is less than 1 % of the numbers seen in the developed world and suggests that there are still large divides to be closed to reach broadband internet ubiquity. However, fibre is already within a 25 km reach for 44 % of the population, indicating promising growth [17].

In South Africa in particular, the popularity of FTTH, and specifically PONs, has been

steadily increasing, with optical fibre deployment surmounting to a potential land grab in the near future [18]. Huawei launched their PON solution in South Africa in March 2015, with a number of companies, including Telkom SA SOC Ltd. and MTN, already deploying trial networks for cost evaluation. Telkom has also stated that they plan to connect 1 million homes via FTTH by 2018 [19], leveraging their extensive fibre network (see [20] for a detailed illustration). In terms of total investment cost in the South African market, PON deployments are believed to be comparable to currently deployed Asymmetric Digital Subscriber Line (ADSL) technologies, making it a strong local contender for next-generation access networks [21].

In conclusion, on both international and national levels, it is clear that contributions toward PON design can have a significant real-world impact, both in terms of cost reduction, which allows more people to benefit from broadband connectivity, and service reliability.

## 1.2 Research goal

The goal of the research presented is to provide a flexible, exact framework capable of producing optimal PON network designs with improvements in two distinct areas:

- **Computational tractability** - Using algorithmic techniques, improve computational performance to allow for the design of larger or more complex network configurations.
- **Modelling accuracy** - Improve the practical relevance and accuracy of the model by incorporating uncertain demand, real-world attenuation effects and refinements such as distributed splitting and network survivability.

## 1.3 Research contributions

The major contributions originating from the research toward the field of PON design in particular and multi-hierarchy networks in general, can be divided into two categories: algorithmic contributions and modelling contributions.

- **Algorithmic contributions**
  1. An existing PON design model is modified and decomposed into a number of independent segments according to its structure in an attempt to improve computational performance.
  2. Strengthening procedures for the PON model are presented and tested to determine efficacy. These are defined and modified for each variant of the model, including standard, stochastic, survivable and distributed splitting configurations.
  3. Results for each variant are presented in a computational study detailing the efficacy of the proposed modifications compared to known approaches in the literature.
- **Modelling contributions**
  1. The PON design model is extended to incorporate demand uncertainty by utilising sets of potential outcomes. This is done using both stochastic programming and robust optimisation principles.
  2. A revenue-based PON model is presented, showcasing how operational considerations can guide the network design.
  3. Attenuation effects are integrated into the presented framework in the form of independent and dependent path length constraints. The implicit handling of these constraints also contributes to multi-hierarchy networks in general.
  4. The modelling framework is extended to incorporate full- and semi-redundancy for the optical fibres between the CO and splitters.

5. A multi-level variant of the PON model, along with the corresponding attenuation considerations, is formulated to design networks with an arbitrary number of cascaded splitters.

Even though the contributions are presented specifically in the PON context, note that the proposed framework can be easily adapted and utilised for any multi-hierarchy network design. Additionally, all relevant industry standards are considered in each step of the modelling process.

## 1.4 Research methodology

In arriving at the contributions listed above, an appropriate methodology for modelling real-world phenomena is followed:

- **Relevant literature** - The relevant literature is studied to determine both the current state of the art for the PON design problem as well as the common approaches followed. Additionally, technical and theoretical concepts concerning PON, optimisation algorithms and variable uncertainty are investigated.
- **Iterative model formulation** - Existing design models are reformulated and a path-based version is decomposed to determine its feasibility in improving computational tractability. This is an iterative step where modifications are done, compared to the original, and adapted according to the analysis of the results. The iterative procedure stops when a sufficient improvement has been achieved.
- **Validation and verification** - All formulations are verified and validated using appropriate techniques for modelling, which, due to it being problem specific, is notoriously difficult (see [22–25] for attempts at addressing this issue). The details of the procedure followed are provided in section 1.4.1.
- **Algorithmic improvements** - Once we have a model formulation that meets the requirements, additional strengthening procedures are applied to further im-

prove computational performance. The techniques applied are external to the model and are dependent on the observations made during the iterative formulation process.

- **Model refinements** - With a scalable framework in hand, we can proceed to refine the model, incorporating additional physical or operational constraints to improve practical relevance. Ideally, the algorithmic improvements should allow for a more complex model to be solved in a similar time-frame as was observed in the initial formulation results. This illustrates how we trade performance for accuracy in mathematical modelling.

### 1.4.1 Validation and verification

As noted above, a complete general validation and verification approach applicable to all models is difficult to formulate, with Sargent [25] suggesting that these be tailored for each specific instance. Furthermore, [24] explains that no model validation technique is absolute, as modelling is inherently only a representation of a system. Therefore, the difficulty lies in determining the level of validation the application requires, which can be exacerbated in environments where comparative solutions from the original system are difficult to obtain. In [23], the author suggests using statistical measures for verification of simulation models, but also argues that this is unnecessary for models having only deterministic inputs. Therefore, we will utilise the following guidelines as set forth by Carson [24]:

- **Face validity** - For a given input scenario, determine if the output produced is reasonable and if the logic behind the model is sound.
- **Input parameter range** - Test the model over a range of input parameters and inspect the output. The idea behind this step is to *stress-test* the model over the expected range of parameters when it is finalised.
- **Comparison** - If applicable, the output of the model should be compared to past

performance of the system or to a baseline model representing an existing system. If a new system is being designed, the model behaviour should be compared to the relevant assumptions and specifications.

At each stage of the modelling process, face validity is ensured by inspecting the topological results and comparing it to what is expected. This includes observing the placement of trenches and splitters and ensuring feasible optical fibre routing paths exist for each ONU. Routing feasibility will be covered in each subsequent chapter and adapted for each model variant. Additionally, the objective is compared to previous iterations to determine if an increase or decrease is warranted according to theoretical expectations.

Next, the models are tested using a large number of data set instances to verify that the model can operate over a wide range of inputs and to reduce bias in conclusions drawn from the result analyses. Instances are derived from both local and international sources to further diversify the test environment.

For the comparison step, existing models in literature are reimplemented to serve as baseline models for inter-model verification, while the behaviours of all subsequently presented models are compared to specifications as standardised in the ITU-T G.984 recommendations. This includes completely characterising every network design solution in terms of optical fibre length and attenuation and ensuring all parameters are within acceptable tolerances.

Finally, since the implementation is done in `C++`, good software engineering principles are followed to ensure correct translation from the mathematical model to programming code. Additionally, modular design principles are used throughout to facilitate code verification.

## 1.5 Thesis overview

The rest of the thesis is organised as follows: chapter 2 aims to familiarise the reader with the technical aspects of the research, introducing concepts specific to optical fibre communication and PONs. As a companion, chapter 3 focusses on the mathematical modelling concepts relevant to the approach followed, in addition to presenting relevant work in the field.

The proposed modelling framework is detailed in chapter 4, explaining the modelling considerations as well as validation and verification techniques. In chapter 5, strengthening procedures are described and tested, aiming to improve the computational performance of the presented model.

Once a scalable formulation has been found, chapter 6 details how the framework is extended to integrate the concept of uncertain demand. Chapter 7 aims to improve the practical relevance of the model through extensions related to splitter dimensioning, survivability and distributed splitting. Finally, chapter 8 concludes the thesis with remarks and recommended future research.

# Chapter 2

# Technical background

*This chapter provides an overview of relevant technical information to contextualise the research. In particular, it outlines optical fibre concepts, Point-to-Point (P2P) and P2MP fibre networks as well as PONs. Finally, after studying the relevant standards, additional specialised versions of PON are detailed, including the use of distributed splitting or multi-level networks and the incorporation of network survivability.*

## 2.1 Introduction

What was initially a research project commissioned by the United States government in the 1960s to study robust inter-computer communication, led to the creation of ARPANET, a regional network connecting academic institutions, and a precursor to the current-day *internet* [26]. Since then, especially as commercial enterprises were incorporated into the network in the early 90s, it has evolved into a global network of networks, interconnecting billions of both public and private devices across all continents.

Standardisation of the functions of these communication networks came in the form of the Open Systems Interconnection (OSI) model [27,28], developed by the International Organisation for Standardisation (ISO), and the Internet protocol suite, also known as TCP/IP, which resulted from research done by the Defense Advanced Research Projects Agency (DARPA) in the 1970s [29]. Both of these models propose a layered protocol stack, each performing a distinct function to enable end-to-end communication. The lowest layer, the *physical* layer in the OSI model and the *link* layer in TCP/IP, standardises communication protocols between physical devices, including electrical and mechanical specifications, synchronisation and the way data is transmitted through the medium.

These link layer protocols are interface specific, which usually falls into one of three main categories: metallic, optical and wireless. All three types are based on the same principle of electromagnetic wave propagation, with the difference lying in the propagation medium. In this sense, the propagation of electrical signals through copper or aluminium wire fall into the metallic category, while radio- or microwaves propagating through the air is categorised as wireless. Finally, optical technologies rely on the propagation of light pulses through a translucent glass or plastic medium. For this thesis, we limit our scope to optical communication mediums.

## 2.2 Optical fibre networks

As stated above, optical fibre networks are communications networks that transfer data between two points by modulating electromagnetic radiation and passing it through a medium that is translucent to the frequency of the radiation. For most optical networks, the frequency is in the near-infrared range, which consists of wavelengths between 780 nm and 3000 nm [30], although a number of bands, known as the *telecommunication windows*, are of special significance, as shown in table 2.1 [3, 4]. In this range, the most widely used medium is an optical fibre made from silica glass, which guide the light by means of total internal reflection.

Table 2.1: Standard wavelength bands for optical fibre [3,4]

| Band | Name | Wavelength range |
|---|---|---|
| O band | Original | 1260-1360 nm |
| E band | Extended | 1360-1460 nm |
| S band | Short wavelengths | 1460-1530 nm |
| C band | Conventional | 1530-1565 nm |
| L band | Long wavelengths | 1565-1625 nm |
| U band | Ultra-long wavelengths | 1625-1675 nm |

### 2.2.1 Optical fibre

Depending on the diameter of the core in the optical fibre, it is referred to as either *multi-mode* or *single-mode*. Multi-mode fibres usually have cores with a thickness of more than 50 $\mu$m, and have multiple light beams propagating through them, originating from a non-coherent source. Single-mode fibres have very thin cores, less than 10 $\mu$m, resulting in an internal reflection angle of close to 90°, which, in conjunction with a coherent light source, means the light propagates essentially horizontally along the fibre [31]. The different construction of single- and multi-mode fibres are shown in figure 2.1.

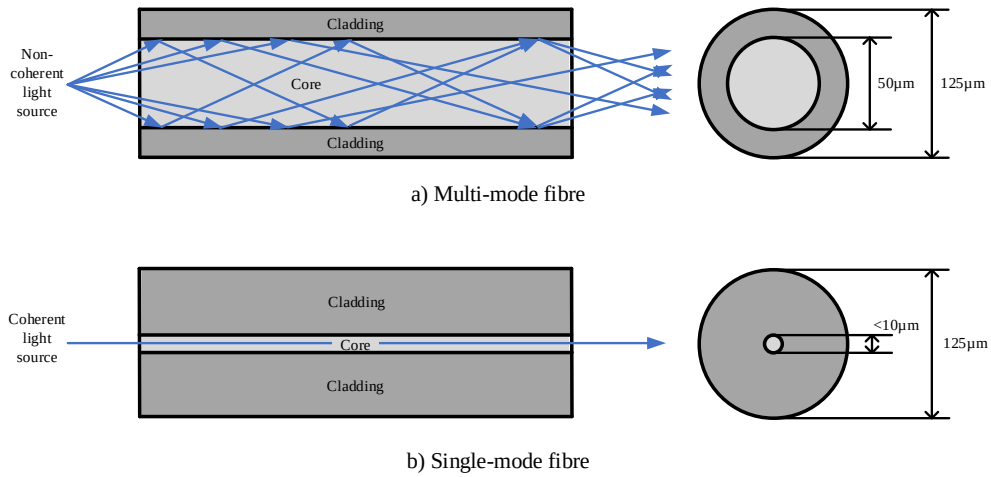



Figure 2.1: Single- and multi-mode fibres

Both single- and multi-mode fibres suffer from two main types of attenuation: *scattering* and *absorption* [32]. Scattering refers to a loss of power due to differences in the

refractive indices of the core and cladding, even on a molecular scale, which can result in light reflecting in arbitrary directions. On the other hand, absorption refers to power loss due to imperfect transparency of the core at specific wavelengths, resulting in the heating of the fibre, similar to how the colour of an object is produced. The dominant attenuation effect depends on the wavelength of the light travelling through the fibre, with scattering having a larger effect at shorter wavelengths.

At some specific wavelengths, namely 1000 nm, 1400 nm and above 1600 nm, leftover doping products used to change the refractive index of the glass, primarily hydroxyl ions ($OH^-$), have major absorption peaks, limiting the useful wavelengths of the fibre [32]. These peaks, illustrated in figure 2.2, are also known as *water peaks*, and special fibre types, known as Low Water Peak Fibre (LWPF), try to minimise them, resulting in a fibre capable of carrying the entire spectrum between 1260 nm and 1675 nm with low attenuation. Normal fibre, on the other hand, are operated at selective wavelengths where scattering and absorption are lowest, usually in the 1310 nm or 1550 nm range.

The ITU-T specifies four different fibre types in their G.652 recommendation, including standard G.652.A/B and extended spectrum LWPF G.652.C/D fibres [33], with typical maximum attenuation figures of 0.3–0.4 dB/km. The ISO also has standards for standard single-mode (B.1.1/OS1) and LWPF single-mode (B.1.3/OS2) fibres in ISO 11801:2002 [34]. Standards for 50 $\mu$m and 62.5 $\mu$m multi-mode fibre is also provided in the form of the G.651.1 recommendation [35] and the OM1–OM4 range in ISO 11801 [34], which has attenuation values of around 3.5 dB/km at 850 nm and 1 dB/km at 1300 nm.

Multi-mode fibres have higher losses due to the more acute internal reflection angle, which results in more scattering, but they are cheaper to produce, making them ideal for short-range interconnects. Additionally, very short multi-mode fibres can be made from plastic instead of silica glass, known as Plastic Optical Fibre (POF). These fibres have much higher absorption losses, making them only feasible for 660 nm non-coherent light sources, but are even cheaper to produce. Conversely, single-mode fibres have very low losses, but due to tight manufacturing tolerances and clarity require-

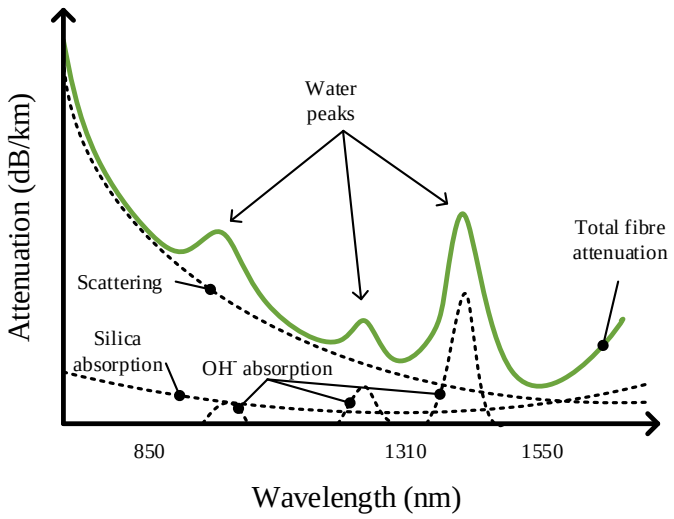



Figure 2.2: Typical optical fibre attenuation curve (adapted from [1])

ments are more expensive to produce, which is why they are used for long-range links where low losses are essential.

### 2.2.2 Fibre penetration

Due to improving technology and cost reduction, optical fibres have been steadily moving from a core-centric role right up to the edge of the network, known as the *access network*. The collective term to refer to the penetration of fibre to the access network is Fibre to the x (FTTx) [36]. While access networks have traditionally been implemented using cheaper, copper-based or wireless technologies, present day telecommunication networks have fibre up to the node (FTTN), the curb (FTTC), the building (FTTB) and even right to the customer premises or home (FTTH). This shift from metallic wires to optical fibres is illustrated in figure 2.3.

One of the technology improvements which allows further penetration of fibre towards the access network, is the P2MP optical fibre network. This type of network shares a single fibre amongst a number of customers in the physical layer, which is less expensive than P2P networks, where a fibre is required between each end-point. In P2MP

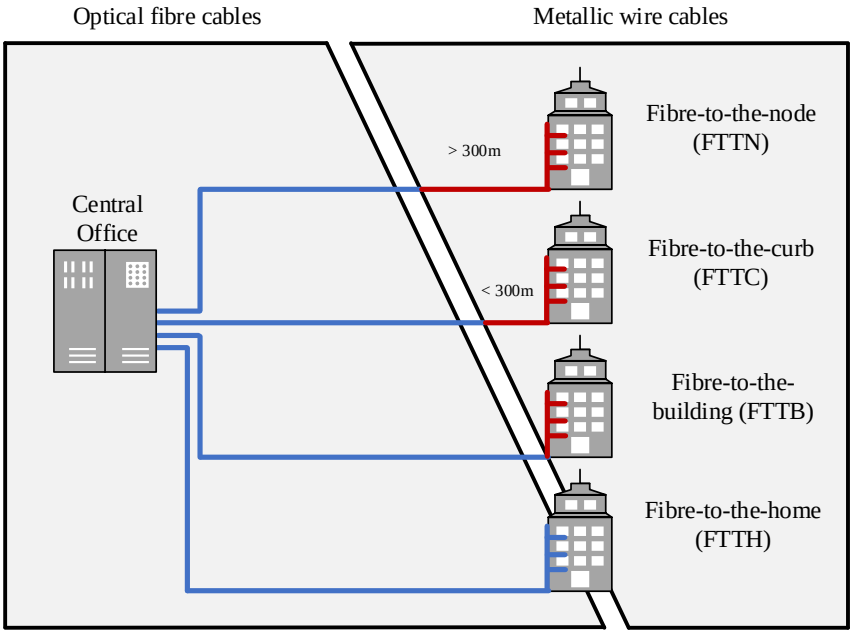



Figure 2.3: Fibre penetration towards the customer premises (FTTx)

networks, a shared fibre exists in what is known as the *feeder network*, connected to an aggregation device or switch. This device then has a number of downstream fibres connected to it, each connected to a customer premises. Utilising a P2MP network and sharing fibres can be up to 50 % cheaper than the equivalent P2P network [9].

### 2.2.3 Active vs. passive optical P2MP networks

In terms of P2MP optical fibre technologies, we can divide between two distinct types: active and passive. This simply refers to the aggregation device, with active networks having active switches, which capture and regenerate the optical signals, in contrast to passive networks which utilises passive splitters, devices that split or aggregate the incoming signals using optical waveguides. Active devices are usually more expensive, but since they regenerate the optical signals, they have longer reach capability. Additionally, the active switches work on Layer 2 or 3 of the network protocol stack, meaning that data only gets forwarded to its intended recipient. Conversely, passive devices require no power in the field and are agnostic to the specific link layer require-

ments for the network, reducing operational costs and resulting in a more future-proof network [9].

## 2.3 Passive Optical Networks

Since passive aggregation devices are much cheaper than active switches and are easier to install and operate, they garner more attention from service providers. The basic topology of the PON consists of the CO, which houses the OLT equipment, and a number of connected passive splitters, each with a number of output ports, connected in turn to ONUs (also known as Optical Network Terminals (ONTs)) at the customer premises. The optical signal originates from the OLT, gets passed through optical waveguides in the passive splitters, which divides the power equally amongst its output ports, therefore forwarding the complete signal to each connected ONU [9]. This forms a tree structure, with the CO as root. In the reverse direction, the splitter acts as a coupler, while a multiple access scheme is used to allow ONUs to communicate in turn with the OLT, usually in the form of Time Division Multiple Access (TDMA). This type of PON is also known as Time Division Multiplexing PON (TDM-PON). More complex forms of PON include Wavelength Division Multiplexing PON (WDM-PON), which has yet to be standardised, but uses more than one wavelength in either the downstream or upstream direction, providing improved privacy and higher bandwidth. The section between the CO and the splitters is called the *feeder network* while the splitter to ONU connections form what is known as the *distribution network*. Figure 2.4 illustrates the topology.

### 2.3.1 Passive splitters

The passive optical splitters are constructed using either Fused Biconical Taper (FBT) or Planar Lightwave Circuit (PLC) technology. FBT splitters are constructed from cascaded sets of fused 3 dB-splitters, each with two input and two output ports, made

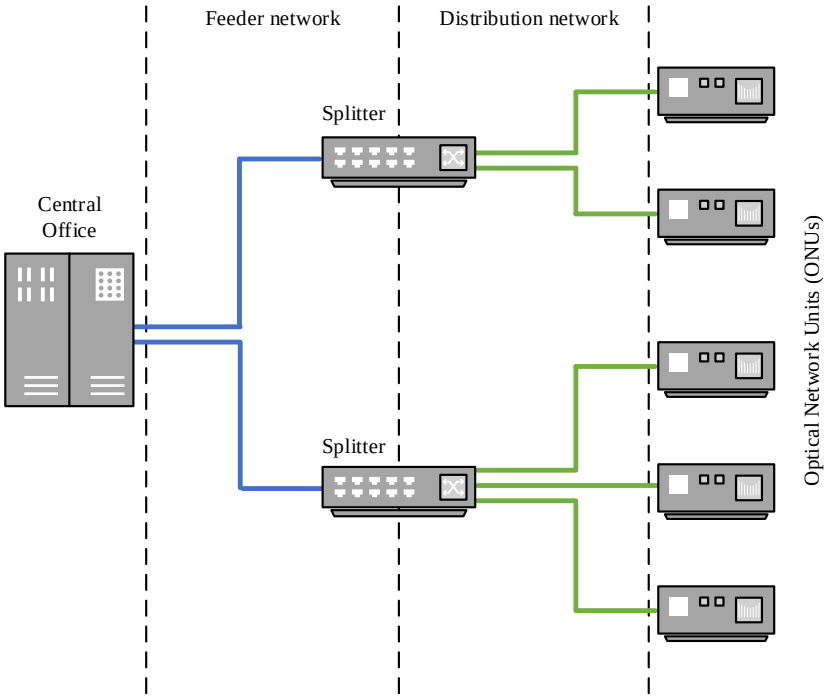



Figure 2.4: Basic topology of a Passive Optical Network (PON)

by fusing two fibres together side-by-side. In these 3 dB-splitter sections, the input power is divided 50:50 between the output ports, with the ratio between them the *split ratio* [37]. PLC splitters are constructed by etching a silica substrate using photolithography techniques, similar to those used in semiconductor manufacturing, creating optical waveguides that distribute the optical power amongst the output ports. For larger split ratios, each output port is then split into two by an additional stage of 3 dB-splitters until we are left with the desired number of final output ports. Due to potential reliability issues, the largest usable FBT passive splitters have 1:32 split ratios, which contain five stages of fused 1:2 splitters, while PLC splitters can be made with split ratios higher than 1:64. Another advantage of PLC splitters is that they are usable across all fibre frequency bands while FBT splitters typically only have operating wavelengths of 850 nm, 1310 nm and 1550 nm [38].

Due to the cascaded nature of passive splitters, the ideal splitter attenuation can be

calculated through the following equation:

$$A_{\text{splitter}} = 10 \times \log_{10} N, \tag{2.1}$$

with $N$ being the number of output ports of the splitter. FBT splitters have internal splices, with an approximate loss of 0.3 dB for each stage. The number of stages $k$ for a splitter with $N$ output ports is $k = \log_2 N$. Therefore, the total attenuation for an FBT splitter can be estimated by:

$$A_{\text{splitter}}^{\text{FBT}} = 10 \log_{10} N + 0.3 \log_2 N. \tag{2.2}$$

In contrast to FBT splitters, which have the same attenuation in both directions, due to the reciprocity of its 3 dB-splitters, PLC splitter stages can have a lower upstream loss, since its 3 dB-splitters may form a coupler with less than 3 dB loss. Additionally, PLC splitters have better *uniformity* (how evenly the power is distributed), and a wider range of operating temperatures.

### 2.3.2 Physical constraints

During deployment planning, a number of additional physical factors become important over and beyond just assigning passive splitters to ONUs. Next, we focus on two of these factors, including PON attenuation and fibre installation.

#### Attenuation

To ensure reliable communication, we have to ensure that the receiver at the ONU can detect the optical signal sent by the OLT. The transmitted power, also known as the *launch power*, is dependent on the OLT optics, while the ONU has a minimum required received power, known as the *sensitivity*. The difference between the transmitted power and the receiver sensitivity is known as the *power budget* or *link budget*, which is the maximum attenuation the network can suffer before communication is no longer possible. Power budgets of 24–29 dB are common for PONs, depending on the standard utilised.

The total PON attenuation consists of a number of components, including the fibre loss, $A_{\text{fibre}}$, and all Channel Insertion Loss (CIL), which consists of splicing ($A_{\text{splice}}$), connector ($A_{\text{connector}}$) and splitter attenuation ($A_{\text{splitter}}$). Therefore, the total attenuation can be written as follows:

$$A_{\text{PON}} = A_{\text{fibre}} + A_{\text{splice}} + A_{\text{connector}} + A_{\text{splitter}}. \tag{2.3}$$

Note that for a given power budget, the maximum network reach, which is determined by $A_{\text{fibre}}$, is dependent on the splitter loss, $A_{\text{splitter}}$ which is the dominant term in (2.3). Stated differently, for higher split ratios, $A_{\text{splitter}}$ increases, which means that for a maximum fixed link budget, and hence a fixed $A_{\text{PON}}$, the maximum fibre loss decreases, limiting the total fibre length.

**Fibre installation**

Since optical fibre is very fragile, the fibre is encapsulated in a number of protective sheaths to prevent breakage. These usually include a tough resin *buffer* surrounding the cladding, as well as a *jacket* to add strength. Additional flexible strengthening fibres, such as kevlar, may also be present between the buffer and jacket layers. Multicore fibres encapsulate a number of fibres in an array of additional sheaths, including waterproofing, metal armour and polyurethane outer jacket layers.

Installation of optical fibre is done in two different ways: aerial and subterranean. As the names imply, aerial fibres are spanned across riser poles, often using existing infrastructure such as standard telephone poles, while subterranean fibres are installed underground. In this thesis, we focus on subterranean fibre, since it is the most common in metropolitan areas. Due to the high cost of initial installation, ducts are usually installed in the ground instead of individual fibres, containing a number of *microducts*, each capable of holding a number of fibres. This allows for future expansion, by using either cable pulling or blowing techniques to install additional fibres in open ducts. Figure 2.5 shows a typical duct for illustration. If no previous network infrastructure exists, the installation is referred to as a *greenfield* network, while in a *brownfield* setting,

ducts or even fibre may already be available for use in the network deployment. It is evident that we want to minimise the number of ducts we install, so *fibre duct sharing*, where a number of fibres are routed in the same duct, is of great concern.

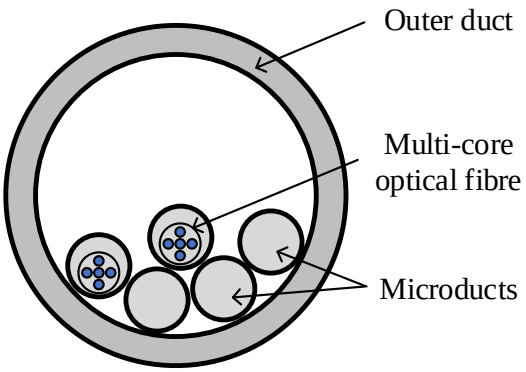



Figure 2.5: Typical fibre duct construction with microducts

### 2.3.3 IEEE 802.3ah/av standards

After the formation of the Ethernet in the First Mile (EFM) study group in November 2000 and the P802.3ah task force in September the following year, a new standard for Ethernet over P2MP fibre emerged, ratified as IEEE 802.3ah in June 2004. This standard was intended to extend Ethernet from the core into the access network, providing a consistent platform for the entire network and improving interoperability.

The new IEEE 802.3ah standard, also known as EPON, concerns itself with the link layer, providing 1 Gb/s bandwidth fibre links in both directions. 1490 nm and 1310 nm wavelengths are used for the down- and upstream directions respectively, along with a reserved 1550 nm wavelength for future expansion or legacy services [10]. In the standard, a conservative power budget of 24 dB is specified, along with a split ratio of 1:16, although in practice, 29 dB power budgets and 1:32 or even 1:64 split ratios are attainable with present day technology [9]. Maximum total network diameter is 20 km, using B.1.3 fibres [34].

Based on the Ethernet structure, EPON has the same packet format as all previous 802.3

compatible devices, even using the same MAC layer. This ensures easy transmission of the frame since EPON frames are not encapsulated before being sent through the network. Instead, a simple preamble, containing a Logical Link ID (LLID) is added to the packet, using the same 802.3 Interpacket Gaps (IPGs), to differentiate between ONU recipients. As all ONUs receive the same packets, this LLID, for which multiple may be registered to a single ONU, is used to filter the data stream. The same LLID is used in the upstream direction at the OLT to determine the origin ONU. Each LLID corresponds to a type of service, which allows data, video and Voice-over-IP (VoIP) to be differentiated at the ONU level.

Even though not included in the standard, all practical implementations of EPON include encryption, usually in the form of Advanced Encryption Standard (AES), since every ONU has access to all the downstream frames. Additionally, frame-based Forward Error Correction (FEC) based on the Reed-Solomon algorithm provides parity information, appended at the end of the frame. This FEC technique can improve bit error rates significantly, using only 6 % additional overhead in the case of RS(255,239) for a coding gain of 4–6 dB [39].

In 2009, an extension to 802.3ah was ratified by the P802.3av task force, known as IEEE 802.3av or 10G-EPON [40]. The standard improved bandwidth capabilities to either symmetrical 10 Gb/s or 1 Gb/s and 10 Gb/s for the up- and downstream links respectively. Interestingly, the use of a different downstream wavelength, 1577 nm, allows both EPON and 10G-EPON to coexist on the same network, while the upstream wavelength is specified as 1270 nm. This ensures seamless upgrades as long as the upstream channels are separated using standard Time Division Multiplexing (TDM).

### 2.3.4 ITU-T G.984/G.987 recommendations

Following definition in the Full Service Access Network (FSAN) consortium in 2001 and ratified in February 2004, the ITU-T G.984 recommendation series specifies the P2MP network known as GPON. This series covers all aspects of the network, includ-

ing basic definitions and architecture [11], the Physical Medium Dependent (PMD) layer [12], the Transmission Convergence (TC) layer [13] and management requirements [14]. As with EPON, these standards are focused on the physical and link layers of the TCP/IP stack.

In its current form, GPON provides high bandwidth, 2.488 Gb/s and 1.244 Gb/s for down- and upstream respectively, using the common wavelengths 1490 nm and 1310 nm [11]. The same 1550 nm wavelength is reserved for downstream video in GPON as specified for EPON. Up to 60 km network reach is supported, using a 28 dB power budget, while the maximum differential distance between ONUs is limited to 20 km. This differential limit avoids synchronisation issues, as the upstream link is accessed using TDM. As such, GPON supports both asynchronous and synchronous services, which includes support for legacy Asynchronous Transfer Mode (ATM). Finally, split ratios of up to 1:128 are supported by the recommendation [9], and G.652 compatible fibres are used throughout [33].

Packets are encapsulated by GPON Encapsulation Method (GEM), a low overhead structure supporting a number of protocols, including Ethernet and TDM. Additionally, similar to the LLIDs of EPON, ONUs can register a number of GEM ports, each with an ID, to run multiple services on a single physical distribution fibre. Similar to ATM, frames are fixed time length, in contrast to Ethernet, with 125 $\mu$s downstream frames providing an 8 kHz reference clock to the ONUs. Encryption is also mandated on the downstream link, utilising AES with either an ONU or a GEM port specific key. The GPON Transmission Convergence (GTC) layer in G.984.3 maps all user data onto the GEM frames and includes a number of interesting features, including a ranging procedure to measure ONU distance in order to set the individual equalisation delay [13]. In addition, it contains a number of configurable parameters, including the type of FEC used and queueing options. As with EPON, frame-based FEC based on the RS(255,239) algorithm is common.

The ITU-T ratified another set of recommendations in 2010, increasing the bandwidth of GPON comparable to 10G-EPON. This 10 Gb/s version of GPON, known as XG-

PON, is specified in G.987.1–G.987.3 [41–43]. As was seen in 10G-EPON, XG-PON provides two bandwidth configurations: XG-PON1, an asymmetrical 2.5 Gb/s and 10 Gb/s up- and downstream version, and XG-PON2, which provides symmetrical 10 Gb/s links. Due to their low chromatic dispersion, the 1577 nm and 1270 nm wavelengths are utilised for the down- and upstream links, the same as specified for 10G-EPON.

For this thesis, the IEEE 802.3ah and ITU-T G.984 standards are similar enough in terms of physical construction that both topologies can be modelled using the same techniques. However, where specific parameters are considered, we will focus on the G.984 recommendation as it is the one service providers in South Africa are most interested in [44].

## 2.4 Multi-level networks

Typically, when deploying PONs, a single splitter level is used, placed strategically close to its connected ONUs and connected directly to the CO. However, both the IEEE 802.3ah and ITU-T G.984 standards and their derivatives allow for what is known as *distributed splitting*. In this scenario, the final effective split ratio is made up of multiple interconnected splitters, which may result in reduced costs as splitters can be placed closer to the customer premises, reducing the total fibre length. This is also known as a *multi-level network*, in reference to the hierarchical configuration of the splitters, each on its own level. The topology of such a network is shown in figure 2.6, along with the effective split ratio at each stage.

As multiple splitters now exist between the OLT and the ONU, the total attenuation for a PON with $M$ different levels is now given by:

$$A_{\text{PON}}^{\text{multi}} = A_{\text{fibre}} + A_{\text{splice}} + A_{\text{connector}} + \sum_{i=1}^{M} A_{\text{splitter}}^{i}, \tag{2.4}$$

where $A_{\text{splitter}}^{i}$ is the insertion loss for a splitter in the $i$-th level. Since splitters have a fixed loss component and due to the additional connectors and splices required, $A_{\text{PON}}^{\text{multi}}$

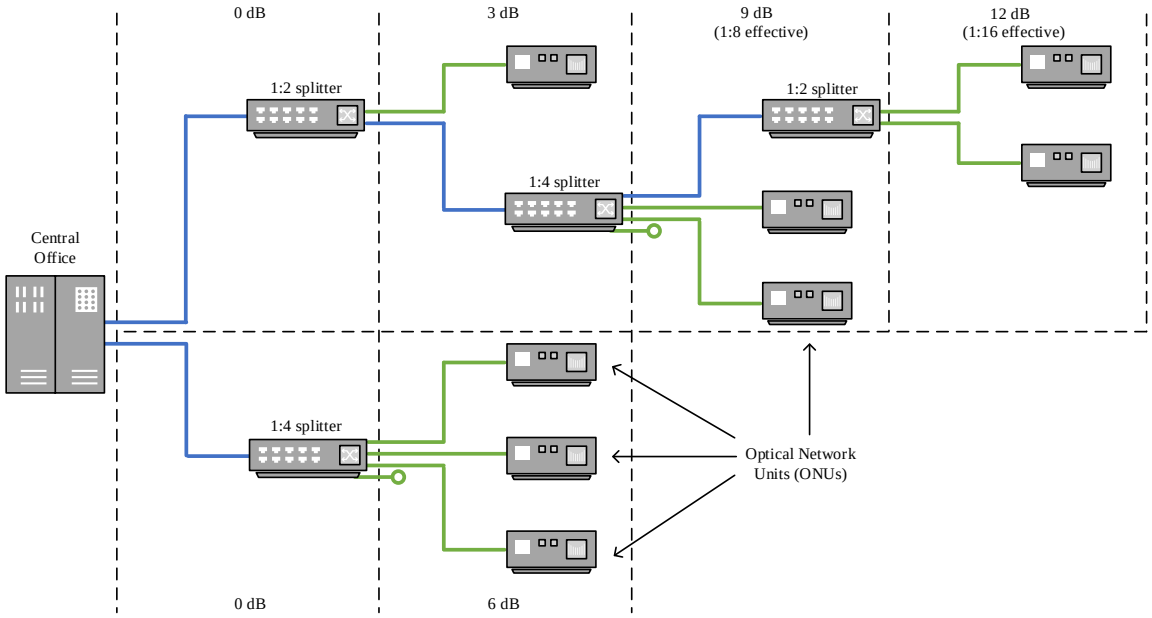



Figure 2.6: Multi-level/distributed splitting configuration of the PON, showing effective split ratios and ideal attenuation for each network region

is usually greater than $A_{\text{PON}}$ for a network with the same effective split ratio, reducing the maximum network diameter. However, the cost benefits may outweigh this disadvantage, especially in high-density networks close to the CO, where the diameter is not a limiting factor.

Service providers may also choose to use either a *homogeneous* or *heterogeneous* multi-level network. Since passive splitters are used, it is perfectly feasible to connect both ONUs and additional downstream splitters to a specific splitter, as long as the network reach and differential distance constraints hold. This configuration is known as a heterogeneous network. From an operational standpoint however, it may be easier to manage a homogeneous network, where only either downstream splitters or ONUs are connected to each splitter. Therefore, a splitter may be designated as a *feeder splitter*, pre-splitting the feeder fibre, or a *distribution splitter*, which serves as the last split point before reaching the customer premises.

## 2.5 Survivable networks

Since PONs aggregate a number of customers on the same fibre, protecting that fibre against disturbances, natural or otherwise, is of great concern. This gives rise to the concept of *survivable* networks, where some redundancy or provisioning strategy is implemented to ensure that when a range of disturbances are encountered, e.g. a feeder fibre is cut, the network survives with some given minimum service level and is restored in an acceptable time frame.

Again, while both EPON and GPON supports the same optional modes of network survivability, we use the GPON recommendation nomenclature. Four different types are specified in G.984, Type A through Type D, each with varying protection characteristics [11]. These are now explained in more detail below, with figures 2.7 through 2.10 illustrating each type.

- **Type A** - In this configuration, only the feeder fibres are duplicated, providing a spare that can be utilised in case of a fibre breakage.
- **Type B** - This type duplicates both the feeder fibres and the OLTs, utilising dual input splitters (2:$N$) to couple the inputs before it reaches the splitting stages.
- **Type C** - Extending on Type B, this configuration also duplicates the distribution fibre and ONU connections, using both dual input ONUs and splitters. This configuration then provides full protection against any single point of failure and is known in G.984.1 as a *full duplex* network.
- **Type D** - In a Type D duplex configuration, partial duplication is realised using a mix of Type B and Type C, providing protection in-between these types. This is usually implemented in multi-level networks where partial protection is incorporated into the intermediate networks.

In Types A, B and D, cold standby of circuits are used, resulting in inevitable signal and frame loss during a disturbance. However, all connections in higher protocol layers

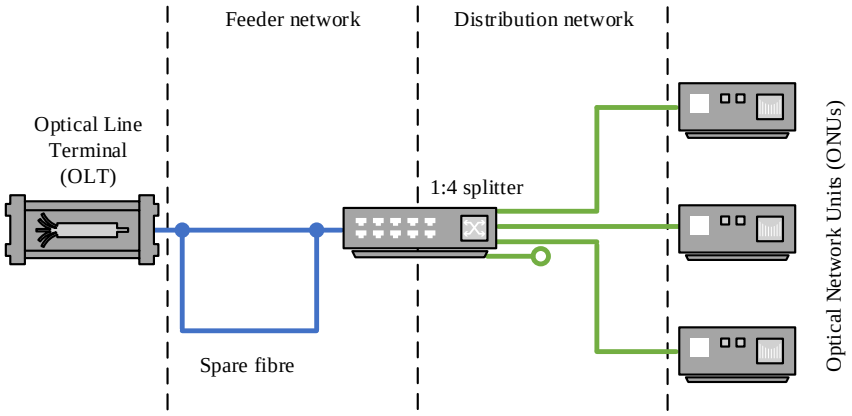



Figure 2.7: G.984 Type A survivable network (adapted from [2])

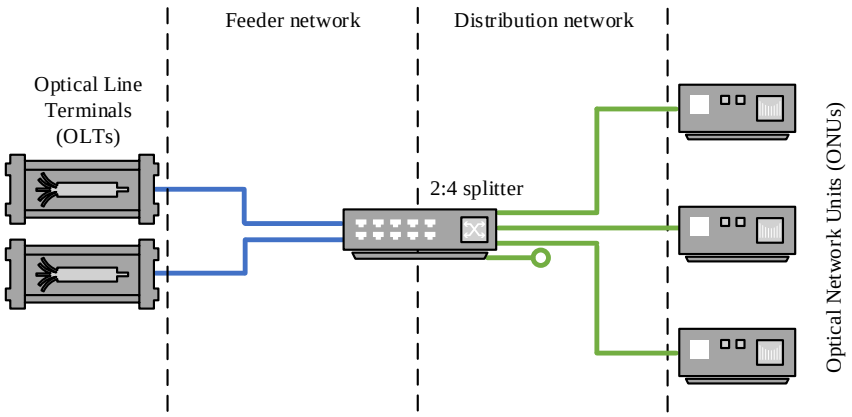



Figure 2.8: G.984 Type B survivable network (adapted from [2])

should still hold, resulting in quick switch-over. Networks in the full duplex Type C configuration can have hot standby of receiver circuits, allowing the implementation of protection modes where even frame loss can be avoided, also known as *hitless switching* [2]. Duplicating all distribution side fibres comes at a prohibitive cost however, and will typically only be used in mission critical networks.

Presently, Types B and C are the most commonly used, with Types A and D deprecated in the 2008 amendment of G.984 [11]. However, Type A remains the least expensive method of providing survivability, since the number of OLTs (or OLT line cards), a large fixed cost component, remains the same.

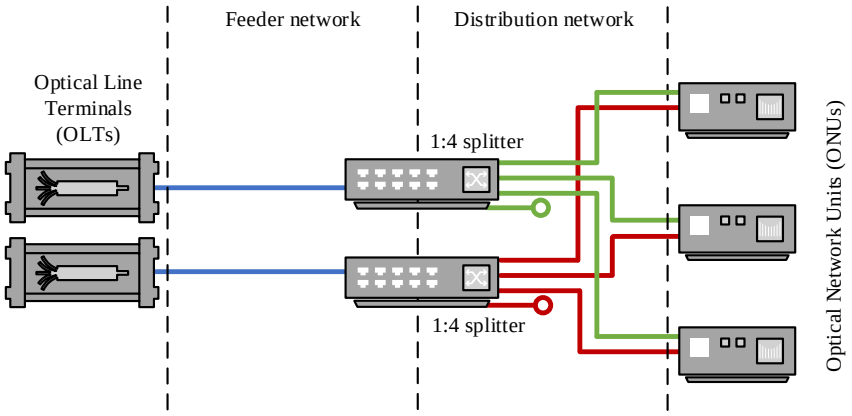



Figure 2.9: G.984 Type C survivable network (adapted from [2])

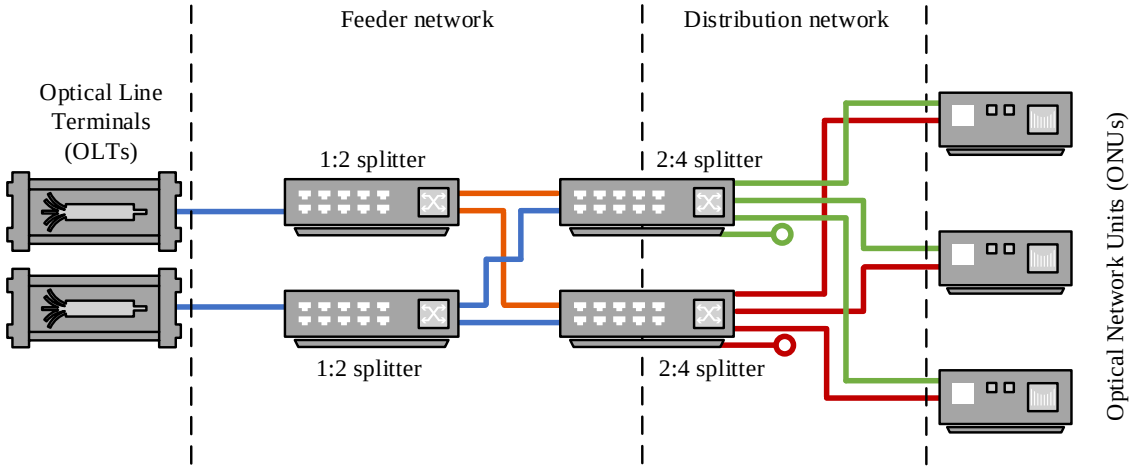



Figure 2.10: G.984 Type D survivable network (adapted from [2])

## 2.6 Conclusion

In this chapter, the technical components of passive optical networks were outlined. This included an introduction into the TCP/IP layers at which PONs operate, optical fibre technical concepts and how this determines the standards, in particular, the different up- and downstream wavelengths, as well as the terminology concerning the various levels of fibre penetration. Next, P2P and P2MP networks were discussed, highlighting the inherent advantages of P2MP in both active and passive forms. The

concepts of passive optical networks were then outlined, focussing on the passive splitter construction, attenuation values and the two prominent standards: IEEE 802.3ah EPON and ITU-T G.984 GPON.

Finally, two optional PON modes were illustrated, including the use of multiple cascaded splitters, known as multi-level networks or distributed splitting, and network survivability, the concept of protecting a network against possible disturbances through duplication of fibres and equipment.

Now that the technical concepts have been detailed, the next chapter will look at the modelling concepts and solution techniques relevant to the thesis.

# Chapter 3

# Modelling and optimisation techniques

---

*This chapter aims to provide contextualisation in terms of modelling and optimisation techniques relevant to the research. These include optimisation techniques, linear programming and its terminology, theoretical equivalents and decomposition. Stochastic and robust optimisation are also discussed, before the Passive Optical Network Design Problem (PONDP) is defined according to its constituent problems. Related work on exact, heuristic and metaheuristic techniques are then provided.*

---

## 3.1 Mathematical modelling

The concept of mathematical modelling can be defined as the mathematical *representation* of the underlying *structure* of some phenomena in the real world. It is a very powerful technique, allowing researchers to visualise interactions between objects in a system before real-world implementation, or change input parameters to test hypotheses. Even though this type of modelling has been done for decades, it is only since the advent of computers that complex models can be formulated with any hope of finding a solution.

A mathematical model is a set of written equations describing the interaction of a number of variables, as well as limiting the variability of some of those variables. Therefore, most models are just a set of equations, known as *constraints*, and a set of variable domains. Stated more generally, a model of a problem $P$, with variables $\boldsymbol{x} \in \mathbb{R}^n$ and a set of allowable variable values S, can be written as follows:

$$P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \in \mathrm{S}\}. \tag{3.1}$$

Any given vector of values $\boldsymbol{x}^* \in \mathrm{S} \cap \mathbb{R}^n$ is known as a *feasible solution* to $P$.

If all the operators in the model exhibit linear behaviour, i.e. if S can be described using only linear equations, $P$ is known as a *linear model*, otherwise it is a *non-linear model*. In general, non-linear models are more difficult to study than linear ones, although it may be possible to *linearise* some non-linear models through various techniques.

Though models describe *how* a system interacts, it is reasonable to ask which state of a system is *best*, which leads us to the topic of optimisation.

## 3.2 Optimisation

Optimisation, i.e. the search for the best or *optimal* state of a system, is a fundamental principle observed in the state of our universe. Minimum energy crystal formation, atom bonding for minimum electron energy as well as biological evolution are all examples of nature trying to find an optimal state for a system [45].

In the field of mathematical modelling, optimisation refers to finding a set of input values that result in an optimal outcome, called the *objective*. The objective, along with a set of constraints defining the variable interaction, is called a *mathematical program*. Stated more formally, given an objective function $f(x) : \mathbb{R}^n \to \mathbb{R}$, which translates the current state of the variable values to some real quantity, and a set $\mathrm{S} \subseteq \mathbb{R}^n$, describing

the model constraints, a mathematical program can be written as follows:

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) \tag{3.2}$$

$$\text{subject to} \quad \boldsymbol{x} \in \mathrm{S}. \tag{3.3}$$

In this case, the minimum value of the objective is considered best, although any program in this form can be transformed to a maximisation problem by minimising the negative of the objective function, $-f(\boldsymbol{x})$, instead.

Differentiating between local and global optima is important in the optimisation field, as some methods are only guaranteed to find local optima. To this end, given an objective function $f : \mathrm{S} \to \mathbb{R}$, $\hat{x}_\ell \in \mathrm{S}$ is a *local optimum* if, for all neighbouring $x \in \mathrm{S}$, it is either a local maximum, where $f(\hat{x}_\ell) \geq f(x)$, or it is a local minimum, with $f(\hat{x}_\ell) \leq f(x)$ [45]. A *global optimum*, $\hat{x}$, is then a generalisation of a local optimum, where $f(\hat{x}) \geq f(x)$ or $f(\hat{x}) \leq f(x)$ for all $x \in \mathrm{S}$ for global maximum and minimum respectively. Figure 3.1 differentiates between the optimality types for a simple objective function $f(x)$, $x \in \mathbb{R}$.

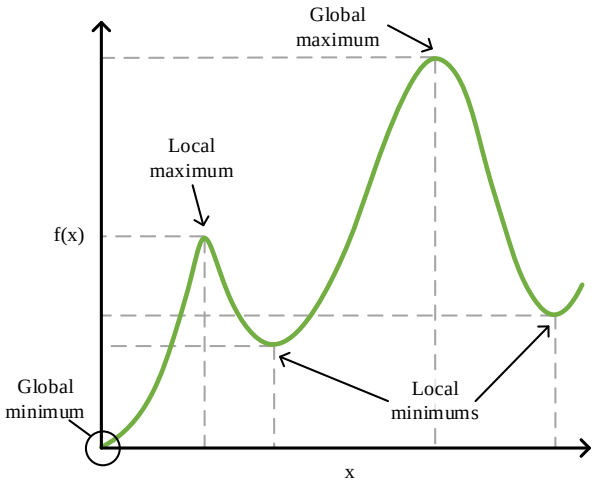



Figure 3.1: Local- and global optima for an objective function $f(x)$

Mathematical programs, as defined in (3.2)–(3.3), can be classified according to their behaviour, similar to models in general. In this sense, if the objective and all constraints are linear functions, i.e. $f(\boldsymbol{x})$ is linear, we have a Linear Program (LP). LPs can be

written in standard form as follows:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{3.4}$$

$$\text{s.t.} \quad A\boldsymbol{x} \leq \boldsymbol{b}, \tag{3.5}$$

$$\boldsymbol{x} \in \mathbb{R}^n_+, \tag{3.6}$$

with $\boldsymbol{c}$ the objective function coefficient vector, $A$ a matrix of constraint coefficients and $\boldsymbol{b}$ a vector containing the right-hand side constraint values. $\boldsymbol{x}$ is the variable vector to be determined. Each constraint of the form $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \leq b$ is known as a *half-space*, i.e. one side of the *hyperplane* introduced by $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = b$, with $\boldsymbol{a}, \boldsymbol{x} \in \mathbb{R}^n$ and $b \in \mathbb{R}$. The constraint set $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$ is then an intersection of a finite set of half-spaces and is known as a *polyhedron*, an $n$-dimensional polygon. If additionally, the polyhedron is bounded, i.e. it can be enclosed by an $n$-ball, which is an $n$-dimensional sphere, of finite radius, it is known as a *polytope*. For bounded polyhedrons, the optimal solution will always be found at one of its vertices, also known as *extreme points*.

If either the objective function or constraints are non-linear in nature, we have a Non-linear Program (NLP). Non-linear formulations are avoided due to possible issues such as non-convexity of the objective function, for which no efficient solution process is known. If an NLP has a quadratic objective function, i.e. if $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$, it is known as a Quadratic Program (QP). If only the constraints are quadratic in nature, it is known as a Quadratically Constrained Program (QCP).

LPs are further differentiated in terms of the domains of its variables. If all variables are integer, i.e. $\boldsymbol{x} \in \mathbb{Z}^n_+$, the resulting program is termed an Integer Linear Program (ILP). If the LP has both integer and real variables, it is known as a Mixed Integer Linear Program (MILP). Generally, ILPs and MILPs are far more difficult to solve than standard LPs, often taking orders of magnitude more computational effort. This is due to the difficulty in contracting the multi-dimensional real *search space*, i.e. all the feasible solution values in the constraint polyhedron, to its integer convex hull. In essence, we need to find an optimal integer solution vector $\boldsymbol{x}^* \in \mathbb{Z}^n \cap \{x \in \mathbb{R}^n_+ : A\boldsymbol{x} \leq \boldsymbol{b}\}$, i.e. the optimal integer point inside the constraint polyhedron. The difference between a real and integer two-dimensional search space is demonstrated in figure 3.2.

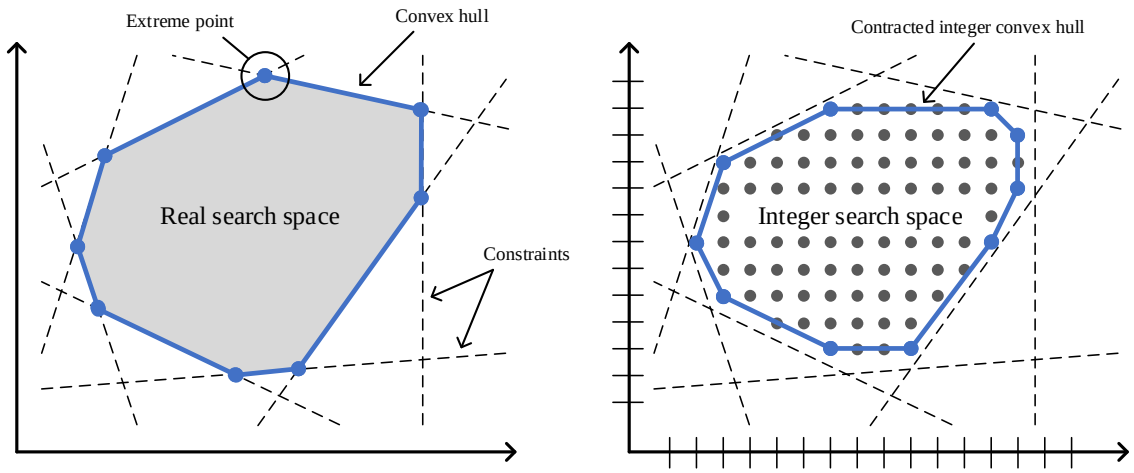



Figure 3.2: LP vs. ILP search spaces

Next, we consider the concept of *duality*, which refers to the fact that every minimisation LP has a complementary maximisation problem, known as a *dual* problem, which has the same optimal solution. Given an LP in standard form as specified in (3.4)–(3.6), known as the *primal* problem, the dual problem can be written as follows:

$$\max \quad \boldsymbol{b}^{\mathrm{T}}\boldsymbol{u} \tag{3.7}$$

$$\text{s.t.} \quad A^{\mathrm{T}}\boldsymbol{u} \geq \boldsymbol{c}, \tag{3.8}$$

$$\boldsymbol{u} \in \mathbb{R}^m_+. \tag{3.9}$$

In general, the relationship between the primal and dual problems can be defined according to these well-known theorems (see [46] for a complete discussion):

**Theorem 3.1** (Weak Duality)**.** *Given feasible solutions $\boldsymbol{x} \in \mathbb{R}^n_+$ for the primal and $\boldsymbol{u} \in \mathbb{R}^m_+$ for the dual in standard form, then the following holds:*

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \leq \boldsymbol{u}^{\mathrm{T}}A\boldsymbol{x} \leq \boldsymbol{b}^{\mathrm{T}}\boldsymbol{u}. \tag{3.10}$$

*Therefore, if either problem is unbounded, the other is necessarily infeasible.*

**Theorem 3.2** (Strong Duality)**.** *If either the primal or dual problem has a finite optimal solution, then so does the other, and they coincide:*

$$\boldsymbol{c}^{\mathrm{T}}\bar{\boldsymbol{x}} = \boldsymbol{b}^{\mathrm{T}}\bar{\boldsymbol{u}}, \tag{3.11}$$

*with $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$ the optimal solutions of the primal and dual problems respectively.*

Weak duality is more general, as it holds even for non-convex problems, while strong duality requires more specific circumstances (in particular it holds for LPs). The difference between the two solutions, $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{u}$, is known as the *duality gap*. To determine if strong duality holds for a convex problem, *Slater's condition* can be used, which states that the feasible region must have an interior point [47]. In addition, optimality can be determined by the complementary slackness theorem [46]:

**Theorem 3.3** (Complementary Slackness)**.** *The solutions $\boldsymbol{x}^* \in \mathbb{R}^n_+$ and $\boldsymbol{u}^* \in \mathbb{R}^m_+$ for the primal and dual problems respectively are optimal if and only if they are feasible and both of the following two statements hold:*

- $x_j^* = 0$ *or* $\sum_{i=1}^m a_{ij}u_i^* = c_j$ *or both,* $j = 1, 2, \ldots, n$,
- $u_i^* = 0$ *or* $\sum_{j=1}^n a_{ij}x_j^* = b_i$ *or both,* $i = 1, 2, \ldots, m$,

*with $a_{ij}$ the entry at the i-th row and j-th column of matrix A.*

Duality is very powerful in some instances, since it may be easier to solve the dual of the problem, either due to the structure or due to the specific solution method employed. Then, if strong duality holds, the dual optimal solution is equivalent to the primal optimal solution.

### 3.2.1 Complexity

Once we have a mathematical model, it can be analysed according to its computational complexity, which translates into the time required to solve the model in relation to the

Table 3.1: Big-O time complexity of a number of well-known problems

| $O(f(n))$ | Time complexity | Problem |
|---|---|---|
| $O(1)$ | Constant | Determining if an input is even/odd |
| $O(\log n)$ | Logarithmic | Binary search algorithm on a sorted vector of size $n$ |
| $O(n)$ | Linear | Finding the average value of a vector |
| $O(n \log n)$ | Linearithmic | Calculating a Fast Fourier Transform (FFT) |
| $O(n^2)$ | Quadratic | Insertion sort of a vector of size $n$ |
| $O(n^3)$ | Cubic | Gaussian elimination for an $n \times n$ matrix |
| $O(n^k)$, $k$ fixed | Polynomial | Interior point LP solution methods |
| $O(k^n)$, $k$ fixed | Exponential | Solving LP using the simplex algorithm |

size of the inputs. For illustration purposes, the complexity measure is often represented in the *big-O* notation, $O(\cdot)$, which describes the asymptotic performance, i.e. the dominant order of how computational effort scales relative to the input size. As an example, finding the sign of an input using the $\text{sgn}(\cdot)$ function is done in constant time, represented as $O(1)$, while sorting a list using insertion sort takes quadratic time, or $O(n^2)$, with $n$ the size of the list. Table 3.1 contains a number of well-known problems with their corresponding time complexity [48].

To further analyse the complexity of mathematical models, we define classes of problems which can be solved in a specified time. For this purpose, a more theoretical approach is followed, using what are known as *Turing machines*, a kind of generic computer algorithm. A theoretical construct proposed by Alan Turing in 1936, a Turing machine is a hypothetical machine capable of reading from and writing to a strip of tape, taking actions depending on its current state. If, during each step of the machine, a unique follow-up state exists for each non-final state, the Turing machine is said to be *deterministic*, while if at any step, multiple successor states exist that should be followed simultaneously, it is said to be *non-deterministic* [48]. While computation in a deterministic Turing machines occurs sequentially, non-deterministic computation can be visualised as an initial state, branching out into a potentially infinite number of paths simultaneously, before stopping at some final state.

We can now define two important complexity classes relevant to mathematical optimisation, which describe problem complexity in terms of Turing machine computation.

**Definition 3.1** (P)**.** *Let* DTIME$(f(n))$ *denote the class of problems solvable by a deterministic Turing machine in time* $O(f(n))$*. Then,*

$$\mathrm{P} = \bigcup_{k \in \mathbb{N}} \mathrm{DTIME}(n^k) \tag{3.12}$$

*is the set of all problems that can be solved by deterministic Turing machines in polynomial time, i.e. efficiently solvable problems.*

**Definition 3.2** (NP)**.** *Let* NTIME$(f(n))$ *denote the class of problems solvable by a non-deterministic Turing machine in time* $O(f(n))$*. Then,*

$$\mathrm{NP} = \bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(n^k) \tag{3.13}$$

*is the set of all problems that can be solved by non-deterministic Turing machines in polynomial time. Additionally it is the class of problems for which solutions can be guessed or verified in polynomial time.*

Next, to compare the complexity of problems, we need to define the concept of *reducibility*. Any problem A is reducible to problem B if we can use an efficient algorithm to solve B as a subroutine to also solve A efficiently. It follows then that problem A is at most as *hard*, i.e. as complex, as B to solve. Additionally, polynomial-time many-one reducibility, also known as *Karp reducibility*, is the ability to compute a function $f(n)$ in polynomial time to map the inputs of one problem to another, so that the output of both problems are the same. Now we can define two new complexity classes, NP-hard and NP-complete as follows:

**Definition 3.3** (NP-complete and NP-hard)**.** *Consider two problems, A and B. If A is polynomial-time many-one reducible to B for all* $A \in NP$*, i.e.* $A \leq_m^p B$*, B is said to be NP-hard. If B is NP-hard and* $B \in NP$*, then B is NP-complete.*

From the definition above, it is evident that NP-hard problems are at least as hard as the hardest NP problems. Additionally, since NP-hard problems are not necessarily in NP, they can also be harder than the hardest NP problems, i.e. harder than all NP-complete problems. This increasing complexity measure is illustrated in figure 3.3, for

both cases of the unproven P versus NP problem, i.e. if a problem can be *verified* in polynomial time, can it also be *solved* in polynomial time?*

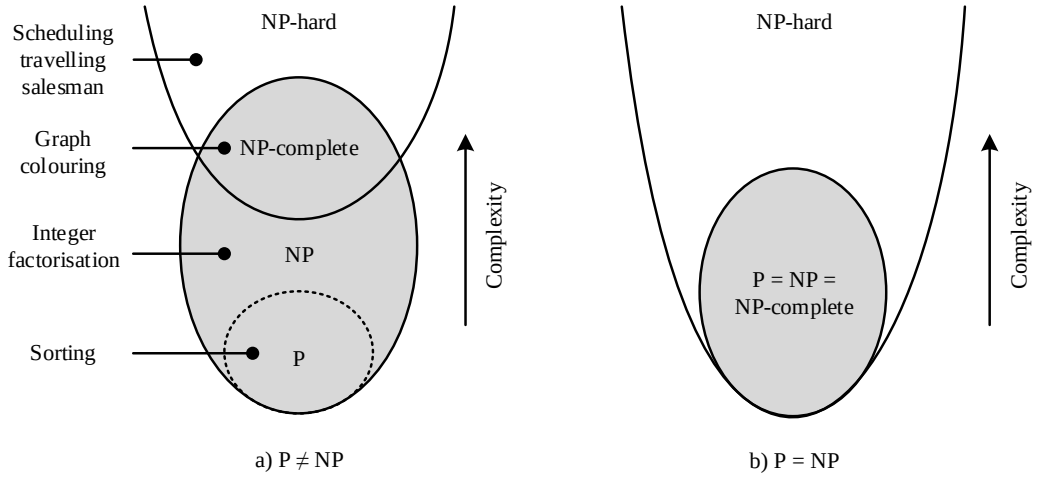



Figure 3.3: Euler diagram of complexity classes for P $\neq$ NP and P $=$ NP

### 3.2.2 Exact solution methods

Now that we know how complex a problem is, we need to go about solving it. For LPs, a number of methods exist to solve them efficiently to global optimality, including interior point and simplex algorithms. ILPs and MILPs require a different technique, known as branch-and-bound, that is capable of finding integer solutions inside a real search space. The branch-and-bound algorithm also uses an LP solver as a subroutine, often solving thousands of LPs before finding an optimal integer solution. A number of commercial LP and ILP solvers exist, including IBM ILOG CPLEX [49], Gurobi [50] and the open-source COIN-OR LP (CLP) [51].

*This is arguably one of the most important unanswered questions in computer science and one of the Millennium Prize problems, for which the Clay Mathematics Institute offers $ 1,000,000 for a correct solution if the reader is willing.

### Simplex method

Developed by George B. Dantzig in 1947, the simplex method, named after the concept of generalising triangles to arbitrary dimensions, is still one of the most popular LP solvers [52]. It is a basis-exchange pivot algorithm, finding improved solutions by means of traversing the edges and vertices of the polyhedron in the direction for which the objective function is strictly increasing. Since the optimal solution is always found at an extreme point, the algorithm will terminate with the optimal solution if no adjacent vertex in any direction yields a better objective value [53]. Figure 3.4 shows how the algorithm moves along edges to extreme points until the optimal solution is found.

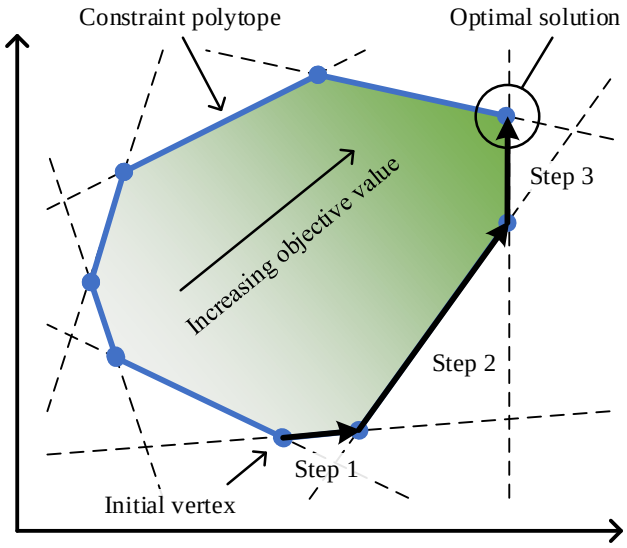



Figure 3.4: Extreme point traversal using the simplex method

A big improvement over the then-used Fourier-Motzkin elimination method, the simplex method is still remarkably efficient in practice, in spite of its worst-case exponential complexity. In fact, on average, it often displays polynomial complexity, making it the fastest LP solver for a wide variety of problems.

**Branch-and-bound algorithm**

The branch-and-bound algorithm is a *divide and conquer* approach to solving ILPs, developed by A. H. Land and A. G. Doig and published in 1960 [54]. In its general form, it works by splitting a candidate set into a number of subsets, known as *branching*, calculating bounds for each subset, and, if they are worse than the best known global bounds, discarding or *pruning* that branch. This is repeated iteratively until the working set is empty or the bounds are equal, resulting in the optimal solution. The pseudocode is given in algorithm 3.1.

**Algorithm 3.1:** General branch-and-bound

**Data**: Set S of candidate solutions.

**Result**: Best lower ($\check{b}$) and upper bound ($\hat{b}$) on S

```
1  function BranchBound(S, b̌, b̂):
2      while S ≠ ∅ and b̌ ≠ b̂ do
3          split S into sets S_1, S_2, ...                    //Branch
4          foreach S_i do
5              calculate bounds b̌_i and b̂_i for S_i           //Bound
6              if b̌_i > b̂ or b̂_i < b̌ then
7                  discard set S_i                            //Prune
8              else
9                  b̂ ← min(b̂, b̂_i)
10                 b̌ ← max(b̌, b̌_i)
11                 BranchBound (S_i, b̌, b̂)
12             end if
13         end foreach
14     end while
15     return b̌, b̂
16 end function
```

When used for solving ILPs, the lower bounds are computed by solving a linear relaxation of the ILP, i.e. the original problem without the integrality constraints. Upper

bounds are computed using heuristics or whenever the relaxation is integer feasible. Branching occurs on the fractional variables, resulting in one problem where the fractional variable is forced less than or equal to its floor value and one where it is forced greater than or equal to the ceiling value.

Note that during the computation of the branch-and-bound algorithm, the bounds are always available, giving a measure known as the *optimality gap*, or the relative difference between the best known lower bound and the best integer feasible solution. Given an objective function $f : \mathbb{R}^n \to \mathbb{R}$, the current best lower bound solution $\check{\boldsymbol{x}}$ and the best integer solution $\hat{\boldsymbol{x}}$, it can be calculated as follows:

$$\text{Optimality gap } \% = \frac{|f(\hat{\boldsymbol{x}}) - f(\check{\boldsymbol{x}})|}{f(\check{\boldsymbol{x}})} \times 100\ \%. \tag{3.14}$$

This allows for the termination of the branch-and-bound algorithm before completion while still providing a guarantee of the worst-case solution quality. A solution with a 5 % optimality gap will be at most 5 % from the optimal solution. As time passes, the bounds improve, and the optimality gap shrinks, as illustrated in figure 3.5.

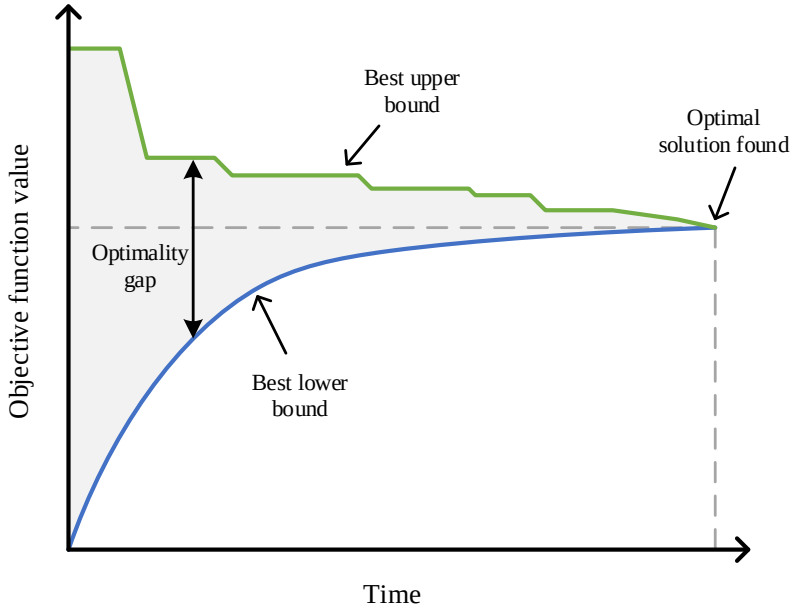



Figure 3.5: Bound convergence during execution of the branch-and-bound algorithm

In the case where branch-and-bound is used in conjunction with a cutting plane algorithm [55], the resulting algorithm is known as *branch-and-cut* [56]. In this approach, after each branching step, the algorithm adds cutting planes in an attempt to reduce the number of fractional variables in the solution. This step serves to separate hyperplanes, or linear *cuts*, that cut off the fractional solution point without cutting off any integer feasible points, thereby ensuring the variables are "less fractional" in subsequent iterations. Common cuts include Gomory Mixed-Integer (GMI), clique, cover, flow-cover and Mixed Integer Rounding (MIR) cuts (see [57] for an explanation of these).

**Benders decomposition**

If a model exhibits a block structure, it can be exploited by projecting out a subset of the variables or constraints into a sub-problem, whose solution can be used to incorporate the variables or constraints back into the original reduced problem. This operation is also known as *decomposition*.

A number of decomposition methods exist, including Dantzig-Wolfe, Lagrangian and Benders. In Dantzig-Wolfe decomposition, a subset of variables are used in the original problem, incorporating them dynamically using what is known as *column generation*, i.e. using information contained in the block structure to decide which variables need to be included to improve the solution [58]. Conversely, Benders decomposition starts with a subset of constraints, which are incorporated back into the original problem implicitly using information projected out at the start [59]. This approach is also known as *row generation*. Finally, Lagrangian relaxation decomposes a problem by partitioning constraints into two sets, substituting the variables in one, and adding a constraint that equates the original and substituted variables to the objective function, punishing violation with a Lagrange multiplier [60]. Then, subgradient methods are utilised to adjust the Lagrange multipliers to find the best solution, which for the Lagrangian decomposition method, is a bound on the original optimal solution.

Consider now Benders decomposition, or just Benders for short, which solves problems in the following form [59]:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + f(\boldsymbol{y}) \tag{3.15}$$

$$\text{s.t.} \quad A\boldsymbol{x} + F(\boldsymbol{y}) \geq \boldsymbol{b}, \tag{3.16}$$

$$\boldsymbol{x} \in \mathbb{R}^n_+,\ \boldsymbol{y} \in \mathrm{S}, \tag{3.17}$$

with the variables $\boldsymbol{y}$ denoting the *hard* variables, e.g. S may be difficult to handle, i.e. $\mathrm{S} \in \{0,1\}^m$, $f$ or $F$ may be non-linear etc.

The idea of Benders is then to temporarily fix $\boldsymbol{y}$ and solve the remaining problem, resulting in a solution for $\boldsymbol{x}$ in terms of $\boldsymbol{y}$. Then, we utilise the structure of the problem to improve the optimal solution bound for $\boldsymbol{y}$. Consider then the problem with fixed $\boldsymbol{y}^*$ (the constant term $f(\boldsymbol{y}^*)$ can be removed from the objective):

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{3.18}$$

$$\text{s.t.} \quad A\boldsymbol{x} \geq \boldsymbol{b} - F(\boldsymbol{y}^*), \tag{3.19}$$

$$\boldsymbol{x} \in \mathbb{R}^n_+. \tag{3.20}$$

Using duality theory, we find the equivalent dual problem:

$$\max_{\boldsymbol{u}} \quad [\boldsymbol{b} - F(\boldsymbol{y}^*)]^{\mathrm{T}}\boldsymbol{u} \tag{3.21}$$

$$\text{s.t.} \quad A^{\mathrm{T}}\boldsymbol{u} \leq \boldsymbol{c}, \tag{3.22}$$

$$\boldsymbol{u} \in \mathbb{R}^m_+. \tag{3.23}$$

We can then write the original problem as follows:

$$\min_{\boldsymbol{y}\in\mathrm{S}} \left[ f(\boldsymbol{y}) + \min_{\boldsymbol{x}\in\mathbb{R}^n_+} \left\{ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} : A\boldsymbol{x} \geq \boldsymbol{b} - F(\boldsymbol{y}) \right\} \right], \tag{3.24}$$

which, by substituting the inner minimisation with the equivalent maximisation dual, is equivalent to:

$$\min_{\boldsymbol{y}\in\mathrm{S}} \left[ f(\boldsymbol{y}) + \max_{\boldsymbol{u}\in\mathbb{R}^m_+} \left\{ [\boldsymbol{b} - F(\boldsymbol{y})]^{\mathrm{T}}\boldsymbol{u} : A^{\mathrm{T}}\boldsymbol{u} \leq \boldsymbol{c}) \right\} \right]. \tag{3.25}$$

In (3.25) then, the outer minimisation problem is known as the *master problem*, while the inner maximisation dual problem is known as the *sub-problem*.

By splitting the dual variables $\boldsymbol{u}$ into extreme rays $\boldsymbol{u}_r$, and extreme points $\boldsymbol{u}_p$, the master problem, now called the *restricted master problem*, can be written equivalently as follows:

$$\begin{aligned} \min \quad & z && (3.26)\\ \text{s.t.} \quad & z \geq f(\boldsymbol{y}) + [\boldsymbol{b} - F(\boldsymbol{y}^*)]^{\mathrm{T}} \boldsymbol{u}_p, && (3.27)\\ & 0 \geq [\boldsymbol{b} - F(\boldsymbol{y}^*)]^{\mathrm{T}} \boldsymbol{u}_r, && (3.28)\\ & \boldsymbol{y} \in \mathrm{S}. && (3.29) \end{aligned}$$

After solving the sub-problem (3.21)–(3.23), if the solution is unbounded, find the extreme ray $\bar{\boldsymbol{u}}_r$ and add the following constraint to the master:

$$0 \geq [\boldsymbol{b} - F(\boldsymbol{y})]^{\mathrm{T}} \bar{\boldsymbol{u}}_r. \tag{3.30}$$

Otherwise, if the solution is finite, add the following constraint:

$$z \geq f(\boldsymbol{y}) + [\boldsymbol{b} - F(\boldsymbol{y})]^{\mathrm{T}} \bar{\boldsymbol{u}}_p, \tag{3.31}$$

with $\bar{\boldsymbol{u}}_p$ the extreme point, or optimal solution, of the sub-problem.

### 3.2.3 Heuristics

Apart from simply solving a problem using the exact techniques described above, a few techniques can be utilised to find *good* sub-optimal solutions. These algorithms, known as *heuristics*, can be defined as a step in an optimisation algorithm that uses the currently available information to determine which candidate solution from the search space should be considered next [45]. Generally then, heuristics are used to guide the optimisation algorithm in the direction of the optimal solution, without having to enumerate all the candidates.

Heuristics are often used to get a fast upper bound solution on the problem, which can be passed directly to the optimisation algorithm as a *warm-start*. Though usually problem specific, a few general heuristics include random search space sampling

[61], Relaxation Induced Neighbourhood Search (RINS) [62] and hill climbing. Integer rounding techniques can also be used to try to get an integer feasible solution from a fractional one. Additionally, Lagrangian decomposition can be seen as a heuristic, since it solves a relaxed version of the problem.

A special class of heuristics, known as *meta-heuristics*, are formulated in a very general way, treating the problem as a black box without insight into its inherent structure [45]. These are often based on natural phenomena, using system behaviour to guide the heuristic to an improved solution (see [63] and [64] for surveys on the topic). In the case of Genetic Algorithms (GAs), sometimes called Evolutionary Computation (EC) algorithms, the heuristic is guided by genetics, in particular, evolution through selection, to improve subsequent generations of solutions through random mutation. The concept of solidifying metallic elements is used in the technique known as Simulated Annealing (SA), where the slow cooling of the substance is interpreted as a reduced probability of accepting worse solutions, a key aspect in avoiding convergence to a local optimum. In ant colony and particle swarm optimisation, creature behaviour is used to find a good solution, using pheromones or local and global best-known positions in a swarm respectively. Finally, some meta-heuristics are not based on natural phenomena at all, e.g. Tabu search and Random optimisation.

Since meta-heuristics are abstracted from the problem structure, they can be used to solve non-linear and non-convex problems, hence their popularity. Unfortunately, even though heuristics provide a bound on the optimal solution of a problem, there is no indication of solution quality. Therefore, even though a number of heuristics give good solutions in practice, there is no guarantee that a specific solution has a low optimality gap, i.e. is a *good* solution. In addition, many heuristics and meta-heuristics can converge to local optimums, missing the global optimum. It is these disadvantages that make exact methods so attractive, even though the computational effort required can be much greater.

## 3.3 Stochastic optimisation

While deterministic models are very powerful, systems in practice often have uncertain influences, e.g. market movement in a pricing model, random perturbations of variables due to measurement error margins or estimation of future variable values. Uncertainty can be handled in a number of ways, although all methods rely on transforming random variables into some deterministic form, usually through a statistical estimate. We now distinguish between four approaches to handle stochastic variables [65]:

- **Two-stage recourse** - In this approach, the modelling is done in two stages. In the first stage, a decision is made, where after a random event realises, affecting the outcome. Then, in the second stage, the model compensates for this realisation using a recourse action, improving the outcome of the first stage.
- **Multi-stage recourse** - Similar in concept to the two-stage approach, the multi-stage approach generalises the recourse periods to an arbitrary number. This modelling approach is suitable for cases where the uncertainty is revealed gradually over a number of time periods, in what is termed a *stochastic process* [65].
- **Risk averse** - In the risk averse approach, the randomness is incorporated into the model via statistical metrics, e.g. ensure that the mean deviation from the $n$-th quartile of the variable value is less than $x$, minimise the expected outcome of a variable as well as the corresponding risk value etc.
- **Robust** - When the random variable has some bounds placed on it, robust optimisation can be used, which aims to make decisions that are feasible to all possible disturbances to a variable [66]. In the case of uncertain estimates, it would minimise the worst-case outcome, ensuring a reasonable solution even in the face of adversity.

The most common approaches used are the two-stage recourse and robust optimisation techniques, both of which will now be discussed in more detail.

### 3.3.1 Stochastic programming

The two- and multi-stage recourse approach, collectively known as *stochastic programming*, was developed by George Dantzig in 1955 [67], not long after the early days of linear programming. In the approach, the first stage variables can be determined without taking the uncertainty into account, i.e. they are deterministic. The second stage variables are uncertain, with their values in either a finite set or a distribution of possibilities. An example in the telecommunication field would be dimensioning router and link capacity given uncertain traffic estimates. In this scenario, the first stage variables would include hardware placement and dimensioning, which occurs based on a given traffic amount, while the second stage variables would be the uncertain traffic flow. Therefore, this approach can be used to make optimal choices for the first stage variables by incorporating the outcome of uncertainty of the second stage variables.

Given variables $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ for the first and second stage respectively, a general linear two-stage recourse problem can be stated as follows:

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + E[Q(\boldsymbol{x}, \xi)] && (3.32)\\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}, && (3.33)\\ & \boldsymbol{x} \geq \boldsymbol{0}, && (3.34) \end{aligned}$$

with $\xi \in \Xi$ the uncertain event and $Q(\boldsymbol{x}, \xi)$, known as the *recourse function*, denoting the optimal value of the second-stage problem. Given functions $q : \Xi \to \mathbb{R}^m$, $W : \Xi \to \mathbb{R}^{q \times m}$, $T : \Xi \to \mathbb{R}^{q \times n}$, and $h : \Xi \to \mathbb{R}^q$, we can write the second-stage problem as follows:

$$\begin{aligned} \min \quad & q(\xi)^{\mathrm{T}}\boldsymbol{y} && (3.35)\\ \text{s.t.} \quad & W(\xi)\boldsymbol{y} = h(\xi) - T(\xi)\boldsymbol{x}, && (3.36)\\ & \boldsymbol{y} \geq \boldsymbol{0}. && (3.37) \end{aligned}$$

In this second-stage problem, we can describe $W(\xi)\boldsymbol{y}$ as a term that compensates for potential inconsistency in the constraint given by $T(\xi)\boldsymbol{x} \leq h(\xi)$. The objective function denotes the cost of the *recourse action*.

The greatest problem in the two-stage recourse approach is determining the expected value of the recourse function, as it is non-linear. Often, however, the uncertain event $\xi$ can be expressed as a finite number of potential realisations, known as *scenarios*, each with a realisation probability. This allows us to approximate the expected value using a linear combination function, by *discretising* the uncertainty. Given a set $\mathrm{S} = \{1, 2, \ldots, S\}$ and scenarios $\xi_1, \xi_2, \ldots, \xi_S$ with probabilities $p_1, p_2, \ldots, p_S$, the expected value in the first-stage problem can be replaced with the following:

$$E[Q(\boldsymbol{x}, \xi)] = \sum_{s \in \mathrm{S}} p_s Q(\boldsymbol{x}, \xi_s). \tag{3.38}$$

This allows us to define an extended equivalent LP, known as the *deterministic equivalent*:

$$\begin{aligned}
\min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \sum_{s \in \mathrm{S}} q_s^{\mathrm{T}} \boldsymbol{y}_s && (3.39)\\
\text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}, && (3.40)\\
& W_s \boldsymbol{y}_s = \boldsymbol{h}_s - T_s \boldsymbol{x}, \quad \forall s \in \mathrm{S}, && (3.41)\\
& \boldsymbol{y}_s \geq \boldsymbol{0}, \quad \forall s \in \mathrm{S}, && (3.42)\\
& \boldsymbol{x} \geq \boldsymbol{0}. && (3.43)
\end{aligned}$$

### 3.3.2 Robust optimisation

In practice, the probability distribution functions for the uncertain variables are very difficult to obtain, if at all possible. Furthermore, the discretised version may result in an infeasible number of scenarios if the required granularity of uncertainty is very fine. To address these issues, the concept of *robust optimisation* is introduced, of which the first well-known formulation was proposed in 1973 by Soyster [68]. The general formulation can be given as follows [66]:

$$\begin{aligned}
\min \quad & f(\boldsymbol{x}) && (3.44)\\
\text{s.t.} \quad & g_i(\boldsymbol{x}, \boldsymbol{u}_i) \leq 0, \quad \forall \boldsymbol{u}_i \in \mathcal{U}_i,\ i = 1, 2, \ldots, n, && (3.45)
\end{aligned}$$

with decision variables $\boldsymbol{x} \in \mathbb{R}^n$, functions $f, g_i : \mathbb{R}^n \to \mathbb{R}$ and uncertainty parameters $\boldsymbol{u}_i \in \mathbb{R}^k$, which take on values in the closed uncertainty sets $\mathcal{U}_i \subseteq \mathbb{R}^k$.

Next we need to define the functions $g_i$. The formulation in the paper by Soyster was done *column-wise* as follows:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{3.46}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \boldsymbol{a}_i x_i \leq \boldsymbol{b}, \quad \forall \boldsymbol{a}_i \in \mathcal{U}_i,\ i = 1, 2, \ldots, n, \tag{3.47}$$

which is equivalent to:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{3.48}$$

$$\text{s.t.} \quad \tilde{A}\boldsymbol{x} \leq \boldsymbol{b}, \tag{3.49}$$

with the entries in $\tilde{A}$ given by $\tilde{a}_{ij} = \sup_{\boldsymbol{a}_i \in \mathcal{U}_i}(a_{ij})$. Note that this forces each variable to take on the worst-case value for each uncertainty set, making it overly conservative.

Ben-Tal and Nemirovsky provided a less conservative approach by formulating the uncertainty information *constraint-wise* [69]. Assuming the uncertainty sets $\mathcal{U}_i$ are convex and closed, this approach leads to the following program, called the *Robust Counterpart (RC)*:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{3.50}$$

$$\text{s.t.} \quad \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i, \quad \forall \boldsymbol{a}_i \in \mathcal{U}_i,\ i = 1, 2, \ldots, n, \tag{3.51}$$

which may be rewritten as:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{3.52}$$

$$\text{s.t.} \quad \max_{\boldsymbol{a}_i \in \mathcal{U}} \{\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}\} \leq b_i, \quad i = 1, 2, \ldots, n, \tag{3.53}$$

with $\mathcal{U}$ the direct product of the partial uncertainty sets $\mathcal{U}_i$, or $\mathcal{U} = \prod_{i=1}^{n} \mathcal{U}_i$. If the uncertainty set $\mathcal{U}$ is ellipsoidal, the RC can be converted to a conic quadratic program. Conversely, if it is a polyhedron, the RC reduces to a linear programming problem [69]. Therefore, by letting $\mathcal{U}_i = \{\boldsymbol{a}_i : G_i\boldsymbol{a}_i \leq \boldsymbol{g}_i\}$, we can write the RC as follows:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{3.54}$$

$$\text{s.t.} \quad \max\{\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} : G_i\boldsymbol{a}_i \leq \boldsymbol{g}_i\} \leq b_i, \quad i = 1, 2, \ldots, n. \tag{3.55}$$

Next, consider the inner maximisation problem for a fixed value of $\boldsymbol{x}$, denoted by $\boldsymbol{x}^*$. By associating variables $\boldsymbol{\zeta}_i$ with the polyhedron constraints $G_i\boldsymbol{a}_i \leq \boldsymbol{g}_i$, we can write the dual problem as follows:

$$\begin{aligned} \min \quad & \boldsymbol{\zeta}_i^{\mathrm{T}}\boldsymbol{g}_i && (3.56)\\ \text{s.t.} \quad & \boldsymbol{\zeta}_i G_i = \boldsymbol{x}^*, && (3.57)\\ & \boldsymbol{\zeta}_i \geq \boldsymbol{0}. && (3.58) \end{aligned}$$

Then, by substituting the dual into the RC formulation (3.54)–(3.55), and unfixing $\boldsymbol{x}^*$, we find a new single linear minimisation problem:

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} && (3.59)\\ \text{s.t.} \quad & \boldsymbol{\zeta}_i\boldsymbol{g}_i \leq b_i, \quad i = 1, 2, \ldots, n, && (3.60)\\ & \boldsymbol{\zeta}_i G_i = \boldsymbol{x}, \quad i = 1, 2, \ldots, n, && (3.61)\\ & \boldsymbol{\zeta}_i \geq \boldsymbol{0}, \quad i = 1, 2, \ldots, n. && (3.62) \end{aligned}$$

For a two-stage robust version of the problem, consider the following two-stage stochastic problem with uncertainty $\xi \in \Xi$ consisting of the elements $A$, $D$ and $\boldsymbol{b}$ [70]:

$$\min_{\boldsymbol{x},\boldsymbol{y}} \{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} : A\boldsymbol{x} + D\boldsymbol{y} \leq \boldsymbol{b}\}_{\xi=[A,D,\boldsymbol{b}]\in\Xi}. \tag{3.63}$$

Then the RC can be written as follows [71]:

$$\min_{\boldsymbol{x}} \{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} : \exists\, \boldsymbol{y}\, \forall \xi = [A, D, \boldsymbol{b}] \in \Xi : A\boldsymbol{x} + D\boldsymbol{y} \leq \boldsymbol{b}\}. \tag{3.64}$$

On the other hand, the Adjustable Robust Counterpart (ARC) is defined as:

$$\min_{\boldsymbol{x}} \{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} : \forall \xi = [A, D, \boldsymbol{b}] \in \Xi\, \exists\, \boldsymbol{y} : A\boldsymbol{x} + D\boldsymbol{y} \leq \boldsymbol{b}\}. \tag{3.65}$$

Unlike the standard RC, the ARC allows for setting variables based on the outcome of a specific realisation scenario, which results in a more flexible problem, i.e. a problem with a larger set of robust feasible solutions, ensuring an improved optimal value while still being feasible for all possible variable realisations.

## 3.4 Network optimisation

As the number of problems modelled using these techniques has increased, a few generalised formulations emerged, one or more of which could be seen in most problems. Specifically in combinatorial optimisation, which can be defined as the subset of problems that has a finite discrete search space, a number of problems are of interest when modelling a PON.

### 3.4.1 Assignment problems

An assignment problem is a problem where all variables are *decision variables*, i.e. binary variables, taking on a value of either 0 or 1. Therefore, it can be described as a problem where each decision has an incurred cost and profit, while operating under some budget constraint. In the general version of the assignment problem, a number of agents can each perform a number of tasks. Each task has a cost and profit, which may be agent specific and each agent has a budget that the sum of its assigned tasks cannot exceed. Let $\boldsymbol{x} \in \{0,1\}^{n\times m}$ be the set of decision variables, and $\boldsymbol{c} \in \mathbb{R}_+^{n\times m}$ and $\boldsymbol{p} \in \mathbb{R}_+^{n\times m}$ be the corresponding cost and profit. Let $\boldsymbol{t} \in \mathbb{R}_+^n$ be the budget of each agent. Then, the *generalised assignment problem* can be defined as follows:

$$\max \quad \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}x_{ij} \tag{3.66}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} c_{ij}x_{ij} \leq t_i, \quad \forall i = 1,2,\ldots,n, \tag{3.67}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \forall j = 1,2,\ldots,m. \tag{3.68}$$

The constraint set (3.67) ensure that each agent does not exceed its budget while (3.68) ensures a task is only assigned to one agent. Even though the formulation is basic, it is deceptively hard to solve, falling into the NP-hard class of problems.

A derivative of the generalised assignment problem, known as the *facility location problem*, is formulated by letting $\boldsymbol{x}$ be the assignment of demand points to facilities. Each

facility has an additional fixed cost $\boldsymbol{f} \in \mathbb{R}_+^n$ to set up, and the allocation of a demand point to a facility incurs a proportional cost, $\boldsymbol{c}$, relative to the distance between them. The problem then becomes the search of an assignment of demand points to facilities for the minimum overall cost [72]. Let $\boldsymbol{y} \in \{0,1\}^n$ be 1 if a facility is to be opened and 0 otherwise and let $p \in \mathbb{R}_+$ be the maximum number of opened facilities. Formulating the facility location problem can then be done as follows [73,74]:

$$\max \quad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{n} f_i y_i \tag{3.69}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ij} = 1, \quad \forall j = 1,2,\ldots,m, \tag{3.70}$$

$$\sum_{i=1}^{n} y_i \leq p, \tag{3.71}$$

$$x_{ij} \leq y_i, \quad \forall i = 1,2,\ldots,n,\ \forall j = 1,2,\ldots,m. \tag{3.72}$$

Constraints (3.70) and (3.71) ensure that a demand point is assigned to one facility and that only $p$ facilities may be opened respectively. The inequalities (3.72) open a facility when a demand point is assigned to it. Like the generalised assignment problem, the facility location problem is also proven to be NP-hard [75].

### 3.4.2 Graph problems

Since a number of real-world problems can be visualised as a connected network of elements, i.e. a graph, the solving of graph problems is a large field in optimisation. In this field, a directed graph, where each arc has a capacity and an assigned flow, is known as a *flow network*. This construction has a number of uses, including applications in routing problems, minimum spanning trees and maximum flow problems. Firstly, in a digraph $\mathcal{G}(\mathrm{V},\mathrm{A})$, with vertices V and arcs A, let $\boldsymbol{c} \in \mathbb{R}_+^{|\mathrm{A}|}$ be the arc capacities. Additionally, two vertices, a *source* $s \in \mathrm{V}$ and a *sink* $t \in \mathrm{V}$ are given. With $\boldsymbol{f} \in \mathbb{R}_+^{|\mathrm{A}|}$ the flow on the arcs, a flow network has two properties:

- **Capacity constraints** - The flow along a specific arc cannot exceed its capacity, i.e. $f_a \leq c_a, \forall a \in \mathrm{A}$.

- **Flow conservation** - For any vertex $u \in \mathrm{V} \setminus \{s, t\}$, the net flow out of a vertex is zero, i.e. $\sum_{v \in \mathrm{V}} f_{uv} = 0$.

Now, if we want the maximum flow between the source and the sink, which is abbreviated *max-flow*, we have the following problem:

$$\max \quad \nabla \tag{3.73}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(v)} f_a - \sum_{a \in \delta^-(v)} f_a = \begin{cases} \nabla, & v = s, \\ -\nabla, & v = t, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall v \in \mathrm{V}, \tag{3.74}$$

$$f_a \leq c_a, \quad \forall a \in \mathrm{A}, \tag{3.75}$$

with $\delta^-(v) : \mathrm{V} \to \mathrm{A}^m$ and $\delta^+(v) : \mathrm{V} \to \mathrm{A}^m$ functions giving all the incoming and outgoing arcs adjacent to vertex $v \in \mathrm{V}$. This problem can be efficiently solved, i.e. in polynomial time, using the Ford-Fulkerson [76] or push-relabel [77] algorithms.

We can generalise network flow problems even further by introducing a number of sources and sinks, known as *commodities*, with a minimum flow requirement or demand between each. This problem is known as the *multi-commodity flow problem* and it is encountered in most telecommunication routing and design problems. To define it, introduce a set of commodities K, with $K_i = (s_i, t_i, d_i)$ the commodity $i$ with source $s_i$, sink $t_i$ and demand $d_i$. Next, redefine $\boldsymbol{f} \in \mathbb{R}_+^{|\mathrm{A}| \times |\mathrm{K}|}$ as the flow of commodity $i \in \mathrm{K}$. Although a number of other variants exist, including maximising the total throughput and maximising concurrent flow, we will look at the minimum cost variant. By introducing a cost $\boldsymbol{h} \in R_+^{|\mathrm{A}|}$ for each unit of flow, it can be written as follows:

$$\min \quad \sum_{a \in \mathrm{A}} \sum_{k \in \mathrm{K}} h_a f_{ak} \tag{3.76}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(v)} f_{ak} - \sum_{a \in \delta^-(v)} f_{ak} = \begin{cases} d_k, & v = s_k, \\ -d_k, & v = t_k, \\ 0, & \text{otherwise}, \end{cases} \qquad \begin{matrix} \forall v \in \mathrm{V}, \\ \forall k \in \mathrm{K}, \end{matrix} \tag{3.77}$$

$$\sum_{k \in \mathrm{K}} f_{ak} \leq c_a, \quad \forall a \in \mathrm{A}. \tag{3.78}$$

Note that for this general formulation, the problem is *capacitated*, i.e. the flow on each arc is limited by some defined capacity $\boldsymbol{c}$. A variant of this problem, where (3.78) is left

out, is called the *uncapacitated* multi-commodity flow problem, and is more commonly used in physical network models. For fractional flows, both variants can be solved in polynomial time using the ellipsoid method [78], while for integer flows, where $\boldsymbol{f} \in \mathbb{Z}_+^{|\mathrm{A}|\times|\mathrm{K}|}$, they are proven to be NP-complete [79].

Another prominent graph problem is known as the *Steiner tree problem*, named after the Swiss mathematician Jakob Steiner. It is a combinatorial optimisation problem where we are tasked with finding a minimum weight subgraph that connects a set of given vertices, known as *Steiner terminals*, to a root, via a number of intermediate edges and vertices known as *Steiner points*. It is encountered in a number of applications, including Very-large-scale Integration (VLSI) circuit layout and communication network design [80]. Given the undirected graph $\mathcal{G}(\mathrm{V},\mathrm{E})$, with vertices V and edges E, introduce a set A of directed arcs obtained from $\mathcal{G}$ by bi-directing each edge of E, as well as binary variables $\boldsymbol{x} \in \{0,1\}^{|\mathrm{E}|}$ denoting the use of an edge in the resulting Steiner tree. Then, with edge costs $\boldsymbol{\ell} \in \mathbb{R}_+^{|\mathrm{E}|}$, root $r \in \mathrm{V}$ and Steiner terminals $\mathrm{T} \subseteq \mathrm{V}$, the Steiner tree problem can be formulated as follows [81]:

$$\min \quad \sum_{e\in\mathrm{E}} \ell_e x_e \tag{3.79}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(v)} f_{ak} - \sum_{a\in\delta^-(v)} f_{ak} = \begin{cases} 1, & v = r, \\ -1, & v = k, \\ 0, & \text{otherwise}, \end{cases} \quad \begin{matrix} \forall v \in \mathrm{V}, \\ \forall k \in \mathrm{T}\setminus\{r\}, \end{matrix} \tag{3.80}$$

$$f_{ijk} \le x_e, \quad \forall e = (i,j) \in \mathrm{E}, \forall k \in \mathrm{T}\setminus\{r\}. \tag{3.81}$$

Most derivatives of the Steiner tree problem are shown to be NP-complete [82] while the general Euclidean Steiner tree problem is NP-hard.

### 3.4.3 Connected Facility Location Problem

The *Connected Facility Location Problem (ConFL)* is a combination of the facility location and Steiner tree problems, first introduced independently by both Guha et al. [83] and Karget and Minkoff [84]. Like the facility location problem, demand points are assigned to open facilities, with proportional costs in relation to the distance between

them. However, the facilities are also connected to a central hub by means of a Steiner tree. Therefore, the central hub serves as the root with Steiner terminals at each open facility. This problem is closely related to designing metropolitan communication networks, where routing capabilities are often only contained in the core network.

Gupta et al. [85] defined ConFL formally as follows. Assume we are given a graph $\mathcal{G}(\mathrm{V}, \mathrm{E})$ with edge costs $\boldsymbol{c} \in \mathbb{R}_+^{|\mathrm{E}|}$, a set of demand points D with corresponding demand $\boldsymbol{d} \in \mathbb{R}_+^{|\mathrm{D}|}$, and facility opening costs $\boldsymbol{f} \in \mathbb{R}_+^{|\mathrm{V}|}$. The objective is to open a set of facilities $\mathrm{F} \subseteq \mathrm{V}$, to assign an open facility $\sigma(j)$ to each demand point $j \in \mathrm{D}$, and to connect all vertices in F with a Steiner tree $T$ rooted at $r \in \mathrm{V}$ so as to minimise:

$$\sum_{i \in \mathrm{F}} f_i + \sum_{j \in \mathrm{V}} d_j c_{\sigma(j),j} + M \sum_{e \in \mathrm{T}} c_e, \tag{3.82}$$

where $c_{uv}$ denotes the length of the shortest path between vertices $u$ and $v$ in V, and $M \geq 1$ is a parameter that scales the cost of the Steiner tree in the core.

Finally, since ConFL requires solving both a facility location and a minimum Steiner tree problem, both of which are NP-hard, it is also NP-hard [85].

## 3.5 Passive Optical Network Design Problem

Now that the constituent problems have been given, we can look at what we call the *Passive Optical Network Design Problem (PONDP)*. As the name implies, it is the problem of designing a PON, including placement of passive splitters, fibres and, since we focus on subterranean fibre, trenches, as detailed in chapter 2. Conceptually, it is similar to ConFL, except that for each opened facility, a Steiner tree is also required to connect its assigned demand points, i.e. the optical fibre trenches. Furthermore, it incorporates a multi-commodity integer flow problem, which determines the number of fibres in each trench. For PONDP, the passive splitters are facilities, ONUs are the demand points and the CO is the central hub. We can reduce the problem to solving three sub-problems simultaneously [61]:

1. Determining the optimal number of opened facilities.
2. Optimal allocation of demand points to facilities.
3. Relocation, reallocation and reconnection of facilities for optimal cost.

Firstly, the optimal number of facilities cannot be determined in advance. Therefore, all combinations should be enumerated in the first sub-problem, which, for a brute-force search, increases computational complexity factorially in relation to the number of potential splitter locations. Given the facility locations, the second sub-problem forms an NP-complete assignment problem [61], while the third sub-problem, which determines facility locations so as to minimise the cost of installing fibre, is known as the Fermat-Weber point problem [86]. Additionally, the third problem also requires optimal trench placement, which entails a number of minimum Steiner tree problems, one rooted at the CO with Steiner terminals at each passive splitter, as well as Steiner trees rooted at each splitter with Steiner terminals at each assigned ONU.

Since solving PONDP entails simultaneously solving a facility location, a multi-commodity integer flow and a number of Steiner tree problems, all of which are NP-hard, PONDP is an NP-hard problem. A formal definition of the problem will be given in the following chapter.

## 3.6 Related work

Over the past few years, a number of authors have studied PONDP and simpler variants of it, including ConFL. In this section, we will give an overview of these works, divided into three categories: heuristics and approximation algorithms, meta-heuristics, and exact methods.

### 3.6.1 Heuristics and approximation algorithms

In [87], Khan proposes a 2-approximation greedy algorithm based on graph clustering techniques using population density functions to solve PONDP. Running in worst case $O(n^2 + m^2)$ time, it provides a 45–65 % deployment cost saving in comparison to the PNLA heuristic proposed in [88].

In [61], Li and Shen introduce a heuristic known as the Recursive Association and Relocation algorithm (RARA) to solve greenfield PONs, which uses random search space sampling and an incremental improvement step similar to SA. It is one of the only papers that incorporates network constraints such as network diameter and differential distance limits into the model, which accounts for attenuation effects. The algorithm is compared to a simple random-cut heuristic to show its efficacy.

The authors in [89] investigate PONDP by implementing a detailed Capital Expenditure (CAPEX) model of the problem that includes non-linear asymmetric fibre costs for the distribution and feeder networks. A greedy heuristic known as the Branch Contracting Algorithm (BCA) is proposed which assigns nearest neighbour splitters to ONUs and computes a Steiner tree in the feeder network. Results are compared to an ILP lower bound for very large graphs (>10,000 nodes) and the authors claim a 10–20 % optimality gap within a few minutes. This result is challenged in [90] however, where the reimplemented algorithm only manages to produce solutions with optimality gaps of 35–44 % using real-world instances. The authors of [89] go on to generalise BCA for Active Ethernet (AE), Very-high-bit-rate Digital Subscriber Line (VDSL) and P2P optical networks in [91].

In [92], the authors compare Steiner tree and A$^*$ heuristics, minimising the trenching and fibre length respectively, in terms of efficacy to connect almost 30,000 households with both P2P and PON technologies. These heuristics are run after a k-means clustering step, in which 1:32 splitters are assumed. Results show that the Steiner tree heuristic and PON architecture provides the lowest deployment cost solution.

Kantarci and Mouftah [93, 94] provide a planning heuristic known as Locate-ONU-

with-Lowest-Availability-Requirement-First (LOWLARF) to solve the survivable variant of PONDP systematically. Three different availability types are considered, including the G.984 Type A, Type B and a hybrid construction of the two for partial redundancy. Since the authors minimise fibre length, results show comparable OLT to ONU distances between LOWLARF and a corresponding MILP model.

In [95], Shi et al. propose search heuristics based on user profile requirements to provide an energy efficient solution for a derivative of PON known as Long-Reach Passive Optical Network (LR-PON), in both tree-and-branch and ring-and-spur topologies. Additionally, a behaviour-aware wavelength assignment strategy is introduced, providing low optimality gaps in comparison to a simple lower bound in terms of the number of wavelengths required.

A number of authors also studied the multi-level PONDP. Kim et al. [96] formulate both single- and dual-level PONs as MILP models, allowing for different splitter types as well as different fibre types. A linearisation of the stepwise fibre cost is done, providing a lower bound relaxation of the model. Then, a heuristic algorithm is utilised which, for a feasible solution, moves splitters towards the CO for maximum cost reduction (upward improvement), and then goes back down the tree to ensure feasibility (downward refinement), resulting in optimality gaps of only 1.3 % on average.

In [97], the authors investigate the multi-level PONDP by considering the remaining split ratio at each level. A heuristic based on cascaded pruning and reallocation is introduced to solve dual-level networks, providing optimality gaps of only 7 % with respect to the optimal solution computed using an ILP model. Results also show that moving from a single- to a multi-level PON can produce CAPEX savings of up to 15 %.

Agata and Nishimura [98] introduce a heuristic algorithm based on k-means clustering to solve a dual-level PON, in which the clusters are connected via a sub-optimal Steiner tree computed using all shortest paths. The Steiner tree heuristic runs in $O(t^3)$ time, with $t$ terminals, instead of the $O(3^t)$ of the exact Dreyfus-Wagner algorithm they compared it with. Then, the authors implemented the algorithm into a design tool with

area-based demand capability.

Swamy and Kumar [99] introduced an 8.55 approximation algorithm for ConFL shortly after its definition, utilising a primal-dual algorithm. The algorithm clusters demand into groups of $M$, before connecting them via a Steiner tree. This improves upon the previously best-known 10.66 [85] and 10.1 approximations [100].

### 3.6.2 Meta-heuristics

In one of the earliest works on PONDP, Bonsma et al. [101] propose an Evolutionary Algorithm (EA) with a novel genetic encoding strategy to avoid sub-optimal convergence. The algorithm uses neutral encoding, heuristically guided evolution as well as a number of different greedy staged optimisation approaches.

A number of multi-level PON meta-heuristics also exist. Poon et al. [102] present a genetic algorithm but use a hybrid approach alongside Minimum Spanning Tree (MST) and clustering steps to improve the solution of a multi-level PON. In contrast to the minimum cost approach most authors take, the algorithm minimises the number of splitters and the total length of fibre. The authors of [103] also propose a genetic algorithm for the multi-level variant, but incorporate a planning heuristic based on min-max distance clustering. Finally, in [104], Kokangul and Ari propose a genetic algorithm to optimise the placement of primary and secondary nodes in a dual-level PON while taking into account different splitter types and attenuation. Results are compared to an ILP, where the GA found the optimal solution for a small real-world deployment in Turkey.

In [105], the authors present an evolutionary algorithm which avoids non-traversable obstacles by means of convex hull mapping. The problem is subdivided using the k-means clustering algorithm and compared to a manual plan in terms of total fibre length and cost, showing a noticeable improvement.

The authors of [106] utilise a genetic algorithm to optimise the position of splitters for

variants of PONDP, including ring, tree and bus topologies. Power budget margins and OLT to ONU distances are compared for small test cases, showing similar results.

Finally, both Chu et al. [107] and Xiong et al. [108] propose ant colony optimisation approaches to PONDP. Apart from incorporating a local search step in their algorithm, Chu et al. also investigates a number of approaches to select drop closure nodes, with a weight-based selection strategy showing the best results. Xiong et al. incorporate a Zone Location partitioning technique in an attempt to avoid local optima, as well as a 0-1 non-linear programming model to optimise network parameters.

### 3.6.3 Exact methods

A number of exact methods have been developed for PONDP, starting with Ouali and Poon [109] and Poon and Ouali [110], who propose standard and growth-aware path-based ILP models to solve deployments with up to 1033 households. In [111], the standard path-based ILP model is extended to account for WDM-PONs, with wavelength allocation and splitter type design capabilities. The resulting model does not include any paths or edges, instead relying on simple variables representing connections between equipment. In addition, rudimentary attenuation effects are integrated by estimating optical fibre loss, and the authors use a cross-layer planning scheme to provision traffic flows.

Grötschel et al. [112] provide a comprehensive paper on deploying PONs, including potential ILP modelling techniques to introduce survivability, operational constraints and path length restrictions. It is noted in the paper that according to the authors, path length constraints in connectivity problems have not yet been studied. Multi-level networks with distributed splitting as well as time-dependent networks are discussed. Finally, it also looks at optimising for a specific coverage.

*K*-level and unconstrained multi-level PONs are studied by Hervet and Chardy [113] and Gouveia et al. [114] respectively. Hervet and Chardy focus on operational constraints, proposing a preprocessing step to group households together for similar

Quality of Service (QoS) per building. Gouveia et al. introduce a number of discretised and disaggregated ILP formulations for a brownfield network, where ducts are assumed to already exist.

Kantarci and Mouftah [115] propose a number of MILP models for Type A, Type B and hybrid Type C survivability configurations using LR-PON technology, solved using CPLEX. The paper shows the efficacy of Type C survivability, showing greatly reduced average ONU service unavailability for a hypothetical small dataset.

In [116], Hervet et al. propose a two-stage robust formulation of the *K*-level PONDP, utilising a column-and-constraint generation approach to generate scenarios on the fly. The uncertainty set is bounded by a parameter denoting the maximum number of clients that will utilise the service in the studied area. All fibres are installed in the first stage while splitters are only installed once the demand is known. The model then minimises the worst-case future splitter cost. Preliminary experimental results use 2,000 randomly generated scenarios with data sets containing between 5 and 10 nodes, displaying the enormous increase in complexity when uncertainty is included.

Bley et al. propose a Lagrangian decomposition approach to solving PONDP in [117], splitting the problem based on two strategies, one based on network structure and one on cost structure. For the network structure decomposition, they propose feeder and distribution models, while for the cost structure, fixed cost and variable cost models. A number of valid inequalities are introduced, known as *connectivity cuts*. These are separated and the convex Lagrange multiplier problem is solved using a bundle subgradient method. Experimental results on real-world large instances with up to 3,862 households show very impressive results with low average optimality gaps.

A number of authors have also studied ConFL and its derivatives, especially in terms of theoretical analysis. Gollowitzer and Ljubić [118] summarise a large number of formulations for ConFL, including both cut-set and flow variants. A polyhedral and computational study on each formulation complete the paper, with the cut-set formulation of Ljubić [119] exhibiting the best computational performance.

In [120], Arulselvan et al. provide a cut formulation for the multi-period incremental ConFL problem, along with a number of cover and cut-set inequalities to improve the relaxation lower bound. The model is then solved using standard branch-and-cut and provides good feasible solutions for smaller instances.

In [121], the authors study the capacitated ConFL problem, developing an ILP model based on single-commodity flows. A number of cut-set and cover inequalities are derived for the problem and separated in a branch-and-cut framework, showing their efficacy.

Ljubić and Gollowitzer introduce two formulations for the hop constrained ConFL problem in [122]: a cut formulation on layered graphs and a Steiner arborescence on layered graphs formulation. A polyhedral study compares the quality of the relaxation lower bounds while a computational study shows optimal solutions for instances up to 1,300 nodes and more than 100,000 edges.

In [123], Leitner et al. study a two-architecture ConFL, i.e. two different FTTx architectures in one design, and gives a cut formulation ILP. Two classes of valid inequalities, named *zl-* and *z-cuts*, are derived, separated and experimentally tested on generated instances.

Bley et al. introduce the survivable hop constrained ConFL problem in [124], providing a multi-commodity flow as well as a hop level multi-commodity flow formulation. Benders decomposition is used to project out the extended flow variables and the approach is experimentally tested on a number of small generated instances with up to 40 facilities.

In [125], Chimani et al. introduce the survivable 2-interconnected ConFL problem, which requires the core network to be 2-connected. The authors provide an ILP model and characterise the problem, showing it is both NP-hard to solve and NP-hard to determine the existence of a feasible solution. Computational results using branch-and-cut suggest efficient computation of solutions for instances with up to 400 nodes.

### 3.6.4 Observations on related work

A number of observations can be made regarding the approaches followed by authors in the field. Firstly, work on both heuristics and meta-heuristics tend to consider a greater number of complex phenomena, including designing networks with multiple splitting stages, attenuation effects or non-linear optical fibre costs. This is due to the fact that they are much easier to integrate into a heuristic environment, as they can be dealt with in an algorithmic manner. Furthermore, in terms of the modelling accuracy vs. computational tractability problem, heuristics show much better computational performance, allowing complexity to be increased without leading to unreasonable computation times. It does come at a price however, producing solutions with no quality guarantee, which is exacerbated by the fact that most papers only compare their results with other heuristics. Where available, optimality gaps tend to exceed 10 % in best-case test environments.

In the exact environment, the opposite is true, with authors tending to include only the essential factors, instead focussing on improving computational performance as-is. Also evident is the apparent lack of proper attenuation integration, with estimations being used in the cases where they are included. For ConFL, hop constraints are utilised to try and estimate attenuation effects, but this is only accurate when all arcs are of equal or comparable length. Where path length constraints are mentioned, they are specified explicitly using global length limits on path formulations. The incorporation of path length constraints dependent on splitter types were absent from the literature, and it is also worthwhile to investigate the efficacy of using a path length constraint generation approach.

Only two papers could be found utilising a decomposition method to solve PONDP, with Benders decomposition only used in the paper by Bley et al. [124] for ConFL to separate additional hop-constrained flow from the standard flow. Therefore, at the time of writing, no research has been done on Benders decomposition for PONDP, which would potentially increase performance and allow for more complexity to be added, resulting in a more relevant and accurate formulation.

Finally, we observe that only a single paper by Hervet et al. [116] addresses demand uncertainty, using robust optimisation techniques to specify an uncertainty set bounded by an expected service usage parameter. This suggests that more work on the topic is required, including explicitly incorporating scenarios utilising the two-stage stochastic programming approach, which have not been studied.

## 3.7 Conclusion

In this chapter, basic concepts surrounding modelling and optimisation techniques were detailed. In terms of modelling, linear, integer and mixed integer models were defined and discussed, along with duality theory and the different applicable complexity classes. Next, solution methods discussed included exact, heuristic and meta-heuristic techniques.

Since exact techniques are able to quantify solution quality, in particular using the optimality gap of the branch-and-bound algorithm, they are superior to heuristics given that their computation time is reasonable. However, heuristics are able to provide solutions quickly, and can therefore be utilised as solution generators in exact algorithms to improve the upper bound.

Next, Benders decomposition was defined as a method to improve computational performance by projecting out parts of a problem and solving them independently. These sub-problems are then used to reincorporate the projected parts back into the original problem. Methods shown to handle uncertain variables included stochastic programming and robust optimisation, both of which will be investigated in a subsequent chapter.

The Passive Optical Network Design Problem (PONDP) was defined, following an overview of its constituent problems, including the facility location, multi-commodity flow and Steiner tree problems. Finally, related work on PONDP and ConFL were outlined and critically evaluated to show three major areas in which contributions can

be made: splitter type dependent path length constraints, Benders decomposition and stochastic programming. In the next chapter, we will look at modelling and decomposing the standard deterministic PONDP.

# Chapter 4

# Mathematical model

---

*This chapter details the approach used to model and decompose PONDP. Two common models, one based on arc-flow and one on paths, are discussed along with their respective shortcomings in terms of scalability. The path formulation is then decomposed using Benders and improved using a column generation approach. All formulations are extended to accommodate path length constraints, verified, validated and tested, showing favourable scalability performance for the decomposed model.*

---

## 4.1 Design motivation

Now that the relevant background has been described, models incorporating the ITU-T G.984 [11] specifications can be designed using the modelling techniques described. First, however, we look at some model considerations present during the entire process.

### 4.1.1 Model considerations

PONDP is modelled in an exact framework, capturing the topology of a PON as described in the ITU-T G.984 [11] specification. As the model contains decision variables, e.g. if a splitter should be installed, if a trench should be dug etc., it will necessarily have integer variables, and in particular, binary, or *0-1 variables*. Therefore, the model falls into the ILP class. Where real variables are used in conjunction with integer variables, the model is classified as a MILP model.

Whenever we model the number of fibres, we should distinguish between the concepts of *splittable* and *unsplittable* flow. If we route a splittable flow between two points, it may follow a number of different paths to its destination, i.e. it can be split up into a number of smaller flows before being routed. An example of this type of flow is the transportation of dirt, which may be split into smaller batches and transported separately along different routes. Routing data packets across telecommunication networks follows a similar approach. Conversely, if a flow is unsplittable, all flow must follow a single path to the destination. A large package for instance, can not be divided into parts, and must be shipped as a single unit along a single route. In the case of PONDP, the flow consists of optical fibre cable, which can not be divided into parts. Hence, for PONDP, all flow is classified as unsplittable.

The objective function of the model is formulated to minimise total deployment cost. A number of models in literature minimise fibre length [61], maximum reach [115] or the number of used splitters [102], but a cost-based approach is more versatile, modelling the interaction between costs more effectively. Furthermore, service providers looking to deploy PONs are interested in minimising their CAPEX in an effort to maximise their revenue, making a cost-based approach more attractive due to the fact that the objective function correlates directly to the real world.

### 4.1.2 Model complexity

To limit modelling complexity, a number of assumptions are made concerning the physical effects that are modelled.

- **ONU connectivity** - In this chapter, we assume that all ONUs are to be connected, i.e. all ONUs have non-zero demand.
- **Single CO** - The number of COs are limited to reduce the resulting modelling complexity. Additionally, the network diameters under investigation are low enough to be covered by a single CO in the real world.
- **Constant fibre and trench costs** - The model assumes constant costs per length for fibre deployment as well as trenching and duct installation. Therefore, no economies of scale effects are included in the model and only linear link costs are considered.
- **Constant equipment costs** - As above, assume constant deployment costs for ONUs, splitters and OLTs, avoiding all economies of scale effects.
- **Single equipment type** - For this chapter, we consider only one type of splitter, ONU and OLT. This reduces the number of possible equipment configurations, simplifying the model and reducing its complexity. However, since multiple splitter types are important in real-world networks, this will be investigated in chapter 7.
- **Trenching limitations** - We assume that all civil trenching limitations are already taken into account in the input data, though models exist that take physical locations and obstacle avoidance into account (see [105]).
- **Port and micro-duct allocation** - Since the model will only consider essential topology related effects, no decisions will be made in terms of port- and micro-duct allocation. Since this will not effect the cost of the deployment to any significant extent, any such decisions can be made subsequent to the model being solved.

- **Physical attenuation** - Due to the fact that path length constraints are required to model attenuation-like effects, they are almost entirely absent from literature (see [61, 112]). Including path length constraints in classical flow models is difficult and often results in non-exploitable forms, invalidating common reduction techniques and algorithms. It is also common for model designers to consider only the *maximum* theoretical distances, e.g. 60 km for GPON, discarding the effect as inconsequential. However, this figure is *greatly* reduced for splitters with larger split ratios or when lower power optics are used. Since attenuation is so strongly linked to the practical feasibility of the resulting network design, it is imperative to include it in the models contained in this thesis. Nevertheless, for this comparative section of the thesis, we will follow the common approach of using the maximum network reach, before studying the physical attenuation effects properly in chapter 7.

## 4.2 Common models

PONDP is modelled in literature using one of two approaches; one based on arc flow and another based on paths. In this section both of these are defined and analysed according to their respective advantages and disadvantages.

As input, we are given an undirected graph $\mathcal{G}(\mathrm{V}, \mathrm{E})$, with vertices V, edges E, and two edge costs $\boldsymbol{c}^{\mathrm{TR}} \in \mathbb{Q}_+^{|\mathrm{E}|}$ and $\boldsymbol{c}^{\mathrm{F}} \in \mathbb{Q}_+^{|\mathrm{E}|}$. $c_e^{\mathrm{TR}}$ is the cost to dig a trench and install a duct while $c_e^{\mathrm{F}}$ is the cost to install fibre in a duct along edge $e \in \mathrm{E}$. Two disjunct subsets, $\mathrm{U} \subset \mathrm{V}$ and $\mathrm{D} \subset \mathrm{V}$, define the vertices that are marked as ONU and splitter locations respectively. Furthermore, we are given a CO location $q \in \mathrm{V}$, an OLT cost $c^{\mathrm{OLT}} \in \mathbb{Q}^+$, splitter cost $c^{\mathrm{SP}} \in \mathbb{Q}^+$ and a fixed ONU deployment cost $c^{\mathrm{ONU}} \in \mathbb{Q}^+$. Let $\kappa \in \mathbb{N}^+$ be the maximum number of ONUs that can connect to any splitter and $\boldsymbol{d} \in \mathbb{N}_+^{|\mathrm{U}|}$ be the demand at each ONU location.

Define a set K of all commodities, i.e. all possible combinations between sources and sinks. For PONDP, this equates to all combinations of splitters and the CO, as well as

all combinations between splitters and ONUs. The index $k_{ij}^{\mathrm{D}} \in \mathrm{K}$ refers to the specific commodity between splitter $i \in \mathrm{D}$ and ONU $j \in \mathrm{U}$, while index $k_i^{\mathrm{F}} \in \mathrm{K}$ refers to the commodity between splitter $i \in \mathrm{D}$ and the CO.

For ease of notation, define the subset $\mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(i) \subset \mathrm{K}$, containing all commodity pairs in the distribution network, i.e. between splitters and ONUs, that contain element $i \in \mathrm{U} \cup \mathrm{D}$. Similarly, define $\mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}(i) \subset \mathrm{K}$ for the feeder network, i.e. between splitters and the CO, with $i \in \{q\} \cup \mathrm{D}$.

The variables $\boldsymbol{x}^{\mathrm{TR}} \in \{0,1\}^{|\mathrm{E}|}$ are used to indicate if a duct is installed along an edge. Therefore, if the variable $x_e^{\mathrm{TR}}$ takes on a value of 1, a trench should be dug along the corresponding edge $e \in \mathrm{E}$ and a duct installed. To indicate which splitters are installed, variables $\boldsymbol{\psi} \in \{0,1\}^{|\mathrm{D}|}$ are used, taking on a value of 1 if a splitter is used and 0 otherwise. A parameter $\Delta \gg 0$ is used as a conceptual upper bound for *indicator* (also known as *big-M*) constraints. To avoid unnecessary slackness and numerical instability, this upper bound is always chosen as small as possible in implementations and is often calculated for every indicator constraint separately.

### 4.2.1 Arc model

The use of directed arcs to model networking design problems is very common in literature as we saw in chapter 3. The biggest advantage of this approach is the tighter LP relaxation bound, improving computational performance. Since performance is one of the main goals, this modelling approach will now be discussed.

For PONDP, the undirected graph $\mathcal{G}$ can be transformed for use with arc flow by substituting two directed edges $(ij)$ and $(ji)$ for every undirected edge $e \in \mathrm{E}$. Let A be the set of all substituted directed edges. Usually, the amount of flow across arcs are indicated by variables $\boldsymbol{f} \in \mathbb{R}_+^{|\mathrm{A}|}$, with $f_{ij}$ indicating the flow across arc $(ij)$. However, since we need the consider the physical network limitations, path lengths are required. To compute path lengths using arc flow, an additional subscript is required for every flow

variable. Therefore, redefine the arc flow variables as $\boldsymbol{f} \in \mathbb{R}_+^{|\mathrm{A}|\times|\mathrm{K}|}$, with $f_{ijk}$ indicating the flow across arc $(ij)$ for the commodity $k \in \mathrm{K}$. Additionally, introduce trench usage variables $\boldsymbol{x}^{\mathrm{TR}} \in \{0,1\}^{|\mathrm{E}|\times|\mathrm{K}|}$. This allows us to sum over all arcs containing flow for a specific commodity to calculate path lengths.

Let $c_{ij}^{\mathrm{F}} = c_{ji}^{\mathrm{F}} = c_e^{\mathrm{F}}, \forall\,(ij) = e \in \mathrm{E}$ be the fibre cost for every arc and $\boldsymbol{N} \in \mathbb{Z}_+^{|\mathrm{D}|}$ the number of ONUs connected to every splitter $i \in \mathrm{D}$. Given a vertex, the function $\phi : \mathrm{V} \to \mathrm{V}^n$ outputs a set of adjacent vertices. The arc model, henceforth known as *ARC*, can now be defined as follows:

*ARC*:

$$\min \quad c^{\mathrm{OLT}} + |\mathrm{U}|c^{\mathrm{ONU}} + \sum_{i\in\mathrm{D}} \psi_i c^{\mathrm{SP}} + \sum_{e\in\mathrm{E}} x_e^{\mathrm{TR}} c_e^{\mathrm{TR}} + \sum_{k\in\mathrm{K}} \sum_{(ij)\in\mathrm{A}} f_{ijk} c_{ij}^{F} \tag{4.1}$$

$$\text{s.t.} \quad \sum_{k\in\mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}} \sum_{j\in\phi(i)} f_{ijk} - \sum_{k\in\mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}} \sum_{j\in\phi(i)} f_{jik} = \begin{cases} -d_i, & i \in \mathrm{U}, \\ N_i, & i \in \mathrm{D}, \\ 0, & \text{otherwise}, \end{cases} \quad \forall i \in \mathrm{V}, \tag{4.2}$$

$$\sum_{k\in\mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}} \sum_{j\in\phi(i)} f_{ijk} - \sum_{k\in\mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}} \sum_{j\in\phi(i)} f_{jik} = \begin{cases} -\psi_i, & i \in \mathrm{D}, \\ \sum_{j\in\mathrm{D}} \psi_j, & i = q, \\ 0, & \text{otherwise}, \end{cases} \quad \forall i \in \mathrm{V}, \tag{4.3}$$

$$N_i \leq \kappa\psi_i, \quad \forall\, i \in \mathrm{D}, \tag{4.4}$$

$$f_{ijk} + f_{jik} \leq \Delta x_{ek}^{\mathrm{TR}}, \quad \forall\,(ij) = e \in \mathrm{E}, \forall\, k \in \mathrm{K}, \tag{4.5}$$

$$x_{ek}^{\mathrm{TR}} \leq x_e^{\mathrm{TR}}, \quad \forall\, e \in \mathrm{E}, \forall\, k \in \mathrm{K}. \tag{4.6}$$

The objective (4.1) minimises the sum of the deployment costs, including equipment (consisting of OLT, ONU and splitter costs), trenching, and fibre costs respectively. Equations (4.2) and (4.3) set the demand and calculate the supply for the distribution and feeder networks respectively, while also ensuring flow conservation. The constraint set (4.4) ensures that splitter capacity is not exceeded while (4.5) and (4.6) set the corresponding trenching variable if flow exists on the arcs traversing the edge.

To implement path length limits, we introduce variables $\boldsymbol{\ell}^{\min} \in \mathbb{R}_+^{|\mathrm{D}|}$ and $\boldsymbol{\ell}^{\max} \in \mathbb{R}_+^{|\mathrm{D}|}$ to calculate the minimum and maximum network reach through every splitter. Let

$\boldsymbol{\ell} \in \mathbb{Q}_+^{|\mathrm{E}|}$ be the length of each edge. Recall that we also require two parameters to model the physical PON constraints, the total network reach, $\ell_{\max}^{\text{total}} \in \mathbb{Q}^+$, and the maximum differential reach, $\ell_{\max}^{\text{diff}} \in \mathbb{Q}^+$. The constraints can then be written as follows:

$$\ell_i^{\min} \leq \Big( \sum_{e \in \mathrm{E}} x_{ek_i^{\mathrm{F}}}^{\mathrm{TR}} \ell_e + \sum_{e \in \mathrm{E}} x_{ek_{ij}^{\mathrm{D}}}^{\mathrm{TR}} \ell_e \Big), \qquad \forall\, i \in \mathrm{D}, \forall\, j \in \mathrm{U}, \tag{4.7}$$

$$\Big( \sum_{e \in \mathrm{E}} x_{ek_i^{\mathrm{F}}}^{\mathrm{TR}} \ell_e + \sum_{e \in \mathrm{E}} x_{ek_{ij}^{\mathrm{D}}}^{\mathrm{TR}} \ell_e \Big) \leq \ell_i^{\max}, \qquad \forall\, i \in \mathrm{D}, \forall\, j \in \mathrm{U}, \tag{4.8}$$

$$\ell_i^{\max} \leq \ell_{\max}^{\text{total}}, \qquad \forall\, i \in \mathrm{D}, \tag{4.9}$$

$$\ell_i^{\max} - \ell_i^{\min} \leq \ell_{\max}^{\text{diff}}, \qquad \forall\, i \in \mathrm{D}. \tag{4.10}$$

Inequalities (4.7) and (4.8) calculate the minimum and maximum network reach for every splitter respectively, summing the up- and downstream fibre lengths, while (4.9) and (4.10) limit the maximum total and differential network reach respectively.

The constraints (4.7)–(4.10) can simply be added to *ARC* to ensure practical feasibility of the PON. In terms of model complexity, the additional subscript requirement on the arc flow $\boldsymbol{f}$ dramatically increases the number of variables, potentially reducing the scalability of this modelling approach. Therefore, it is worthwhile to investigate alternatives, which may outperform *ARC* whenever path lengths are required, as is the case for PONDP.

### 4.2.2 Path model

Another common approach to network flow modelling is using paths instead of arc flow. In this approach, the paths are calculated prior to solving the model, greatly simplifying the model definition. Additionally, since we already have the paths and their corresponding lengths, physical constraints are trivial to model.

First, define the set P containing all possible simple paths through graph $\mathcal{G}$ between all commodities. Again for ease of notation, define the subset $\mathrm{P}(k) \subset \mathrm{P}$ containing all paths for commodity $k \in \mathrm{K}$, and the subset $\mathrm{P}(k, e) \subset \mathrm{P}(k)$ containing all paths for commodity $k \in \mathrm{K}$ that traverse edge $e \in \mathrm{E}$. In the model implementation, the paths

between every commodity are calculated using an $m$-shortest* simple path algorithm, in this case Yen's algorithm [126]. By ensuring $m > |\mathrm{P}(k)|$, we can generate a ranked list of all possible paths for commodity $k \in \mathrm{K}$. Limiting $m$, and therefore the number of paths, results in a heuristic method, and is investigated in [90].

Introduce variables $\boldsymbol{f}^{\mathrm{B}} \in \{0,1\}^{|\mathrm{P}|}$ to indicate the installation of fibre along a path and variables $\boldsymbol{f} \in \mathbb{Z}^{|\mathrm{P}|}$ to indicate the flow. When a path $p \in \mathrm{P}$ is used, $f_p^{\mathrm{B}}$ will be 1 and $f_p$ will take on the value of the flow crossing it, and both will be 0 otherwise. Assume that all demand is routed via the same path for every ONU. Fibre cost is redefined as $\boldsymbol{c}^{\mathrm{F}} \in \mathbb{Q}_+^{|\mathrm{P}|}$, the cost of installing a fibre in a duct along a path. The path model, henceforth known as *PATH*, can now be defined as follows:

*PATH*:

$$\min \quad c^{\mathrm{OLT}} + |\mathrm{U}|c^{\mathrm{ONU}} + \sum_{i \in \mathrm{D}} \psi_i c^{\mathrm{SP}} + \sum_{e \in \mathrm{E}} x_e^{\mathrm{TR}} c_e^{\mathrm{TR}} + \sum_{p \in \mathrm{P}} f_p c_p^F \tag{4.11}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j)} \sum_{p \in \mathrm{P}(k)} f_p = d_j, \quad \forall\, j \in \mathrm{U}, \tag{4.12}$$

$$\sum_{k \in \mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}(i)} \sum_{p \in \mathrm{P}(k)} f_p = \psi_i, \quad \forall\, i \in \mathrm{D}, \tag{4.13}$$

$$\sum_{k \in \mathrm{K}} \sum_{p \in \mathrm{P}(k,e)} f_p \leq \Delta x_e^{\mathrm{TR}}, \quad \forall\, e \in \mathrm{E}, \tag{4.14}$$

$$f_p \leq \Delta f_p^{\mathrm{B}}, \quad \forall p \in \mathrm{P}, \tag{4.15}$$

$$\sum_{k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(i)} \sum_{p \in \mathrm{P}(k)} f_p \leq \kappa \psi_i, \quad \forall\, i \in \mathrm{D}. \tag{4.16}$$

The objective function (4.11) is exactly equivalent to the *ARC* objective function (4.1), and minimises the deployment cost. The constraint set (4.12) ensures that ONU demand is satisfied by the used paths, while (4.13) ensures a used splitter is connected to the CO. Inequality set (4.14) sets the trench usage variable $x_e^{\mathrm{TR}}$ when a path traverses the edge $e \in \mathrm{E}$, while (4.15) sets $f_p^{\mathrm{B}}$ to 1 if any flow exists on the path $p \in \mathrm{P}$. Finally, (4.16) limits the number of ONUs that can connect to a single splitter while simultaneously setting the splitter usage variable $\psi_i$. For a more in-depth explanation on the

*In general, the problem is known as the *k-shortest simple path problem*, but since $k$ is already used as an index of the commodity set K, we use $m$ here to avoid confusion.

formulation of the *PATH* model, refer to [90].

The path model equivalent of the physical PON constraints specified for the arc model in (4.7)–(4.10) can be defined by introducing $\ell_p \in \mathbb{Q}_+^{|\mathrm{P}|}$, the length of a path $p \in \mathrm{P}$.

$$\ell_i^{\min} \leq \Big( \sum_{p \in \mathrm{P}(k_i^{\mathrm{F}})} f_p^{\mathrm{B}} \ell_p + \sum_{p \in \mathrm{P}(k_{ij}^{\mathrm{D}})} f_p^{\mathrm{B}} \ell_p \Big), \qquad \forall\, i \in \mathrm{D}, \forall\, j \in \mathrm{U}, \tag{4.17}$$

$$\Big( \sum_{p \in \mathrm{P}(k_i^{\mathrm{F}})} f_p^{\mathrm{B}} \ell_p + \sum_{p \in \mathrm{P}(k_{ij}^{\mathrm{D}})} f_p^{\mathrm{B}} \ell_p \Big) \leq \ell_i^{\max}, \qquad \forall\, i \in \mathrm{D}, \forall\, j \in \mathrm{U}, \tag{4.18}$$

$$\ell_i^{\max} \leq \ell_{\max}^{\text{total}}, \qquad \forall\, i \in \mathrm{D}, \tag{4.19}$$

$$\ell_i^{\max} - \ell_i^{\min} \leq \ell_{\max}^{\text{diff}}, \qquad \forall\, i \in \mathrm{D}. \tag{4.20}$$

Unfortunately, it is evident that the set P will be infeasible for large input graphs, even when the graph is sparse, since the number of possible simple paths increases dramatically as the number of edges increases. Therefore it is suspected that this modelling approach may also have limited scalability potential. From this and the previous section, it is therefore considered worthwhile to investigate decomposing either *ARC* or *PATH*.

## 4.3 Decomposition

Recall from section 3.2.2 that there are three major decomposition methods: Dantzig-Wolfe or *column* generation, Benders or *row* generation, and Lagrangian decomposition. All of these exploit the independent block structure of a model to split it into multiple independent sub-problems. Though *ARC* can be decomposed using Benders, the sheer number of variables in the resulting sub-problems still limits scalability. Therefore, in this section, we use Benders to decompose a relaxation of *PATH* and get rid of the complicating constraint (4.14), which links the edge variables $\boldsymbol{x}^{\mathrm{TR}}$ and path variables $\boldsymbol{f}$.

It is a well known issue that you can not project out integer variables using Benders decomposition, as classical duality theory does not extend to integer programming

[127]. This means we can not project out the flow variables $\boldsymbol{f} \in \mathbb{Z}_+^{|\mathrm{P}|}$. However, if we assume that the optimal solution to PONDP is a *tree*, i.e. it contains no cycles and by extension no splittable flows, this limiting issue can be overcome.

**Theorem 4.1.** *The distribution and feeder networks in the optimal solution of PONDP are trees.*

*Proof.* Consider the *ARC* formulation. Furthermore, without loss of generality, consider the distribution network and let ONUs be Steiner tree terminals. Since ONU demand $\boldsymbol{d}$ is integral, it can be substituted for the corresponding number of terminals, each with a demand of 1. For any possible fixed assignment of integral supply to splitters, $\boldsymbol{N}^*$, and therefore any assignment of Steiner tree roots, the multi-commodity flow conservation constraints (4.2) along with (4.5) are equivalent to the classical bi-directed Steiner arborescence constraints [81]. Since the bi-directed Steiner arborescence problem is equivalent to the Steiner tree problem, the resulting distribution network will be a tree.

The same reasoning applies to the feeder network, by letting $q$ be the root node and observing that for any fixed assignment of splitters, $\boldsymbol{\psi}^*$, each with a demand of 1, the constraints (4.3) are equivalent to the bi-directed Steiner arborescence constraints. Therefore, the feeder network is also a tree. □

Note that the *overall* network will not form a tree if we assume bounded splitter capacities, as two splitters might need to serve the same demand point. However, even with this assumption, when viewed separately, both the feeder and distribution networks will still be trees.

Consider *PATH* with relaxed path variables $\boldsymbol{f} \in \mathbb{R}_+^{|\mathrm{P}|}$, named $PATH_R$. This relaxation does not ensure unsplittable flow, but since we now know the optimal solution of the problem will not contain splittable flow in either the distribution or feeder networks, *PATH* and $PATH_R$ can be used interchangeably for our purposes. Therefore, we can project the relaxed path variables out of $PATH_R$, which can then be reincorporated implicitly by utilising cut generation.

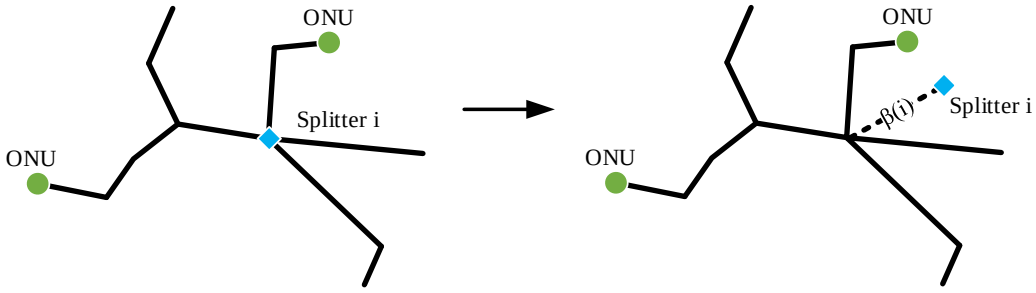
The master problem is defined exclusively using edge variables, while the sub-problems only contain path variables. This allows us to further decompose the path variables into two independent sub-problems, one for the distribution network and another for the feeder network. While the hierarchical structure of PONDP logically supports the distinction between the two flow types, this further decomposition also ensures that the sub-problems have a specific structure we exploit in section 4.3.3.

Consider the objective function (4.11) of *PATH*. Since the path variables are projected out of $PATH_R$, we do not have access to them to define an equivalent objective function for the master problem. While the fibre usage will be incorporated using cuts, the splitter usage variables $\boldsymbol{\psi}$ can not be set as done in (4.16). Therefore, splitter usage is inferred implicitly from additional edge flow variables, strategically placed to eliminate splitter usage ambiguity through preprocessing of the input graph.

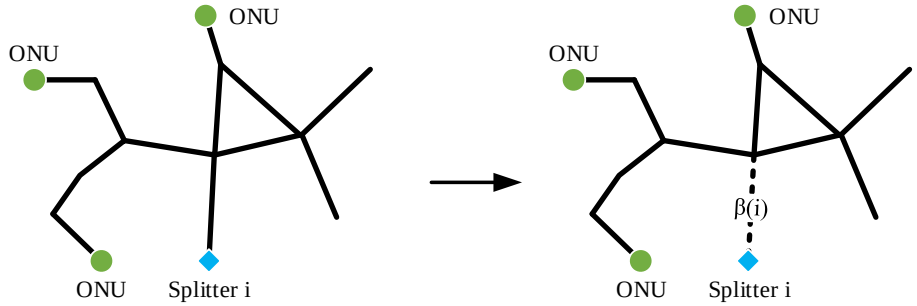
### 4.3.1 Graph preprocessing

To be able to unambiguously infer splitter usage from edge flow, the input graph $\mathcal{G}$ is preprocessed. Additional vertices are created for every non-leaf, or *internal*, Point of Interest (POI) vertex, i.e. CO, splitter and ONU locations, and connected via an auxiliary edge of length $\epsilon \ll 1$ to the original location. These new vertices are then substituted for the corresponding vertices in the sets U, D and point $q$, effectively ensuring that all vertices in these sets are leaves. Figure 4.1 illustrates this processing step for both internal and leaf splitter vertices.

Next, we introduce the concept of *indicator* edges, which is just the single edge connecting a leaf in the modified graph. The function $\beta : \mathrm{V} \to \mathrm{E}$ outputs the unique indicator edge for a vertex if it is a leaf, and is undefined for all other vertices. All flow through an indicator edge will be destined for, or will originate from, the corresponding leaf, as no simple path will traverse this edge otherwise. Therefore, if flow exists on edge $\beta(i) \in \mathrm{E}, i \in \mathrm{D}$, we can infer that the splitter $i$ is connected and, by extension, in use.




a) Internal splitter preprocessing




b) Leaf splitter preprocessing

Figure 4.1: Graph preprocessing for internal and leaf splitters

### 4.3.2 Benders formulation

We now have all the components to construct the Benders decomposed model, starting with the master problem consisting only of edge flow variables.

#### Master problem

Introduce edge flow variables $\boldsymbol{x}^{\mathrm{FD}} \in \mathbb{Z}_{+}^{|\mathrm{E}|}$ for the fibre usage in the distribution network and $\boldsymbol{x}^{\mathrm{FF}} \in \mathbb{Z}_{+}^{|\mathrm{E}|}$ for the fibre usage in the feeder network. We can now define the master problem, henceforth known as $PON_M$, as follows:

$PON_M$:

$$\min \qquad c^{\mathrm{OLT}} + |\mathrm{U}|c^{\mathrm{ONU}} + \sum_{e \in \mathrm{E}} (x_e^{\mathrm{TR}} c_e^{\mathrm{TR}} + x_e^{\mathrm{FD}} c_e^{\mathrm{F}} + x_e^{\mathrm{FF}} c_e^{\mathrm{F}}) + \sum_{i \in \mathrm{D}} \psi_i c^{\mathrm{SP}} \tag{4.21}$$

$$\text{s.t.} \quad x^{\text{FD}}_{\beta(i)} \leq \kappa \psi_i, \quad \forall i \in \text{D}, \tag{4.22}$$

$$x^{\text{FF}}_e \leq \Delta x^{\text{TR}}_e, \quad \forall e \in \text{E}, \tag{4.23}$$

$$x^{\text{FD}}_e \leq \Delta x^{\text{TR}}_e, \quad \forall e \in \text{E}. \tag{4.24}$$

The objective (4.21) once again minimises the same total deployment cost parameters as in both *ARC* and *PATH*. Constraint set (4.22) infers splitter usage if distribution fibres exist on an indicator edge while simultaneously limiting the number of fibres that can connect to any splitter. Inequalities (4.23) and (4.24) set the trench usage variable if at least one fibre exists on an edge.

**Distribution sub-problem**

Sub-problems in the Benders decomposition framework ensure feasibility by cutting off infeasible regions in the master search space. To this end, consider first the feasibility in the distribution network, i.e. all ONUs need to be connected to a splitter.

For the distribution sub-problem, a similar approach to Terblanche [128] is followed (see also [129]), in which the total capacity shortfall $\alpha \in \mathbb{R}^+$ is minimised, given a fixed edge flow capacity vector $\boldsymbol{x}^{\text{FD}*}$ from the master. Therefore, the sub-problem finds a feasible solution, ensuring every ONU is connected, which requires the least total capacity increase in the master. This sub-problem formulation allows us to derive tight cuts to project the path variables into the master problem.

To avoid ambiguity, we separate the path flow variables $\boldsymbol{f}$ into their distribution and feeder components, $\boldsymbol{f}^{\text{D}} \in \mathbb{R}_+^{|\text{P}|}$ and $\boldsymbol{f}^{\text{F}} \in \mathbb{R}_+^{|\text{P}|}$ respectively. Now we can define the distribution sub-problem, henceforth known as $PON_D$, as follows:

$PON_D$:

$$\min \quad \alpha \tag{4.25}$$

$$\text{s.t.} \quad \sum_{k \in \text{K}^{\text{SP}}_{\text{ONU}}(j)} \sum_{p \in \text{P}(k)} f^{\text{D}}_p = d_j, \quad \forall j \in \text{U}, \tag{4.26}$$

$$\sum_{k \in \mathrm{K}} \sum_{p \in \mathrm{P}(k,e)} f_p^{\mathrm{D}} - \alpha \leq x_e^{\mathrm{FD}*}, \quad \forall e \in \mathrm{E}. \tag{4.27}$$

Constraint set (4.26) ensures that the ONU demand is met while (4.27) calculates the capacity shortfall of the sub-problem solution given the currently reserved capacity in the master problem.

Next, we need to derive a cut based on the solution vector of $PON_D$. To that end, we consider its dual. Associate dual variables $\boldsymbol{\pi}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{U}|}$ and $\boldsymbol{\mu}^{\mathrm{D}} \in \mathbb{R}_+^{|\mathrm{E}|}$ with constraints (4.26) and (4.27) respectively. Introduce the subset $\mathrm{E}(p) \subset \mathrm{E}$, consisting of all edges traversed by a path $p \in \mathrm{P}$. In terms of the dual, $\boldsymbol{\pi}^{\mathrm{D}}$ plays the role of the demand variable and $\boldsymbol{\mu}^{\mathrm{D}}$ the capacity variable. The corresponding dual (illustrated with a prime) can then be written as follows:

$PON_D$':

$$\max \quad \sum_{j \in \mathrm{U}} d_j \pi_j^{\mathrm{D}} - \sum_{e \in \mathrm{E}} x_e^{\mathrm{FD}*} \mu_e^{\mathrm{D}} \tag{4.28}$$

$$\text{s.t.} \quad \sum_{e \in \mathrm{E}} \mu_e^{\mathrm{D}} \leq 1, \tag{4.29}$$

$$\pi_j^{\mathrm{D}} - \sum_{e \in \mathrm{E}(p)} \mu_e^{\mathrm{D}} \leq 0, \qquad \begin{array}{c} \forall p \in \mathrm{P}(k), \forall k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j), \\ \forall j \in \mathrm{U}. \end{array} \tag{4.30}$$

Since $\alpha \geq 0$ in the minimisation primal problem, a feasible solution to the maximisation dual will have a non-positive objective value. This fact is stated more formally in the so-called *Japanese theorem*, a special case of Farkas' lemma, developed independently by Iri [130], and Onaga and Kakusho [131].

**Theorem 4.2** (Japanese theorem [130, 131])**.** *For a given capacity vector* $\boldsymbol{x}^{\mathrm{FD}*}$ *and* $\alpha = 0$*, the polyhedron given by* (4.26)–(4.27) *will not be empty iff*

$$\sum_{e \in \mathrm{E}} x_e^{\mathrm{FD}} \mu_e^{\mathrm{D}} \geq \sum_{j \in \mathrm{U}} d_j \pi_j^{\mathrm{D}} \tag{4.31}$$

*for all* $\mu_e^{\mathrm{D}} \geq 0$*, and where* $\pi_j^{\mathrm{D}} = \min_{k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \mu_e^{\mathrm{D}}$.

The value of $\pi_j^{\mathrm{D}}$ is found by rearranging equation (4.30) as follows:

$$\pi_j^{\mathrm{D}} \leq \sum_{e \in \mathrm{E}(p)} \mu_e^{\mathrm{D}}, \qquad \begin{array}{c} \forall p \in \mathrm{P}(k), \forall k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j), \\ \forall j \in \mathrm{U}. \end{array} \tag{4.32}$$

The minimum value of $\pi_j^{\mathrm{D}}$ across all applicable sums of $\mu_e^{\mathrm{D}}$ will be the largest feasible value in $PON_D{}'$, yielding the maximum objective function value and consequently, the deepest cut.

When inequality (4.31) does not hold, we add the following cut to the master:

$$\sum_{e \in \mathrm{E}} x_e^{\mathrm{FD}} \mu_e^{\mathrm{D}*} \geq \sum_{j \in \mathrm{U}} d_j \pi_j^{\mathrm{D}*}, \tag{4.33}$$

with $\boldsymbol{x}^{\mathrm{FD}}$ the corresponding variables in $PON_M$, and $\boldsymbol{\mu}^{\mathrm{D}*}$ and $\boldsymbol{\pi}^{\mathrm{D}*}$ the solution values from the dual $PON_D{}'$. This type of cut is also known as a *metric inequality*.

**Feeder sub-problem**

Following the same logic applied in deriving the distribution sub-problem, we can derive the feeder sub-problem, ensuring that all used splitters are connected to the CO.

Start by defining the feeder sub-problem, which minimises the total capacity shortfall $\alpha \in \mathbb{R}^+$, given a fixed edge flow capacity vector $\boldsymbol{x}^{\mathrm{FF}*}$ from the master. In contrast to the distribution sub-problem, where the demand is a parameter, the feeder sub-problem demand is the value of the variable $\boldsymbol{\psi}$, denoted with an asterisk (*). Therefore, if a splitter $i$ is used, i.e. $\psi_i = 1$, the demand for a fibre between splitter $i$ and the CO is 1, ensuring that the splitter is connected to the CO. The feeder sub-problem, henceforth known as $PON_F$, can now be defined as follows:

$PON_F$:

$$\min \qquad \alpha \tag{4.34}$$

$$\text{s.t.} \qquad \sum_{k \in \mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}(i)} \sum_{p \in \mathrm{P}(k)} f_p^{\mathrm{F}} = \psi_i^*, \quad \forall i \in \mathrm{D}, \tag{4.35}$$

$$\sum_{k \in \mathrm{K}} \sum_{p \in \mathrm{P}(k,e)} f_p^{\mathrm{F}} - \alpha \leq x_e^{\mathrm{FF}*}, \quad \forall e \in \mathrm{E}. \tag{4.36}$$

By associating dual variables $\boldsymbol{\pi}^{\mathrm{F}} \in \mathbb{R}^{|\mathrm{D}|}$ and $\boldsymbol{\mu}^{\mathrm{F}} \in \mathbb{R}_{+}^{|\mathrm{E}|}$ with constraints (4.35) and (4.36) respectively, we can derive the dual problem $PON_F{}'$:

$PON_F{}'$:

$$\max \quad \sum_{i \in \mathrm{D}} \psi_i^* \pi_i^{\mathrm{F}} - \sum_{e \in \mathrm{E}} x_e^{FF*} \mu_e^{\mathrm{F}} \tag{4.37}$$

$$\text{s.t.} \quad \sum_{e \in \mathrm{E}} \mu_e^{\mathrm{F}} \leq 1, \tag{4.38}$$

$$\pi_i^{\mathrm{F}} - \sum_{e \in \mathrm{E}(p)} \mu_e^{\mathrm{F}} \leq 0, \qquad \begin{array}{c} \forall p \in \mathrm{P}(k), \forall k \in \mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}(i), \\ \forall i \in \mathrm{D}. \end{array} \tag{4.39}$$

Again using the *Japanese theorem*, the corresponding feeder network metric inequality can be derived:

$$\sum_{e \in \mathrm{E}} x_e^{\mathrm{FF}} \mu_e^{\mathrm{F}*} \geq \sum_{i \in \mathrm{D}} \psi_i \pi_i^{\mathrm{F}*}, \tag{4.40}$$

with $\boldsymbol{x}^{\mathrm{FF}}$ the corresponding variables in $PON_M$, and $\boldsymbol{\mu}^{\mathrm{D}*}$ and $\boldsymbol{\pi}^{\mathrm{D}*}$ the solution values from the dual $PON_F{}'$. Similarly, $\pi_i^{\mathrm{F}} = \min_{k \in \mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}(i)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \mu_e^{\mathrm{F}}$, which was derived by rearranging inequality (4.39).

In conclusion, by separating violated cuts (4.33) and (4.40) and adding them to the master, the path variables $\boldsymbol{f}$ are projected back into $PON_M$, ensuring connectivity in both the distribution and feeder networks.

### 4.3.3 Column generation

The separation of the metric inequalities (4.33) and (4.40) is trivial: solve the corresponding sub-problem ($PON_D$ or $PON_F$), get the dual values and add the resulting violated cuts. However, both $PON_D$ and $PON_F$ require a subset of P, which, as mentioned in section 4.2.2, may be infeasibly large for large input graphs. This means that the same scalability limitations of *PATH* are present in its Benders decomposed formu-

lation. Fortunately, the derivation of the metric inequalities yields a property we can exploit to reduce the size of the set P: the calculation of values $\boldsymbol{\pi}^{\mathrm{D}}$ and $\boldsymbol{\pi}^{\mathrm{F}}$.

Consider the distribution sub-problem. Instead of using the entire set $\mathrm{P}(k)$, use a subset of the paths between each commodity, $\mathrm{P_R}(k) \subset \mathrm{P}(k), k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}$. Start by solving $PON_D$ using this reduced path set.

Recall that the value of $\boldsymbol{\pi}^{\mathrm{D}}$ was given as

$$\pi^{\mathrm{D}}_j = \min_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \mu^{\mathrm{D}}_e, \quad \forall j \in \mathrm{U}. \tag{4.41}$$

This is exactly equivalent to finding the length of the shortest path between ONU $j \in \mathrm{U}$ and any splitter, given the input graph $\mathcal{G}$ and edge costs $\boldsymbol{\mu}^{\mathrm{D}}$.

Therefore, use the values $\boldsymbol{\mu}^{\mathrm{D}*}$ we got from solving $PON_D$ and calculate the shortest path between every ONU and its closest splitter using Dijkstra's algorithm [132]. For ONU $j \in \mathrm{U}$, denote the calculated path length as $\pi^{\mathrm{D}\prime}_j$. If $\pi^{\mathrm{D}\prime}_j < \pi^{\mathrm{D}*}_j$, add the corresponding calculated path to $\mathrm{P_R}$ and resolve $PON_D$. Repeat this process until no shorter path exists for any ONU $j \in \mathrm{U}$.

As we are adding new variables, or *columns*, to the model at each iteration, this is known as *column generation*. Since the metric inequality we add to the master only takes the minimum path lengths into account, this procedure is still optimal, ensuring that no shorter path will *not* be considered.

For the feeder sub-problem, the exact same procedure is used, except that we calculate the shortest path lengths $\pi^{\mathrm{F}\prime}_i$ between every splitter $i \in \mathrm{D}$ and the CO at point $q$ using Dijkstra.

In the rest of the thesis, the Benders formulation which includes column generation will be referred to as $BENDERS_C$.

### 4.3.4 Path length constraints

To comply with the PON physical constraints, path lengths need to be constrained similar to (4.7)–(4.10) for *ARC* and (4.17)–(4.20) for *PATH*. These constraints can not be added explicitly to the master, since paths are not available, and can not be added to the sub-problems since it requires access to both distribution and feeder paths simultaneously. Therefore, the path length constraints are added as an additional cut, separated when needed.

Introduce variables $\boldsymbol{x}^{\mathrm{TD}} \in \{0,1\}^{|\mathrm{E}|}$ and $\boldsymbol{x}^{\mathrm{TF}} \in \{0,1\}^{|\mathrm{E}|}$, indicating the usage of a trench for the distribution and feeder network respectively. This allows us to disaggregate edge usage in the master and add cuts to limit edge usage in the distribution or feeder networks exclusively if required. Add the following constraints to $PON_M$ to set them accordingly:

$$x_e^{\mathrm{FF}} \leq \Delta x_e^{\mathrm{TF}}, \qquad \forall e \in \mathrm{E}, \tag{4.42}$$

$$x_e^{\mathrm{FD}} \leq \Delta x_e^{\mathrm{TD}}, \qquad \forall e \in \mathrm{E}, \tag{4.43}$$

$$x_e^{\mathrm{TF}} \leq x_e^{\mathrm{FF}}, \qquad \forall e \in \mathrm{E}, \tag{4.44}$$

$$x_e^{\mathrm{TD}} \leq x_e^{\mathrm{FD}}, \qquad \forall e \in \mathrm{E}, \tag{4.45}$$

$$x_e^{\mathrm{TF}} \leq x_e^{\mathrm{TR}}, \qquad \forall e \in \mathrm{E}, \tag{4.46}$$

$$x_e^{\mathrm{TD}} \leq x_e^{\mathrm{TR}}, \qquad \forall e \in \mathrm{E}. \tag{4.47}$$

Inequalities (4.42)–(4.43) set the trench variables if flow exists in the respective networks, while (4.44) and (4.45) do the opposite. Constraint sets (4.46) and (4.47) set the global trench usage variable if a trench is used in either the feeder or distribution network.

By saving the feasible solutions of the distribution and feeder sub-problems, i.e. the values of $\boldsymbol{f}^{\mathrm{D}}$ and $\boldsymbol{f}^{\mathrm{F}}$, we can check for path length violations for every splitter. First, evaluate the length of every used distribution path, $p_i^{\mathrm{D}}$, and feeder path, $p_i^{\mathrm{F}}$, for every splitter $i \in \mathrm{D}$. If any combination violates the maximum path length $\ell_{\max}^{\mathrm{total}}$, it may be

tempting to add the following cut to $PON_M$:

$$\sum_{e \in \mathrm{E}(p_i^{\mathrm{D}})} \ell_e x_e^{\mathrm{TD}} + \sum_{e \in \mathrm{E}(p_i^{\mathrm{F}})} \ell_e x_e^{\mathrm{TF}} - \ell_{\max}^{\mathrm{total}} \leq \Delta(1 - \psi_i), \quad \forall i \in \mathrm{D}, \tag{4.48}$$

with $\Delta = \sum_{e \in \mathrm{E}(p^{\mathrm{D}}) \cup \mathrm{E}(p^{\mathrm{F}})} \ell_e$. This cut has a similar form to the explicit cuts we saw in equations (4.17)–(4.20). Unfortunately, if a longer path is chosen at the start, it may cut off subsequent paths that share some subset of edges with it, resulting in a sub-optimal solution. For example, if a path exists that crosses almost all edges in the graph, a cut in the form above will effectively ensure that a combination of shorter paths that each share a single edge with the long path will also be infeasible, as the cut does not distinguish between edges used in a single path or the same edges used in multiple paths.

Therefore, we need to retreat to *canonical cuts* (often referred to as *no-good cuts*) [133]. A canonical cut is an inequality which cuts of a specific solution $\boldsymbol{x}^*$ from a problem $P$. In general, the cut can be formulated as follows:

$$||\boldsymbol{x} - \boldsymbol{x}^*|| \geq \epsilon, \tag{4.49}$$

with $||\cdot||$ the $p$-norm and $\epsilon \geq 0$ the radius around the point $\boldsymbol{x}^*$ chosen to ensure that (4.49) does not cut off any other feasible point. In general, (4.49) is clearly non-linear, as well as being very inefficient to model. However, for the binary case where $\boldsymbol{x} \in \{0,1\}^{|\mathrm{M}|}$, note that the 1-norm (also known as the taxicab norm or Manhattan distance) takes on the value:

$$||\boldsymbol{x} - \boldsymbol{x}^*||_1 = \sum_{i \in \mathrm{M}} |x_i - x_i^*|, \tag{4.50}$$

where

$$|x_i - x_i^*| = \begin{cases} x_i, & \text{if } x_i^* = 0, \\ (1 - x_i), & \text{if } x_i^* = 1. \end{cases} \tag{4.51}$$

Therefore, with $\epsilon = 1$, we can derive the canonical cut:

$$\sum_{i \in \mathrm{M}: x_i^* = 0} x_i + \sum_{i \in \mathrm{M}: x_i^* = 1} (1 - x_i) \geq 1. \tag{4.52}$$

Even though this can be extended to general integer variables, it leads to an extremely inefficient formulation, requiring $2|\mathrm{M}|$ additional continuous variables, $|\mathrm{M}|$ additional binary variables and $3|\mathrm{M}|+1$ additional constraints [133].

In canonical cut form therefore, by utilising equation (4.52) and setting $\epsilon = \psi_i$, we can define the path length cut as follows:

$$\sum_{\substack{e \in \mathrm{E}: \\ x_e^{\mathrm{TD}*}=0}} x_e^{\mathrm{TD}} + \sum_{\substack{e \in \mathrm{E}: \\ x_e^{\mathrm{TD}*}=1}} (1 - x_e^{\mathrm{TD}}) + \sum_{\substack{e \in \mathrm{E}: \\ x_e^{\mathrm{TF}*}=0}} x_e^{\mathrm{TF}} + \sum_{\substack{e \in \mathrm{E}: \\ x_e^{\mathrm{TF}*}=1}} (1 - x_e^{\mathrm{TF}}) \geq \psi_i, \quad \forall i \in \mathrm{D}. \tag{4.53}$$

This cut is extremely poor, ensuring that when $\psi_i = 1$, one or more of the variables $\boldsymbol{x}^{\mathrm{TD}}$ and $\boldsymbol{x}^{\mathrm{TF}}$ need to toggle between 0 and 1. Due to the nature of the path length cut however, we know that turning an edge on that is not contained in $\mathrm{E}(\boldsymbol{p}^{\mathrm{D}}) \cup \mathrm{E}(\boldsymbol{p}^{\mathrm{F}})$, will still result in a violated cut as adding an edge to a path will not make it shorter. Similarly, turning an edge off which is not contained in the same subset will also result in a violated cut, as the paths will still be the same length. Therefore, we can constrain the cut to the edges contained in the paths $\boldsymbol{p}^{\mathrm{D}}$ and $\boldsymbol{p}^{\mathrm{F}}$:

$$\sum_{\substack{e \in \mathrm{E}(p_i^{\mathrm{D}}): \\ x_e^{\mathrm{TD}*}=0}} x_e^{\mathrm{TD}} + \sum_{\substack{e \in \mathrm{E}(p_i^{\mathrm{D}}): \\ x_e^{\mathrm{TD}*}=1}} (1 - x_e^{\mathrm{TD}}) + \sum_{\substack{e \in \mathrm{E}(p_i^{\mathrm{F}}): \\ x_e^{\mathrm{TF}*}=0}} x_e^{\mathrm{TF}} + \sum_{\substack{e \in \mathrm{E}(p_i^{\mathrm{F}}): \\ x_e^{\mathrm{TF}*}=1}} (1 - x_e^{\mathrm{TF}}) \geq \psi_i, \; \forall i \in \mathrm{D}. \tag{4.54}$$

Since $\boldsymbol{x}^{\mathrm{TD}}$ and $\boldsymbol{x}^{\mathrm{TF}}$ will never be 0 in these specific subsets, as all edges in $\mathrm{E}(\boldsymbol{p}^{\mathrm{D}}) \cup \mathrm{E}(\boldsymbol{p}^{\mathrm{F}})$ are currently used, we can simplify (4.54):

$$\sum_{e \in \mathrm{E}(p_i^{\mathrm{D}})} x_e^{\mathrm{TD}} + \sum_{e \in \mathrm{E}(p_i^{\mathrm{F}})} x_e^{\mathrm{TF}} + \psi_i \leq |\mathrm{E}(p_i^{\mathrm{D}})| + |\mathrm{E}(p_i^{\mathrm{F}})|, \quad \forall i \in \mathrm{D}. \tag{4.55}$$

This path length cut will therefore ensure that when $\psi_i = 1$, at least one edge in $\mathrm{E}(\boldsymbol{p}^{\mathrm{D}}) \cup \mathrm{E}(\boldsymbol{p}^{\mathrm{F}})$ must be switched off. Since no two splitters will share *all* the edges in both $\boldsymbol{p}^{\mathrm{D}}$ and $\boldsymbol{p}^{\mathrm{F}}$, (4.55) will not cut off other feasible paths.

Next we need to ensure that the differential path length constraints are not violated. Note that since the feeder path $p_i^{\mathrm{F}}$ is fixed for any splitter $i \in \mathrm{D}$, the difference in path lengths will only depend on the difference between distribution path lengths. To this

end, evaluate the length of every used distribution path connected to every splitter $i \in \mathrm{D}$, and determine the minimum and maximum length paths. Denote these $p_i^{\min}$ and $p_i^{\max}$ respectively. If the difference between the lengths exceeds the maximum differential length, the naive approach would again be to add the following cut:

$$\sum_{e \in \mathrm{E}(p_i^{\max})} \ell_e x_e^{\mathrm{TD}} - \sum_{e \in \mathrm{E}(p_i^{\min})} \ell_e x_e^{\mathrm{TD}} - \ell_{\max}^{\mathrm{diff}} \leq \Delta(1 - \psi_i), \quad \forall i \in \mathrm{D}. \tag{4.56}$$

Unfortunately, if $p_i^{\min}$ is removed from the problem, this may cut off other valid paths that share a subset of edges with $p_i^{\max}$ as the cut will revert to:

$$\sum_{e \in \mathrm{E}(p_i^{\max})} \ell_e x_e^{\mathrm{TD}} \leq \ell_{\max}^{\mathrm{diff}}, \quad \forall i \in \mathrm{D}, \tag{4.57}$$

which, if $\ell_{\max}^{\mathrm{diff}} \ll \ell_{\max}^{\mathrm{total}}$ for example, would essentially disallow the use of any edges in $p_i^{\max}$, even if the solution could be feasible.

Switching to canonical cut format and again setting $\epsilon = \psi_i$, we can define the differential path length canonical cut as follows:

$$\sum_{\substack{e \in \mathrm{E}: \\ x_e^{\mathrm{TD}*} = 0}} x_e^{\mathrm{TD}} + \sum_{\substack{e \in \mathrm{E}: \\ x_e^{\mathrm{TD}*} = 1}} (1 - x_e^{\mathrm{TD}}) \geq \psi_i, \quad \forall i \in \mathrm{D}. \tag{4.58}$$

This ensures that when $\psi_i = 1$, one or more of the variables $\boldsymbol{x}^{\mathrm{TD}}$ need to toggle between 0 and 1. We know that subtracting an edge from $\boldsymbol{p}^{\min}$ or adding an edge to $\boldsymbol{p}^{\max}$ will still result in a violated cut, as we either need to increase the length of $\boldsymbol{p}^{\min}$ or decrease the length of $\boldsymbol{p}^{\max}$ to be within the differential limit. Therefore, we can constrain the cut to the edges contained in the paths $\boldsymbol{p}^{\max}$ and those that are not in paths $\boldsymbol{p}^{\min}$:

$$\sum_{\substack{e \in \mathrm{E}(p_i^{\max}) \cup \mathrm{E} \setminus \mathrm{E}(p_i^{\min}): \\ x_e^{\mathrm{TD}*} = 0}} x_e^{\mathrm{TD}} + \sum_{\substack{e \in \mathrm{E}(p_i^{\max}) \cup \mathrm{E} \setminus \mathrm{E}(p_i^{\min}): \\ x_e^{\mathrm{TD}*} = 1}} (1 - x_e^{\mathrm{TD}}) \geq \psi_i, \; \forall i \in \mathrm{D}. \tag{4.59}$$

Since all edges in $\mathrm{E}(\boldsymbol{p}^{\min}) \cup \mathrm{E}(\boldsymbol{p}^{\max})$ are currently used, $\boldsymbol{x}^{\mathrm{TD}}$ will never be 0 in these specific subsets. Furthermore, since turning off an edge in $\mathrm{E} \setminus \mathrm{E}(p_i^{\min})$ will still result in a violated cut, we can simplify (4.59):

$$\sum_{e \in \mathrm{E}(p_i^{\max})} x_e^{\mathrm{TD}} - \sum_{\substack{e \in \mathrm{E} \setminus \mathrm{E}(p_i^{\min}): \\ x_e^{\mathrm{TD}*} = 0}} x_e^{\mathrm{TD}} + \psi_i \leq |\mathrm{E}(p_i^{\max})|, \; \forall i \in \mathrm{D}. \tag{4.60}$$

This differential path length cut will therefore ensure that when $\psi_i = 1$, at least one edge in $\mathrm{E}(\boldsymbol{p}^{\max})$ must be switched off or one edge not in $\mathrm{E}(\boldsymbol{p}^{\min})$ must be switched on. In this case, since no two paths will share *all* the edges with both $\boldsymbol{p}^{\min}$ and $\boldsymbol{p}^{\max}$ unless they are connected to the same splitters and ONUs, (4.60) will not cut off other feasible paths.

## 4.4 Experimental methodology

To evaluate the computational performance of the Benders decomposition approach, a number of experiments are conducted. Before the results are presented however, we will explain the experimental methodology, including the choice of data used as input, how we interpret the results and how the results are verified and validated.

### 4.4.1 Input data sets and parameters

Geographic Information System (GIS)-mapped real-world data is used for all testing purposes since it ensures that performance measures are valid for practical network designs. These data sets consist partly of experimental data provided by atesio GmbH [134], and partly of custom generated data sets derived from GIS mapping data of South Africa.

The testing approach is two-fold: to assess theoretical performance and to ensure practical feasibility. To that end, two different categories of input data sets are used as input, one containing a variety of different graph configurations as one would expect to encounter in real-world deployments, which we call *baseline* sets, and one containing artificially constrained graphs to test scalability with different numbers of ONUs and splitters. All input graphs are manufactured from much larger input graphs by starting at the CO location and running Dijkstra's algorithm, marking all vertices and edges encountered, until the requisite number of ONUs and splitters are contained in the resulting marked subgraph.

Model parameters are either approximated from data provided by industry sources or derived from the G.984 standards [11–14]. To preserve relative domination between the costs terms in the objective function, the relative accuracy of cost parameters must be as high as possible. For example, the extent of fibre duct sharing is determined by the relation between the parameters $\boldsymbol{c}^{\mathrm{F}}$ and $\boldsymbol{c}^{\mathrm{TR}}$ [135]. When $\boldsymbol{c}^{\mathrm{TR}} \gg \boldsymbol{c}^{\mathrm{F}}$, more ducts will be shared, as it becomes more cost effective to install a longer fibre that requires less additional trenching.

The path length limit $\ell_{\mathrm{total}}^{\max}$ can be derived to comply with the ITU-T G.984.2 [12] standard, and is conservative in its estimation. Let $\mathrm{P_b}$ be the total optical power budget in dB. Splitter loss, $a_{\mathrm{s}}$, is estimated through the equation $a_{\mathrm{s}} = 11 \log_{10} \kappa$, which includes 10 % additional loss due to heat and reflection over the minimum attainable by a passive splitter with $\kappa$ output ports.

For a given connector loss, $a_{\mathrm{c}}$ (dB), splice loss, $a_\ell$ (dB), and fibre loss $a_{\mathrm{f}}$ (dB/km), the maximum path length (in metres) can then be computed as follows:

$$\ell_{\mathrm{total}}^{\max} = \frac{\mathrm{P_b} - a_{\mathrm{c}} - a_\ell - a_{\mathrm{s}}}{a_{\mathrm{f}}} \times 1000. \tag{4.61}$$

This equation subtracts all equipment attenuation effects from the total power budget and uses the remainder as the maximum attenuation that may arise from the fibre itself. Even though typical values for $\ell_{\mathrm{total}}^{\max}$ are less than the 60 km maximum theoretical network reach as specified in the ITU-T G.984.2 standard, we will utilise the maximum network reach for this chapter, as alluded to in section 4.1.2.

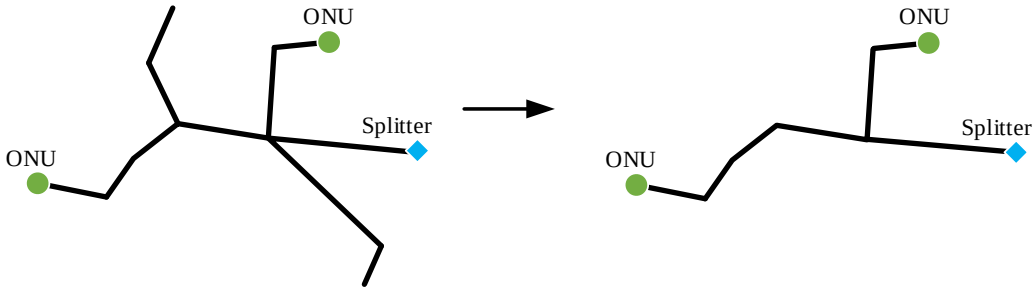
**Data set reduction**

Typically, the input graph $\mathcal{G}$ has a large potential for edge reduction, by removing excess leaves or substituting a number of edges with a single equivalent edge. Trimming excess leaves from $\mathcal{G}$ will be called *first-order reductions* while edge substitution will be known as *second-order reductions*. Since both the number of variables and the number of constraints in $PON_M$ are proportional to the number of edges in $\mathcal{G}$, any reduction in $|\mathrm{E}|$ can have a drastic impact on computational performance. Furthermore, since

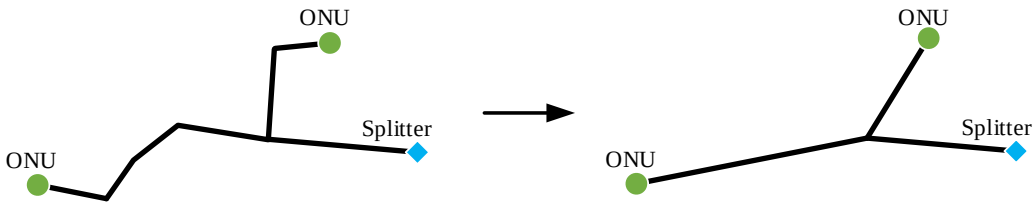
we are calculating shortest paths using Dijkstra's algorithm, which has a complexity of $O(|E| + |V| \log |V|)$ when implemented using Fibonacci heaps [136], a reduction in the size of sets E and V will result in faster column generation on average.

An excess leaf is defined as a vertex $v \in V \setminus (U \cup D \cup \{q\})$ where $\deg(v) = 1$. All excess leaves, as well as their incident edges, are trimmed iteratively until no more exist in the graph.

Edge substitution is done for all vertices $w \in V \setminus (U \cup D \cup \{q\})$ where $\deg(w) = 2$. For vertex $w$, let $e_{w1} = (a, w)$ and $e_{w2} = (b, w)$ be its incident edges with lengths $\ell_{w1}$ and $\ell_{w2}$ respectively. Then, remove $w$, $e_{w1}$ and $e_{w2}$ from $\mathcal{G}$, and add a new equivalent edge $e_w = (a, b)$ with length $\ell_w = \ell_{w1} + \ell_{w2}$. Repeat this procedure until no more candidate edges exist. Additionally, after the substitution, if more than two edges exist with the same end vertices, all except the shortest edge are removed. Both operations are demonstrated in figure 4.2.




a) First-order graph reduction (excess leaf removal)




b) Second-order graph reduction (edge substitution)

Figure 4.2: First- and second-order reductions

### 4.4.2 Result interpretation

During experimentation, results will be interpreted with particular emphasis placed on the following:

- **Solution feasibility** - The resulting design needs to conform to the G.984 specifications in terms of physical topology feasibility, i.e. the network should be able to be connected in the real world as shown in the solution, as well as be numerically feasible, e.g. the number of fibres in a duct should be sensible, fibres should follow a path etc. This latter feasibility check is also known as *face validity* [24].
- **Solution optimality** - The great advantage of the exact framework is the concept of optimality and solution quality. All results should be either optimal or have a guaranteed measure of solution quality, i.e. a low positive optimality gap.
- **Computational performance** - Since scalability is one of the addressed issues in this thesis, computational performance is of utmost importance. In the context of mathematical modelling, this is either a measure of the time or computational effort expended to solve the model to optimality, or the quality of the solution attained given a fixed period of time.

### 4.4.3 Validation and verification

The models under test are verified and validated using a variety of techniques. Most importantly, they are verified using inter-model comparison [24], i.e. if all model formulations give the exact same optimal objective function and variable values across a wide range of input graphs, we can be confident that our decomposition approach is equivalent to the standard *ARC* and *PATH* formulations and that the implementation is done correctly. All models are validated using face validation [24], i.e. checking that solution values are reasonable and the model decisions are logical and in accordance to specifications. In addition, solutions are externally validated using an auxiliary ILP model and checked for physical constraint feasibility using solution *metrics*.

**Routing feasibility checking**

To ensure the solution is feasible in terms of fibre routing, an external feasibility checker is used. This consists of solving an aggregated arc-flow model with fixed edge capacities and no objective function. Since we minimise the null vector $\mathbf{0}$, the solution to the model will only indicate feasibility, providing no information regarding optimality. However, this will drastically increase computational performance. By re-solving an infeasible model under some relaxed assumptions, we can also detect sub-optimality in certain cases.

Introduce arc-flow variables $\boldsymbol{f}^{\mathrm{D}} \in \mathbb{Z}_{+}^{|\mathrm{A}|}$ and $\boldsymbol{f}^{\mathrm{F}} \in \mathbb{Z}_{+}^{|\mathrm{A}|}$ for the distribution and feeder networks respectively. Solution values $\boldsymbol{x}^{\mathrm{FD}*}$ and $\boldsymbol{x}^{\mathrm{FF}*}$ are found directly from $PON_M$, or converted from *ARC* and *PATH* variables. For *ARC*, the values are calculated according to the following equations (with asterisks denoting solution values):

$$x_e^{\mathrm{FD}*} = \sum_{k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}} (f_{ijk}^* + f_{jik}^*), \qquad \forall (ij) = e \in \mathrm{E}, \tag{4.62}$$

$$x_e^{\mathrm{FF}*} = \sum_{k \in \mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}} (f_{ijk}^* + f_{jik}^*), \qquad \forall (ij) = e \in \mathrm{E}, \tag{4.63}$$

while, for *PATH*, they are converted from the solution values $\boldsymbol{f}^*$:

$$x_e^{\mathrm{FD}*} = \sum_{k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}} \sum_{p \in \mathrm{P}(k,e)} f_p^*, \qquad \forall e \in \mathrm{E}, \tag{4.64}$$

$$x_e^{\mathrm{FF}*} = \sum_{k \in \mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}} \sum_{p \in \mathrm{P}(k,e)} f_p^*, \qquad \forall e \in \mathrm{E}. \tag{4.65}$$

These conversions ensure a standardised flow that can be used to check feasibility. We can now derive a feasibility checking model, henceforth known as *FEAS*, which is given in (4.66)–(4.70).

*FEAS*:

$$\min \qquad \mathbf{0} \tag{4.66}$$

$$\text{s.t.} \qquad \sum_{j \in \phi(i)} f_{ij}^{\mathrm{D}} - \sum_{j \in \phi(i)} f_{ji}^{\mathrm{D}} = \begin{cases} -x_{\beta(i)}^{\mathrm{FD}*}, & i \in \mathrm{U}, \\ x_{\beta(i)}^{\mathrm{FD}*}, & i \in \mathrm{D}, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall\, i \in \mathrm{V}, \tag{4.67}$$

$$\sum_{j\in\phi(i)} f_{ij}^{\mathrm{F}} - \sum_{j\in\phi(i)} f_{ji}^{\mathrm{F}} = \begin{cases} -x_{\beta(i)}^{\mathrm{FF}*}, & i \in \mathrm{D}, \\ x_{\beta(i)}^{\mathrm{FF}*}, & i = q, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall i \in \mathrm{V}, \tag{4.68}$$

$$f_{ij}^{\mathrm{D}} + f_{ji}^{\mathrm{D}} = x_e^{\mathrm{FD}*}, \quad \forall (ij) = e \in \mathrm{E}, \tag{4.69}$$

$$f_{ij}^{\mathrm{F}} + f_{ji}^{\mathrm{F}} = x_e^{\mathrm{FF}*}, \quad \forall (ij) = e \in \mathrm{E}. \tag{4.70}$$

Constraint sets (4.67) and (4.68) set the supply and demand in the distribution and feeder networks respectively, while also ensuring flow conservation. To ensure the flow does not exceed the allocated capacity, equations (4.69) and (4.70) are included for the distribution and feeder networks respectively.

If there exists a feasible solution to *FEAS*, the PONDP solution is feasible in terms of connectivity. If the model is infeasible, relax the capacity constraints and re-solve the model. In other words, replace equations (4.69) and (4.70) with the following equations respectively:

$$f_{ij}^{\mathrm{D}} + f_{ji}^{\mathrm{D}} \leq x_e^{\mathrm{FD}*}, \qquad \forall (ij) = e \in \mathrm{E}, \tag{4.71}$$

$$f_{ij}^{\mathrm{F}} + f_{ji}^{\mathrm{F}} \leq x_e^{\mathrm{FF}*}, \qquad \forall (ij) = e \in \mathrm{E}. \tag{4.72}$$

If this relaxed model is feasible, it means that the PONDP solution is *feasible* in terms of connectivity, but *sub-optimal* due to excess reserved capacity on one or more edges.

**Metric calculation**

Since *FEAS* can only determine feasibility in terms of fibre routing, we need to ensure all physical constraints, i.e. path length constraints, hold. To this end, we calculate a number of metrics for the feasible solution. First, calculate all used paths between ONUs and splitters and between splitters and the CO. For *ARC*, the paths can be determined for every commodity $k \in \mathrm{K}$ by adding all edges for which $x_{ek}^{\mathrm{TR}} = 1, e \in \mathrm{E}$. For *PATH*, we already have access to all used paths, i.e. all paths $p \in \mathrm{P}$ for which $f_p \geq 1$. For the Benders decomposed formulation, we can re-use the last feasible solutions $\boldsymbol{f}^{\mathrm{D}}$

and $\boldsymbol{f}^{\mathrm{F}}$ from the sub-problems, or re-solve the sub-problems with the solution vectors $\boldsymbol{x}^{\mathrm{FD}*}$ and $\boldsymbol{x}^{\mathrm{FF}*}$.

Once all the used paths are available, go through all feeder and distribution paths connected to every splitter, and calculate the minimum, maximum and average of each of the following metrics:

- **Distribution network reach** - The lengths of fibre in the distribution network in meters.
- **Feeder network reach** - The lengths of fibre in the feeder network in meters.
- **Global network reach** - The combined lengths of fibre in the distribution and feeder networks for every splitter in meters.
- **Differential distance** - The difference in fibre length from CO to ONU through each splitter in meters.
- **Attenuation at ONU** - The calculated attenuation at every ONU, including connector, splice, splitter and fibre losses in dB.
- **Spare link budget at ONU** - Spare link budget is calculated by subtracting the attenuation at an ONU from the total optical power budget $\mathrm{P_b}$.

By ensuring that all maximum network reach parameters are below the path length limits calculated beforehand, we can ensure physical constraint feasibility. Additionally, any negative spare attenuation values indicate an infeasible solution, since the received power at one or more ONUs is lower than the receiver sensitivity.

## 4.5 Results and analysis

All models are implemented in `C++` using the *Concert* extensions of IBM ILOG CPLEX 12.6 [49] and the Qt 5.2.1 framework [137]. Experiments are run on an Intel i7 with

8 threads, running at 3.5 GHz, while a total of 16 GiB of main memory are allocated for every run. When memory usage exceeds 24 GiB, the run is stopped with an *out-of-memory* error. If the time limit of 2 hours is exceeded, which includes all path calculation times, the current best solution is saved, along with the respective optimality gap, if one exists. Path calculation is done concurrently on all available cores to improve computational performance.

Both the baseline and scalability tests are done using three different formulations: *ARC*, *PATH* and $BENDERS_C$. For the baseline tests, nine different real-world data sets are used, all hypothetical deployments within South Africa, ranging in size from 33 to 190 ONUs and 8 to 52 splitters, and classified in one of three categories: urban, suburban or rural. These data sets will give real-world results and will be used to verify and validate the models in question. Since we want to compare the different models with regard to their numerical values and resulting topology, the baseline input graphs considered are small. Larger graphs will likely be plagued by the scalability issues mentioned in each model section, obfuscating the desired test outcome.

Scalability tests are then conducted using a range of input graphs, with an increasing number of ONUs and splitters, manufactured from five hypothetical PON deployments within Germany. These tests are used to determine the average sensitivity of each model to the sizes of sets D, U, V and E. Design parameters used for all formulations are given in table 4.1.

The baseline numerical results are illustrated in table 4.2, showing dataset parameters, computation times, $t_s$, and optimality gaps for each formulation. $|\mathrm{V_R}|$ and $|\mathrm{E_R}|$ denote the number of vertices and edges in $\mathcal{G}$ *after* first- and second-order data set reductions have been performed, respectively. In all results, bold font indicates the entry with the best computational performance. As expected, the formulations are equivalent, with all optimal solution values identical across the three formulations where available. Furthermore, all solutions were validated as feasible using *FEAS*.

Interestingly, *PATH* provides the best computational performance where the path cal-

Table 4.1: Design parameters

| Parameter | Symbol | Value |
|---|---|---|
| Fixed OLT setup cost | $c^{\mathrm{CO}}$ | R 10,000 |
| Splitter capacity | $\kappa$ | 64 |
| Splitter unit cost | $c^{\mathrm{SP}}$ | R 3,000 |
| ONU unit cost | $c^{\mathrm{ONU}}$ | R 500 |
| Average trenching cost | $c^{\mathrm{TR}}$ | R 300/m |
| Average fiber cost | $c^{\mathrm{F}}$ | R 50/m |
| Maximum differential distance | $\ell_{\max}^{\mathrm{diff}}$ | $20 \times 10^3$ m |
| Maximum network reach | $\ell_{\max}^{\mathrm{total}}$ | $60 \times 10^3$ m |
| Total connector loss ($\times 4$) | $a_{\mathrm{c}}$ | 2 dB |
| Total splice loss ($\times 4$) | $a_{\ell}$ | 0.4 dB |
| Fibre attenuation | $a_{\mathrm{f}}$ | 0.4 dB/km |
| Power budget | $\mathrm{P_b}$ | 23 dB |

Table 4.2: Baseline numerical results

| | | | | | *ARC* | | *PATH* | | *BENDERS*$_C$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data set | $\lvert V_R\rvert$ | $\lvert E_R\rvert$ | $\lvert U\rvert$ | $\lvert D\rvert$ | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] |
| `urban1r` | 256 | 280 | 92 | 26 | **12.7** | **>2 h** | >2 h | | 25.6 | >2 h |
| `urban2r` | 198 | 207 | 87 | 13 | – | 495 | – | **402** | – | 3,272 |
| `urban3r` | 113 | 118 | 44 | 18 | – | 366 | – | **16.5** | – | 266 |
| `suburb1r` | 113 | 133 | 33 | 17 | – | 93.9 | >24 GiB | | – | **13.9** |
| `suburb2r` | 118 | 126 | 51 | 8 | – | 85.6 | – | **53.9** | – | 178 |
| `suburb3r` | 295 | 314 | 103 | 41 | >24 GiB | | >24 GiB | | **3.84** | **>2 h** |
| `rural1r` | 431 | 474 | 166 | 38 | >24 GiB | | >24 GiB | | >24 GiB | |
| `rural2r` | 440 | 450 | 190 | 52 | >24 GiB | | – | **4,084** | >24 GiB | |
| `rural3r` | 409 | 409 | 188 | 23 | – | 2,440 | – | **1.75** | – | 2,050 |

culation can be done in a feasible time, while simultaneously failing to provide a solution in four cases, more than both *ARC* and *BENDERS*$_C$. Its performance using `rural3r` is particularly impressive. Upon further investigation, it is apparent that due to the tree-like structure of the data set, the number of possible paths are comparatively low, resulting in superior performance. *ARC* provides the best solution for the `urban1r` data set, due to its tighter LP relaxation, and therefore, better lower bound, in comparison to *BENDERS*$_C$. Finally, *BENDERS*$_C$ shows decent performance, on average only 23.6 % behind *ARC* in terms of optimal solution computation times, while also being the only formulation to give a solution for `suburb3r`.

Table 4.3: Scalability numerical results

| Data set | $\|V_R\|$ | $\|E_R\|$ | $\|U\|$ | $\|D\|$ | *ARC* | | *PATH* | | $BENDERS_C$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] |
| micronet | 27 | 31 | 10 | 3 | – | 0.64 | **–** | **0.13** | – | 0.58 |
| citynet1 | 90 | 122 | 10 | 10 | – | 7.99 | – | 308 | **–** | **1.35** |
| citynet2 | 145 | 169 | 50 | 5 | **–** | **11.0** | >24 GiB | | – | 15.1 |
| citynet3 | 465 | 655 | 50 | 50 | >24 GiB | | >2 h | | **4.48** | **>2 h** |
| citynet4 | 305 | 364 | 100 | 10 | **1.91** | **>2 h** | >24 GiB | | 7.02 | >2 h |
| citynet5 | 565 | 755 | 100 | 50 | >24 GiB | | >2 h | | **45.9** | **>2 h** |
| citynet6 | 766 | 959 | 200 | 50 | >24 GiB | | >2 h | | **79.7** | **>2 h** |
| mednet1 | 76 | 107 | 10 | 10 | – | 3.02 | – | 220 | **–** | **1.20** |
| mednet2 | 124 | 138 | 50 | 5 | **–** | **11.4** | – | 415 | – | 27.5 |
| mednet3 | 350 | 467 | 50 | 50 | >24 GiB | | >2 h | | **7.43** | **>2 h** |
| mednet4 | 253 | 287 | 100 | 10 | **–** | **3,533** | >24 GiB | | 11.6 | >2 h |
| mednet5 | 450 | 568 | 100 | 50 | >24 GiB | | >2 h | | **18.0** | **>2 h** |
| mednet6 | 650 | 770 | 200 | 50 | >24 GiB | | >2 h | | >24 GiB | |
| hugenet1 | 61 | 77 | 10 | 10 | – | 13.7 | – | 72.6 | **–** | **1.92** |
| hugenet2 | 118 | 128 | 50 | 5 | – | 78.6 | – | 74.7 | **–** | **11.1** |
| hugenet3 | 517 | 775 | 50 | 50 | >24 GiB | | >2 h | | **–** | **6,975** |
| hugenet4 | 239 | 258 | 100 | 10 | **–** | **4,818** | >24 GiB | | 3.45 | >2 h |
| hugenet5 | 617 | 875 | 100 | 50 | >24 GiB | | >2 h | | **28.5** | **>2 h** |
| hugenet6 | 817 | 1078 | 200 | 50 | >24 GiB | | >2 h | | **85.8** | **>2 h** |
| subnet1 | 90 | 126 | 10 | 10 | – | 3.56 | – | 101 | **–** | **1.25** |
| subnet2 | 124 | 137 | 50 | 5 | **–** | **23.6** | – | 164 | – | 43.8 |
| subnet3 | 467 | 681 | 50 | 50 | >24 GiB | | >2 h | | **11.1** | **>2 h** |
| subnet4 | 271 | 314 | 100 | 10 | **0.18** | **>2 h** | >24 GiB | | 6.18 | >2 h |
| subnet5 | 567 | 783 | 100 | 50 | >24 GiB | | >2 h | | **26.3** | **>2 h** |
| subnet6 | 767 | 985 | 200 | 50 | >24 GiB | | >2 h | | **44.2** | **>2 h** |

Scalability numerical results are given in table 4.3, displaying major scalability issues for both *ARC* and *PATH*. While *ARC* mostly suffers from memory issues, providing solutions for only half of the data sets tested, calculating all possible paths for *PATH* in the time limit proves difficult, resulting in solutions only 32 % of the time. Conversely, $BENDERS_C$ fares much better, providing solutions in all but one of the test cases. *ARC* clearly has an advantage when the commodity count is low, i.e. when $|U| \times |D|$ is low, giving the best solution times for all datasets with 100 ONUs and 10 splitters. A summary of the qualitative results is presented in table 4.4.

Table 4.4: Qualitative baseline and scalability results

| Formulation | Percentage of instances with | | | |
|---|---|---|---|---|
| | No solution (time limit) [%] | No solution (memory limit) [%] | Sub-optimal solution [%] | Optimal solution [%] |
| *ARC* | – | 44.1 | 8.82 | 47.1 |
| *PATH* | 38.2 | 23.5 | – | 38.2 |
| $BENDERS_C$ | – | 8.82 | 47.1 | 44.1 |

## 4.6 Conclusion

In this chapter, two common approaches to modelling PONDP were discussed, one based on arc-flow, *ARC*, and another based on paths, *PATH*. Since the inclusion of path length constraints is crucial to implement physical network constraints and ensure feasible network designs, both models were extended with this in mind. The flow variables in *ARC* were redefined for all commodity pairs, increasing the number of variables in the model significantly. *PATH* allowed for easy integration of path length constraints but required the calculation of all possible paths between commodities.

To alleviate these scalability issues, *PATH* was decomposed using Benders into a master and two sub-problems, one for the feeder network and one for the distribution network. The sub-problem division resulted in structures we could exploit to implement column generation, avoiding the need to calculate all possible paths at the onset and resulting in a formulation known as $BENDERS_C$. Path length constraints were then implemented implicitly through the separation of cuts.

Results were checked for feasibility, verified using inter-model verification and validated using both face validity and the calculation of network metrics. As expected, *ARC* and *PATH* suffered from severe scalability issues in terms of memory usage and path calculation time respectively. Each formulation excelled in different cases, with the number of commodities and the graph structure being the biggest indicators of computation times for *ARC* and *PATH* respectively.

Even though $BENDERS_C$ did not suffer from the same scalability issues, the resulting solutions had large optimality gaps for large graphs, indicating either weak lower- or upper bound calculation. To improve lower bounds, valid inequalities can be separated, while upper bounds can be improved through solution generation. By utilising both of these techniques, both computational performance and solution quality can be improved, resulting in a more scalable framework. For this reason, chapter 5 will look at deriving strengthening cuts and primal heuristics.

# Chapter 5

# Solution improvement

---

*This chapter proposes techniques to improve both the lower and upper bounds of PONDP, including strengthening cuts and a primal heuristic. Additionally, algorithmic implementation improvements, which increase the number and quality of cuts separated, are proposed. Finally, a complete computational study is provided which compares results with that obtained in chapter 4, showing the efficacy of the proposed additions.*

---

## 5.1 Introduction

The results for $BENDERS_C$ in chapter 4 were promising, offering better scalability than the common formulations without sacrificing solution quality. However, scalability is still an issue, with it still not being able to provide good quality solutions for medium-sized networks, i.e. in the order of 100 ONUs. Therefore, in this chapter we discuss improving computational performance of $BENDERS_C$ through the use of strengthening cuts and heuristic solution generation.

To strengthen the lower bound during computation, two sets of valid inequalities are

separated at every node of the branch-and-cut procedure: one based on Steiner tree connectivity and one on network flow. Additionally, to improve the upper bound during the solution process, i.e. the best integer solution found thus far, feasible solutions will be generated using fractional variable values as well as path information contained in the sub-problems. Combined, these techniques should improve the convergence rate of the solver, resulting in increased computational performance.

## 5.2 Connectivity cuts

The aptly named *connectivity cuts* are cut inequalities for the Steiner tree problem that ensure connectivity in the Steiner subgraph [138]. These cuts were implemented by Bley et al. for a Lagrangian decomposed version of *ARC* [117] as well as for ConFL [124]. Since we proved that both the distribution and feeder networks are Steiner trees in section 4.3, these cuts can be utilised to strengthen the model. Furthermore, cuts can be derived for the *global network*, i.e. the sum of the distribution and feeder networks, since we know that the CO will ultimately be connected to each ONU.

Every connectivity cut has the form as shown in equation (5.1) [138], ensuring that components are connected to their commodity pairs either within the subset W or via edges crossing the subset boundary.

$$x(\delta(W)) \geq 1, \quad \forall\, W \subset V, \tag{5.1}$$

with $\delta(W)$ the cut induced by subset $W \subset V$, i.e. the set of edges $(v, w)$, with $w \in W$ and $v \in V \setminus W$. By adapting these for the distribution network using undirected edges, we can ensure connectivity between ONUs and splitters. Therefore, the distribution connectivity cuts can be derived as follows:

$$\text{CON}_D: \quad \sum_{i \in W \cap D} \psi_i + \sum_{e \in \delta(W)} x_e^{TD} \geq 1, \quad \forall\, W \subset V, W \cap U \neq \varnothing. \tag{5.2}$$

The first term on the left-hand side of (5.2) counts the number of active splitters in the subset while the second term counts the number of active distribution edges crossing

the subset boundary. Therefore, by ensuring that this adds to at least one, we ensure that there are sources for connectivity for each subset that contains at least one ONU.

Similarly, for the feeder network, we need to ensure connectivity between used splitters and the CO. Adapting (5.1) for the feeder network, we can derive feeder connectivity cuts (5.3), ensuring that the connected path to a splitter must either originate from within the subset or it must cross the boundary edges.

$$\begin{aligned} \text{CON}_\text{F}: \quad & \sum_{\text{W}\cap\{q\}} 1 + \sum_{e\in\delta(\text{W})} x_e^{\text{TF}} \geq \psi_i, \quad \forall\, \text{W} \subset \text{V}, \forall i \in \text{W} \cap \text{D} \neq \varnothing, \\ & \therefore \sum_{e\in\delta(\text{W})} x_e^{\text{TF}} \geq \psi_i, \quad \forall\, \text{W} \subset \text{V} \setminus \{q\}, \forall i \in \text{W} \cap \text{D} \neq \varnothing. \end{aligned} \tag{5.3}$$

To avoid a large number of slack cuts, we define (5.3) for all vertices except $q$. Furthermore, note that (5.3) is defined for each splitter contained in the subset W, which means $|\text{W} \cap \text{D}|$ cuts are separated for every subset W.

Finally, to ensure global connectivity, we derive a cut that ensures that either the CO is present in the subset or that there must be some active boundary edges in the global network. These global connectivity cuts are given below:

$$\begin{aligned} \text{CON}_\text{G}: \quad & \sum_{\text{W}\cap\{q\}} 1 + \sum_{e\in\delta(\text{W})} x_e^{\text{TR}} \geq 1, \quad \forall\, \text{W} \subset \text{V}, \text{W} \cap \text{U} \neq \varnothing, \\ & \therefore \sum_{e\in\delta(\text{W})} x_e^{\text{TR}} \geq 1, \quad \forall\, \text{W} \subset \text{V} \setminus \{q\}, \text{W} \cap \text{U} \neq \varnothing. \end{aligned} \tag{5.4}$$

### 5.2.1 Separation

It is widely known that Steiner tree inequalities can be separated in polynomial time [117, 138] using the max-flow algorithm. Therefore, we utilise the same procedure to separate connectivity cuts. In particular, the procedure to separate distribution connectivity cuts are as follows:

1. **Fractional values** - Fractional variable values $\psi_i^*$ and $x_e^{\text{TD}*}$ are found from the LP relaxation of $PON_M$.

2. **Augment graph** - Create an auxiliary graph from $\mathcal{G}$ and add an artificial root node $r$ connected to all facilities $i \in \mathrm{D}$ via edges $(i, r)$. This root node will serve as a super source.

3. **Max-flow** - Calculate the maximum flow, using the highest label preflow-push algorithm by Goldberg and Tarjan [81], from each ONU sink to the artificial root node using $x_e^{\mathrm{TD}*}$ as capacity on edge $e \in \mathrm{E}$ and $\psi_i^*$ as capacity on edge $(i, r)$.

4. **Min-cut** - If the maximum flow is less than 1, it constitutes a violated cut. Project the maximum flow to a minimum *s-t* cut and add (5.2) by setting W to the source subset $s$.

The feeder and global connectivity cut separation follow the same idea, except that they have no artificial root nodes. Both separation procedures use the CO as the source, but the feeder connectivity separation uses splitters as sinks and edge capacities $x_e^{\mathrm{TF}*}$ while the global connectivity separation procedure uses ONUs as sinks with edge capacities $x_e^{\mathrm{TR}*}$. In each case, whenever the maximum flow is less than 1, the projected min-cut is used as the subset W and the corresponding connectivity cut is added to $PON_M$.

## 5.3 Flow-cutset inequalities

Flow-cutset inequalities are cuts widely used in network flow modelling to ensure that the required demand is met by the allocated supply. These inequalities are specified across a subset of vertices, and the reasoning is simple: given a partitioning of vertices $\mathrm{W} \subset \mathrm{V}$ and $\mathrm{W}' \subset \mathrm{V} \setminus \mathrm{W}$, the demand in W must either be supplied by sources in W or by sources in W′. If the sources are located in W′, the flow must cross the boundary edges $(v, w), v \in \mathrm{W}, w \in \mathrm{W}'$. Therefore, they are similar to the connectivity cuts, except that they use flow instead of just connectivity.

Recall from sections 4.3.1 and 4.3.2 that $x_{\beta(i)}^{\mathrm{FD}}$ and $x_{\beta(j)}^{\mathrm{FD}}$ represent the flow coming in to a splitter $i \in \mathrm{D}$ and going out of an ONU $j \in \mathrm{U}$ in the distribution network respectively.

In addition, in the feeder network, $x^{\mathrm{FF}}_{\beta(q)}$ represents the number of fibres going in to the CO and $x^{\mathrm{FF}}_{\beta(i)}$ is the number of fibres leaving a splitter $i \in \mathrm{D}$ (which is also set by $\psi_i$).

In the distribution network, our demand is the ONU fibre demand and the supply is the capacity reserved at each splitter. Therefore, for our flow-cutset inequalities, we want to ensure that enough supply is reserved at splitters to meet the total ONU demand. The resulting inequality is as follows:

$$\mathrm{CUT_D}: \quad \sum_{i \in \mathrm{W} \cap \mathrm{D}} x^{\mathrm{FD}}_{\beta(i)} + \sum_{e \in \delta(\mathrm{W})} x^{\mathrm{FD}}_{e} \geq \sum_{j \in \mathrm{W} \cap \mathrm{U}} x^{\mathrm{FD}}_{\beta(j)}, \quad \forall\, \mathrm{W} \subset \mathrm{V}, \mathrm{W} \cap \mathrm{U} \neq \varnothing. \tag{5.5}$$

The first term on the left-hand side counts the total supply in the subset while the second term adds the flow over all boundary edges. This is equivalent to the total supply, which must be greater or equal to the term on the right-hand side of (5.5), i.e. the total ONU demand.

For the feeder network, used splitters have a demand for fibre, which the CO can supply. Hence, we want to ensure enough fibre is reserved at the CO so that all used splitters can be connected. The feeder flow-cutset inequality can therefore be written as follows:

$$\mathrm{CUT_F}: \quad \sum_{\mathrm{W} \cap \{q\}} x^{\mathrm{FF}}_{\beta(q)} + \sum_{e \in \delta(\mathrm{W})} x^{\mathrm{FF}}_{e} \geq \sum_{i \in \mathrm{W} \cap \mathrm{D}} x^{\mathrm{FF}}_{\beta(i)}, \quad \forall\, \mathrm{W} \subset \mathrm{V}, \mathrm{W} \cap \mathrm{D} \neq \varnothing. \tag{5.6}$$

Similarly, the total supply consists of the supply at the CO (if it is contained in the subset) as well as the flow crossing the subset boundary. This must then be greater or equal to the total splitter demand, which is calculated on the right-hand side of (5.6).

### 5.3.1 Separation

The process to separate the flow-cutset inequalities is analogous to that of the connectivity cuts, with some differences in the source and sink node locations and the violation criteria. Without loss of generality, consider again the distribution flow-cutset separation:

1. **Fractional values** - Variable values $x_e^{\text{FD}*}$ are found from the LP relaxation of $PON_M$.

2. **Augment graph** - Create an auxiliary graph from $\mathcal{G}$ and add super source node $r$ connected to all facilities $i \in \text{D}$ via edges $(i, r)$ and super sink node $u$ connected to all ONUs $j \in \text{U}$ via edges $(j, u)$.

3. **Max-flow** - Calculate the maximum flow between the artificial nodes using $x_e^{\text{FD}*}$ as capacity on edge $e$, $x_{\beta(i)}^{\text{FD}*}$ as capacity on edge $(i, r)$ and $x_{\beta(j)}^{\text{FD}*}$ as capacity on edge $(j, u)$.

4. **Min-cut** - Let $d_{\text{net}} = \sum_{j \in \text{W} \cap \text{U}} x_{\beta(j)}^{\text{FD}*} - \sum_{i \in \text{W} \cap \text{D}} x_{\beta(i)}^{\text{FD}*}$ be the total net demand. If the maximum flow is less than $d_{\text{net}}$, a violated cut can be derived by projecting the maximum flow to a minimum *s-t* cut and adding (5.5) using the subset induced by the cut.

For the feeder flow-cutset separation, the idea is similar, except that a single source (the CO) and a super sink consisting of all used splitters with auxiliary edge capacities $x_{\beta(i)}^{\text{FF}*}$ are used. Additionally, a violated cut is found whenever the maximum flow is less than $d_{\text{net}} = \sum_{i \in \text{W} \cap \text{D}} x_{\beta(i)}^{\text{FF}*} - \sum_{\text{W} \cap \{q\}} x_{\beta(q)}^{\text{FF}*}$.

Along with this method, it is also possible to include the flow-cutset inequalities using the same cut subsets determined while separating the connectivity cuts. This allows us to simultaneously set the connectivity and increase the flow with one max-flow calculation, the caveat being that we may potentially add a number of redundant flow-cutset inequalities. This set of inequalities, derived from the connectivity cuts, we call *connectivity flow* cuts.

## 5.4 Implementation improvements

As well as the more theoretical strengthening cuts detailed in the previous section, a number of algorithmic improvements are also implemented. These include variable

management as well as separation improvement techniques.

### 5.4.1 Non-basic variable removal

When a column generation approach is followed, it is often necessary to manage the enormous number of variables added to the problem to avoid variable inflation. During computation, it is common to add a large number of variables only to never use them again in subsequent iterations. To avoid this, we can remove variables that have not entered the simplex basis for $n$ successive iterations, keeping the LPs small.

Finding a value of $n$ that is small enough to result in scalability gains while being large enough to avoid the costly operation of regenerating the variable later is extremely problem specific however, and can even depend on the order in which nodes are processed in the branch-and-cut tree. Therefore, this feature is not included by default in the computations, but is implemented to illustrate the efficacy of the approach.

### 5.4.2 Epsilon max-flow

During separation of the connectivity and flow-cutset cuts, the push-relabel max-flow algorithm is utilised to determine minimum cut subsets in a subgraph. Due to the nature of the variables used in the capacity initialisation of the subgraph, a large number of values are zero, resulting in a minimum cut that is not necessarily arc-minimal. Therefore, for the Steiner tree problem, Koch and Martin [138] proposed adding a *creep flow* value, a small $\epsilon$ value in the order of $10^{-6}$, to each arc capacity to improve the quality of the max-flow solution. While this change increases the run time of the algorithm since more arcs have to be considered, the resulting cut is vastly improved, reducing the number of LP iterations required to solve the model. This implementation approach is followed in all max-flow computations.

### 5.4.3 Nested and reverse cuts

During separation of Steiner inequalities for the Steiner tree problem, Koch and Martin [138] and Chopra et al. [139] proposed two techniques to increase the number of cuts separated in each iteration. Since the connectivity cuts are directly derived from these cuts, these techniques apply to them as well.

The first technique proposed by Koch and Martin is the separation of so called *nested cuts*. After a minimum cut has been found by the max-flow algorithm, set all arc capacities in the cut to 1 and re-run the algorithm. The resulting cut will therefore contain the next minimum cut if it is assumed that the LP raises the capacities accordingly. By repeating this process until the max-flow is equal to the required demand, we can separate more cuts at each iteration and potentially reduce the required number of LP iterations.

Secondly, Chopra et al. proposed the separation of *back cuts* or *reverse cuts*, where, after finding the initial cut, the flow on each arc, as well as the source and sink nodes, are swapped and the algorithm re-run, resulting in a minimum cut in the opposite direction. Similar to the nested cuts, this also increases the number of cuts separated at each iteration, potentially improving overall computational performance.

Both the nested and reverse cuts are tested to determine their efficacy for PONDP, where the input graphs are ultimately undirected.

## 5.5 Primal heuristic

As mentioned in the introduction, the improvement of the upper bound is analogous to generating higher quality solutions earlier during computation. To this end, we note that even though the path variables were projected out of the master, the subproblem solutions still contain feasible routings, in particular, the values of $f^{\mathrm{D}}$ and $f^{\mathrm{F}}$, which we can potentially use to construct a feasible solution. This idea is used to formulate

the following greedy primal heuristic, henceforth known as *PRIMAL*:

1. **Path grouping** - Group all used distribution and feeder paths (where $f_p^{\mathrm{D}*} \geq \epsilon$ and $f_p^{\mathrm{F}*} \geq \epsilon$ respectively, $0 < \epsilon \ll 1$) according to their connected splitter.

2. **Path reduction** - Since multiple paths may be used for a commodity in the subproblems, we now go through all connected distribution and feeder paths and remove all but the shortest path between each commodity.

3. **Capacity check** - Check the number of distribution paths connected to each splitter and discard the solution if it violates the splitter capacity $\kappa$.

4. **Path length check** - Iterate through all splitters and calculate the minimum and maximum total fibre length, i.e. the sum of the distribution and feeder fibre lengths. Furthermore, calculate the maximum differential length by subtracting the minimum total fibre length from the maximum. If the maximum length exceeds the path length limit, $\ell_{\max}^{\mathrm{total}}$, or the maximum differential length exceeds the differential limit, $\ell_{\max}^{\mathrm{diff}}$, discard the solution.

5. **Edge usage count** - For each edge, count the number of distribution and feeder paths traversing it. This then becomes the values of $\boldsymbol{x}^{\mathrm{FD}}$ and $\boldsymbol{x}^{\mathrm{FF}}$ respectively. If for any edge $e \in \mathrm{E}$, $x_e^{\mathrm{FD}} \geq 1$, set $x_e^{\mathrm{TD}}$ to 1, otherwise set it to 0. Set $\boldsymbol{x}^{\mathrm{TF}}$ in the same way.

6. **Solution construction** - All additional variable values are then derived from the used edges, including setting $\psi_i$ to 1 if $x_{\beta(i)}^{\mathrm{FD}} \geq 1$.

The primal heuristic should be able to generate a feasible solution before edge capacity has been sufficiently lifted in $PON_M$ to form contiguous paths, ensuring that a good upper bound is found early in the computation process.

## 5.6 Computational study

Since the aim of this chapter is to improve computational performance, a computational study is now given to determine the efficacy of the strengthening and solution generation procedures detailed in the preceding sections.

### 5.6.1 Experimental methodology

To ensure comparability, the experimental methodology as described in 4.4 is followed for this section as well, using the same baseline and scalability instances (with the same number of elements) to determine feasibility and computational performance of the proposed strengthening procedures. Therefore, results presented in this section is comparable to the results presented in section 4.5. However, in terms of result interpretation, the focus is moved towards computational performance in this section. All experiments are run on the same hardware and implemented in `C++` using the *Concert* extensions of IBM ILOG CPLEX 12.6 and the Qt 5.2.1 framework. In addition, all results are verified and validated using the same procedure as discussed in section 4.4.3, including feasibility checking and solution metric verification.

Strengthening cuts are separated at each node of the branch-and-cut tree, both for fractional and integer solutions. A number of strategies can be followed to improve convergence in this regard, including separating only at nodes with quadratic index (e.g. see [117]), or every $n$-th node, but as these are more trial-and-error approaches, being instance specific, they will not be used in this section.

All combinations of the connectivity cuts and flow-cutset inequalities are tested to determine their efficacy. In the case where both connectivity and flow-cutset inequalities are used simultaneously, connectivity flow cuts are also included. The *PRIMAL* heuristic is tested separately as well as alongside the best configuration of connectivity and flow-cutset inequalities. The nested and reverse cuts implementation improvements are tested using the same configuration.

Finally, to showcase the improvement in removing non-basic variables from the LP after a set number of iterations, a small case study is done using data set `subnet3`. For this experiment, we include all connectivity and flow-cutset inequalities and adjust the threshold $n$, i.e. the number of successive iterations a path must be out of the simplex basis before it is removed from the problem, between 20 and 300. An instance where columns are not removed, i.e. $n = \infty$, is also included. The number of columns generated and removed are recorded, along with the time to solve, the peak memory usage, and the number of nodes visited in the branch-and-cut tree.

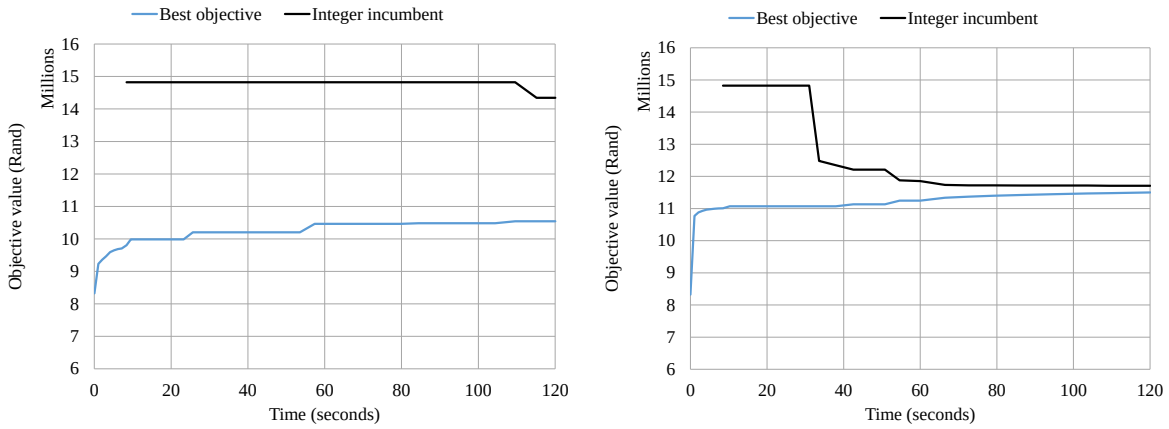
### 5.6.2 Results and analysis

#### Connectivity and flow-cutset inequalities

An example illustrating the effect of strengthening cuts is given in figure 5.1, showing the first 2 minutes of branch-and-cut convergence for the `rural3r` data set. It is evident that with the cuts included, the lower bound increases almost immediately to a value only attained after 2 minutes by the standard model, and at the termination point it is 9.1 % higher. Additionally, the number of violated cuts added increased from 835 to 3,231, which, apart from improving the lower bound, allows CPLEX to quickly find a better integer feasible solution using rounding heuristics.

The first six entries in table 5.1 show the baseline test results for the strengthening cuts, with dramatic improvements across the board. Entries in bold highlight the best result for each data set. It is evident that the connectivity cuts are very effective, resulting in almost an order of magnitude improvement across all instances, with an average computation time reduction of 86 %. Data set `suburb3r` is solved to optimality in only 2 minutes, in comparison to the sub-optimal solution found in chapter 4 in 2 hours.

Even though including just the flow-cutset inequalities improves the computational performance of $BENDERS_C$ in all but two instances, combining them with the connectivity cuts yields much better performance, displaying the best results in half of the




(a) No strengthening cuts

(b) With connectivity and flow-cutset cuts

Figure 5.1: MIP convergence plot for the first 2 minutes of the `rural3r` data set

instances. Note that `suburb3r` is solved in just over a minute, which is more than two orders of magnitude faster than with no cuts added. Finally, even though the overall computational effort has been reduced by 89 %, `rural1r` proved to still be unsolvable under the specified memory constraints.

The results of the scalability test is given in the rest of table 5.1, showing the same performance increases we saw in the baseline test. Connectivity cuts resulted in a 77 % decrease in computational effort for instances that could be solved to optimality by the standard model. Additionally, it could optimally solve 6 additional instances, with an average time of 20 minutes. In the instances where $BENDERS_C$ with connectivity cuts could not find the optimal solution, the solution quality increased substantially, with an optimality gap reduction of 71 % on average.

Including just the flow-cutset inequalities did not show promising results, beating the standard model in only 9 of the instances, however, combining the flow and connectivity cuts resulted in the best overall configuration, providing the best performance in 15 of the 25 scalability instances. An 82 % decrease in computational effort is seen in comparison to the model with no cuts added, and an additional 6 instances could be solved to optimality. A number of data sets showed better performance using only the connectivity cuts, suggesting that the gains achieved in lifting the flow are eclipsed

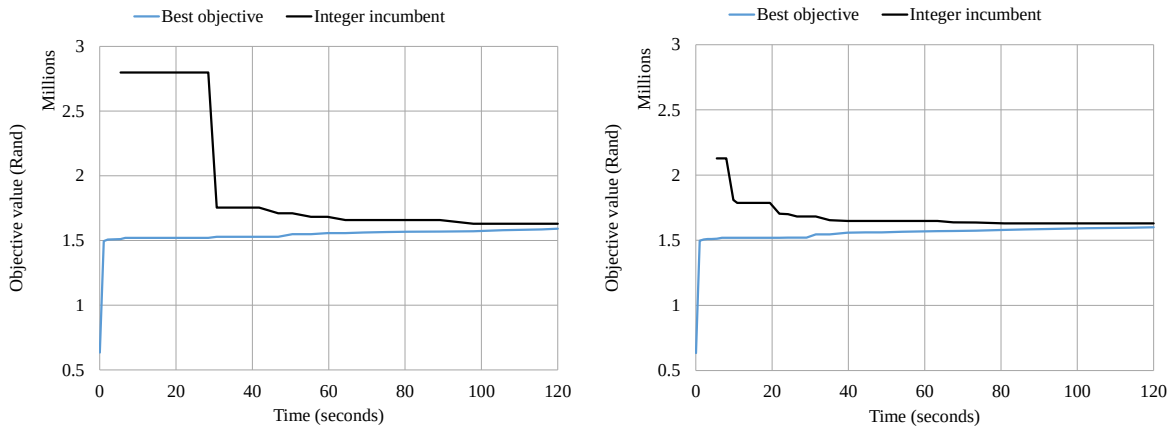
Table 5.1: Computational results - connectivity and flow-cutset inequalities

| Data set | *BENDERS*$_C$ including | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | No cuts | | (5.2)–(5.4) | | (5.5)–(5.6) | | (5.2)–(5.6) | |
| | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] |
| *Section 1: Baseline data sets* | | | | | | | | |
| urban1r | 25.6 | >2 h | **6.69** | **>2 h** | 22.2 | >2 h | 8.96 | >2 h |
| urban2r | – | 3,272 | – | **120** | – | 647 | – | 189 |
| urban3r | – | 266 | – | 30.8 | – | 190 | – | **16.0** |
| suburb1r | – | 13.9 | – | **3.37** | – | 14.3 | – | 3.68 |
| suburb2r | – | 178 | – | 25.1 | – | 167 | – | **12.3** |
| suburb3r | 3.84 | >2 h | – | 139 | – | 5,079 | – | **62.5** |
| rural1r | >24 GiB | | >24 GiB | | >24 GiB | | >24 GiB | |
| rural2r | >24 GiB | | **12.1** | **>2 h** | >24 GiB | | 12.8 | >2 h |
| rural3r | – | 2,050 | – | 526 | – | 3,215 | – | **360** |
| *Section 2: Scalability data sets* | | | | | | | | |
| micronet | – | 0.58 | – | 0.21 | – | 0.62 | – | **0.08** |
| citynet1 | – | 1.35 | – | 0.29 | – | 1.74 | – | **0.21** |
| citynet2 | – | 15.1 | – | **4.45** | – | 16.1 | – | 4.59 |
| citynet3 | 4.48 | >2 h | – | 150 | 2.63 | >2 h | – | **75.8** |
| citynet4 | 7.02 | >2 h | – | 355 | 3.02 | >2 h | – | **177** |
| citynet5 | 45.9 | >2 h | **3.49** | **>2 h** | 49.0 | >2 h | 6.45 | >2 h |
| citynet6 | 79.7 | >2 h | **71.2** | **>2 h** | 80.1 | >2 h | 71.5 | >2 h |
| mednet1 | – | 1.20 | – | 0.18 | – | 0.43 | – | **0.05** |
| mednet2 | – | 27.5 | – | 8.52 | – | 34.3 | – | **7.96** |
| mednet3 | 7.43 | >2 h | – | **149** | 9.99 | >2 h | – | 157 |
| mednet4 | 11.6 | >2 h | **1.70** | **>2 h** | 24.4 | >2 h | 3.05 | >2 h |
| mednet5 | 18.0 | >2 h | 3.25 | >2 h | 29.3 | >2 h | **3.02** | **>2 h** |
| mednet6 | >24 GiB | | **8.83** | **>2 h** | >24 GiB | | 21.7 | >2 h |
| hugenet1 | – | 1.92 | – | **0.45** | – | 3.00 | – | 0.51 |
| hugenet2 | – | 11.1 | – | 4.60 | – | 8.33 | – | **3.27** |
| hugenet3 | – | 6,975 | – | 108 | 2.35 | >2 h | – | **78.4** |
| hugenet4 | 3.45 | >2 h | – | **1,059** | 11.0 | >2 h | 0.37 | >2 h |
| hugenet5 | 28.5 | >2 h | 3.13 | >2 h | 24.0 | >2 h | – | **5,315** |
| hugenet6 | 85.8 | >2 h | >2 h | | 85.8 | >2 h | **20.1** | **>2 h** |
| subnet1 | – | 1.25 | – | **0.07** | – | 1.25 | – | 0.08 |
| subnet2 | – | 43.8 | – | **9.82** | – | 37.9 | – | 10.7 |
| subnet3 | 11.1 | >2 h | – | 267 | 6.90 | >2 h | – | **105** |
| subnet4 | 6.18 | >2 h | 1.23 | >2 h | 2.11 | >2 h | – | **1,500** |
| subnet5 | 26.3 | >2 h | – | 5,710 | 22.6 | >2 h | – | **3,046** |
| subnet6 | 44.2 | >2 h | 16.4 | >2 h | 87.3 | >2 h | **14.7** | **>2 h** |

by either the computational effort expended to separate the cuts or the additional time required in solving a larger master LP relaxation. Finally, even with the strengthening cuts, the citynet6 instance still shows low solution quality, seemingly indicating a poor upper bound.

**Primal heuristic**

Figure 5.2 shows an example of the effect the *PRIMAL* heuristic has on the upper bound during the computation of the `citynet4` instance. As expected, the first incumbent found after 5 seconds is already 24 % lower than the one CPLEX generates, and faster upper bound convergence is seen overall. However, it should also be noted that CPLEX gets good results as time progresses using its built-in general solution generation heuristics.




(a) Without *PRIMAL* (b) With *PRIMAL*

Figure 5.2: MIP convergence plot of the `citynet4` data set

The addition of improved upper bounds results in better performance for the majority of instances, as can be seen in table 5.2, which contains both baseline and scalability results. For 21 of the 34 data sets, the addition of the *PRIMAL* heuristic resulted in better performance than previously attained by using only the connectivity and flow-cutset cuts. On average, these datasets showed a 30 % improvement in computational performance where optimal solutions could be obtained, with a maximum of 69.9 % for `mednet3`. For sub-optimal solutions, solution quality improved by 25 % on average.

Most notably, a good solution is found for the `rural1r` data set, which could not be solved previously, and the solution quality of `citynet6` improved by 71.8 %. The remaining instances likely did not suffer from poor upper bounds, with the additional overhead inherent in finding a heuristic solution exceeding the gains achieved.

Table 5.2: Computational results - *PRIMAL* heuristic

| | *BENDERS*$_C$ including | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | No cuts | | *PRIMAL* | | (5.2)–(5.4) | | (5.2)–(5.4) & *PRIMAL* | | (5.2)–(5.6) | | (5.2)–(5.6) & *PRIMAL* | |
| Data set | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] |
| *Section 1: Baseline data sets* | | | | | | | | | | | | |
| urban1r | 25.6 | >2 h | 22.2 | >2 h | 6.69 | >2 h | **6.08** | **>2 h** | 8.96 | >2 h | 8.92 | >2 h |
| urban2r | – | 3,272 | – | 3,280 | – | **120** | – | 142 | – | 189 | – | 351 |
| urban3r | – | 266 | – | 203 | – | 30.8 | – | 24.1 | – | **16.0** | – | 22.3 |
| suburb1r | – | 13.9 | – | 13.3 | – | 3.37 | – | **1.54** | – | 3.68 | – | 2.50 |
| suburb2r | – | 178 | – | 160 | – | 25.1 | – | 26.4 | – | 12.3 | – | **9.94** |
| suburb3r | 3.84 | >2 h | – | 1,256 | – | 139 | – | 78.2 | – | **62.5** | – | 71.1 |
| rural1r | >24 GiB | | >24 GiB | | >24 GiB | | >24 GiB | | >24 GiB | | **9.92** | **>2 h** |
| rural2r | >24 GiB | | >24 GiB | | 12.1 | >2 h | **11.3** | **>2 h** | 12.8 | >2 h | 12.1 | >2 h |
| rural3r | – | 2,050 | – | 4,360 | – | 526 | – | 637 | – | 360 | – | **300** |
| *Section 2: Scalability data sets* | | | | | | | | | | | | |
| micronet | – | 0.58 | – | 0.61 | – | 0.21 | – | 0.62 | – | **0.08** | – | **0.08** |
| citynet1 | – | 1.35 | – | 1.42 | – | 0.29 | – | 0.29 | – | **0.21** | – | 0.25 |
| citynet2 | – | 15.1 | – | 13.7 | – | 4.45 | – | **3.55** | – | 4.59 | – | 4.00 |
| citynet3 | 4.48 | >2 h | 4.07 | >2 h | – | 150 | – | 118 | – | 75.8 | – | **68.2** |
| citynet4 | 7.02 | >2 h | 3.14 | >2 h | – | 355 | – | 497 | – | **177** | – | 296 |
| citynet5 | 45.9 | >2 h | 33.8 | >2 h | **3.49** | **>2 h** | 8.74 | >2 h | 6.45 | >2 h | 8.12 | >2 h |
| citynet6 | 79.7 | >2 h | 46.7 | >2 h | 71.2 | >2 h | **20.1** | **>2 h** | 71.5 | >2 h | 29.5 | >2 h |
| mednet1 | – | 1.20 | – | 0.42 | – | 0.18 | – | 0.12 | – | **0.05** | – | 0.06 |
| mednet2 | – | 27.5 | – | 31.5 | – | 8.52 | – | 8.20 | – | 7.96 | – | **6.21** |
| mednet3 | 7.43 | >2 h | – | 4,459 | – | 149 | – | 85.0 | – | 157 | – | **44.8** |
| mednet4 | 11.6 | >2 h | 12.3 | >2 h | 1.70 | >2 h | 1.12 | >2 h | 3.05 | >2 h | **1.06** | **>2 h** |
| mednet5 | 18.0 | >2 h | 24.5 | >2 h | 3.25 | >2 h | 2.67 | >2 h | 3.02 | >2 h | **2.37** | **>2 h** |
| mednet6 | >24 GiB | | 41.0 | >2 h | **8.83** | **>2 h** | 11.0 | >2 h | 21.7 | >2 h | 11.9 | >2 h |
| hugenet1 | – | 1.92 | – | 1.61 | – | **0.45** | – | 0.65 | – | 0.51 | – | 0.62 |
| hugenet2 | – | 11.1 | – | 10.9 | – | 4.60 | – | 4.11 | – | 3.27 | – | **2.41** |
| hugenet3 | – | 6,975 | 1.16 | >2 h | – | 108 | – | **33.6** | – | 78.4 | – | 40.3 |
| hugenet4 | 3.45 | >2 h | 4.21 | >2 h | – | **1,059** | – | 1,134 | 0.37 | >2 h | – | 3,264 |
| hugenet5 | 28.5 | >2 h | 28.1 | >2 h | 3.13 | >2 h | 1.07 | >2 h | – | 5,315 | – | **2,584** |
| hugenet6 | 85.8 | >2 h | 43.3 | >2 h | >2 h | | 19.9 | >2 h | 20.1 | >2 h | **14.9** | **>2 h** |
| subnet1 | – | 1.25 | – | 0.46 | – | **0.07** | – | 0.13 | – | 0.08 | – | 0.08 |
| subnet2 | – | 43.8 | – | 34.0 | – | 9.82 | – | 13.1 | – | 10.7 | – | **8.78** |
| subnet3 | 11.1 | >2 h | 4.45 | >2 h | 6.90 | 267 | – | 164 | – | 105 | – | **101** |
| subnet4 | 6.18 | >2 h | 2.44 | >2 h | 1.23 | >2 h | – | **929** | – | 1,500 | – | 1,012 |
| subnet5 | 26.3 | >2 h | 26.6 | >2 h | – | 5,710 | – | 3,225 | – | **3,046** | – | 5,102 |
| subnet6 | 44.2 | >2 h | >24 GiB | | 16.4 | >2 h | 16.5 | >2 h | 14.7 | >2 h | **14.3** | **>2 h** |

Removing all strengthening cuts and using only the *PRIMAL* heuristic resulted in two additional solved instances, with an overall improvement of 5.6 %. Where both ap-

proaches resulted in sub-optimal solutions, the solution quality improved by 19.4 %. In particular, the worst-case optimality gap when using *PRIMAL* was only 46.7 % in comparison to 85.8 % without it. Therefore, it is evident that improving the upper bound has noticeable effects in terms of solution quality and convergence, even without improving the lower bound.

**Nested and reverse cuts**

The results for the nested and reverse cut separation techniques are shown in table 5.3. Separating nested cuts performed worse on average, with an 11.9 % increase in computation time required to solve the data sets. For a few instances however, performance increased drastically, with `subnet4` and `hugenet3` showing improvements of 44.5 % and 38.1 % respectively. Additionally, using nested cuts allows `rural1r` to be solved within the memory constraints. For sub-optimal solutions, nested cuts provided only slightly lower quality solutions, with optimality gaps 7.9 % higher on average. The maximum optimality gap decreased from 71.5 % to 28.4 % however, suggesting that the nested cuts may improve convergence at the start, but lose their effectiveness further down the branch-and-cut tree as the effort to separate them exceeds their strengthening effect.

Reverse cuts performed even worse on average, showing a 47.6 % increase in computation time. However, as with the nested cuts, a few instances benefited from their inclusion, with a 33.5 % decrease for `hugenet3`, and solution quality improved in 7 instances, also reducing the maximum optimality gap to 34.4 %.

For the sake of completeness, the nested cuts were also tested alongside the *PRIMAL* heuristic, which improved upon the previous results in 16 of the instances, even though average performance decreased by 15.6 %. `urban2r` for example, showed an 86.9 % improvement, improving upon the previous best configuration by more than 60 %.

Table 5.3: Computational results - nested and reverse cuts

| | *BENDERS*$_C$ including | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | (5.2)–(5.6) | | (5.2)–(5.6) & nested | | (5.2)–(5.6) & reverse | | (5.2)–(5.6) & *PRIMAL* | | (5.2)–(5.6) & nested & *PRIMAL* | |
| Data set | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] | gap [%] | $t_s$ [sec] |
| *Section 1: Baseline data sets* | | | | | | | | | | |
| urban1r | 8.96 | >2 h | 9.25 | >2 h | **8.10** | **>2 h** | 8.92 | >2 h | 9.77 | >2 h |
| urban2r | – | 189 | – | 179 | – | 391 | – | 351 | – | **45.9** |
| urban3r | – | **16.0** | – | 29.1 | – | 20.1 | – | 22.3 | – | 29.2 |
| suburb1r | – | 3.68 | – | **2.40** | – | 3.29 | – | 2.50 | – | 2.88 |
| suburb2r | – | 12.3 | – | 20.6 | – | 40.1 | – | **9.94** | – | 63.7 |
| suburb3r | – | 62.5 | – | 155 | – | 82.2 | – | 71.1 | – | **39.2** |
| rural1r | >24 GiB | | 10.8 | >2 h | **7.65** | **>2 h** | 9.92 | >2 h | >24 GiB | |
| rural2r | 12.8 | >2 h | 13.5 | >2 h | 12.5 | >2 h | **12.1** | **>2 h** | 13.1 | >2 h |
| rural3r | – | 360 | – | **253** | – | 486 | – | 300 | – | 325 |
| *Section 2: Scalability data sets* | | | | | | | | | | |
| micronet | – | **0.08** | – | 0.10 | – | 0.11 | – | **0.08** | – | 0.11 |
| citynet1 | – | 0.21 | – | 0.14 | – | 0.25 | – | 0.25 | – | **0.12** |
| citynet2 | – | 4.59 | – | 4.01 | – | 10.0 | – | 4.00 | – | **3.03** |
| citynet3 | – | 75.8 | – | 65.1 | – | 118 | – | 68.2 | – | **38.6** |
| citynet4 | – | 177 | – | 204 | – | 526 | – | 296 | – | **107** |
| citynet5 | 6.45 | >2 h | **3.24** | **>2 h** | 7.01 | >2 h | 8.12 | >2 h | 4.43 | >2 h |
| citynet6 | 71.5 | >2 h | **28.4** | **>2 h** | 34.4 | >2 h | 29.5 | >2 h | 36.9 | >2 h |
| mednet1 | – | **0.05** | – | 0.07 | – | 0.06 | – | 0.06 | – | 0.09 |
| mednet2 | – | 7.96 | – | 9.75 | – | 8.65 | – | **6.21** | – | 9.83 |
| mednet3 | – | 157 | – | 148 | – | 137 | – | **44.8** | – | 94.6 |
| mednet4 | 3.05 | >2 h | 1.22 | >2 h | 1.55 | >2 h | 1.06 | >2 h | **0.88** | **>2 h** |
| mednet5 | 3.02 | >2 h | 5.25 | >2 h | 3.14 | >2 h | **2.37** | **>2 h** | 5.53 | >2 h |
| mednet6 | 21.7 | >2 h | 12.6 | >2 h | 12.3 | >2 h | 11.9 | >2 h | **10.9** | **>2 h** |
| hugenet1 | – | 0.51 | – | 0.57 | – | 0.49 | – | 0.62 | – | **0.38** |
| hugenet2 | – | 3.27 | – | 2.69 | – | 3.60 | – | **2.41** | – | 3.26 |
| hugenet3 | – | 78.4 | – | 48.5 | – | 102 | – | 40.3 | – | **34.5** |
| hugenet4 | 0.37 | >2 h | 1.15 | >2 h | 1.91 | >2 h | – | 3,264 | – | **2,756** |
| hugenet5 | – | 5,315 | 4.20 | >2 h | 2.31 | >2 h | – | **2,584** | 3.91 | >2 h |
| hugenet6 | 20.1 | >2 h | 18.1 | >2 h | 17.3 | >2 h | **14.9** | **>2 h** | 18.5 | >2 h |
| subnet1 | – | 0.08 | – | 0.08 | – | **0.07** | – | 0.08 | – | **0.07** |
| subnet2 | – | 10.7 | – | 9.44 | – | 19.8 | – | 8.78 | – | **2.94** |
| subnet3 | – | 105 | – | 194 | – | 196 | – | **101** | – | 139 |
| subnet4 | – | 1,500 | – | 833 | – | 998 | – | 1,012 | – | **457** |
| subnet5 | – | 3,046 | 0.68 | >2 h | – | 3,856 | – | 5,102 | – | **2,620** |
| subnet6 | 14.7 | >2 h | >2 h | | 16.2 | >2 h | **14.3** | **>2 h** | 17.6 | >2 h |

**Non-basic variable removal**

The results for the non-basic variable removal test is shown in figure 5.3. Even though for 80 % of the values of $n$ tested, the computation time decreased, no clear relation can be seen, with peaks at $n = 140$ and $n = 180$ likely the result of branch-and-cut node order traversal differences. A decrease of 56 % in computation time in comparison to the run with no variable removal is seen at $n = 280$ however, along with a memory usage decrease of 14 %, illustrating that keeping the LPs small definitely has benefits.

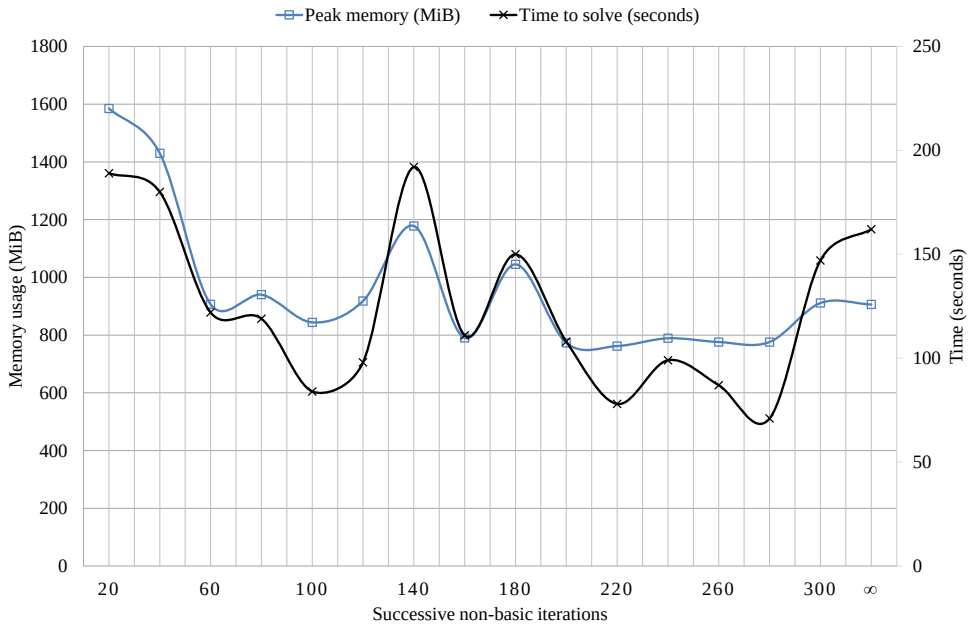
For small values of $n$, the number of columns generated increases dramatically, peaking at a factor of 9, indicating that a large number of columns have to be regenerated after they have been removed. At $n = 280$, the number of columns generated is only 15 % higher than without non-basic variable removal while almost 70 % of those are removed, illustrating that most variables removed were in fact redundant.

## 5.7 Conclusion

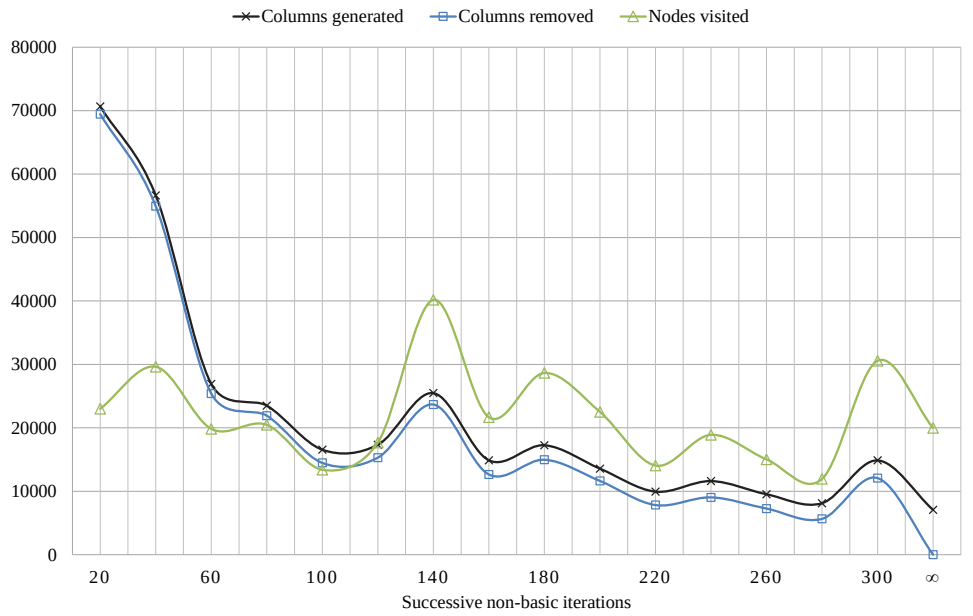
In this chapter, several techniques were tested to improve both the lower and upper bounds of PONDP during computation. In terms of strengthening the lower bound, the connectivity cuts were extremely effective, increasing computational performance by almost an order of magnitude. This improvement is likely due to the enforcement of a tree structure from the onset, which the standard formulation only takes on in the optimal solution.

Adding flow-cutset inequalities alongside the connectivity cuts improved the performance once again for the majority of instances, showing the efficacy of strengthening both the trenching and fibre components of the model. For some data sets however, the additional separation effort exceeds the strengthening gains achieved.

Separating additional inequalities using nested and reverse cuts showed mixed results, with some data sets benefiting quite dramatically from their inclusion and others tak-




(a) Computation time and memory results




(b) Nodes and column results

Figure 5.3: Computational results for the `subnet3` data set with non-basic variable removal

ing twice as long to solve. It is suspected that the additional cuts improve the bound during the first stages of computation, where we only have limited knowledge in terms of the optimal structure, but lose effectiveness later on as they become redundant.

To strengthen the upper bound, a primal heuristic known as *PRIMAL* was proposed, which results in a 30 % improvement in computational performance, as well as a 25 % reduction in the optimality gap on average for sub-optimal solutions. This suggests that the heuristic effectively lowers the upper bound early during computation, as well as outperforming the general heuristics CPLEX employs.

Since we now have a scalable approach that improves computational performance of the general PONDP, the next chapter will consider incorporating demand uncertainty into the modelling framework to improve numerical performance.

# Chapter 6

# Demand uncertainty

*In this chapter we extend the modelling framework to handle demand uncertainty, utilising both two-stage recourse stochastic programming and robust optimisation techniques. Relevant strengthening cuts and heuristics are modified to account for the changes, and a computational study shows the resulting deployment cost improvements as well as the increased computational effort to solve the models. A revenue model is then formulated to maximise the expected return on investment by connecting a subset of ONUs.*

## 6.1 Introduction

In chapter 4 we introduced a minimum cost model where all ONUs had a fixed, positive demand. In reality however, when design and deploying PONs, this demand is rarely known, with the service provider having to rely on past experience or social metrics to estimate the demand. To address this phenomenon, we approach the field of stochastic optimisation, where we endeavour to incorporate and leverage randomness to improve the accuracy of our model.

In this chapter, we incorporate the two most common approaches in stochastic optimisation, the two-stage recourse and robust approaches, into the modelling framework. This showcases the flexibility of the framework and ensures that uncertain demand can be incorporated into PONDP, improving the practical relevance of the model.

Additionally, we investigate a strategy to connect ONUs using a revenue model. Instead of minimising deployment cost, the revenue model maximises expected revenue by deciding which ONUs to connect based on expected realisation probabilities and potential return on investment. This approach can be used concurrently with either the two-stage recourse or robust models, effectively giving the model more design freedom to find the best solution.

## 6.2 Two-stage recourse formulation

The two-stage recourse approach utilises the distribution of random variables to form a deterministic model [65]. When the random variable is integer, as in the case of PONDP where the demand can not take on a fractional value, we discretise the uncertainty using a set of realisation values, known as *scenarios*. Each scenario therefore contains a set of values for all ONU demands as well as a corresponding realisation probability. Scenarios are also mutually independent, i.e. we assume that no two scenarios can realise at the same time. This independence can be exploited to save costs in terms of fibre and splitter dimensioning.

Scenarios can be generated in a number of ways, all depending on the data available. The basic case would be to generate a scenario for every possible combination of realisation values for every demand. Unfortunately, the possible combinations increase exponentially with regards to $|U|$, in particular $2^{|U|}$ for the binary case, crippling scalability for all but the smallest of networks. Monte Carlo simulation can be used to reduce the number of scenarios, sampling a subset of the possible configurations from a given distribution. Other techniques to reduce the number of scenarios include domination [128], scenario clustering, Quasi-Monte Carlo sampling as well as selection

heuristics [140, 141].

In this thesis, we do not investigate the different scenario generation methods, instead assuming we are given a set of generated scenarios S, each containing a subset of ONU demands that realises. Furthermore, we assume each scenario has an identical probability to realise and that each ONU is contained in at least one scenario. For subsequent sections, the two-stage recourse formulation will be known as $BENDERS_S$. As a reference, appendix A contains a list of all named formulations and problems.

### 6.2.1 Model modification

Recall from section 3.3.1 that the recourse formulation relies on two interacting stages. For PONDP, the first stage decision is where to trench, which is based on the current deterministic realisation of the uncertain second stage decision, in this case the fibre routing based on ONU demand. Stated differently, after a realisation of the second stage uncertainty becomes available, i.e. we know what ONU demand is, we can optimise the first stage, i.e. where to trench. This interaction is bi-directional, and routing paths in the second stage are also chosen given the current trench assignment.

Since the uncertainty is contained in the distribution network, only the distribution sub-problem $PON_D$ is affected. Redefine the parameter $\boldsymbol{d} \in \mathbb{N}_+^{|\mathrm{U}|\times|\mathrm{S}|}$ as the demand at each ONU in a specific scenario. To transform $PON_D$ into a two-stage recourse sub-problem, redefine both the shortfall variable $\boldsymbol{\alpha} \in \mathbb{R}_+^{|\mathrm{S}|}$ and the flow variable $\boldsymbol{f}^{\mathrm{D}} \in \mathbb{R}_+^{|\mathrm{P}|\times|\mathrm{S}|}$ to account for the independent nature of the scenarios. Then formulate the two-stage recourse distribution sub-problem, $PON_{DS}$ as follows:

$PON_{DS}$:

$$\min \quad \sum_{s\in\mathrm{S}} \alpha_s \tag{6.1}$$

$$\text{s.t.} \quad \sum_{k\in\mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j)} \sum_{p\in\mathrm{P}(k)} f_{ps}^{D} = d_j^s, \quad \forall j \in \mathrm{U}, \forall s \in \mathrm{S}, \tag{6.2}$$

$$\sum_{k\in\mathrm{K}} \sum_{p\in\mathrm{P}(k,e)} f_{ps}^{D} - \alpha_s \leq x_e^{\mathrm{FD}*}, \quad \forall e \in \mathrm{E}, \forall s \in \mathrm{S}. \tag{6.3}$$

Similar to $PON_D$, we will again derive the metric inequality associated with $PON_{DS}$. To this end, associate dual variables $\boldsymbol{\pi}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{U}|\times|\mathrm{S}|}$ and $\boldsymbol{\mu}^{\mathrm{D}} \in \mathbb{R}_{+}^{|\mathrm{E}|\times|\mathrm{S}|}$ with constraints (6.2) and (6.3) respectively. The corresponding dual can then be written as follows:

$PON_{DS}$':

$$\max \quad \sum_{s\in\mathrm{S}} \Big( \sum_{j\in\mathrm{U}} d_j^s \pi_{js}^{\mathrm{D}} - \sum_{e\in\mathrm{E}} x_e^{\mathrm{FD}*} \mu_{es}^{\mathrm{D}} \Big) \tag{6.4}$$

$$\text{s.t.} \quad \sum_{e\in\mathrm{E}} \mu_{es}^{\mathrm{D}} \leq 1, \quad \forall s \in \mathrm{S}, \tag{6.5}$$

$$\pi_{js}^{\mathrm{D}} - \sum_{e\in\mathrm{E}(p)} \mu_{es}^{\mathrm{D}} \leq 0, \qquad \begin{array}{c} \forall p \in \mathrm{P}(k), \forall k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j), \\ \forall j \in \mathrm{U}, \forall s \in \mathrm{S}. \end{array} \tag{6.6}$$

Using the *Japanese theorem* again as was done in section 4.3.2, the two-stage recourse distribution metric inequality can be derived:

$$\sum_{e\in\mathrm{E}} x_e^{\mathrm{FD}} \mu_{es}^{\mathrm{D}*} \geq \sum_{j\in\mathrm{U}} d_j^s \pi_{js}^{\mathrm{D}*}, \quad \forall s \in \mathrm{S}, \tag{6.7}$$

with $\boldsymbol{x}^{\mathrm{FD}}$ the corresponding variables in $PON_M$, and $\boldsymbol{\mu}^{\mathrm{D}*}$ and $\boldsymbol{\pi}^{\mathrm{D}*}$ the solution values from the dual $PON_{DS}$'. We can also do column generation by noting that (6.6) can be rearranged to get $\pi_{js}^{\mathrm{D}} = \min_{k\in\mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j)} \min_{p\in\mathrm{P}(k)} \sum_{e\in\mathrm{E}(p)} \mu_{es}^{\mathrm{D}}$. The value of $\pi_{js}^{\mathrm{D}}$ then denotes the length of the shortest path between ONU $j \in \mathrm{U}$ and a splitter, given edge costs $\mu_{es}^{\mathrm{D}}$.

Furthermore, note that (6.7) is defined for each scenario, which means that $|\mathrm{S}|$ cuts are generated after each separation of the sub-problem. This already gives an indication as to possible scalability problems due to the increase in the number of distribution metric inequalities.

### 6.2.2 Algorithmic modification

Since the strengthening cuts of chapter 5 assumed fixed ONU demand, we need to adapt them to accommodate for demand uncertainty. Connectivity cuts are still valid

for $BENDERS_S$, but the flow-cutset inequalities as defined in (5.5) and (5.6) are not, since they assume all ONU demands need to be met simultaneously.

While we still need to ensure that the capacity is sufficient for all demands, the demands have to be met non-concurrently, i.e. for each scenario independently. Therefore, the flow-cutset inequality becomes:

$$\sum_{i \in W \cap D} x^{FD}_{\beta(i)} + \sum_{e \in \delta(W)} x^{FD}_e \geq \sum_{j \in W \cap U} d^s_j, \quad \forall W \subset V, W \cap U \neq \varnothing, \forall s \in S, \tag{6.8}$$

which is equivalent to:

$$CUT_{DS}: \quad \sum_{i \in W \cap D} x^{FD}_{\beta(i)} + \sum_{e \in \delta(W)} x^{FD}_e \geq \max_{s \in S} \Big\{ \sum_{j \in W \cap U} d^s_j \Big\}, \quad \forall W \subset V, W \cap U \neq \varnothing. \tag{6.9}$$

*PRIMAL* functions almost identically, except for Step 1, where a distribution path is now set as used if $f^D_{ps} \geq \epsilon$ for any $s \in S$.

Finally, to validate the model, the routing feasibility checker *FEAS* is solved iteratively for each scenario, with alternate distribution flow conservation constraints. Introducing the subset $U_s \subseteq U, s \in S$ of all ONUs contained in the scenario $s \in S$, we replace (4.67) with:

$$\sum_{j \in \phi(i)} f^D_{ij} - \sum_{j \in \phi(i)} f^D_{ji} = \begin{cases} -x^{FD*}_{\beta(i)}, & i \in U_s, \\ x^{FD*}_{\beta(i)}, & i \in D, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall i \in V. \tag{6.10}$$

This alternate version of *FEAS* with scenario specific ONU demand is henceforth called $FEAS_S$.

## 6.3 Robust formulation

The robust optimisation approach attempts to protect a solution from disturbances in the input data by ensuring that it is feasible for all possible values of the random variables [66]. In the PONDP environment, this equates to being able to connect the

worst-case scenario of ONU demand realisation. In other environments where the realisation of worst-case scenarios are relatively rare (*black swans* if you will [142]), solutions may deviate substantially from average, possibly rendering it less relevant in practice as CAPEX is increased without regarding the corresponding risk.

While the two-stage recourse approach incorporated the scenarios directly into the distribution sub-problem $PON_{DS}$, we could also define the demand as belonging to some uncertainty set $\mathfrak{D}$. Therefore, we can write the ONU demand as $\boldsymbol{d} \in \mathfrak{D}$. Though the robust and two-stage recourse formulations are not equivalent in general, by including the same scenarios and defining the uncertainty set carefully, we can, for the sake of comparison, arrive at an equivalent robust formulation to $BENDERS_S$, henceforth known as $BENDERS_R$.

### 6.3.1 Model modification

As the uncertainty still only affects the distribution network, we only need to adapt $PON_D$. Additionally, for the sake of the robust derivation, we switch to formulating proportional flow by separating the demand quantity from the variable value (thereby scaling the variable *proportionally* using the demand). This is only equivalent to the absolute flow approach in the uncapacitated case if no splittable flows exist in the optimal solution, as was proven for PONDP, and allows us to consolidate all the demand into a single constraint set. To this end, introduce proportional flow variables $\boldsymbol{h}^{\mathrm{D}} \in [0,1]^{|\mathrm{P}|}$ and consider the robust distribution sub-problem:

$PON_{DR}$:

$$\min \quad \alpha \tag{6.11}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k)} h^{\mathrm{D}}_p = 1, \quad \forall j \in \mathrm{U}, \tag{6.12}$$

$$\sum_{j \in \mathrm{U}} \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k,e)} d_j h^{\mathrm{D}}_p - \alpha \leq x^{\mathrm{FD}*}_e, \quad \forall e \in \mathrm{E}, \forall d_j \in \mathfrak{D}. \tag{6.13}$$

The deterministic version of the above problem, also known as the *robust counterpart*,

can be defined by noting that we want to consider the worst-case ONU demand. Therefore, with $conv(\cdot)$ denoting the convex hull, we add an inner maximisation problem to (6.13) as follows:

$PON_{DRC}$:

$$\min \quad \alpha \tag{6.14}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k)} h^{\mathrm{D}}_p = 1, \quad \forall j \in \mathrm{U}, \tag{6.15}$$

$$\max_{\boldsymbol{d} \in conv(\mathfrak{D})} \left\{ \sum_{j \in \mathrm{U}} \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k,e)} d_j h^{\mathrm{D}}_p \right\} - \alpha \leq x^{\mathrm{FD}*}_e, \quad \forall e \in \mathrm{E}. \tag{6.16}$$

Consider next the inner maximisation problem in (6.16). It will ensure that given a fixed routing capacity $x^{\mathrm{FD}*}_e$, the demand is increased to take on the bounds of $conv(\mathfrak{D})$. Furthermore, we now describe $\mathfrak{D}$ as a polyhedron, i.e. given a fixed flow $h^{\mathrm{D}*}_p$ and an index $e^*$, the demand is a convex combination of the extreme points of $conv(\mathfrak{D})$. Next, to derive an equivalent formulation to $BENDERS_S$, let $\mathfrak{D}$ contain all demands specified in all scenarios. Thus, with coefficients $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathrm{S}|}_+$, the inner maximisation problem can be rewritten as follows:

$$\max \quad \sum_{j \in \mathrm{U}} \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k,e)} d_j h^{\mathrm{D}*}_p \tag{6.17}$$

$$\text{s.t.} \quad d_j - \sum_{s \in \mathrm{S}} \lambda_s d^s_j = 0, \quad \forall j \in \mathrm{U}, \tag{6.18}$$

$$\sum_{s \in \mathrm{S}} \lambda_s = 1. \tag{6.19}$$

Next, associate dual variables $\boldsymbol{\zeta}^{e^*} \in \mathbb{R}^{|\mathrm{U}|}$ and $\eta_{e^*} \in \mathbb{R}$ with constraints (6.18) and (6.19) respectively. This allows us to write the dual:

$$\min \quad \eta_{e^*} \tag{6.20}$$

$$\text{s.t.} \quad \sum_{j \in \mathrm{U}} \zeta^{e^*}_j d^s_j - \eta_{e^*} \leq 0, \quad \forall s \in \mathrm{S}, \tag{6.21}$$

$$\zeta^{e^*}_j \geq \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k,e^*)} h^{\mathrm{D}*}_p, \quad \forall j \in \mathrm{U}. \tag{6.22}$$

Substituting this minimisation dual for (6.16) in $PON_{DRC}$ and unfixing $h^{\mathrm{D}*}_p$ and the index $e^*$, we are left with the *adjustable robust counterpart*:

$PON_{DARC}$:

$$\min \quad \alpha \tag{6.23}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k)} h^{\mathrm{D}}_p = 1, \quad \forall j \in \mathrm{U}, \tag{6.24}$$

$$\eta_e - \alpha \leq x^{\mathrm{FD}*}_e, \quad \forall e \in \mathrm{E}, \tag{6.25}$$

$$\sum_{j \in \mathrm{U}} \zeta^e_j d^s_j - \eta_e \leq 0, \quad \forall s \in \mathrm{S}, \forall e \in \mathrm{E}, \tag{6.26}$$

$$\zeta^e_j \geq \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k,e)} h^{\mathrm{D}}_p, \quad \forall j \in \mathrm{U}, \forall e \in \mathrm{E}. \tag{6.27}$$

Finally, we need to derive a metric inequality for the robust distribution sub-problem. Associate dual variables $\boldsymbol{\pi}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{U}|}$, $\boldsymbol{\mu}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{E}|}_+$, $\boldsymbol{\rho} \in \mathbb{R}^{|\mathrm{S}|\times|\mathrm{E}|}_+$ and $\boldsymbol{\gamma} \in \mathbb{R}^{|\mathrm{U}|\times|\mathrm{E}|}_+$ with constraints (6.24)–(6.27) respectively. The corresponding dual can then be written as follows:

$PON_{DARC}{}'$:

$$\max \quad \sum_{j \in \mathrm{U}} \pi^{\mathrm{D}}_j - \sum_{e \in \mathrm{E}} x^{\mathrm{FD}*}_e \mu^{\mathrm{D}}_e \tag{6.28}$$

$$\text{s.t.} \quad \sum_{e \in \mathrm{E}} \mu^{\mathrm{D}}_e \leq 1, \tag{6.29}$$

$$\pi^{\mathrm{D}}_j - \sum_{e \in \mathrm{E}(p)} \gamma^e_j \leq 0, \quad \begin{array}{c} \forall p \in \mathrm{P}(k), \forall k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j), \\ \forall j \in \mathrm{U}, \end{array} \tag{6.30}$$

$$\mu^{\mathrm{D}}_e - \sum_{s \in \mathrm{S}} \rho^s_e = 0, \quad \forall e \in \mathrm{E}, \tag{6.31}$$

$$\gamma^e_j - \sum_{s \in \mathrm{S}} d^s_j \rho^s_e = 0, \quad \forall j \in \mathrm{U}, \forall e \in \mathrm{E}. \tag{6.32}$$

Utilising the *Japanese theorem*, we now derive the metric inequality for the robust distribution sub-problem:

$$\sum_{e \in \mathrm{E}} x^{\mathrm{FD}}_e \mu^{\mathrm{D}*}_e \geq \sum_{j \in \mathrm{U}} \pi^{\mathrm{D}*}_j, \tag{6.33}$$

with solution values $\boldsymbol{\mu}^{\mathrm{D}*}$ and $\boldsymbol{\pi}^{\mathrm{D}*}$ from the dual $PON_{DARC}{}'$ and $\boldsymbol{x}^{\mathrm{FD}}$ the variable vector in $PON_M$. Again, constraint (6.30) can be rearranged to find the length of the shortest path between an ONU $j \in \mathrm{U}$ and a splitter, given edge costs $\gamma^e_j$, which is defined as $\pi^{\mathrm{D}}_j = \min_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \gamma^e_j$.

For the robust problem, note that only a single metric inequality is derived during separation, in contrast to |S| inequalities for the two-stage recourse approach. This is because the scenarios are incorporated implicitly into the master problem, instead of explicitly.

### 6.3.2 Algorithmic modification

Again, the connectivity cuts are valid as-is for the robust formulation. The same changes to the flow-cutset inequalities are made for the robust approach as detailed in section 6.2.2 for the two-stage recourse formulation, where the right-hand side is redefined as the maximum demand for ONUs over all scenarios.

The solution generation heuristic *PRIMAL* is used exactly as specified in section 5.5.

For validation purposes, $FEAS_S$, the scenario specific version of *FEAS*, as defined in section 6.2.2, is solved iteratively over all scenarios $s \in S$.

## 6.4 Revenue formulation

Both the two-stage recourse and robust formulations take the uncertainty inherent in the demand into account, i.e. connecting all ONUs if they exist in at least one scenario, while saving costs through reduced fibre and splitter dimensioning. Consider now the possibility of not connecting an ONU at all, either because its probability of realising is too low, or because income generated from that ONU can not offset the cost to connect it. We can combine both possibilities in a model that, instead of minimising cost, maximises revenue. This type of model is also of great value to service providers, as return on investment is their primary business goal.

Since it remains a stochastic model, instead of the true revenue (which is unknown), we maximise expected revenue.

### 6.4.1 Model modification

To allow the master problem to decide whether an ONU should be connected, we introduce a binary ONU usage variable, $\boldsymbol{\theta} \in \{0,1\}^{|\mathrm{U}|}$, that takes on a value of 1 if an ONU should be connected, and 0 otherwise. This allows the master to interact with the required demand in the distribution sub-problem, similar to the interaction between $\boldsymbol{\psi}$ and the feeder sub-problem. Since a situation might arise where no ONUs are connected, we also introduce a CO usage variable, $\omega \in \{0,1\}$. This ensures that when no splitters are used, the OLT cost can be set to zero, leading to a zero objective function.

Furthermore, to model the expected income, introduce parameters $\boldsymbol{r} \in \mathbb{Q}_{+}^{|\mathrm{U}|}$ and $\boldsymbol{\varrho} \in [0,1]^{|\mathrm{U}|}$, the total average income during the network lifetime and the service requirement probability respectively. Now we can define our new master problem, $PON_M^{rev}$, as follows:

$PON_M^{rev}$:

$$\begin{aligned}
\max \quad & \sum_{j\in\mathrm{U}} \varrho_j r_j \theta_j - \omega c^{\mathrm{OLT}} - \sum_{j\in\mathrm{U}} \theta_j c^{\mathrm{ONU}} - \\
& \sum_{e\in\mathrm{E}} (x_e^{\mathrm{TR}} c_e^{\mathrm{TR}} + x_e^{\mathrm{FD}} c_e^{\mathrm{F}} + x_e^{\mathrm{FF}} c_e^{\mathrm{F}}) - \sum_{i\in\mathrm{D}} \psi_i c^{\mathrm{SP}} && (6.34)\\
\text{s.t.} \quad & x_{\beta(i)}^{\mathrm{FD}} \leq \kappa \psi_i, \quad \forall i \in \mathrm{D}, && (6.35)\\
& x_{\beta(q)}^{\mathrm{FF}} \leq |\mathrm{D}|\omega, && (6.36)\\
& x_e^{\mathrm{FF}} \leq \Delta x_e^{\mathrm{TR}}, \quad \forall e \in \mathrm{E}, && (6.37)\\
& x_e^{\mathrm{FD}} \leq \Delta x_e^{\mathrm{TR}}, \quad \forall e \in \mathrm{E}. && (6.38)
\end{aligned}$$

The objective function (6.34) maximises total expected revenue by subtracting all deployment costs from the expected income from the used ONUs. Constraints (6.35) and (6.36) set the splitter and CO usage variables respectively, while constraints (6.37)–(6.38) set the trench usage variables. To incorporate path length constraints into $PON_M^{rev}$, we follow the same process as detailed previously in section 4.3.4.

In terms of the sub-problems, we now have a variable ONU demand $\boldsymbol{\theta}$. Therefore, the

distribution sub-problem is affected. Three different formulations will now be given, including adapted versions of the fixed demand distribution sub-problem $PON_D$, the two-stage recourse distribution sub-problem $PON_{DS}$ and the robust distribution sub-problem $PON_{DARC}$. First, consider the variable demand distribution sub-problem:

$PON_D^{rev}$:

$$\min \quad \alpha \tag{6.39}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j)} \sum_{p \in \mathrm{P}(k)} f_p^{\mathrm{D}} = d_j \theta_j^*, \quad \forall j \in \mathrm{U}, \tag{6.40}$$

$$\sum_{k \in \mathrm{K}} \sum_{p \in \mathrm{P}(k,e)} f_p^{\mathrm{D}} - \alpha \leq x_e^{\mathrm{FD}*}, \quad \forall e \in \mathrm{E}, \tag{6.41}$$

with $\boldsymbol{\theta}^*$ the current variable values from $PON_M^{rev}$. A positive objective function indicates insufficient capacity, for which the corresponding metric inequality can be derived by considering the dual. Associating dual variables $\boldsymbol{\pi}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{U}|}$ and $\boldsymbol{\mu}^{\mathrm{D}} \in \mathbb{R}_+^{|\mathrm{E}|}$ with constraints (6.40) and (6.41) respectively, we get the metric inequality:

$$\sum_{e \in \mathrm{E}} x_e^{\mathrm{FD}} \mu_e^{\mathrm{D}*} \geq \sum_{j \in \mathrm{U}} d_j \theta_j \pi_j^{\mathrm{D}*}. \tag{6.42}$$

To implement column generation, the value of $\pi_j^{\mathrm{D}}$ is found through the shortest path calculation:

$$\pi_j^{\mathrm{D}} = \min_{k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \mu_e^{\mathrm{D}}, \quad \forall j \in \mathrm{U}. \tag{6.43}$$

Note that similar to the feeder metric inequality (4.40), after being added to the master problem, the cut in (6.42) can be turned on or off based on the value of $\boldsymbol{\theta}$. The model formed by combining $PON_M^{rev}$, $PON_F$ and $PON_D^{rev}$ will henceforth be known as $BENDERS^{rev}$.

For the two-stage recourse formulation, we use $PON_M^{rev}$ and $PON_F$ but adapt $PON_{DS}$ to form the formulation $BENDERS_S^{rev}$ as follows:

$PON_{DS}^{rev}$:

$$\min \quad \sum_{s \in \mathrm{S}} \alpha_s \tag{6.44}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k)} f^{D}_{ps} = d^{s}_{j} \theta^{*}_{j}, \quad \forall j \in \mathrm{U}, \forall s \in \mathrm{S}, \tag{6.45}$$

$$\sum_{k \in \mathrm{K}} \sum_{p \in \mathrm{P}(k,e)} f^{D}_{ps} - \alpha_{s} \leq x^{\mathrm{FD}*}_{e}, \quad \forall e \in \mathrm{E}, \forall s \in \mathrm{S}. \tag{6.46}$$

Again, by associating dual variables $\boldsymbol{\pi}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{U}| \times |\mathrm{S}|}$ and $\boldsymbol{\mu}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{E}| \times |\mathrm{S}|}_{+}$ with constraints (6.45) and (6.46), we derive the metric inequality:

$$\sum_{e \in \mathrm{E}} x^{\mathrm{FD}}_{e} \mu^{\mathrm{D}*}_{es} \geq \sum_{j \in \mathrm{U}} d^{s}_{j} \theta_{j} \pi^{\mathrm{D}*}_{js}, \quad \forall s \in \mathrm{S}. \tag{6.47}$$

Paths are generated by solving the following shortest path calculation for every scenario $s \in \mathrm{S}$, using edge costs $\mu^{\mathrm{D}}_{es}$:

$$\pi^{\mathrm{D}}_{js} = \min_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \mu^{\mathrm{D}}_{es}, \quad \forall j \in \mathrm{U}. \tag{6.48}$$

Finally, for the robust formulation, $PON_{DARC}$ is adapted to incorporate variable demand as shown in (6.49)–(6.53) below.

$PON^{rev}_{DARC}$:

$$\min \quad \alpha \tag{6.49}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k)} f^{\mathrm{D}}_{p} = \theta^{*}_{j}, \quad \forall j \in \mathrm{U}, \tag{6.50}$$

$$\eta_{e} - \alpha \leq x^{\mathrm{FD}*}_{e}, \quad \forall e \in \mathrm{E}, \tag{6.51}$$

$$\sum_{j \in \mathrm{U}} \zeta^{e}_{j} d^{s}_{j} - \eta_{e} \leq 0, \quad \forall s \in \mathrm{S}, \forall e \in \mathrm{E}, \tag{6.52}$$

$$\zeta^{e}_{j} \geq \sum_{k \in \mathrm{K}^{\mathrm{SP}}_{\mathrm{ONU}}(j)} \sum_{p \in \mathrm{P}(k,e)} f^{\mathrm{D}}_{p}, \quad \forall j \in \mathrm{U}, \forall e \in \mathrm{E}. \tag{6.53}$$

To derive the metric inequality for $PON^{rev}_{DARC}$, associate dual variables $\boldsymbol{\pi}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{U}|}$, $\boldsymbol{\mu}^{\mathrm{D}} \in \mathbb{R}^{|\mathrm{E}|}_{+}$, $\boldsymbol{\rho} \in \mathbb{R}^{|\mathrm{S}| \times |\mathrm{E}|}_{+}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{|\mathrm{U}| \times |\mathrm{E}|}_{+}$ with constraints (6.50) through (6.53) respectively. Then, whenever the dual objective value is negative, we can add the following cut:

$$\sum_{e \in \mathrm{E}} x^{\mathrm{FD}}_{e} \mu^{\mathrm{D}*}_{e} \geq \sum_{j \in \mathrm{U}} \theta_{j} \pi^{\mathrm{D}*}_{j}, \tag{6.54}$$

with its shortest path generation calculation:

$$\pi_j^{\mathrm{D}} = \min_{k \in \mathrm{K}_{\mathrm{ONU}}^{\mathrm{SP}}(j)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \gamma_j^e, \quad \forall j \in \mathrm{U}, \tag{6.55}$$

which is equivalent to finding the shortest path between an ONU $j \in \mathrm{U}$ and any splitter, given edge costs $\gamma_j^e$. The robust formulation, containing $PON_M^{rev}$, $PON_F$ and $PON_{DARC}^{rev}$ will henceforth be known as $BENDERS_R^{rev}$.

### 6.4.2 Algorithmic modification

Since ONU connectivity is no longer a given, we can not utilise the connectivity cuts as derived in chapter 5. Therefore, the distribution connectivity cut (5.2) is redefined to incorporate the ONU usage variable $\boldsymbol{\theta}$ as follows:

$$\mathrm{CON}_{\mathrm{D}}^{\mathrm{rev}}: \quad \sum_{i \in \mathrm{W} \cap \mathrm{D}} \psi_i + \sum_{e \in \delta(\mathrm{W})} x_e^{\mathrm{TD}} \geq \theta_j, \quad \forall\, \mathrm{W} \subset \mathrm{V}, \forall j \in \mathrm{W} \cap \mathrm{U} \neq \varnothing. \tag{6.56}$$

Similarly the global connectivity cut (5.4) can be redefined as follows:

$$\mathrm{CON}_{\mathrm{G}}^{\mathrm{rev}}: \quad \sum_{e \in \delta(\mathrm{W})} x_e^{\mathrm{TR}} \geq \theta_j, \quad \forall\, \mathrm{W} \subset \mathrm{V} \setminus \{q\}, \forall j \in \mathrm{W} \cap \mathrm{U} \neq \varnothing. \tag{6.57}$$

Therefore, for the revenue model, the separation of either the distribution or global connectivity cuts yields $|\mathrm{W} \cap \mathrm{U}|$ inequalities.

If the stochastic formulations $BENDERS_S^{rev}$ or $BENDERS_R^{rev}$ are used, the distribution flow-cutset inequalities are adapted as shown in (6.9). Additionally, the solution generation heuristic *PRIMAL* is used as-is for $BENDERS_R^{rev}$ and with the alteration detailed in section 6.2.2 for $BENDERS_S^{rev}$. $FEAS_S$ is then utilised to validate the solutions.

Conversely, if the deterministic formulation $BENDERS^{rev}$ is used, the standard flow-cutset inequalities and the *PRIMAL* heuristic introduced in chapter 5, as well as the initial validation model *FEAS* are used.

## 6.5 Computational study

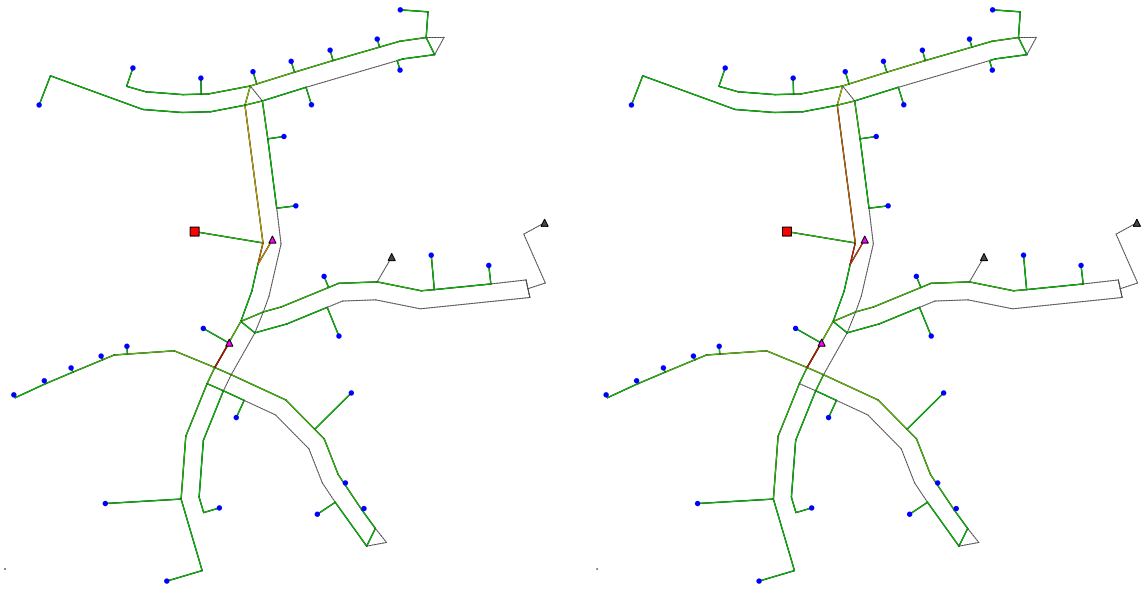
In contrast to the computational studies in chapters 4 and 5, this section focusses on the topological results of the stochastic models in question. The aim of this computational study is to demonstrate the flexibility of the modelling framework, as well as the inherent differences in the approaches introduced in this chapter. Therefore, a detailed performance analysis will not form part of this section.

To illustrate the advantage of both stochastic approaches and revenue models, case studies and numerical results will be given. Then, to determine the scalability of the two-stage recourse and robust approaches, small data sets will be solved using a variable number of random demand scenarios.

### 6.5.1 Stochastic numerical study

#### Case study

To illustrate how scenarios function as well as the difference between deterministic and stochastic formulations, we will now look at a small case study using a data set with 4 potential splitter locations and 30 ONUs, named `stochnet`. The optimal deterministic solution using $BENDERS_C$ is shown in figure 6.1a, with circles representing ONUs, triangles representing splitters and the square representing the CO. In addition, magenta triangles are locations where splitters are installed while black triangles denote unused potential splitter locations. Thick coloured lines indicate installed fibres in ducts, with yellow ducts housing more fibres than green and red more than yellow. Grey lines denote unused potential duct locations. Using the same model parameters as detailed in chapter 4, the optimal objective value for the deterministic model is R 599,767.72, requiring two splitters, one with 12 and one with 18 connected ONUs. Therefore both a 1:16 and a 1:32 splitter would be required to connect this network. The average and maximum number of fibres in each duct is 2.29 and 18 respectively.

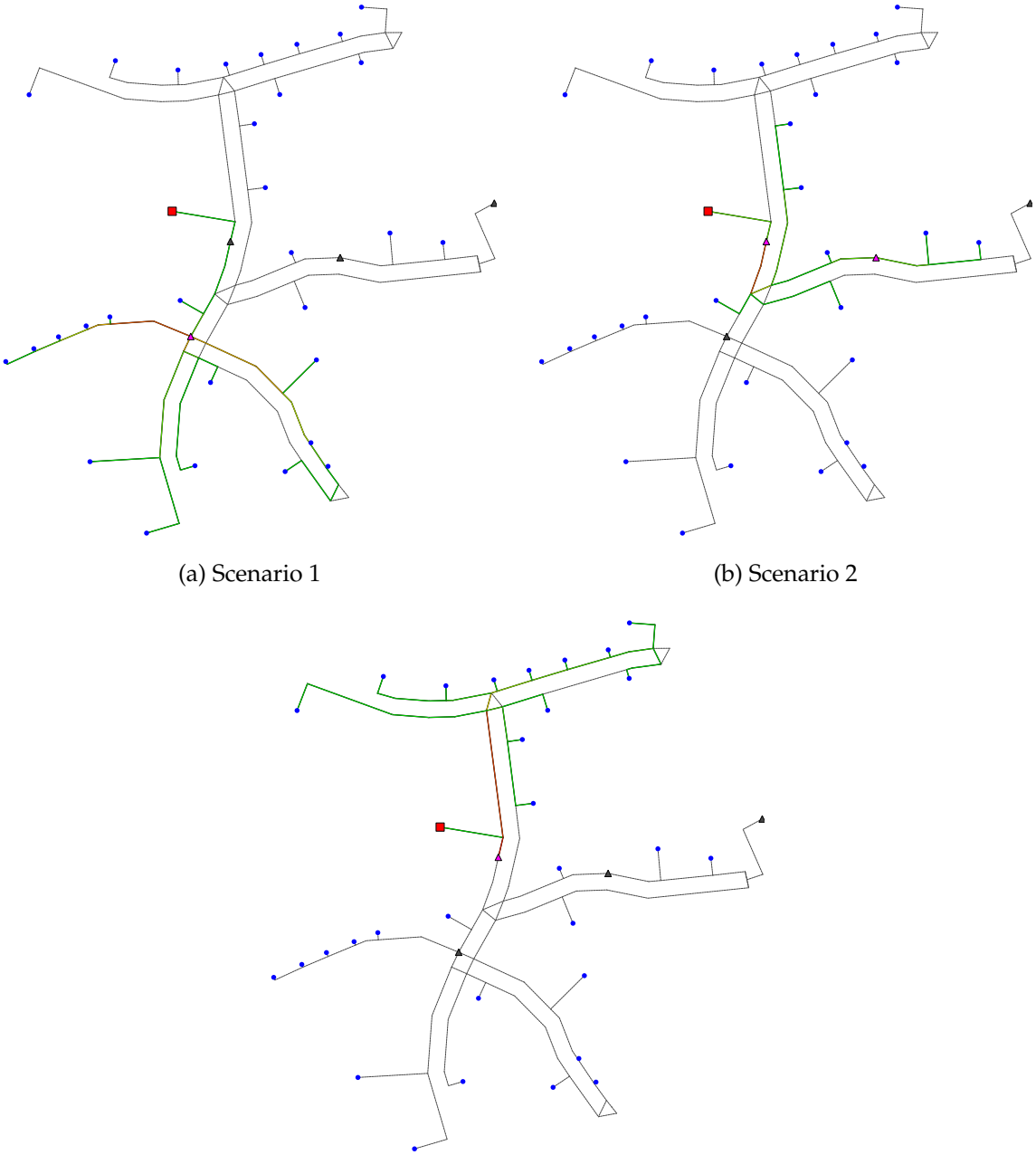


(a) Deterministic optimal solution (b) Two-stage recourse optimal solution

Figure 6.1: Optimal deterministic and stochastic solutions of the `stochnet` data set

Next, suppose we have 20 independent scenarios, each with a number of ONU demands. Three examples of such scenarios are illustrated in figure 6.2, showing different realisation possibilities for the same data set. Note also that scenarios may overlap, with one ONU shared between scenarios 1 and 2 and two ONUs shared between 2 and 3. Such realisations can occur whenever the scenarios are generated using operational thresholds, e.g. overlapping income brackets or fuzzy neighbourhood borders.

The fact that only one scenario can realise at a time is then exploited by the $BENDERS_S$ formulation (or $BENDERS_R$ equivalently). Even though the resulting optimal solution topology is almost the same as that of the deterministic solution (with the exception of two edges in the lower intersection), as seen in figure 6.1b, the objective value is only R 586,025.37, a 2.3 % decrease. In particular, the average number of fibres in each edge is 2.06, with a maximum of 11, which shows how ducts can be dimensioned lower in the stochastic solution due to re-use across scenarios. Additionally, the same splitters from the deterministic solution now only have 10 and 11 connected ONUs respectively, which means 1:16 splitters can be used throughout, resulting in additional cost savings.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Figure 6.2: Examples of potential scenario realisations for the `stochnet` data set

**Experimental methodology**

We now aim to generalise the results by looking at the objective values for more data sets. Even though these are highly instance and scenario dependent, it should provide a benchmark for what can be expected on average.

Scenarios are generated randomly by forming subsets $U_s \subset U, |U_s| = k, \forall s \in S$ from the data set, where $k$ is chosen as 50 % of the number of ONUs. Furthermore, we select random scenarios such that all ONUs are contained in at least one subset $U_s$ to avoid skewing the results in terms of model complexity. In practice, it is realistic to assume a small number of scenarios for access network design, as it is operationally expensive to accurately evaluate and compile demand uncertainty. To ensure an accurate representation, 20 scenarios for each instance are randomly generated, and only data sets from the scalability and baseline sets with fewer than 50 ONUs are considered to keep solution quality high.

Unless stated otherwise, the same procedure is followed as detailed in section 4.4.1, including all data set reduction techniques and parameters. The relevant modified cuts introduced are used for each formulation.

All models were again implemented in `C++` using the *Concert* extension of IBM ILOG CPLEX and the Qt 5.2.1 framework and run on the same hardware as the original model to ensure results were comparable. The $BENDERS_S$ formulation is used throughout, and a 2-hour time limit and 24 GiB memory limit is enforced for each test. The objective function value, gap, time to solve, $t_s$, and peak memory usage, $M_{peak}$, are recorded for each instance.

**Results and analysis**

The numerical results are given in table 6.1, showing an average deployment cost reduction of 2.77 % across all instances. Larger data sets seem to show a larger reduction since more opportunities for lower fibre and duct dimensioning exist, with the exception of `suburb1r`. For this improvement, computational effort and memory usage increased by a factor of 23.7 and 21.2 on average respectively. However, for these data set sizes, the time to solve remains reasonable, with an average computation time of only 3.5 minutes. Since the small cost saving percentage can translate into a large absolute saving, e.g. a R 49,214 saving for `mednet2`, the solution is still of great value.

Table 6.1: Two-stage recourse numerical results

| Data set | $BENDERS_C$ | | | | $BENDERS_S$ | | | | Objective |
|---|---|---|---|---|---|---|---|---|---|
| | Obj [R] | gap [%] | $t_s$ [sec] | $M_{peak}$ [MiB] | Obj [R] | gap [%] | $t_s$ [sec] | $M_{peak}$ [MiB] | reduction [%] |
| `urban3r` | 617,071 | – | 30.7 | 215 | 603,745 | – | 733 | 5,865 | 2.16 |
| `suburb1r` | 732,521 | – | 3.37 | 117 | 729,613 | – | 124 | 2,082 | 0.40 |
| `micronet` | 147,181 | – | 0.21 | 43.1 | 145,708 | – | 0.26 | 133 | 1.00 |
| `citynet1` | 209,991 | – | 0.29 | 77.8 | 209,569 | – | 3.32 | 158 | 0.20 |
| `citynet2` | 818,381 | – | 4.45 | 180 | 786,014 | – | 108 | 3,682 | 3.95 |
| `mednet1` | 279,219 | – | 0.18 | 49.2 | 269,148 | – | 0.30 | 122 | 3.61 |
| `mednet2` | 1,230,739 | – | 8.52 | 134 | 1,181,526 | – | 194 | 4,366 | 4.00 |
| `hugenet1` | 212,700 | – | 0.45 | 49.4 | 203,935 | – | 2.28 | 178 | 4.12 |
| `hugenet2` | 978,609 | – | 4.60 | 137 | 936,564 | – | 113 | 2,950 | 4.30 |
| `subnet1` | 141,740 | – | 0.07 | 44.8 | 137,055 | – | 0.36 | 99.2 | 3.31 |
| `subnet2` | 869,116 | – | 9.82 | 239 | 839,604 | 0.36 | 1,024 | >24 GiB | 3.40 |

### 6.5.2 Stochastic scalability study

#### Experimental methodology

For the scenario scalability test, the `suburb1r` and `suburb2r` instances shown in table 4.2 will be used, along with the modified cuts. Since introducing uncertainty into a model is expected to drastically increase the required computational effort, we deliberately choose smaller data sets to showcase scalability when scenarios are included. Therefore, for these data sets of less than 50 ONUs, we test using between 10 and 50 scenarios.

The same 2 hour time limit and 24 GiB memory limit are enforced for all tests, after which the run is stopped and the optimality gap of the best solution is recorded. To determine scalability performance, the solution time, $t_s$, and peak memory usage, $M_{peak}$, are recorded as the number of scenarios increase.

Table 6.2: Stochastic computational results - `suburb1r`

| | $BENDERS_S$ | | | $BENDERS_R$ | | |
|---|---|---|---|---|---|---|
| \|S\| | gap [%] | $t_s$ [sec] | $M_{peak}$ [MiB] | gap [%] | $t_s$ [sec] | $M_{peak}$ [MiB] |
| 10 | – | 39.2 | 736 | – | 330 | 327 |
| 20 | – | 124 | 2,082 | – | 387 | 348 |
| 30 | – | 161 | 2,736 | – | 463 | 405 |
| 40 | – | 246 | 3,974 | – | 358 | 408 |
| 50 | – | 131 | 2,829 | 72.6 | >2 h | 3,024 |

Table 6.3: Stochastic computational results - `suburb2r`

| | $BENDERS_S$ | | | $BENDERS_R$ | | |
|---|---|---|---|---|---|---|
| \|S\| | gap [%] | $t_s$ [sec] | $M_{peak}$ [MiB] | gap [%] | $t_s$ [sec] | $M_{peak}$ [MiB] |
| 10 | – | 317 | 3,513 | – | 2,065 | 1,074 |
| 20 | – | 497 | 6,845 | 23.8 | >2h | 6,508 |
| 30 | – | 1,045 | 12,730 | 18.3 | >2h | 6,712 |
| 40 | – | 909 | 12,985 | 20.1 | >2h | 10,014 |
| 50 | – | 2,645 | 21,034 | 14.0 | >2h | 4,879 |

**Results and analysis**

The computational results for the `suburb1r` and `suburb2r` data sets are shown in tables 6.2 and 6.3 respectively. Compared to the deterministic results, it is immediately evident that the computational effort required to solve the two-stage recourse model is much higher, increasing by an order of magnitude for `suburb1r` with 10 scenarios over the deterministic model. With 50 scenarios, this increases to almost two orders of magnitude. Memory usage for the deterministic model with the `suburb1r` and `suburb2r` data sets peaked at 110 MiB and 227 MiB respectively, resulting in at least an 11 fold increase when 50 scenarios are considered.

For the robust formulation, the `suburb1r` results suggest that it trades computational performance for memory efficiency, which is as expected due to the condensed nature of the cuts, i.e. one cut is generated across all scenarios. Additionally, even though the number of overall cuts separated is similar to the two-stage formulation, fewer

columns were generated, which illustrates that the process is more efficient. Unfortunately, $PON_{DARC}$ takes much longer to solve than $PON_{DS}$, which severely impacts the performance of the robust formulation, with individual nodes taking more than 10 minutes to separate in some instances. On further investigation, including switching LP solvers, it seems that the structure of $PON_{DARC}$ leads to poor computational performance with the simplex algorithm.

### 6.5.3 Revenue case study

We now show the impact of a revenue-based model using the `citynet2` data set and the $BENDERS_S^{rev}$ formulation. 10 scenarios are generated randomly as detailed above while each ONU is assigned a realisation probability of $\varrho_j \sim U(0.01, 1), \forall j \in \text{U}$. The probabilities assigned to each ONU will usually be related to the scenario generation process, but as the scenario and revenue parts of the formulation are independent, they need not be.

Next, we assume a fixed income per ONU, $r_i = r_j, \forall i, j \in \text{U}$, and increase it from R 10,000 to R 40,000, noting the resulting objective function value as well as the number of ONUs the model decides to connect. Results can be seen in figure 6.3. At R 10,000, no ONUs are connected, as the expected revenue would be negative otherwise. At R 11,000, 30 ONUs are suddenly connected, illustrating how the expected income they can generate only just offsets the cost to connect them, resulting in a small positive objective function, i.e. expected revenue.

This trend continues, and at R 17,000, another splitter can be added to connect additional ONUs. The maximum number of splitters used is 4, which occurs when the income per ONU reaches R 36,000. Then, the number of splitters drops back to 3 until all ONUs are connected. Note also that even though the topology changes quite drastically as $r_j$ is varied, the objective function shows almost a linear increase. Since the relation between the number of ONUs connected and the objective function is not at all obvious however, it is unlikely that network designers will find the correct config-

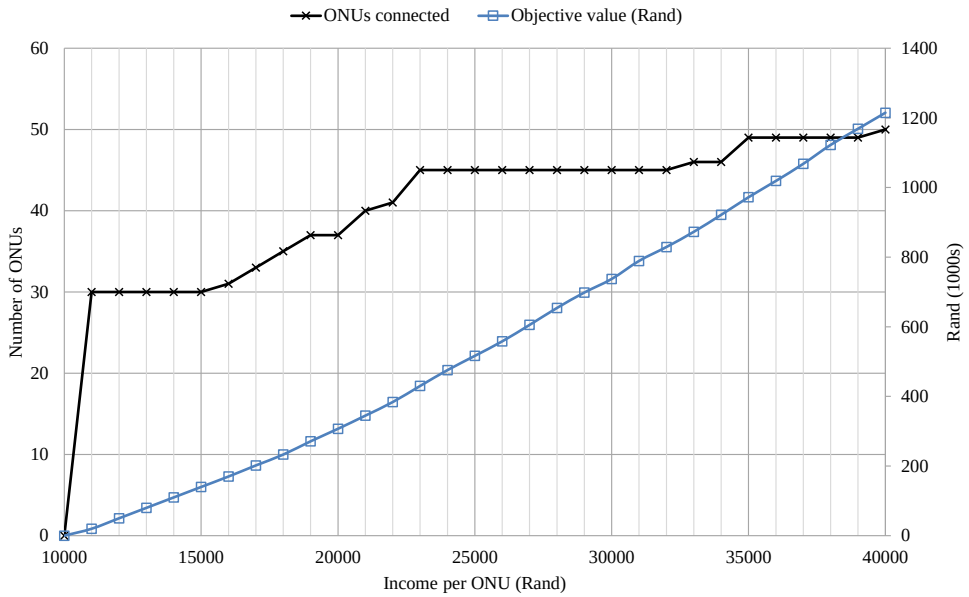



Figure 6.3: Revenue and topology results for `citynet2` data set

uration for maximum revenue using manual design methods.

Figures 6.4 and 6.5 show examples from the results at different income per ONU levels, illustrating how the model connects an increasing number of ONUs as the income per ONU increases.

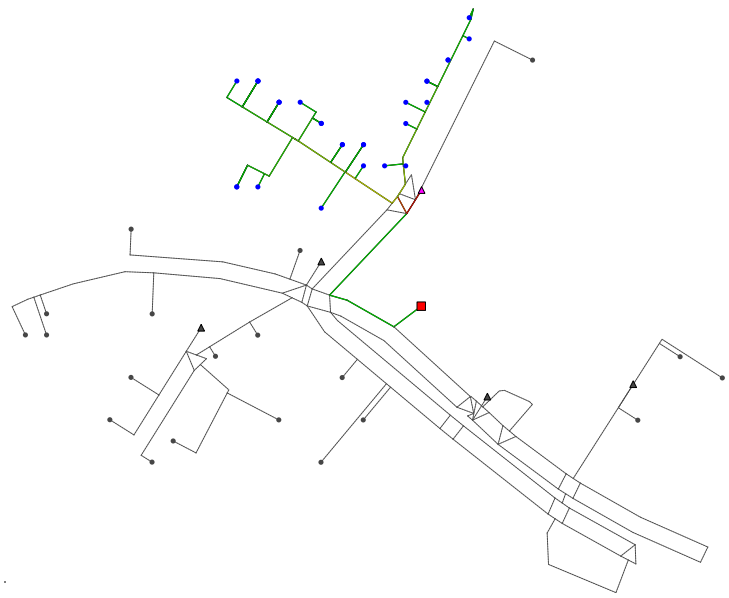
## 6.6 Conclusion

In this chapter, we detailed the process of incorporating demand uncertainty into the modelling framework using both stochastic programming and robust optimisation approaches to illustrate its flexibility. All relevant strengthening cuts and heuristics were then modified to account for these modelling changes.

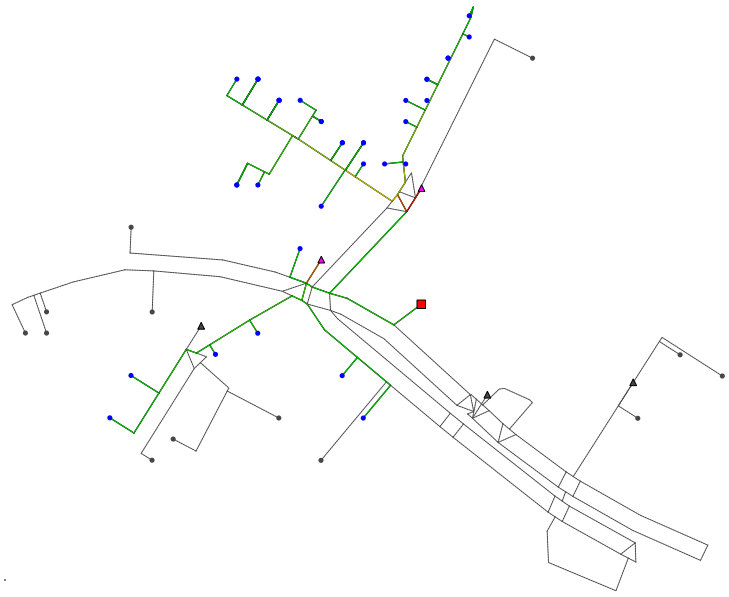
Next, a revenue-based model was formulated, illustrating how service providers can design networks directly with revenue in mind. This combines the operational and network design layers, removing the guesswork between divisions and potentially resulting in a higher quality solution for the provider. A case study was also done to showcase the behaviour of the revenue model under varying income scenarios.

The computational study of the stochastic approaches showed a 3 % reduction in total deployment cost on average, with larger data sets producing bigger gains. This reduction comes at a cost however, with computational effort and memory usage increasing by more than an order of magnitude. A scalability test where the number of scenarios is increased shows an increase in the required computational effort to solve the model, exhibiting a linear trend, with memory usage becoming a major limiting factor.

Now that uncertain demand can be accounted for in the modelling framework, we will look at extending the formulation to allow for common variations of PONDP, including accounting for multiple splitter types, survivable networks and multi-level networks.

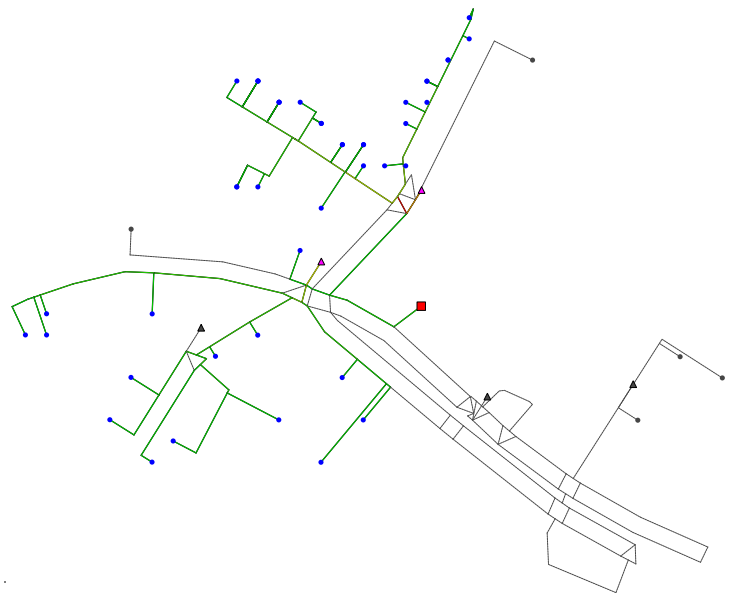


(a) Income per ONU = R 11,000

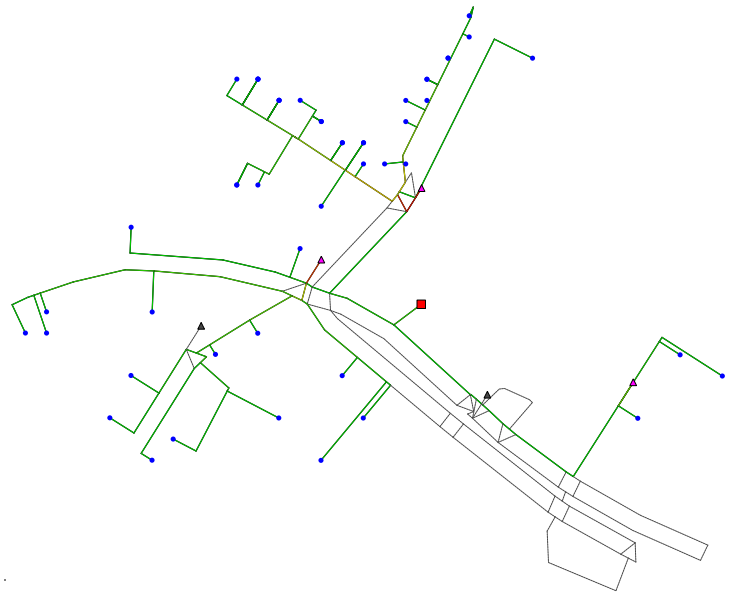


(b) Income per ONU = R 20,000

Figure 6.4: Optimal solutions for the `citynet2` data set under different revenue conditions



(a) Income per ONU = R 30,000



(b) Income per ONU = R 40,000

Figure 6.5: Optimal solutions for the `citynet2` data set under different revenue conditions (continued)

# Chapter 7

# Model refinements

*This chapter aims to illustrate the flexibility of the modelling framework by incorporating additional practical extensions of PONDP into the formulation. This includes modifying the basic formulation, relevant cuts and heuristics to account for multiple splitter types, fully- and semi-redundant survivability as well as distributed splitting in the form of a multi-level network. A computational study shows the numerical and computational performance of each extension, illustrating how computation times increase with the additional model complexity.*

## 7.1 Introduction

The previous chapter illustrated how commonly used theoretical methods to incorporate uncertainty into a model apply directly to the modelling framework introduced in chapter 4. In this chapter, we focus on refining the framework to accommodate some specific practical extensions of PONDP found in the ITU-T G.984 [11–14] recommendations, which will increase the relevance and practicality of the model for real-world PON deployments. These include dimensioning splitters in terms of the number of output ports, designing with network survivability in mind, as well as allowing mul-

tiple level networks, where splitters are cascaded to bring the feeder network even closer to the customer premises.

## 7.2 Splitter types

Dimensioning splitters can have a significant impact on the PON design as a whole. Though correctly dimensioned splitters will cost less and hence reduce the total deployment cost, the biggest impact concerns the link attenuation. Larger splitters with more output ports introduce more attenuation loss in the link between the CO and the ONU, potentially limiting the resulting network diameter. Therefore, there may be cases where a single 1:64 splitter, which has, for example, a loss of 19.8 dB, will be replaced by dual 1:32 splitters, each with a loss of 16.5 dB, to ensure the power received at each ONU is sufficient for reliable connectivity.

Extending the $BENDERS_C$ model to dimension the splitters is easily done by introducing a few additional variables and adding them to the master problem $PON_M$. First, let T be the set of all splitter types. Introduce binary variables $\boldsymbol{\varphi} \in \{0,1\}^{|\mathrm{D}|\times|\mathrm{T}|}$, which take on a value of 1 if splitter $i \in \mathrm{D}$ is of type $t \in \mathrm{T}$, and 0 otherwise. Let $\boldsymbol{n} \in \mathbb{Z}_+^{|\mathrm{T}|}$ be the number of output ports for a splitter of type $t \in \mathrm{T}$ and replace the splitter cost parameter $c^{\mathrm{SP}}$ with $\boldsymbol{c}^{\mathrm{SP}} \in \mathbb{Q}_+^{|\mathrm{T}|}$, the cost of a splitter of type $t \in \mathrm{T}$. Then, add the following constraints to $PON_M$:

$$x^{\mathrm{FD}}_{\beta(i)} \leq \sum_{t\in\mathrm{T}} n_t \varphi_i^t, \qquad \forall i \in \mathrm{D}, \tag{7.1}$$

$$\sum_{t\in\mathrm{T}} \varphi_i^t = \psi_i, \qquad \forall i \in \mathrm{D}. \tag{7.2}$$

Constraint (7.1) ensures that the number of output ports on the allocated splitter is sufficient to connect its allocated ONUs while (7.2) ensures that when a splitter is used, it only takes on a single type.

Additionally, to ensure the correct costs for the chosen splitter types are taken into account, replace the splitter cost term $\sum_{i\in\mathrm{D}} \psi_i c^{\mathrm{SP}}$ in the objective function (4.21) of $PON_M$

with the following:

$$\sum_{i \in \mathrm{D}} \sum_{t \in \mathrm{T}} \varphi_i^t c_t^{\mathrm{SP}}. \tag{7.3}$$

The combined formulation is henceforth referred to as $BENDERS_T$ and includes $PON_M$ with modifications (7.1)–(7.3), $PON_F$ and $PON_D$.

### 7.2.1 Path length cuts modification

Even though (7.1)–(7.3) is sufficient to implement splitter types into the model, the inclusion has some significant additional consequences, particularly when implementing path length constraints. Recall from section 4.3.4 that the following path length cut is added to $PON_M$ whenever paths are found that violate the network reach limit:

$$\sum_{e \in \mathrm{E}(p_i^{\mathrm{D}})} x_e^{\mathrm{TD}} + \sum_{e \in \mathrm{E}(p_i^{\mathrm{F}})} x_e^{\mathrm{TF}} + \psi_i \leq |\mathrm{E}(p_i^{\mathrm{D}})| + |\mathrm{E}(p_i^{\mathrm{F}})|, \quad \forall i \in \mathrm{D}. \tag{7.4}$$

This cut is based on the assumption that the network reach limit $\ell_{\max}^{\text{total}}$ is global for all splitters. However, since different splitter types have different loss profiles, the network reach limit now depends on the splitter type used. In particular, let $\boldsymbol{\ell}^{\max} \in \mathbb{Q}_+^{|\mathrm{T}|}$ denote the network reach limit for the different splitter types.

To cater for the different length limits, we need to be able to cut off a solution whenever the total length of paths $p_i^{\mathrm{F}}$ and $p_i^{\mathrm{D}}$ exceeds $\ell_{t^*}^{\max}$, with $t^*$ the type used for splitter $i \in \mathrm{D}$. Modifying equation (7.4) by setting the original canonical offset $\epsilon$ equal to $\varphi_i^{t^*}$, we can define the splitter type aware path length canonical cut as follows:

$$\sum_{e \in \mathrm{E}(p_i^{\mathrm{D}})} x_e^{\mathrm{TD}} + \sum_{e \in \mathrm{E}(p_i^{\mathrm{F}})} x_e^{\mathrm{TF}} + \varphi_i^{t^*} \leq |\mathrm{E}(p_i^{\mathrm{D}})| + |\mathrm{E}(p_i^{\mathrm{F}})|, \quad \forall i \in \mathrm{D}. \tag{7.5}$$

This path length cut will therefore ensure that when $\varphi_i^{t^*} = 1$, at least one edge in $\mathrm{E}(\boldsymbol{p}^{\mathrm{D}}) \cup \mathrm{E}(\boldsymbol{p}^{\mathrm{F}})$ must be switched off. By utilising $\varphi_i^{t^*}$ instead of $\psi_i$, the violation can be cleared by switching to a different splitter type with a longer reach. Additionally, since no two splitters will share *all* the edges in both $\boldsymbol{p}^{\mathrm{D}}$ and $\boldsymbol{p}^{\mathrm{F}}$, (7.5) will not cut off other

feasible paths. Note that differential cuts are not impacted, since the differential limit is not dependent on the splitter ratio.

## 7.3 Survivability

Network survivability in the PON environment attempts to safeguard customer connectivity against faults and cable breakages by deploying redundant equipment or optical fibre links between the CO and the ONU. The ITU-T G.984.1 [11] recommendation includes four types of survivability configurations, depending on the operational requirements, as was illustrated in section 2.5.

In this section, to illustrate the modelling framework flexibility, we implement full edge-disjoint G.984 Type B survivability, as well as a hybrid Type B we call *λ-disjoint* survivability. Both these types have redundant fibres in the feeder network, dual input splitters and multiple OLTs in the CO.

### 7.3.1 Full edge-disjoint survivability

To implement full edge-disjoint survivability, we utilise a layered graph approach. In this approach, we have multiple, independent layers, each containing a fully connected network, as illustrated in figure 7.1. The G.984 Type B duplex configuration only incorporates redundancy in the feeder network, so only the master and feeder sub-problem are affected. For our approach, we assume fully disjoint network layers, i.e. dedicated path protection [143].

Assume we want $N$ redundant paths for every used splitter in the feeder network. Substitute $\boldsymbol{x}^{\text{TF}}$ and $\boldsymbol{x}^{\text{FF}}$ with layered trench and flow variables $\boldsymbol{x}^{\text{TF}} \in \{0,1\}^{|\text{E}|\times N}$ and $\boldsymbol{x}^{\text{FF}} \in \mathbb{Z}_{+}^{|\text{E}|\times N}$ respectively. Next, replace $PON_M$ with the following master problem which incorporates the new variables:

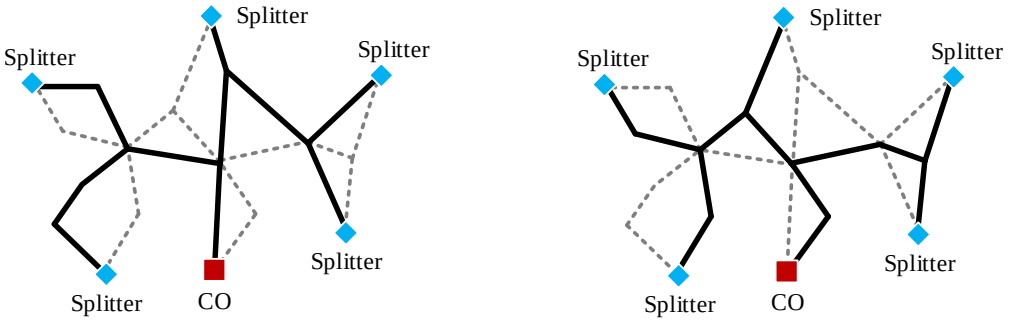



Figure 7.1: Independent connected graph layers in the feeder network

$PON_M^{surv}$:

$$\min \quad c^{\mathrm{OLT}} + |\mathrm{U}|c^{\mathrm{ONU}} + \sum_{e\in\mathrm{E}}(x_e^{\mathrm{TR}}c_e^{\mathrm{TR}} + x_e^{\mathrm{FD}}c_e^{\mathrm{F}}) + \sum_{n=0}^{N-1}\sum_{e\in\mathrm{E}} x_{en}^{\mathrm{FF}}c_e^{\mathrm{F}} + \sum_{i\in\mathrm{D}}\psi_i c^{\mathrm{SP}} \tag{7.6}$$

$$\text{s.t.} \quad x_{\beta(i)}^{\mathrm{FD}} \leq \kappa\psi_i, \quad \forall i \in \mathrm{D}, \tag{7.7}$$

$$x_{en}^{\mathrm{FF}} \leq \Delta x_e^{\mathrm{TR}}, \quad \forall e \in \mathrm{E}, n = 0, 1, \ldots, N-1, \tag{7.8}$$

$$x_e^{\mathrm{FD}} \leq \Delta x_e^{\mathrm{TR}}, \quad \forall e \in \mathrm{E}. \tag{7.9}$$

Additionally, the disaggregated trench variables for the path length constraints (4.42), (4.44) and (4.46) are replaced with:

$$x_{en}^{\mathrm{FF}} \leq \Delta x_{en}^{\mathrm{TF}}, \qquad \forall e \in \mathrm{E}, n = 0, 1, \ldots, N-1, \tag{7.10}$$

$$x_{en}^{\mathrm{TF}} \leq x_{en}^{\mathrm{FF}}, \qquad \forall e \in \mathrm{E}, n = 0, 1, \ldots, N-1, \tag{7.11}$$

$$x_{en}^{\mathrm{TF}} \leq x_e^{\mathrm{TR}}, \qquad \forall e \in \mathrm{E}, n = 0, 1, \ldots, N-1. \tag{7.12}$$

For fully edge-disjoint paths, we need to ensure that $x^{\mathrm{TF}}$ can only take on a value of 1 in one of the layers. However, we introduced indicator edges for both splitters and the CO in chapter 4, which are shared amongst all network layers. Therefore, define the subset $\mathrm{E}_{\mathrm{ind}} = \mathrm{E} \setminus \cup_{i\in\mathrm{D}\cup\{q\}}\beta(i)$, containing all edges that are not indicator edges of either a splitter or the CO, and add the following constraint to $PON_M^{surv}$:

$$\sum_{n=0}^{N-1} x_{en}^{\mathrm{TF}} \leq 1, \quad \forall e \in \mathrm{E}_{\mathrm{ind}}. \tag{7.13}$$

The feeder sub-problem is then separated for each layer, for a fixed index $n^*$, using capacity vectors $\boldsymbol{x}^{\mathrm{FF}}_{n^*}$ from the master problem. Therefore, we solve the following sub-problem:

$PON^{surv}_{F}$:

$$\min \quad \alpha \tag{7.14}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}^{\mathrm{CO}}_{\mathrm{SP}}(i)} \sum_{p \in \mathrm{P}(k)} f^{\mathrm{F}}_{p} = \psi^{*}_{i}, \quad \forall i \in \mathrm{D}, \tag{7.15}$$

$$\sum_{k \in \mathrm{K}} \sum_{p \in \mathrm{P}(k,e)} f^{\mathrm{F}}_{p} - \alpha \leq x^{\mathrm{FF}*}_{en^*}, \quad \forall e \in \mathrm{E}, \tag{7.16}$$

which, by associating dual variables $\boldsymbol{\pi}^{\mathrm{F}}_{n^*} \in \mathbb{R}^{|\mathrm{D}|}$ and $\boldsymbol{\mu}^{\mathrm{F}}_{n^*} \in \mathbb{R}^{|\mathrm{E}|}_{+}$ with constraints (7.15) and (7.16) respectively, and using the same procedure as discussed in section 4.3.2, leads to the set of metric inequalities:

$$\sum_{e \in \mathrm{E}} x^{\mathrm{FF}}_{en} \mu^{\mathrm{F}*}_{en} \geq \sum_{i \in \mathrm{D}} \psi_i \pi^{\mathrm{F}*}_{in}, \quad n = 0, 1, \ldots, N-1, \tag{7.17}$$

as well as shortest path problems for column generation:

$$\pi^{\mathrm{F}}_{in} = \min_{k \in \mathrm{K}^{\mathrm{CO}}_{\mathrm{SP}}(i)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \mu^{\mathrm{F}}_{en}, \tag{7.18}$$

which is equivalent to finding the shortest path between splitter $i \in \mathrm{D}$ and the CO in the $n$-th redundant layer of the feeder network, using $\mu^{\mathrm{F}}_{en}$ as edge weights. For subsequent sections, the full edge-disjoint formulation, consisting of $PON^{surv}_{M}$, $PON^{surv}_{F}$, $PON_{D}$ and constraints (7.10)–(7.13), will be known as $BENDERS^{surv}_{full}$.

### 7.3.2 $\lambda$-disjoint survivability

It is evident that full edge-disjoint survivability is strict in terms of redundancy, leaving no leeway for edge sharing amongst layers. This is by design to survive any edge failure, allowing service providers to provide services with high Service Level Agreements (SLAs) to customers. However, service providers may also be interested in providing a service with *some* redundancy, for a lower cost, to customers with lower

SLAs. A network with limited redundancy has what are known as *shared risk groups*, which are often distinguished by type, e.g. Shared Risk Link Group (SRLG) for link sharing and Shared Risk Equipment Group (SREG) for equipment sharing [144]. A design with shared edges would have a failure risk in-between full edge-disjoint redundant networks and networks with no redundancy. Additionally, such an approach may reduce costs in clustered graphs, e.g. two neighbourhoods on either side of a river, where fully redundant fibre will be much more expensive, as seen in figure 7.2. To that end, we introduce a semi-redundant network known as a $\lambda$-survivable network.

### $\lambda$-edge disjoint survivability

For a limited edge-disjoint survivable network, the parameter $\lambda \in \mathbb{Z}_+$ is the maximum number of edges any two redundant paths in the feeder network layers may share. Therefore, as with the edge-disjoint survivable formulation, we use $PON_M^{surv}$, $PON_F^{surv}$, $PON_D$ and (7.10)–(7.12). However, we do not add constraint (7.13).

Additionally, add variables $\boldsymbol{x}^{\mathrm{SF}} \in \{0,1\}^{|\mathrm{E}|}$ to the master problem, each of which takes on the value 1 if the corresponding edge is used in more than one layer. Since $\boldsymbol{x}^{\mathrm{TF}}$ is binary, this relation can be linearised as follows and added to the master problem:

$$x_e^{\mathrm{SF}} \leq \sum_{n=0}^{N-1} x_{en}^{\mathrm{TF}}, \qquad \forall e \in \mathrm{E}_{\mathrm{ind}}, \tag{7.19}$$

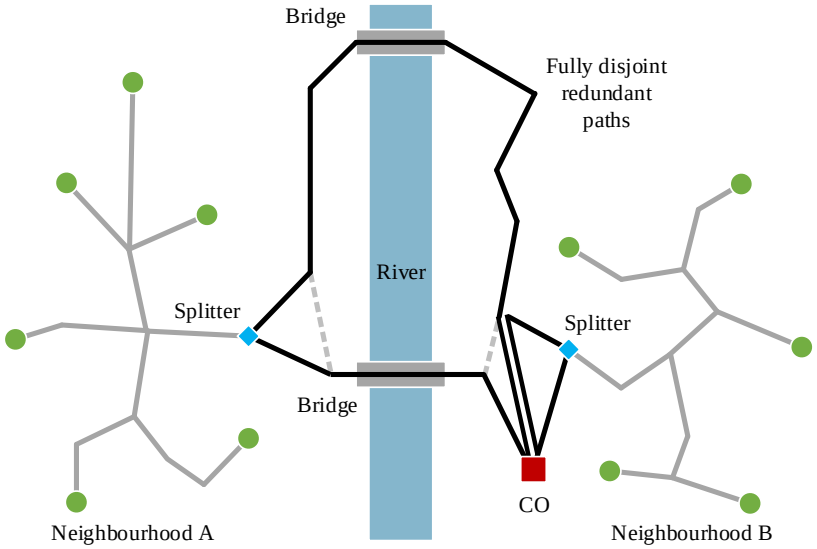
$$x_{ei}^{\mathrm{TF}} + x_{ej}^{\mathrm{TF}} \leq x_e^{\mathrm{SF}} + 1, \qquad \begin{array}{c} \forall \{i,j \,|\, 0 \leq i < j \leq N-1\}, \\ \forall e \in \mathrm{E}_{\mathrm{ind}}. \end{array} \tag{7.20}$$

Constraint (7.19) ensures that when an edge is unused in all of the layers, $\boldsymbol{x}^{\mathrm{SF}}$ takes on a value of 0, while (7.20) sets $\boldsymbol{x}^{\mathrm{SF}}$ to 1 if an edge is used in more than one layer. In the case where $N = 2$, (7.19) and (7.20) can be simplified:

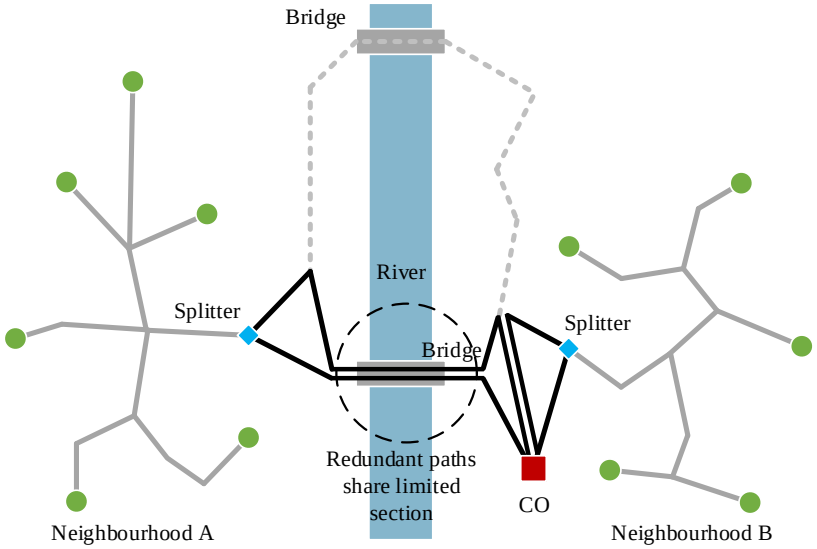
$$x_e^{\mathrm{SF}} \leq x_{e0}^{\mathrm{TF}}, \qquad \forall e \in \mathrm{E}_{\mathrm{ind}}, \tag{7.21}$$

$$x_e^{\mathrm{SF}} \leq x_{e1}^{\mathrm{TF}}, \qquad \forall e \in \mathrm{E}_{\mathrm{ind}}, \tag{7.22}$$

$$x_{e0}^{\mathrm{TF}} + x_{e1}^{\mathrm{TF}} \leq x_e^{\mathrm{SF}} + 1, \qquad \forall e \in \mathrm{E}_{\mathrm{ind}}. \tag{7.23}$$




a) Full edge-disjoint redundancy




b) λ-disjoint redundancy

Figure 7.2: Full edge-disjoint vs. $\lambda$-disjoint survivability

Next, whenever $\boldsymbol{x}^{\mathrm{SF}} \neq \mathbf{0}$, calculate the number of shared edges $w(p_u, p_v)$ for each splitter $i \in \mathrm{D}$ in every combination of layers $n$ and $m \neq n$ as follows:

$$w(p_{in}^{\mathrm{F}}, p_{im}^{\mathrm{F}}) = \sum_{e \in \mathrm{E}(p_{in}^{\mathrm{F}}) \cap \mathrm{E}(p_{im}^{\mathrm{F}})} x_e^{\mathrm{SF}}, \qquad \begin{array}{l} \forall i \in \mathrm{D}, 0 < n < N-1, \\ 0 < m < N-1, m \neq n. \end{array} \tag{7.24}$$

If, for any feeder path $p^{\mathrm{F}}_{in}$ and a redundant path $p^{\mathrm{F}}_{im}$ as specified above, $w(p^{\mathrm{F}}_{in}, p^{\mathrm{F}}_{im}) > \lambda$, we know that the feeder path shares more than $\lambda$ edges with the redundant path in another layer, and therefore need to derive a cut to remove this infeasibility. To avoid cutting off feasible paths that share a subset of the feeder edges, we once again retreat to the canonical cut form to derive the necessary cut. In this case, since we involve edge variables from multiple layers, we have to consider the following form with $\epsilon = \psi_i$:

$$\sum_{n=0}^{N-1} \sum_{\substack{e \in \mathrm{E}: \\ x^{\mathrm{TF}*}_{en} = 0}} x^{\mathrm{TF}}_{en} + \sum_{n=0}^{N-1} \sum_{\substack{e \in \mathrm{E}: \\ x^{\mathrm{TF}*}_{en} = 1}} (1 - x^{\mathrm{TF}}_{en}) \geq \psi_i. \tag{7.25}$$

Like the other canonical cuts in basic form, this cut is exceptionally weak, and only ensures that at least one variable in one of the layers needs to toggle between 0 and 1, removing the infeasible solution. However, with some basic assumptions, we can improve the strength of this cut significantly. Firstly, for a given combination of feeder path $p^{\mathrm{F}}_{in}$ and a redundant partner $p^{\mathrm{F}}_{im}$, we know that turning on an edge in the same layer as the feeder path will still result in a violation as it will not reduce the number of shared edges. Similarly, turning on an edge in the redundant layer will not reduce the number of shared edges. This results in the following cut:

$$\sum_{\substack{e \in \mathrm{E}: \\ x^{\mathrm{TF}*}_{en} = 1}} (1 - x^{\mathrm{TF}}_{en}) + \sum_{\substack{e \in \mathrm{E}: \\ x^{\mathrm{TF}*}_{em} = 1}} (1 - x^{\mathrm{TF}}_{em}) \geq \psi_i. \tag{7.26}$$

Next, because the number of shared edges cannot be reduced by turning off an edge that is not part of $p^{\mathrm{F}}_{in}$ or $p^{\mathrm{F}}_{im}$, we can constrain the space to only edges contained in these paths. Furthermore, note that turning off an edge in the redundant layer that is not shared with the feeder path will also not reduce the number of shared edges. However, if we were to constrain the cut to just these intersecting edges, there may be rare cases where all subsequent redundant paths for splitter $i$ are feasible, but feasible redundant paths for other splitters cross *all* intersecting edges $\mathrm{E}(p^{\mathrm{F}}_{in}) \cap \mathrm{E}(p^{\mathrm{F}}_{im})$, which would result in a violated cut. Therefore, we can simplify the cut to:

$$\sum_{e \in \mathrm{E}(p^{\mathrm{F}}_{in})} x^{\mathrm{TF}}_{en} + \sum_{e \in \mathrm{E}(p^{\mathrm{F}}_{im})} x^{\mathrm{TF}}_{em} + \psi_i \leq |\mathrm{E}(p^{\mathrm{F}}_{in})| + |\mathrm{E}(p^{\mathrm{F}}_{im})|. \tag{7.27}$$

Since no other feeder path will share all edges with $p^{\mathrm{F}}_{in}$ in layer $n$, and since any other redundant path in layer $m$ will not share all edges with $p^{\mathrm{F}}_{im}$, (7.27) will not cut off other feasible solutions. When (7.27) is separated within the Benders formulation according to (7.24), it is known as $BENDERS^{surv}_{\lambda\text{-}edge}$.

#### $\lambda$-length disjoint survivability

Another way of handling semi-redundant network links is to limit the length of the shared risk group. In this case, $\lambda \in \mathbb{Q}_+$ indicates the maximum length (in metres) that may be shared between redundant paths in different layers. The same approach is taken for the length-disjoint version as was illustrated for the $\lambda$-edge disjoint version above, except that we replace the violation criteria of the cut.

Again, if $\boldsymbol{x}^{\mathrm{SF}} \neq \mathbf{0}$, we now calculate the total length of the shared edges $\ell(p_u, p_v)$ for each splitter $i \in \mathrm{D}$ in every combination of layers $n$ and $m \neq n$ as follows:

$$\ell(p^{\mathrm{F}}_{in}, p^{\mathrm{F}}_{im}) = \sum_{e \in \mathrm{E}(p^{\mathrm{F}}_{in}) \cap \mathrm{E}(p^{\mathrm{F}}_{im})} \ell_e x^{\mathrm{SF}}_e, \qquad \begin{array}{l} \forall i \in \mathrm{D}, 0 < n < N-1, \\ 0 < m < N-1, m \neq n. \end{array} \tag{7.28}$$

In this case, if for any feeder path $p^{\mathrm{F}}_{in}$ and a redundant path $p^{\mathrm{F}}_{im}$ as specified above, $\ell(p^{\mathrm{F}}_{in}, p^{\mathrm{F}}_{im}) > \lambda$, we know that the redundant path shares a section with the feeder path that is longer than $\lambda$, and therefore need to remove this infeasibility by adding (7.27).

When criteria (7.28) is used instead of (7.24), the formulation is called $BENDERS^{surv}_{\lambda\text{-}length}$.

### 7.3.3 Algorithmic modification

Since additional feeder network layers exist in both the full edge-disjoint and $\lambda$-disjoint formulations, all strengthening cuts related to the feeder network are duplicated for each network layer. First, the feeder connectivity cut, $\mathrm{CON}_{\mathrm{F}}$, is redefined as follows

and separated for each feeder layer:

$$\text{CON}_{\text{F}}^{\text{surv}}: \quad \sum_{e\in\delta(\text{W})} x_{en}^{\text{TF}} \geq \psi_i, \qquad \begin{array}{c}\forall\,\text{W}\subset\text{V}\setminus\{q\},\forall i\in\text{W}\cap\text{D}\neq\varnothing,\\ n=0,1,\ldots,N-1.\end{array} \tag{7.29}$$

Similarly, the feeder flow-cutset inequalities are redefined for each feeder layer:

$$\text{CUT}_{\text{F}}^{\text{surv}}: \quad \sum_{\text{W}\cap\{q\}} x_{\beta(q)n}^{\text{FF}} + \sum_{e\in\delta(\text{W})} x_{en}^{\text{FF}} \geq \sum_{i\in\text{W}\cap\text{D}} x_{\beta(i)n}^{\text{FF}}, \qquad \begin{array}{c}\forall\,\text{W}\subset\text{V},\text{W}\cap\text{D}\neq\varnothing,\\ n=0,1,\ldots,N-1.\end{array} \tag{7.30}$$

The *PRIMAL* heuristic presented in section 5.5 is easily modified to derive solutions for each feeder layer by duplicating all operations done on the first layer in subsequent layers.

Finally, to ensure routing feasibility, the *FEAS* model from section 4.4.3 is modified to try to route flow on all redundant layers. Therefore, the feeder flow and capacity constraints, given in (4.68) and (4.70) respectively, are replaced by the following:

$$\sum_{j\in\phi(i)} f_{ijn}^{\text{F}} - \sum_{j\in\phi(i)} f_{jin}^{\text{F}} = \begin{cases} -x_{\beta(i)n}^{\text{FF}*}, & i\in\text{D},\\ x_{\beta(i)n}^{\text{FF}*}, & i=q,\\ 0, & \text{otherwise,}\end{cases} \qquad \begin{array}{c}\forall i\in\text{V},\\ n=0,1,\ldots,N-1,\end{array} \tag{7.31}$$

$$f_{ijn}^{\text{F}} + f_{jin}^{\text{F}} = x_{en}^{\text{FF}*}, \quad \forall(ij)=e\in\text{E}, n=0,1,\ldots,N-1. \tag{7.32}$$

The resulting formulation is known as *FEAS*$^{surv}$. When the model is infeasible, replace (7.32) with (7.33) and resolve the model to determine sub-optimal routing feasibility.

$$f_{ijn}^{\text{F}} + f_{jin}^{\text{F}} \leq x_{en}^{\text{FF}*}, \quad \forall(ij)=e\in\text{E}, n=0,1,\ldots,N-1. \tag{7.33}$$

## 7.4 Multi-level networks

In section 2.4, multi-level PONs were introduced, which allow for the cascading of splitters, i.e. splitters connecting to other splitters. This increases the effective feeder fibre penetration, pushing it closer to the ONU and hence providing further savings due to reduced overall fibre usage.

Adapting $PON_M$ to accommodate multi-level splitters is not trivial, but due to the structure of the modelling framework, can be done by including an arbitrary number of independent *intermediate* networks. These intermediate networks are connected between the feeder and distribution networks, and are separated by level, i.e. the number of splitter hops to the CO. To this end, for an $M$-level PON, which means there can be a maximum of $M > 1$ splitters between any ONU and the CO, let $\mathrm{L} = \{0, 1, \ldots, M-1\}$ be the set of all level indices. If $M = 1$, the standard single-level model formulation detailed in chapter 4 can be used, as no intermediate networks exist.

### 7.4.1 Preprocessing

To differentiate between splitters of different levels, we add $M - 1$ auxiliary splitters to the graph for every original splitter present, similar to the indicator edge operation in chapter 4, with each splitter $i \in \mathrm{D}$ having an independent indicator edge $\beta(i)$ and a specific level $\ell \in \mathrm{L}$. Therefore, adjacent to every original splitter location (before preprocessing), we now have $M$ splitters, with levels 0 through $M - 1$, as illustrated in figure 7.3.

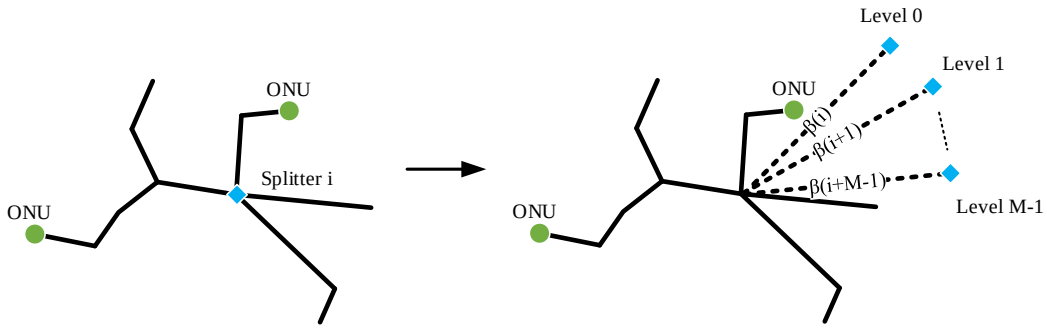



Figure 7.3: Graph preprocessing for multi-level PON

The newly added auxiliary splitters have to be linked to the original, since they are virtual in nature, representing the same splitter connected at different levels. To this end, let $\mathrm{D}_i^{\mathrm{A}} \subset \mathrm{D}$ be the subset of splitters from other levels that are associated with splitter $i \in \mathrm{D}$, including splitter $i$ itself. In other words, for any splitter $i$ of a level $m$,

$\mathrm{D}_i^{\mathrm{A}}$ contains all splitters of levels $0 < \ell < M-1$ that are adjacent to splitter $i$, as well as splitter $i$. Therefore, $|\mathrm{D}_i^{\mathrm{A}}| = M, \forall i \in \mathrm{D}$.

### 7.4.2 Model modification

Introduce intermediate feeder variables $\boldsymbol{x}^{\mathrm{FI}} \in \mathbb{Z}_+^{|\mathrm{E}|\times|\mathrm{L}|}$, with $\boldsymbol{x}_\ell^{\mathrm{FI}}$ denoting the variables for the level $\ell$ intermediate network, where $\ell \in \mathrm{L}, \ell \neq M-1$. Therefore, level 0 splitters connect to the CO via $\boldsymbol{x}^{\mathrm{FF}}$, level 1 splitters connect to level 0 splitters via $\boldsymbol{x}_0^{\mathrm{FI}}$ and ONUs connect to any level splitters through $\boldsymbol{x}^{\mathrm{FD}}$. In general then, level $\ell$ splitters connect to level $\ell-1$ splitters through $\boldsymbol{x}_{(\ell-1)}^{\mathrm{FI}}, 1 < \ell \leq M-1$, as illustrated in figure 7.4.

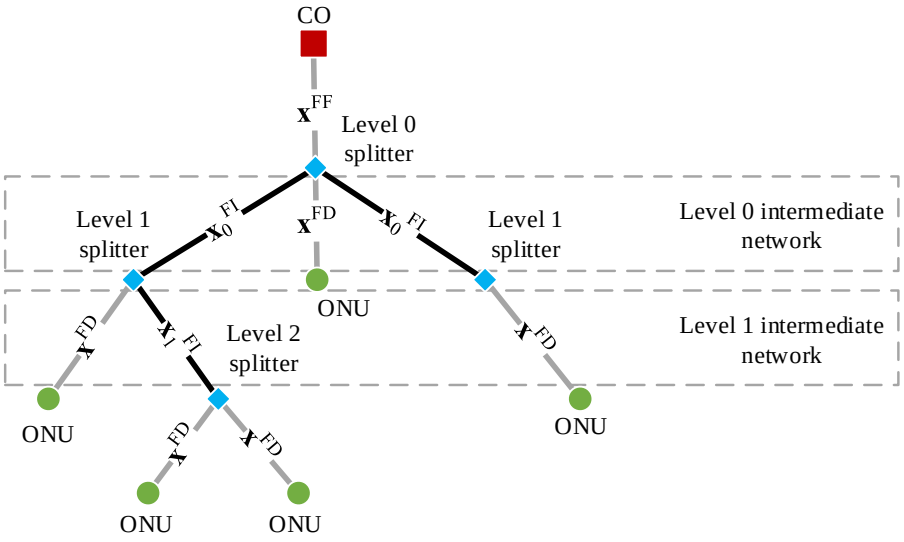



Figure 7.4: Intermediate networks in the multi-level formulation

Next, let $\mathrm{D}_\ell \subseteq \mathrm{D}$ be the set of all splitters of level $\ell \in \mathrm{L}$ and let $\mathrm{K}_{\mathrm{SP}(u)}^{\mathrm{SP}(v)}(i) \subset \mathrm{K}$ denote all commodities between splitter $i \in \mathrm{D}_u$ in level $u$ and a splitter in level $v$.

Furthermore, let $\boldsymbol{\Psi}^{\mathrm{I}} \in \mathbb{Z}_+^{|\mathrm{D}\setminus\mathrm{D}_{M-1}|}$ and $\boldsymbol{\Psi}^{\mathrm{D}} \in \mathbb{Z}_+^{|\mathrm{D}|}$ respectively denote the number of downstream fibres from other splitters and ONUs connected to a splitter. For ease of notation, let $\boldsymbol{\Psi} \in \mathbb{Z}_+^{|\mathrm{D}|}$ be the total number of downstream fibres connected to a splitter.

Then, by aggregating the fixed cost, with $c^{\text{fixed}} = c^{\text{OLT}} + |\text{U}|c^{\text{ONU}}$, we can write the multi-level master problem $PON_{ML}$ as follows:

$PON_{ML}$:

$$\min \quad c^{\text{fixed}} + \sum_{e \in \text{E}} \left( x_e^{\text{TR}} c_e^{\text{TR}} + c_e^{\text{F}} \left[ x_e^{\text{FD}} + x_e^{\text{FF}} + \sum_{\ell \in \text{L}} x_{e\ell}^{\text{FI}} \right] \right) + \sum_{i \in \text{D}} \psi_i c^{\text{SP}} \tag{7.34}$$

$$\text{s.t.} \quad x_{\beta(i)}^{\text{FD}} = \Psi_i^{\text{D}}, \quad \forall i \in \text{D}, \tag{7.35}$$

$$x_{\beta(i)}^{\text{FF}} = \psi_i, \quad \forall i \in \text{D}_0, \tag{7.36}$$

$$x_{\beta(i)(\ell-1)}^{\text{FI}} = \psi_i, \quad \forall i \in \text{D}_\ell, \forall \ell \in \text{L}, \ell \neq 0, \tag{7.37}$$

$$x_{\beta(i)(\ell)}^{\text{FI}} = \Psi_i^{\text{I}}, \quad \forall i \in \text{D}_\ell, \forall \ell \in \text{L}, \ell \neq M-1, \tag{7.38}$$

$$\Psi_i^{\text{I}} + \Psi_i^{\text{D}} = \Psi_i, \quad \forall i \in \text{D} \setminus \text{D}_{M-1}, \tag{7.39}$$

$$\Psi_i^{\text{D}} = \Psi_i, \quad \forall i \in \text{D}_{M-1}, \tag{7.40}$$

$$\Psi_i \leq \kappa \psi_i, \quad \forall i \in \text{D}, \tag{7.41}$$

$$\sum_{i \in \text{D}_0} \psi_i \geq 1, \tag{7.42}$$

$$\sum_{j \in \text{D}_i^{\text{A}}} \psi_j \leq 1, \quad \forall i \in \text{D}, \tag{7.43}$$

$$x_e^{\text{FF}} \leq \Delta x_e^{\text{TR}}, \quad \forall e \in \text{E}, \tag{7.44}$$

$$x_{e\ell}^{\text{FI}} \leq \Delta x_e^{\text{TR}}, \quad \forall e \in \text{E}, \forall \ell \in \text{L}, \tag{7.45}$$

$$x_e^{\text{FD}} \leq \Delta x_e^{\text{TR}}, \quad \forall e \in \text{E}. \tag{7.46}$$

Apart from all the deployment costs from $PON_M$, the objective (7.34) now also includes fibre cost for the intermediate networks. Constraints (7.35) and (7.36) count the number of ONUs and COs connected to a splitter, noting that only level 0 splitters can connect to the CO. Next, constraints (7.37) and (7.38) count the number of incoming and outgoing fibres in each splitter of level $\ell$. (7.39)–(7.41) adds the number of splitters and ONUs to $\boldsymbol{\Psi}$ and sets the splitter capacity. Constraint (7.42) ensures that there should be at least one level 0 splitter, while (7.43) ensures that at a specific splitter location, only one level may be active. Finally, constraints (7.44) through (7.46) respectively set the trench usage variables of the feeder, intermediate and distribution networks based on the number of optical fibres routed through an edge.

Next, we look at the sub-problems. $PON_D$ is left as-is since ONUs are allowed to connect to any level splitter. Since only level 0 splitters may connect to the CO however, $PON_F$ is altered to reflect this:

$PON_{FL}$:

$$\min \quad \alpha \tag{7.47}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}(i)} \sum_{p \in \mathrm{P}(k)} f_p^{\mathrm{F}} = \psi_i^*, \quad \forall i \in \mathrm{D}_0, \tag{7.48}$$

$$\sum_{k \in \mathrm{K}} \sum_{p \in \mathrm{P}(k,e)} f_p^{\mathrm{F}} - \alpha \leq x_e^{\mathrm{FF}*}, \quad \forall e \in \mathrm{E}, \tag{7.49}$$

which, after associating dual variables $\boldsymbol{\pi}^{\mathrm{F}} \in \mathbb{R}^{|\mathrm{D}_0|}$ and $\boldsymbol{\mu}^{\mathrm{F}} \in \mathbb{R}_+^{|\mathrm{E}|}$ to (7.48) and (7.49) respectively, leads to the metric inequality:

$$\sum_{e \in \mathrm{E}} x_e^{\mathrm{FF}} \mu_e^{\mathrm{F}*} \geq \sum_{i \in \mathrm{D}_0} \psi_i \pi_i^{\mathrm{F}*}, \tag{7.50}$$

as well as the shortest path calculation for column generation:

$$\pi_i^{\mathrm{F}} = \min_{k \in \mathrm{K}_{\mathrm{SP}}^{\mathrm{CO}}(i)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \mu_e^{\mathrm{F}}, \quad \forall i \in \mathrm{D}_0. \tag{7.51}$$

Since we also need to ensure feasible routing for the intermediate networks, we introduce the intermediate sub-problem for a given level $\ell^* \in \mathrm{L}, \ell^* > 0$. By letting $\boldsymbol{f}_{\ell^*}^{\mathrm{I}} \in \mathbb{R}^{|\mathrm{P}|}$ be the flow in the intermediate network, and $\boldsymbol{x}_{\ell^*-1}^{\mathrm{FI}*}$ and $\boldsymbol{\psi}^*$ be the fixed capacity and splitter usage vectors from the master problem respectively, we can define the sub-problem as follows:

$PON_I^{\ell^*}$:

$$\min \quad \alpha \tag{7.52}$$

$$\text{s.t.} \quad \sum_{k \in \mathrm{K}_{\mathrm{SP}(\ell^*)}^{\mathrm{SP}(\ell^*-1)}(i)} \sum_{p \in \mathrm{P}(k)} f_{p\ell^*}^{\mathrm{I}} = \psi_i^*, \quad \forall i \in \mathrm{D}_{\ell^*}, \tag{7.53}$$

$$\sum_{k \in \mathrm{K}} \sum_{p \in \mathrm{P}(k,e)} f_{p\ell^*}^{\mathrm{F}} - \alpha \leq x_{e(\ell^*-1)}^{\mathrm{FI}*}, \quad \forall e \in \mathrm{E}. \tag{7.54}$$

Associating dual variables $\boldsymbol{\pi}_{\ell*}^{\mathrm{I}} \in \mathbb{R}^{|\mathrm{D}_{\ell^*}|}$ and $\boldsymbol{\mu}_{\ell^*}^{\mathrm{I}} \in \mathbb{R}_+^{|\mathrm{E}|}$ with constraints (7.53) and (7.54) respectively, we can write the corresponding dual:

$PON_I^{\ell^*}$':

$$\max \quad \sum_{i \in \mathrm{D}_{\ell^*}} \psi_i^* \pi_{i\ell^*}^{\mathrm{I}} - \sum_{e \in \mathrm{E}} x_{e(\ell^*-1)}^{FI*} \mu_{e\ell^*}^{\mathrm{I}} \tag{7.55}$$

$$\text{s.t.} \quad \sum_{e \in \mathrm{E}} \mu_{e\ell^*}^{\mathrm{I}} \leq 1, \tag{7.56}$$

$$\pi_{i\ell^*}^{\mathrm{I}} - \sum_{e \in \mathrm{E}(p)} \mu_{e\ell^*}^{\mathrm{I}} \leq 0, \qquad \begin{array}{c} \forall p \in \mathrm{P}(k), \forall k \in \mathrm{K}_{\mathrm{SP}(\ell^*)}^{\mathrm{SP}(\ell^*-1)}(i), \\ \forall i \in \mathrm{D}_{\ell^*}. \end{array} \tag{7.57}$$

Next, for a positive objective value of $PON_I^{\ell^*}$, signifying insufficient capacity for the level $\ell^*$ intermediate network, we use the *Japanese Theorem* to derive the set of metric inequalities:

$$\sum_{e \in \mathrm{E}} x_{e(\ell-1)}^{FI} \mu_{e\ell}^{\mathrm{I}*} \geq \sum_{i \in \mathrm{D}_\ell} \psi_i \pi_{i\ell}^{\mathrm{I}*}, \quad \forall \ell \in \mathrm{L}, \ell \neq 0, \tag{7.58}$$

with $\boldsymbol{\mu}_\ell^{\mathrm{I}*}$ and $\boldsymbol{\pi}_\ell^{\mathrm{I}*}$ solution vectors from $PON_I^{\ell}$', the dual of the level $\ell$ intermediate sub-problem.

Column generation for a level $\ell^*$ intermediate problem is done by calculating the shortest paths between each splitter $i \in \mathrm{D}_{\ell^*}$ in level $\ell^*$ and any other splitter in level $\ell^* - 1$ using edge weights $\mu_{e\ell^*}^{\mathrm{I}}$ as follows:

$$\pi_{i\ell^*}^{\mathrm{I}} = \min_{k \in \mathrm{K}_{\mathrm{SP}(\ell^*)}^{\mathrm{SP}(\ell^*-1)}(i)} \min_{p \in \mathrm{P}(k)} \sum_{e \in \mathrm{E}(p)} \mu_{e\ell^*}^{\mathrm{I}}, \quad \forall i \in \mathrm{D}_{\ell^*}. \tag{7.59}$$

The complete multi-level formulation, henceforth known as $BENDERS_{ML}$, then consists of $PON_{ML}$, $PON_{FL}$, $PON_D$ and $PON_I^{\ell^*}$, with $\ell^* \in \mathrm{L}, \ell^* \neq 0$. Hence, we have a master problem and $M + 1$ sub-problems.

### 7.4.3 Homogeneous and heterogeneous networks

As it is defined in the previous section, $BENDERS_{ML}$ allows both splitters and ONUs to connect to the same splitter, which results in a heterogeneous network. Recall from section 2.4 that service providers may, to simplify the network from an operational point of view, enforce that when a splitter is used, either other splitters or ONUs may

connect to it, but non-simultaneously. This type of network configuration is then homogeneous in nature.

Most of the variables and sets to enforce a homogeneous multi-level network are already defined, so, after introducing binary variables $\boldsymbol{\psi}^{\mathrm{D}} \in \{0,1\}^{|\mathrm{D}\setminus\mathrm{D}_{M-1}|}$, indicating whether a splitter has ONUs connected to it, we only need to add the following two constraints to $PON_{ML}$:

$$\Psi_i^{\mathrm{D}} \leq \Delta\psi_i^{\mathrm{D}}, \qquad \forall i \in \mathrm{D}\setminus\mathrm{D}_{M-1}, \tag{7.60}$$

$$\Psi_i^{\mathrm{I}} \leq \Delta(1-\psi_i^{\mathrm{D}}), \qquad \forall i \in \mathrm{D}\setminus\mathrm{D}_{M-1}. \tag{7.61}$$

Constraints (7.60) and (7.61) ensure that when $\Psi_i^{\mathrm{D}} > 0$ for any splitter $i \in \mathrm{D}\setminus\mathrm{D}_{M-1}$, $\Psi_i^{\mathrm{I}} = 0$ must hold, and vice versa. This ensures that either only ONUs or only splitters can connect to a splitter $i$. Note that level $M-1$ splitters are not included in the cut, as they are already homogeneous, i.e. only ONUs can connect to them.

### 7.4.4 Path length cuts modification

Since additional splitters can now exist between an ONU and the CO, the derivation of path length cuts is affected once again. Introduce binary trench usage variables $\boldsymbol{x}^{\mathrm{TI}} \in \{0,1\}^{|\mathrm{E}|\times|\mathrm{L}|}$ for the intermediate network and, in addition to the disaggregated trench variable constraints (4.42)–(4.47) described in section 4.3.4, add the following to $PON_{ML}$:

$$x_{e\ell}^{\mathrm{FI}} \leq \Delta x_{e\ell}^{\mathrm{TI}}, \quad \forall e \in \mathrm{E}, \forall \ell \in \mathrm{L}, \tag{7.62}$$

$$x_{e\ell}^{\mathrm{TI}} \leq x_{e\ell}^{\mathrm{FI}}, \quad \forall e \in \mathrm{E}, \forall \ell \in \mathrm{L}, \tag{7.63}$$

$$x_{e\ell}^{\mathrm{TI}} \leq x_{e}^{\mathrm{TR}}, \quad \forall e \in \mathrm{E}, \forall \ell \in \mathrm{L}. \tag{7.64}$$

Next, we need to enforce path length constraints on all possible combinations of used feeder, intermediate and distribution paths between the CO and ONUs utilising their effective split ratios. This is done by following the steps below:

1. **Path grouping** - For each splitter, group all used distribution, feeder and intermediate paths connected to it, i.e. if $f_p^{\mathrm{F}*} \geq \epsilon$, $f_p^{\mathrm{D}*} \geq \epsilon$ and $f_{p\ell}^{\mathrm{I}*} \geq \epsilon$ respectively. When the capacity vectors from the master are integer, i.e. when separating path length cuts for integer feasible solutions, $\epsilon = 1$, otherwise, $0 < \epsilon \ll 1$.

2. **Auxiliary graph** - Construct an auxiliary digraph with vertices $\hat{\mathrm{D}} \cup \{q\}$, where $\hat{\mathrm{D}} = \{i \in \mathrm{D} | \psi_i^* \geq \epsilon\}$ is the set of all used splitters. Arcs are added between vertices where a used path exists, orientated towards the CO or the lower level splitter. Since path length cuts are separated for sub-optimal solutions, multiple paths may be present between commodities, forming a Directed Acyclic Graph (DAG).

3. **Path enumeration** - All possible simple paths are now enumerated between each splitter and the CO using depth-first search with backtracking. To avoid confusion, these paths will henceforth be called *upstream links*. The upstream links therefore contain all *upstream* paths towards the CO, crossing both the intermediate and feeder networks.

4. **Split ratio computation** - For every upstream link, calculate the effective split ratio by multiplying the split ratios of each splitter contained in the link. If, for example, a 1:2 splitter is connected via two other 1:4 splitters to the CO, the effective split ratio becomes 1:32 ($2 \times 4 \times 4 = 32$).

5. **Reach limit** - For each upstream link of each splitter, determine or calculate (using equation (4.61)) its network reach limit based on its effective split ratio or the sum of the splitter insertion losses in dB.

For each distribution splitter $i$, if the length of any combination of a distribution path and an upstream link exceeds the network reach limit, we derive violated path length cuts for each connected ONU. Therefore, for a combination of a distribution path $p_i^{\mathrm{D}}$, intermediate paths $p_{i\ell}^{\mathrm{I}}$ (contained in the upstream link of splitter $i$) and a feeder path $p_i^{\mathrm{F}}$ that violates the corresponding length limit $\ell_i^{\max}$, we extend the path length canonical

cut as follows:

$$\sum_{e\in \mathrm{E}(p_i^{\mathrm{D}})} x_e^{\mathrm{TD}} + \sum_{e\in \mathrm{E}(p_i^{\mathrm{F}})} x_e^{\mathrm{TF}} + \sum_{j\in \mathrm{L}, j<\ell} \sum_{e\in \mathrm{E}(p_{ij}^{\mathrm{I}})} x_{ej}^{\mathrm{TI}} + \psi_i \leq \Delta, \quad \forall i \in \mathrm{D}_\ell, \ell \in \mathrm{L}, \tag{7.65}$$

with $\Delta = |\mathrm{E}(p^{\mathrm{D}})| + |\mathrm{E}(p^{\mathrm{F}})| + \sum_{j\in \mathrm{L}, j<\ell} |\mathrm{E}(p_{ij}^{\mathrm{I}})|$. Therefore, for a violated cut, i.e. a path that exceeds the length limit, at least one edge in the distribution, feeder or intermediate network needs to be turned off. Again, since no other path will share all edges in all the corresponding networks, the equation above will not cut off subsequent feasible paths.

**Multiple splitter types**

Since multi-level networks will typically be used alongside multiple splitter types to allow for the substitution of a large splitter with cascaded smaller ones, we also need to incorporate the splitter types of section 7.2. Utilising the canonical cut form, we can derive path length cuts for each connected ONU, again for each combination of $p_i^{\mathrm{D}}$, $p_{i\ell}^{\mathrm{I}}$ and $p_i^{\mathrm{F}}$ that violates the corresponding length limit $\ell_i^{\max}$.

Let the function $g$ output a unique sum of all type-aware splitter usage variables, $\varphi_i^t$, used in a link, or a given combination of intermediate and feeder paths. Furthermore, let $|g|$ denote the number of different variables in the function output. As an example to illustrate the output of $g$, assume for a given link, a splitter 1 is used, which is connected via an intermediate path to a splitter 2. Furthermore, assume splitter 1 is of type $a$ and splitter 2 is of type $b$. Then, $g = \varphi_1^a + \varphi_2^b$ and $|g| = 2$.

Now we can derive the canonical multi-level path length cut as follows:

$$\sum_{e\in \mathrm{E}(p_i^{\mathrm{D}})} x_e^{\mathrm{TD}} + \sum_{e\in \mathrm{E}(p_i^{\mathrm{F}})} x_e^{\mathrm{TF}} + \sum_{j\in \mathrm{L}, j<\ell} \sum_{e\in \mathrm{E}(p_{ij}^{\mathrm{I}})} x_{ej}^{\mathrm{TI}} + g(p_i^{\mathrm{F}}, \boldsymbol{p}_i^{\mathrm{I}}) \leq \Delta + |g(p_i^{\mathrm{F}}, \boldsymbol{p}_i^{\mathrm{I}})| - 1, \quad \begin{matrix} \forall i \in \mathrm{D}_\ell, \\ \ell \in \mathrm{L}, \end{matrix} \tag{7.66}$$

with $\Delta = |\mathrm{E}(p_i^{\mathrm{D}})| + |\mathrm{E}(p_i^{\mathrm{F}})| + \sum_{j\in \mathrm{L}, j<\ell} |\mathrm{E}(p_{ij}^{\mathrm{I}})|$ the number of edges contained in the distribution, feeder and intermediate paths. Conceptually then, (7.66) ensures that for a violated combination of paths and a given set of splitters, at least one edge or splitter

must change. Additionally, changing a splitter type will also satisfy the violated cut, i.e. switching to a type which will lead to a feasible connection. It is evident that if $i$ is a level 0 splitter, i.e. there are no intermediate paths in its upstream link, (7.66) simplifies to the standard multiple splitter type canonical path length cut given in (4.55).

#### Differential path length cuts

For the differential path length cuts, we keep track of the paths with the minimum and maximum lengths for each splitter. Let $p_i^{\mathrm{D,max}}$ and $p_{i\ell}^{\mathrm{I,max}}$ be the maximum length paths in the distribution and intermediate networks respectively. Similarly, let $p_i^{\mathrm{D,min}}$ and $p_{i\ell}^{\mathrm{I,min}}$ be the corresponding minimum paths. If the difference between the maximum and minimum distance between an ONU and the CO exceeds the differential distance limit $\ell_{\mathrm{diff}}^{\max}$, add the following canonical cut:

$$\sum_{e\in \mathrm{E}(p_i^{\mathrm{D,max}})} x_e^{\mathrm{TD}} + \sum_{j\in \mathrm{L}, j<\ell} \sum_{e\in \mathrm{E}(p_{ij}^{\mathrm{I,max}})} x_{ej}^{\mathrm{TI}} - \sum_{\substack{e\in \mathrm{E}\setminus \mathrm{E}(p_i^{\mathrm{D,min}}):\\ x_e^{\mathrm{TD}*}=0}} x_e^{\mathrm{TD}} - \sum_{j\in \mathrm{L}, j<\ell} \sum_{\substack{e\in \mathrm{E}\setminus \mathrm{E}(p_{ij}^{\mathrm{I,min}}):\\ x_{ej}^{\mathrm{TI}*}=0}} x_{ej}^{\mathrm{TI}} +$$

$$\psi_i \le |\mathrm{E}(p_i^{\max})| + \sum_{j\in \mathrm{L}, j<\ell} |\mathrm{E}(p_{ij}^{\mathrm{I,max}})|, \quad \forall i \in \mathrm{D}_\ell, \ell \in \mathrm{L}. \tag{7.67}$$

For this cut, if the differential length limit is exceeded, either $\boldsymbol{p}^{\mathrm{D,max}}$ or $\boldsymbol{p}^{\mathrm{I,max}}$ need to lose one or more edges, reducing the maximum path length, or $\boldsymbol{p}^{\mathrm{D,min}}$ or $\boldsymbol{p}^{\mathrm{I,min}}$ need to gain (or swap out) one or more edges, increasing the minimum path length. Only these options may clear the violation, reducing the differential length.

### 7.4.5 Strengthening cuts modification

The incorporation of intermediate networks requires additional strengthening cuts to be derived. Firstly however, a simple valid inequality is included to help the master problem quickly build a hierarchical network. It simply states that all used splitters in level $\ell \neq 0$ must be served by splitters in the level above it, $\ell - 1$:

$$\sum_{i\in \mathrm{D}_{\ell-1}} \Psi_i^{\mathrm{I}} = \sum_{i\in \mathrm{D}_\ell} \psi_i, \quad \forall \ell \in \mathrm{L}, \ell \neq 0. \tag{7.68}$$

**Connectivity cuts**

In terms of connectivity cuts, the distribution and global versions are used as-is. The feeder connectivity cut must now only ensure connectivity between the CO and splitters of level 0. Therefore, $\text{CON}_\text{F}$ is replaced by:

$$\begin{aligned}\text{CON}_\text{FL}: \quad & \sum_{\text{W}\cap\{q\}} 1 + \sum_{e\in\delta(\text{W})} x_e^{\text{TF}} \geq \psi_i, \quad \forall\, \text{W} \subset \text{V}, \forall i \in \text{W}\cap\text{D}_0 \neq \varnothing, \\ & \therefore \sum_{e\in\delta(\text{W})} x_e^{\text{TF}} \geq \psi_i, \quad \forall\, \text{W} \subset \text{V}\setminus\{q\}, \forall i \in \text{W}\cap\text{D}_0 \neq \varnothing. \end{aligned} \tag{7.69}$$

which are separated in the same way as $\text{CON}_\text{FL}$, as detailed in section 5.2.1, but all sets D are replaced with $\text{D}_0$.

Furthermore, we introduce new connectivity cuts for the intermediate networks, which ensures either the connected path to a splitter from another splitter in a lower level originates within the subset W, or it must come from outside the subset:

$$\text{CON}_\text{I}: \quad \sum_{j\in\text{W}\cap\text{D}_{\ell-1}} \psi_j + \sum_{e\in\delta(\text{W})} x_{e(\ell-1)}^{\text{TI}} \geq \psi_i, \qquad \begin{matrix}\forall\, \text{W} \subset \text{V}, \forall i \in \text{W}\cap\text{D}_\ell \neq \varnothing, \\ \forall \ell \in \text{L}, \ell \neq 0.\end{matrix} \tag{7.70}$$

The separation of $\text{CON}_\text{I}$ consists of the same steps as for $\text{CON}_\text{D}$ but, in this case, we calculate the max-flow between a supersource of all level $\ell-1$ splitters and each level $\ell$ splitter individually. The supersource auxiliary edges $(j, r)$ connected to splitters $j \in \text{D}_{\ell-1}$ have capacities $\psi_j^*$ and all other edge capacities are set to $x_{e(\ell-1)}^{\text{TI}*}$. If the maximum flow is less than 1, the projected violated min-cut is added to $PON_{ML}$.

**Flow-cutset inequalities**

Similarly, for the flow-cutset inequalities, the feeder version $\text{CUT}_\text{F}$ needs to be modified to only count the demand of level 0 splitters. Therefore, it can be rewritten as follows:

$$\text{CUT}_\text{FL}: \quad \sum_{\text{W}\cap\{q\}} x_{\beta(q)}^{\text{FF}} + \sum_{e\in\delta(\text{W})} x_e^{\text{FF}} \geq \sum_{i\in\text{W}\cap\text{D}_0} \psi_i, \quad \forall\, \text{W} \subset \text{V}, \text{W}\cap\text{D}_0 \neq \varnothing. \tag{7.71}$$

$\text{CUT}_\text{FL}$ is separated as detailed in section 5.3.1, and a projected violated min-cut is added whenever the max-flow is less than $d_{\text{net}} = \sum_{i\in\text{W}\cap\text{D}_0} \psi_i^* - \sum_{\text{W}\cap\{q\}} x_{\beta(q)}^{\text{FF}*}$.

Additionally, flow-cutset inequalities for the intermediate network are derived, ensuring that enough capacity exists in the level $\ell - 1$ network to satisfy the demand of level $\ell$ splitters, with $\ell > 0$:

$$\text{CUT}_\text{I}: \quad \sum_{i \in \text{W} \cap \text{D}_{\ell-1}} \Psi_i^\text{I} + \sum_{e \in \delta(\text{W})} x_{e(\ell-1)}^\text{FI} \geq \sum_{j \in \text{W} \cap \text{D}_\ell} \psi_i, \qquad \begin{matrix} \forall\, \text{W} \subset \text{V}, \text{W} \cap \text{D}_\ell \neq \varnothing, \\ \forall \ell \in \text{L}, \ell \neq 0. \end{matrix} \tag{7.72}$$

The intermediate flow-cutset inequalities are separated for each level $\ell \in \text{L}, \ell \neq 0$, using a similar process as the one detailed in section 5.3.1:

1. **Fractional values** - Variable values $x_{e\ell}^{\text{FD}*}$, $\psi_i^*$ and $\Psi_i^{\text{I}*}$ are found from the LP relaxation of $PON_{ML}$.
2. **Augment graph** - Create an auxiliary graph from $\mathcal{G}$ and add super source node $r$ connected to all level $\ell - 1$ splitters $i \in \text{D}_{\ell-1}$ via edges $(i, r)$ and super sink node $u$ connected to all level $\ell$ splitters $j \in \text{D}_\ell$ via edges $(j, u)$.
3. **Max-flow** - Calculate the maximum flow between the artificial nodes using $x_{e(\ell-1)}^{\text{FI}*}$ as capacity on edge $e$, $\Psi_i^{\text{I}*}$ as capacity on edge $(i, r)$ and $\psi_i^*$ as capacity on edge $(j, u)$.
4. **Min-cut** - Let $d_\text{net} = \sum_{j \in \text{W} \cap \text{D}_\ell} \psi_i^* - \sum_{i \in \text{W} \cap \text{D}_{\ell-1}} \Psi_i^{\text{I}*}$ be the total net demand. If the maximum flow is less than $d_\text{net}$, a violated cut can be derived by projecting the maximum flow to a minimum $s$-$t$ cut and adding (7.72) using the subset induced by the cut.

### 7.4.6 Primal heuristic modification

Generating integer feasible solutions for $PON_{ML}$ is done via a modified version of the *PRIMAL* heuristic presented in section 5.5. As with the other modifications, the inclusion of intermediate networks have the biggest impact. To adapt the heuristic for multi-level networks, we utilise the same idea, but more paths need to be taken into account, requiring a path enumeration procedure similar to that of the path length

cuts in section 7.4.4. Therefore, by saving the values of $\boldsymbol{f}^{\mathrm{D}}$, $\boldsymbol{f}^{\mathrm{I}}$ and $\boldsymbol{f}^{\mathrm{F}}$, we can generate solutions with the following heuristic, henceforth known as $PRIMAL_{ML}$:

1. **Path grouping** - For each used splitter $i \in \hat{\mathrm{D}} = \{i \in \mathrm{D} | \psi_i^* \geq \epsilon\}$, group all used distribution, feeder and intermediate paths (where $f_p^{\mathrm{D}*} \geq \epsilon$, $f_{p\ell}^{\mathrm{I}*} \geq \epsilon$ and $f_p^{\mathrm{F}*} \geq \epsilon$ respectively, $0 < \epsilon \ll 1$) connected to it.

2. **Path reduction** - In the case where a fractional solution to $PON_{ML}$ is used, multiple paths may exist between commodities. Therefore, remove all but the shortest path between each used commodity.

3. **Capacity check** - If the number of distribution and downstream intermediate paths connected to each splitter exceeds the used splitter type capacity, discard the solution.

4. **Auxiliary graph** - Construct an auxiliary DAG with vertices $\hat{\mathrm{D}} \cup \{q\}$. Arcs are added between vertices where a used path exists, orientated towards the CO or the lower level splitter.

5. **Path enumeration** - All simple paths are now enumerated between each splitter and the CO using depth-first search with backtracking. These combinations of intermediate and feeder paths are called upstream links.

6. **Path length check** - For every upstream link, calculate the effective split ratio by multiplying the split ratios of each splitter contained in the link. For each upstream link of each used splitter, determine or calculate its reach limit based on its effective split ratio or the sum of the splitter losses. Then, iterate through all combinations of distribution paths and upstream links for each splitter, keeping track of the minimum and maximum total fibre length between an ONU and the CO. The differential distance is calculated by subtracting the minimum fibre length from the maximum. If the maximum fibre length exceeds the path length limit calculated for that splitter, or if the differential distance exceeds the differential limit, discard the solution.

7. **Edge usage count** - Count the number of distribution, intermediate (for each level individually) and feeder paths that traverse each edge. These counts are the new solution values of $\boldsymbol{x}^{\mathrm{FD}}$, $\boldsymbol{x}^{\mathrm{FI}}$ and $\boldsymbol{x}^{\mathrm{FF}}$ respectively. If $x_e^{\mathrm{FD}} \geq 1, e \in \mathrm{E}$, set $x_e^{\mathrm{TD}}$ to 1, otherwise set it to 0. Similarly, if $x_{e\ell}^{\mathrm{FI}} \geq 1, e \in \mathrm{E}$, set $x_{e\ell}^{\mathrm{TI}}$ to 1, for every level $\ell \in \mathrm{L}$. $\boldsymbol{x}^{\mathrm{TF}}$ is set in the same way.

8. **Solution construction** - All other variable values are derived from the number of used edges, including setting $\psi_i$ to 1 if $x_{\beta(i)}^{\mathrm{FD}} \geq 1$. Additionally, $\Psi_i^{\mathrm{D}}$ and $\Psi_i^{\mathrm{I}}$ are set according to equations (7.35) and (7.38) respectively. The value of $\Psi_i$ is set according to (7.39) and (7.40).

### 7.4.7 Routing feasibility checking

The ILP used to check routing feasibility, *FEAS*, introduced in section 4.4.3, needs some modifications to be able to determine feasibility for the multi-level formulation. Most importantly, it needs to account for the additional intermediate networks that are now present. To this end, introduce arc-flow variables $\boldsymbol{f}^{\mathrm{I}} \in \mathbb{Z}_+^{|\mathrm{A}|\times|\mathrm{L}|}$ for the intermediate network flow. With solution values $\boldsymbol{x}^{\mathrm{FD}*}$, $\boldsymbol{x}^{\mathrm{FI}*}$ and $\boldsymbol{x}^{\mathrm{FF}*}$ from $PON_{ML}$, we get the multi-level feasibility checking model, henceforth known as $FEAS_{ML}$:

$FEAS_{ML}$:

$$\min \quad \mathbf{0} \tag{7.73}$$

$$\text{s.t.} \quad \sum_{j\in\phi(i)} f_{ij}^{\mathrm{D}} - \sum_{j\in\phi(i)} f_{ji}^{\mathrm{D}} = \begin{cases} -x_{\beta(i)}^{\mathrm{FD}*}, & i \in \mathrm{U}, \\ x_{\beta(i)}^{\mathrm{FD}*}, & i \in \mathrm{D}, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathrm{V}, \tag{7.74}$$

$$\sum_{j\in\phi(i)} f_{ij\ell}^{\mathrm{I}} - \sum_{j\in\phi(i)} f_{ji\ell}^{\mathrm{I}} = \begin{cases} -x_{\beta(i)(\ell)}^{\mathrm{FI}*}, & i \in \mathrm{D}_{\ell+1}, \\ x_{\beta(i)(\ell)}^{\mathrm{FI}*}, & i \in \mathrm{D}_{\ell}, \\ 0, & \text{otherwise}, \end{cases} \quad \begin{array}{c} \forall\, i \in \mathrm{V}, \forall \ell \in \mathrm{L}, \\ \ell \neq M-1, \end{array} \tag{7.75}$$

$$\sum_{j\in\phi(i)} f_{ij}^{\mathrm{F}} - \sum_{j\in\phi(i)} f_{ji}^{\mathrm{F}} = \begin{cases} -x_{\beta(i)}^{\mathrm{FF}*}, & i \in \mathrm{D}_0, \\ x_{\beta(i)}^{\mathrm{FF}*}, & i = q, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall i \in \mathrm{V}, \tag{7.76}$$

$$f_{ij}^{\mathrm{D}} + f_{ji}^{\mathrm{D}} = x_e^{\mathrm{FD}*}, \quad \forall (ij) = e \in \mathrm{E}, \tag{7.77}$$

$$f_{ij\ell}^{\mathrm{I}} + f_{ji\ell}^{\mathrm{I}} = x_{e\ell}^{\mathrm{FI}*}, \quad \forall (ij) = e \in \mathrm{E}, \forall \ell \in \mathrm{L}, \tag{7.78}$$

$$f_{ij}^{\mathrm{F}} + f_{ji}^{\mathrm{F}} = x_e^{\mathrm{FF}*}, \quad \forall (ij) = e \in \mathrm{E}. \tag{7.79}$$

Recall from section 4.4.3 that we use the null objective (7.73) to find a feasible solution, without any information regarding optimality. Constraints (7.74)–(7.76) implement flow conservation for the distribution, intermediate and feeder networks respectively, while (7.77)–(7.79) sets the edge capacity. If there exists a feasible solution to $FEAS_{ML}$, the $PON_{ML}$ solution is feasible in terms of connectivity. If the model is infeasible, we relax the capacity constraints and re-solve the model. Therefore, replace equations (7.77)–(7.79) with the following equations respectively:

$$f_{ij}^{\mathrm{D}} + f_{ji}^{\mathrm{D}} \leq x_e^{\mathrm{FD}*}, \qquad \forall (ij) = e \in \mathrm{E}, \tag{7.80}$$

$$f_{ij\ell}^{\mathrm{I}} + f_{ji\ell}^{\mathrm{I}} \leq x_{e\ell}^{\mathrm{FI}*}, \qquad \forall (ij) = e \in \mathrm{E}, \forall \ell \in \mathrm{L}, \tag{7.81}$$

$$f_{ij}^{\mathrm{F}} + f_{ji}^{\mathrm{F}} \leq x_e^{\mathrm{FF}*}, \qquad \forall (ij) = e \in \mathrm{E}. \tag{7.82}$$

As stated before in section 4.4.3, if the relaxed model is feasible, it means that the solution is feasible in terms of connectivity, but sub-optimal due to excess reserved capacity on one or more edges.

## 7.5 Computational study

The computational study of this chapter aims to illustrate the compromise between representational accuracy and computational feasibility when introducing additional modelling complexity. Therefore, the focus is split between numerical and computational performance. As all the refinements in this chapter add complexity, only data sets from the baseline and scalability sets that could be solved in chapter 5 in less than

Table 7.1: Splitter design parameters

| Type $[n_t]$ | Attenuation (dB) $[A_t^{SP}]$ | Cost $[c_t^{SP}]$ |
|---|---|---|
| 1:2 | 3.31 | R 500 |
| 1:4 | 6.62 | R 1,000 |
| 1:8 | 9.93 | R 1,500 |
| 1:16 | 13.25 | R 2,000 |
| 1:32 | 16.56 | R 2,500 |
| 1:64 | 19.87 | R 3,000 |
| 1:128 | 23.18 | R 5,000 |

an hour will be used. This ensures that solution quality is high in order to effectively compare numerical results. Additionally, all tests include strengthening cuts as well as the *PRIMAL* heuristic. All cuts are separated in a nested fashion, and in doing so, the computational results of this section can be compared to the right-most column of table 5.3.

### 7.5.1 Splitter types

#### Experimental methodology

Recall that we only used the maximum network reach in chapter 4 to model attenuation. Since only a single splitter type was involved, splitter attenuation effects could be separated from the overall attenuation since all paths will have the same splitter attenuation. Effectively then, the attenuation only depended on the path length itself. When multiple splitter types are included however, the separation of path length cuts becomes much more involved, as path length limits are dependent on the connected splitter.

Therefore, along with the multiple splitter types, the path length cuts are tested in more detail in this section, using power budgets of 21 dB and 19 dB. These are typical of class A and B GPONs, and since the tested networks are small, attenuation effects are low on average. A number of splitter types are used, from 1:2 to 1:128, as shown in table 7.1,

Table 7.2: Model design parameters for multiple splitter type formulation

| Parameter | Symbol | Value |
|---|---|---|
| Fixed OLT setup cost | $c^{\text{CO}}$ | R 10,000 |
| ONU unit cost | $c^{\text{ONU}}$ | R 500 |
| Average trenching cost | $c^{\text{TR}}$ | R 300/m |
| Average fiber cost | $c^{\text{F}}$ | R 50/m |
| Maximum differential distance | $\ell_{\max}^{\text{diff}}$ | $20 \times 10^3$ m |
| Total connector loss ($\times 4$) | $a_{\text{c}}$ | 2 dB |
| Total splice loss ($\times 4$) | $a_\ell$ | 0.4 dB |
| Fibre attenuation | $a_{\text{f}}$ | 0.4 dB/km |
| Power budget | $\text{P}_{\text{b}}$ | 21 dB, 19 dB |

along with the parameters given in table 7.2. Splitter attenuation is calculated using the approximation $a_t^{\text{s}} = 11 \log_{10} n_t$ and the path length is calculated using equation (4.61), which will be restated for convenience for each splitter type $t \in \text{T}$:

$$\ell_t^{\max} = \frac{\text{P}_{\text{b}} - a_{\text{c}} - a_\ell - a_t^{\text{s}}}{a_{\text{f}}} \times 1000. \tag{7.83}$$

If, for any splitter type $t \in \text{T}$, $\ell_t^{\max}$ is negative, the splitter type in question is removed in preprocessing, as it will never be used in a feasible solution.

In addition to showing the objective value, optimality gap and time to solve, we also include the maximum PON attenuation, $A_{\text{PON}}^{\max}$, and the number of path length cuts added, $N_{\text{path}}$. For this experiment, the $BENDERS_T$ formulation is used along with the cuts introduced in section 7.2.

**Results and analysis**

From the results, shown in table 7.3, it is immediately evident that including multiple splitter types and using the adapted path length cuts increases the computational effort required to solve the instances to optimality. However, even though thousands of path length cuts are required, average computation time increases of only 62 % and 279 % for the 21 dB and 19 dB runs are observed respectively. This demonstrates the relatively minor increase in complexity with multiple splitter types, as well as the efficacy of the modified path length cuts, even though they are derived from very weak canonical

cuts. Since the 19 dB power budget constraint is just short of the maximum attenuation of a number of instances in the 21 dB run, more path length cuts are required, resulting in increased computation times. Stated differently, a complete network topology change, which includes using smaller splitters and reallocating ONUs, is required when enforcing a 19 dB limit on a network with for instance a 19.76 dB attenuation, as is the case for `rural3r`. This change is enforced only through the separation of path length cuts, which for large networks can be computationally intensive.

Table 7.3: Computational results - multiple splitter types

| | *BENDERS*$_T$ including (5.2)–(5.6), nested cuts and *PRIMAL* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $P_b = 21$ dB | | | | | $P_b = 19$ dB | | | | |
| Data set | Obj [R 1k] | gap [%] | $t_s$ [sec] | $A^{max}_{PON}$ [dB] | $N_{path}$ | Obj [R 1k] | gap [%] | $t_s$ [sec] | $A^{max}_{PON}$ [dB] | $N_{path}$ |
| *Section 1: Baseline data sets* | | | | | | | | | | |
| `urban2r` | 953 | – | 83.2 | 19.04 | 0 | 962 | – | 215 | 15.74 | 10,680 |
| `urban3r` | 612 | – | 50.0 | 19.09 | 0 | 615 | – | 35.2 | 15.81 | 15,148 |
| `suburb1r` | 729 | – | 5.93 | 15.75 | 0 | 729 | – | 5.96 | 15.75 | 886 |
| `suburb2r` | 1,234 | – | 37.9 | 15.85 | 0 | 1,234 | – | 26.8 | 15.85 | 3,068 |
| `suburb3r` | 2,340 | – | 46.7 | 19.21 | 0 | 2,748 | – | 103 | 15.97 | 11,103 |
| `rural3r` | 11,704 | – | 826 | 19.76 | 0 | 12,013 | – | 5,957 | 16.45 | 693,050 |
| *Section 2: Scalability data sets* | | | | | | | | | | |
| `micronet` | 146 | – | 0.09 | 15.70 | 0 | 146 | – | 0.07 | 15.70 | 0 |
| `citynet1` | 209 | – | 0.26 | 15.72 | 0 | 209 | – | 0.20 | 15.72 | 0 |
| `citynet2` | 813 | – | 3.87 | 19.04 | 0 | 871 | – | 3.80 | 15.74 | 3,304 |
| `citynet3` | 782 | – | 114 | 19.05 | 0 | 783 | – | 86.4 | 15.77 | 7,530 |
| `citynet4` | 1,632 | – | 227 | 19.08 | 0 | 1,878 | – | 1,046 | 15.78 | 6,474 |
| `mednet1` | 278 | – | 0.15 | 15.76 | 0 | 278 | – | 0.08 | 15.76 | 0 |
| `mednet2` | 1,230 | – | 5.26 | 19.14 | 0 | 1,360 | – | 25.4 | 15.88 | 3,029 |
| `mednet3` | 1,230 | – | 81.2 | 19.14 | 274 | 1,360 | – | 899 | 15.88 | 10,375 |
| `hugenet1` | 212 | – | 0.48 | 15.72 | 0 | 212 | – | 0.39 | 15.72 | 0 |
| `hugenet2` | 978 | – | 3.02 | 19.07 | 0 | 1,045 | – | 3.90 | 15.76 | 1,142 |
| `hugenet3` | 978 | – | 36.1 | 19.07 | 0 | 1,045 | – | 184 | 15.76 | 5,170 |
| `hugenet4` | 1,767 | 1.28 | >2 h | 19.17 | 0 | 1,847 | – | 3,788 | 15.88 | 5,767 |
| `subnet1` | 141 | – | 0.12 | 15.70 | 0 | 141 | – | 0.10 | 15.70 | 0 |
| `subnet2` | 867 | – | 9.07 | 19.06 | 0 | 926 | – | 10.1 | 15.77 | 795 |
| `subnet3` | 867 | – | 164 | 19.06 | 0 | 926 | – | 1,089 | 15.77 | 11,688 |
| `subnet4` | 2,011 | – | 1,496 | 19.10 | 0 | 2,258 | 0.69 | >2 h | 16.03 | 54,243 |
| `subnet5` | 2,011 | – | 1,646 | 19.10 | 720 | 2,254 | 9.60 | >2 h | 16.01 | 36,659 |

In terms of numerical results, including multiple splitter types results in a total deployment cost increase of 0.16 % and 5.24 % for a 21 dB and 19 dB power budget respectively. Additionally, a run using the same 24 dB power budget as specified in

chapter 4 showed a 0.37 % decrease in deployment cost, which would be attributed mainly to reductions in splitter cost. This illustrates that lower power budgets yield more expensive networks, as smaller splitters have to be utilised, decreasing the total length of feeder fibre shared amongst ONUs. However, lower power budgets are usually accompanied by lower OLT costs, which could result in a less expensive network overall. This is not the case for the tested instances using the given parameters though, as a 19 dB network results in an average absolute deployment cost increase of R 95,443, which is higher than an OLT cost reduction can offset.

Looking at the maximum PON attenuation, we can see most instances have values around either 19.1 dB or 15.7 dB. Since splitter attenuation is the dominant variable factor in the total attenuation and the networks are small, resulting in low fibre loss, this is to be expected. The `rural3r` instance is the exception however, with maximum fibre lengths of over 2 km, in contrast to the average of 421 m.

### 7.5.2 Survivability

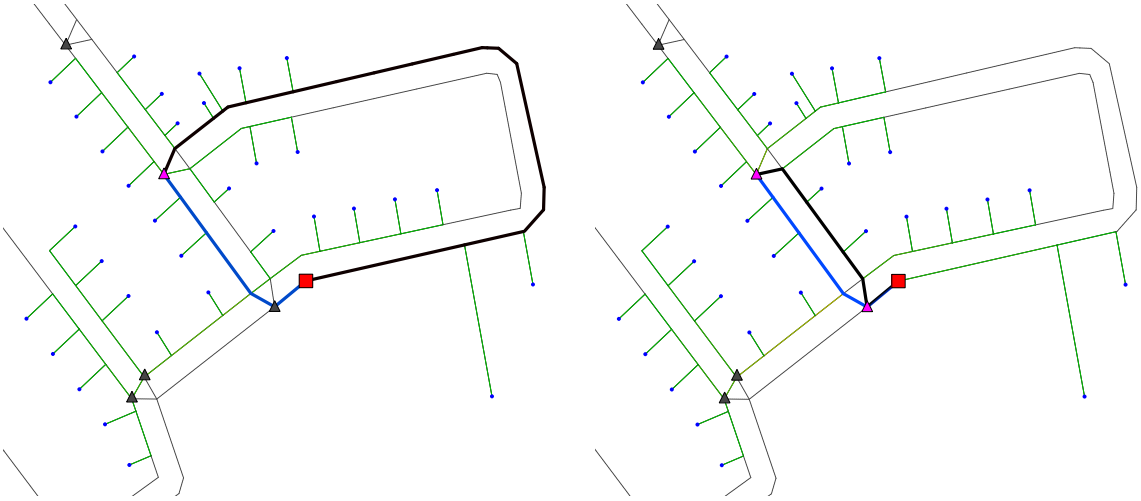
#### Experimental methodology

Next, we will consider the survivable PONDP, using the $BENDERS^{surv}_{full}$, $BENDERS^{surv}_{\lambda\text{-}edge}$ and $BENDERS^{surv}_{\lambda\text{-}length}$ formulations. In each case, we assume dual redundant feeder fibres are required, i.e. $N = 2$. This is the most common case, with $N > 2$ only applying to absolute mission critical applications or where multiple simultaneous fibre breakages are expected. In the case of $\lambda$-disjoint survivability, we assume sharing 2 edges or 50 m worth of feeder fibre is acceptable.

Where not specified, the experimental methodology described in section 4.4 is followed for this experiment, including relevant parameters and data set reduction techniques. Additionally, a single splitter type is again assumed for this experiment to isolate only the influences relevant to the survivable formulation.

**Results and analysis**

To illustrate the difference between full edge-disjoint and $\lambda$-disjoint survivability, we consider the `suburb1r` instance. Optimal solutions for both types are shown in figure 7.5, with the feeder paths highlighted. In the full edge-disjoint case, paths may not share any edges, resulting in the second path taking a long detour to the used splitter. When relaxing the full edge-disjoint constraints, we find that by sharing only a single edge, the redundant path is significantly shortened, resulting in a 16.4 % reduction in total deployment cost. Furthermore, a second splitter can now be used since its feeder path cost can now be offset by the distribution fibre cost savings.



(a) Full edge-disjoint (b) $\lambda$-edge disjoint, with $\lambda = 2$

Figure 7.5: Optimal edge-disjoint solutions for the `suburb1r` data set, showing redundant feeder paths for a splitter

The rest of the computational and numerical results are contained in table 7.4, with all survivable formulations producing a significant increase in deployment cost as expected. The full edge-disjoint variant is on average 24.9 % more expensive than the non-survivable PON, with the semi-redundant PONs increasing deployment cost by 9.47 % and 11.9 % for the $\lambda$-edge and $\lambda$-length disjoint cases respectively. Results also show that it can be up to 96.3 % more expensive to provide a fully redundant network, up to three times as much as a $\lambda$-edge disjoint network which shares only a few edges.

Table 7.4: Computational results - full edge-disjoint and $\lambda$-edge disjoint survivability

| | *BENDERS*$_C$ including (5.2)–(5.6), nested cuts and *PRIMAL* | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Full edge-disjoint | | | $\lambda$-edge disjoint ($\lambda = 2$) | | | | $\lambda$-length disjoint ($\lambda = 50$ m) | | | |
| Data set | Obj [R 1k] | gap [%] | $t_s$ [sec] | Obj [R 1k] | gap [%] | $t_s$ [sec] | $N_\lambda$ | Obj [R 1k] | gap [%] | $t_s$ [sec] | $N_\lambda$ |
| *Section 1: Baseline data sets* | | | | | | | | | | | |
| urban2r | 1,070 | – | 277 | 1,057 | 5.81 | >2 h | 24,417 | 1,057 | 3.39 | >2 h | 80,815 |
| urban3r | 745 | – | 15.5 | 730 | – | 2,535 | 2,847 | 731 | – | 1,214 | 3,013 |
| suburb1r | 943 | – | 11.9 | 788 | – | 164 | 4,836 | 788 | – | 58.8 | 1,828 |
| suburb2r | 1,431 | – | 35.7 | 1,431 | – | 2,309 | 22,234 | 1,431 | – | 2,017 | 26,316 |
| suburb3r | 2,912 | 4.70 | >2 h | 2,868 | 18.7 | >2 h | 1,011 | 2,677 | 10.62 | >2 h | 825 |
| rural3r | Infeasible | | | >2 h | | | | >2 h | | | |
| *Section 2: Scalability data sets* | | | | | | | | | | | |
| micronet | 166 | – | 0.27 | 151 | – | 0.08 | 0 | 151 | – | 0.07 | 0 |
| citynet1 | 270 | – | 2.77 | 220 | – | 0.69 | 2 | 216 | – | 0.16 | 0 |
| citynet2 | 919 | – | 15.0 | 851 | – | 8.43 | 93 | 866 | – | 15.8 | 258 |
| citynet3 | 919 | 2.88 | >2 h | 851 | – | 5,551 | 31,893 | 866 | 2.51 | >2 h | 6,362 |
| citynet4 | 1,910 | 0.52 | >2 h | 1,832 | 10.7 | >2 h | 55,302 | 1,858 | 12.5 | >2 h | 22,768 |
| mednet1 | 477 | – | 2.39 | 364 | – | 0.97 | 17 | 477 | – | 18.7 | 578 |
| mednet2 | 1,566 | – | 10.2 | 1,293 | – | 19.9 | 321 | 1,380 | – | 99.4 | 872 |
| mednet3 | 1,491 | 11.8 | >2 h | 1,293 | – | 1,031 | 1,646 | 1,380 | 7.23 | >2 h | 3,270 |
| hugenet1 | 248 | – | 0.93 | 217 | – | 0.37 | 0 | 217 | – | 0.37 | 0 |
| hugenet2 | 1053 | – | 7.00 | 1,000 | – | 4.89 | 135 | 1,000 | – | 4.44 | 66 |
| hugenet3 | 1053 | – | 99.1 | 1,000 | – | 270 | 199 | 1,000 | – | 260 | 390 |
| hugenet4 | 2082 | 1.05 | >2 h | 1,949 | 8.37 | >2 h | 32,804 | 1,978 | 9.53 | >2 h | 32,673 |
| subnet1 | 278 | – | 0.85 | 146 | – | 0.09 | 0 | 146 | – | 0.08 | 0 |
| subnet2 | 1041 | – | 16.8 | 954 | 0.57 | 121 | 187,752 | 954 | – | 3,106 | 219,557 |
| subnet3 | 1041 | – | 717 | 954 | 5.77 | 2,119 | 20,438 | 954 | 6.90 | >2 h | 5,227 |
| subnet4 | 2483 | – | 4,428 | 2,142 | 6.82 | 2,477 | 11,776 | 2,147 | 4.90 | >2 h | 17,421 |
| subnet5 | >24 GiB | | | >2 h | | | | >2 h | | | |

Note that in the full edge-disjoint case, rural3r proved to be infeasible, suggesting two fully redundant feeder paths could not be found for one or more of the used splitters. If enough edges exist in the core of the graph, close to the CO, this type of infeasibility may be fixed in a real-world scenario by including more boundary edges so that longer detour paths may be considered.

Comparing the two $\lambda$-disjoint formulations, we see that for citynet1, the $\lambda$-edge disjoint case yielded a more expensive solution, while beating the $\lambda$-length disjoint formulation in most other instances. This is easily explained by seeing how the average

length of 2 edges compares to the 50 m length specified for each individual graph, with the longer one yielding potentially less expensive solutions since more optical fibre can be shared between the redundant layers.

The computational results show increases in computation time across the board, with an average increase of *at least* a factor of 30 for the full edge-disjoint case and a factor of 100 for the $\lambda$-disjoint cases. Since a number of the solutions are sub-optimal, only a lower bound can be determined for the computational effort increase, but since solution quality is generally high, it should not be grossly conservative. Regardless, designing a survivable network clearly increases computational effort quite substantially.

### 7.5.3 Multi-level networks

#### Experimental methodology

To test the multi-level formulation, $BENDERS_{ML}$ is used with up to 3 levels of splitters, i.e. $M = 3$. In addition, the multi-level formulation makes the most sense when incorporating splitter types as well, as a single large splitter can be replaced by two cascaded splitters with the same equivalent split ratio. Therefore, the splitter types shown in table 7.1 is used in this section, along with the splitter type aware path length cut formulation shown in (7.66). Both hetero- and homogeneous variants are considered.

To better illustrate the gains of a multi-level PON, the power budget is increased to 29 dB, which is typical for a class C GPON network. This gives the model more freedom to use cascaded splitters, even though the resulting total splitter attenuation may be slightly higher. Stated differently, a higher power budget allows us to focus on the cost benefits of the formulation. This would only be required for small data sets, as larger data sets covering clustered regions will benefit more from distributed splitting, even when working with more constrained power budgets. Unfortunately, solving larger instances may increase the computational effort beyond reasonable limits. To

ensure a fair comparison, $BENDERS_{ML}$ is compared to a single-level formulation using the same 29 dB power budget.

Finally, the maximum number of levels utilised, $M_{\max}$, is recorded for each instance, along with the objective value, gap and computation time.

**Results and analysis**

Both numerical and computational results are shown in table 7.5. It is evident that computational effort increases drastically, even when considering only a three-level network, with more than half of the instances showing sub-optimal solutions and the rest showing computation times up to two orders of magnitude higher. Since we saw in chapter 4 that computational effort scales strongly with the number of commodities, this is not unexpected, as a multi-level network effectively has $M \times |\mathrm{D}|$ splitters. Additionally, the large number of potential inter-splitter connections increase the total number of commodities to $O(M|\mathrm{U}||\mathrm{D}| + M^2|\mathrm{D}|^2)$, in contrast to just $O(|\mathrm{U}||\mathrm{D}|)$ for the single level formulation, which increases modelling complexity as well.

The difference in computational performance between the hetero- and homogeneous variants were slight, with some selected instances solved faster using the latter. This is likely due to the fact that an extra splitter is required to make the network homogeneous in cases where both splitters and ONUs connect to a single splitter in the heterogeneous case. Therefore, the multi-level solution is cut off early due to its cost. On average however, the homogeneous variant produces 67 % higher optimality gaps under the same time constraints.

In terms of numerical results, a maximum of two levels were deployed in the heterogeneous multi-level solutions, showing that the instances are potentially too small for higher level networks. In fact, in 9 of the 23 instances the best solution is a single level network. For those instances where a heterogeneous multi-level solution proved superior to a single level, objective values dropped by at least 0.93 %, although when taking the solution quality into account, it could be up to 5.65 %.

Table 7.5: Computational results - multi-level networks

| | *BENDERS*$_C$ including (5.2)–(5.6), nested cuts and *PRIMAL* | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Single level ($M = 1$) | | | Multi-level ($M = 3$) heterogeneous | | | | Multi-level ($M = 3$) homogeneous | | | |
| Data set | Obj [R 1k] | gap [%] | $t_s$ [sec] | Obj [R 1k] | gap [%] | $t_s$ [sec] | $M_{\max}$ | Obj [R 1k] | gap [%] | $t_s$ [sec] | $M_{\max}$ |
| *Section 1: Baseline data sets* | | | | | | | | | | | |
| urban2r | 953 | – | 73.0 | 933 | 6.61 | >2 h | 2 | 949 | 8.47 | >2 h | 2 |
| urban3r | 612 | – | 154 | 585 | 8.04 | >2 h | 2 | 582 | 7.93 | >2 h | 3 |
| suburb1r | 729 | – | 4.90 | 725 | – | 617 | 2 | 729 | – | 614 | 1 |
| suburb2r | 1,234 | – | 31.4 | 1,204 | 5.72 | >2 h | 2 | 1,211 | 3.99 | >2 h | 2 |
| suburb3r | 2,236 | – | 69.3 | 2,266 | 7.45 | >2 h | 2 | 3,240 | 35.9 | >2 h | 3 |
| rural3r | 11,701 | – | 5,090 | >2 h | | | | 12,126 | 10.6 | >2 h | 3 |
| *Section 2: Scalability data sets* | | | | | | | | | | | |
| micronet | 146 | – | 0.12 | 146 | – | 0.43 | 1 | 146 | – | 1.09 | 1 |
| citynet1 | 209 | – | 0.25 | 209 | – | 3.44 | 1 | 209 | – | 4.99 | 1 |
| citynet2 | 813 | – | 3.58 | 807 | – | 1,533 | 2 | 813 | – | 75.9 | 1 |
| citynet3 | 782 | – | 102 | 765 | 5.38 | >2 h | 2 | 863 | 16.5 | >2 h | 3 |
| citynet4 | 1,619 | – | 223 | 1614 | 6.70 | >2 h | 2 | 1,624 | 7.68 | >2 h | 2 |
| mednet1 | 278 | – | 0.09 | 278 | – | 0.49 | 1 | 278 | – | 0.27 | 1 |
| mednet2 | 1,230 | – | 4.45 | 1,230 | – | 242 | 1 | 1,230 | – | 62.9 | 1 |
| mednet3 | 1,230 | – | 115 | 1,230 | 6.60 | >2 h | 1 | 1,230 | 8.45 | >2 h | 1 |
| hugenet1 | 212 | – | 0.60 | 212 | – | 2.28 | 1 | 212 | – | 4.34 | 1 |
| hugenet2 | 978 | – | 3.05 | 976 | – | 274 | 2 | 978 | – | 46.9 | 1 |
| hugenet3 | 978 | – | 35.6 | 976 | 0.73 | >2 h | 2 | 978 | 1.90 | >2 h | 1 |
| hugenet4 | 1,758 | – | 5,879 | 1,771 | 14.1 | >2 h | 2 | 1,762 | 11.9 | >2 h | 1 |
| subnet1 | 141 | – | 0.10 | 141 | – | 0.28 | 1 | 141 | – | 0.28 | 1 |
| subnet2 | 867 | – | 10.0 | 867 | – | 229 | 1 | 867 | – | 110 | 1 |
| subnet3 | 867 | – | 154 | 867 | 5.96 | >2 h | 1 | 867 | 4.37 | >2 h | 1 |
| subnet4 | 2,001 | – | 772 | 1,998 | 7.35 | >2 h | 2 | 2,001 | 7.02 | >2 h | 1 |
| subnet5 | 2,001 | 1.05 | >2 h | >2 h | | | | >2 h | | | |

The homogeneous multi-level formulation deployed all available levels in 4 of the instances, illustrating that more splitters are required to ensure network homogeneity. The number of instances where a single-level network is the best-known solution also increases to 15. Deployment cost improved in only 3 instances, where a 2.42 % reduction is observed on average, while the lower bounds indicate that the average improvement could be up to 5.7 % when all instances are considered. It is expected that larger networks will lead to larger deployment cost reductions for multi-level networks, although as-is these networks would potentially require unreasonable computation times.

## 7.6 Conclusion

In this chapter, a number of physical extensions to PONDP contained in the G.984 standard were investigated and incorporated into the modelling framework. First, the capability to handle multiple splitter types was added, as correctly dimensioning splitters may lead to a small cost reduction. More importantly, attenuation effects can be limited by using splitters with lower split ratios, which is crucial in environments with tight power budget constraints.

Next, we looked at designing a PON with survivability in mind, using both fully redundant and semi-redundant formulations. In contrast to the expensive fully redundant network, a semi-redundant network can share a limited portion between redundant fibres, forming what is known as a shared risk group. Both $\lambda$ edge- and length-limited semi-redundant networks were formulated, ensuring high availability at only a third of the additional cost required to implement a fully survivable network. Computational results suggest that designing a survivable network increases computation time by at least a factor of 20 while deployment cost increases by 10–25 %.

Finally, distributed splitting was utilised to design a multi-level network, using additional sections known as intermediate networks in-between the distribution and feeder segments of a PON. This required modifications to the $BENDERS_C$ formulation, the strengthening cuts, heuristics and feasibility checks. The resulting hetero- and homogeneous formulations required substantial effort to solve, with computation times increasing by up to two orders of magnitude, for a deployment cost reduction of between 1 % and 6 %.

# Chapter 8

# Conclusions and recommendations

*This chapter concludes the thesis, summarising the research conducted and how it ties in with the research goal, as well as revisiting the contributions made. Finally, recommended future work, with the research conducted as foundation, is discussed.*

## 8.1 Concluding summary

In the next few years, the number of PON deployments is expected to increase significantly, as service providers opt for access level optical fibre networks to address the exponential rise in consumer bandwidth requirements. To contribute to improving network designs, this thesis focussed on addressing two important aspects: modelling accuracy and computational tractability, as stated in the research goal in chapter 1.

To this end, an overview of the technical literature was provided in chapter 2 to familiarise the reader with the concepts relevant to the physical deployment of a PON. This included the physical construction of P2MP networks, the specifics of PONs and the relevant IEEE 802.3ah and ITU-T G.984 standards. Optional variants of the topology

in cases where redundancy is required were also introduced. As a companion, chapter 3 contributed to acquainting the reader with the mathematical modelling concepts, where techniques on formulating and solving models were presented. Furthermore, we showed the constituent problems and the resulting complexity of the Passive Optical Network Design Problem (PONDP). Related work on the subject was presented and analysed, illustrating gaps in the literature in terms of decomposing the problem for improved computational performance, integrating realistic attenuation effects and considering demand uncertainty.

Chapter 4 started with defining the common arc- and path-based models employed in literature and illustrated how attenuation effects, in the form of path length constraints, could be added. It was observed that in an environment with path length constraints, the number of variables required to formulate the problem as an arc-based model resulted in poor computational scalability. Path length constraints were easily added to the path-based model, but since the computation of all possible paths proved infeasible for large graphs, this too suffered from poor computational scalability. However, these formulations served as a baseline for the next step, where the path-based model was decomposed into a master problem and two independent sub-problems using Benders decomposition. This resulted in a formulation with a well-behaved structure and a limited number of variables, improving scalability. In addition, column generation was applied in the sub-problems to generate feasible routing paths on the fly, removing the need for generating all paths. Path length constraints were integrated into the model implicitly through the separation of cuts. Computational results suggested vastly improved scalability, providing solutions for 91 % of the tested instances, as opposed to only 56 % and 38 % for the arc- and path-based models respectively. However, it was observed that solution quality for large graphs was low, indicating either poor lower or upper bounds (or both).

Aiming to improve computational performance, chapter 5 presented techniques to strengthen both the lower and upper bounds of the Benders formulation. Lower bounds were improved by separating strengthening cuts in the form of connectivity cuts and flow-cutset inequalities. Connectivity cuts were derived from Steiner tree

cut inequalities, ensuring a connected set of trenches exist in the feeder and distribution networks. Additionally, global connectivity cuts were derived to ensure the CO is connected via trenches to each ONU. To strengthen the model by lifting the flow in each trench, flow-cutset inequalities were derived for the feeder and distribution networks, ensuring the number of optical fibres in each trench is sufficient to meet the ONU and splitter demands. A primal heuristic was proposed to improve the upper bound, utilising feasible routing paths from the sub-problems to construct a feasible solution. Implementation improvements concerning the separation of these cuts were considered, including separating them in a nested fashion and adding creep-flow to the capacity graph. Finally, computational results were provided for a number of different configurations, with the best configuration decreasing computational effort by 84 % and solving 33 % more instances to optimality. Where sub-optimal solutions were still produced, solution quality improved by 74 %.

Since computational performance was increased, modelling complexity could potentially be added while operating under the same resource and time constraints. To this end, and to address the modelling accuracy aspect of the research goal, chapter 6 illustrated how demand uncertainty could be integrated into the Benders formulation. Both a two-stage recourse formulation based on stochastic programming and an equivalent robust optimisation formulation were presented. Since the utilised techniques could not solve models with explicitly defined stochastic variables, the uncertainty in demand was discretised through the use of independent scenarios which could be leveraged to reduce dimensioning costs for both splitters and fibres. All strengthening procedures introduced in the previous chapter were adapted to account for the new distribution sub-problems, but the computational effort required in solving instances with even a modest number of scenarios proved to be substantial, easily eclipsing the improvement we saw in chapter 5 by an order of magnitude. Unfortunately, the robust formulation showed poor performance, which upon further investigation proved to be due to the resulting LP being particularly hard to solve using the simplex algorithm. In terms of numerical results, a deployment cost reduction of up to 4 % was observed in the larger instances. Finally, we demonstrated how the formulation could be modified

to optimise for operational aspects directly by presenting a revenue-based model that connects ONUs if the cost can be offset by the expected income generated.

Further modelling accuracy improvements related to the physical standards were presented in chapter 7, starting with dimensioning splitters according to the number of connected ONUs. Additionally, we introduced splitter type dependent path length cuts, since the maximum network reach, derived from the power budget and insertion losses, is highly dependent on the attenuation introduced by splitters with different split ratios. This ensured the model provided a feasible network design in the presence of variable splitter loss, which was confirmed by the computational results, with lower power budgets resulting in slightly increased deployment costs. Even though the dependent cuts are not as strong as the original cuts, computational effort increased by less than a factor of 3, resulting in reasonable computation times.

Next, both full edge-disjoint redundancy and $\lambda$ edge- and length-disjoint semi-redundancy were included in the network design, improving the overall survivability of the network. Semi-redundancy allows for a section of the redundant paths to be shared, reducing cost while limiting the vulnerability. Introducing full edge-disjoint optical fibres increased deployment cost by 25 % on average, in contrast to only 9–13 % for networks with small shared risk groups. Even with modified strengthening procedures, computational effort increased by at least a factor of 20 however, again surpassing the improvements of chapter 5.

Finally, we looked at distributed splitting through the formulation of a multi-level model with an arbitrary number of intermediate networks. As formulated, the network was heterogeneous, with splitters and ONUs connecting to the same splitter. In the homogeneous variant, we differentiated between splitters connected only to other splitters and those connected only to ONUs, enforcing this distinction across the network. All strengthening procedures and verification and validation approaches were adapted to work with the new formulation. Similar to the results seen in chapter 6, multi-level networks proved difficult to solve, with computational effort increasing by up to two orders of magnitude. Both hetero- and homogeneous variants produced

cost benefits of between 1 % and 6 %, with homogeneous networks being slightly more expensive to deploy due to the additional splitting levels required.

## 8.2 Research contributions made

Now that the research has been presented, we can revisit the main contributions listed in section 1.3, describing them in more accurate technical terms and referencing where the work was done.

- **Algorithmic contributions**

  1. A novel path-based PONDP model, decomposed using Benders into a master problem and two independent sub-problems, is proposed in chapter 4 to improve computational tractability.
  2. Strengthening procedures for the Benders formulation, including connectivity cuts, flow-cutset inequalities and a novel primal heuristic, are derived in chapter 5 to strengthen the upper and lower bounds during computation. All relevant modifications are done to ensure operation for each variant of the model, including the stochastic formulations of chapter 6, and the survivable and distributed splitting configurations of chapter 7.
  3. Results for each variant are presented in a set of computational studies in chapters 4 through 7, detailing the efficacy of the proposed modifications compared to the baseline Benders formulation as well as arc- and path-based formulations from literature.

- **Modelling contributions**

  1. The Benders formulation is extended in chapter 6 to incorporate uncertainty through the use of discretised scenarios containing potential demand realisations. The distribution sub-problem is reformulated to derive both a

two-stage recourse and a robust optimisation model in sections 6.2 and 6.3 respectively.

2. In section 6.4, a revenue-based model is formulated by redefining the master problem and making ONU demand variable, showcasing the flexibility of the modelling framework and how operational considerations can guide the network design by connecting ONUs when their costs offset the expected income they will generate.
3. Real-world attenuation effects are integrated into the presented modelling framework in the form of novel splitter type independent (section 4.3.4) and dependent (sections 7.2.1 and 7.4.4) path length constraints. The implicit handling of these constraints through violated cut separation also contributes to multi-hierarchy networks in general.
4. The modelling framework is extended in section 7.3 to incorporate full- and semi-redundancy for the optical fibres between the CO and the splitters. New semi-redundant variants limiting either the number of shared edges or the length of optical fibre shared between redundant layers ensure that the shared risk groups remain small.
5. A distributed splitting variant of the model, along with the corresponding attenuation considerations in the form of modified splitter type dependent path length cuts, is formulated as a multi-level network in section 7.4 to allow for an arbitrary number of cascaded splitters, reducing deployment cost.

## 8.3 Recommendations for future work

Unsurprisingly, given the overarching theme of the thesis, the possibilities for future work with this research as foundation can be divided into one of two categories: computational tractability and modelling accuracy.

### 8.3.1 Computational tractability

In this thesis, progress was made to dramatically improve the computational performance of the PONDP formulation. However, there are a few implementation details that were considered out of the scope of the research.

Firstly, a proper separation strategy should be devised to deal with the enormous number of cuts added. As mentioned in chapter 5, this could entail separating cuts only every $n$ nodes of the branch-and-cut tree, or only for quadratic node indices. Furthermore, it was noted that the majority of cuts added were connectivity cuts, whose separation, while greatly improving the lower bound, may exceed the benefit gained later on during computation. By keeping record of the ONUs for which cuts have been separated, we can reduce the number of distribution connectivity cuts while ensuring cut diversity by separating for a fixed subset of ONUs in a round-robin fashion (see [117] for a similar approach taken). The same can be done for global and feeder connectivity cuts by considering a subset of ONUs and splitters respectively.

In chapter 5, we saw that computational performance can be improved by removing columns if they have not entered the simplex basis in a number of successive iterations. While this led to an improvement, an instance independent strategy to determine when the variables should be removed is required to ensure predictable performance. Candidates include making the factor $n$ dependent on the number of variables in the basis or making $n$ dynamic according to current LP performance.

Another area which can be addressed to improve performance is that of providing warm-start solutions for the formulation prior to solving it. For the single-level formulation, a $k$-shortest path heuristic (see [90]) can be used to provide a very good upper bound at the onset, although any heuristic or meta-heuristic would suffice. This could lead to quick pruning of a number of branches in the branch-and-cut tree, reducing the number of required node visits and improving performance. Similarly, since the multi-level formulation is just a generalisation, an optimal or good quality sub-optimal solution for the single-level model can be used as a warm-start for the multi-level model.

Finally, we saw that both the two-stage recourse and robust models suffered from severe performance issues, suggesting more work in this area is required. In particular, the robust formulation performed poorly due to the resulting LP, so a reformulation of the uncertainty set may result in a model with improved performance.

### 8.3.2 Modelling accuracy

A number of additional practical PON design choices can be integrated into the formulations to improve the practical relevance of the model. Amongst others, these include the placement of optical fibre junction boxes, doing OLT and splitter port allocation and dimensioning hardware in the CO. Furthermore, the model can be adapted to distinguish between a number of fixed optical fibre cable types, e.g. 24-core, 48-core and 96-core, as well as allowing both subterranean and aerial fibre.

As the research in this thesis only focusses on greenfield deployment, where no existing infrastructure exists, a model capable of designing brownfield deployments could be useful in urban environments. In this case, ducts with empty capacity could already exist, wherein optical fibre can be installed for a reduction in cost. It may also be of practical relevance to consider leasing already installed *dark* or unlit fibre from another company instead of deploying all new infrastructure in segments of the network. This can be modelled similar to the brownfield case, but with modified costs to estimate leasing costs over the network lifetime.

Since deploying PON networks can take a number of years, a multi-period formulation of PONDP could be of great significance, providing service providers with a network design over time to maximise return on investment. However, since an additional subscript will be required for each flow variable to denote the deployment at a specific time, the model will likely be incredibly hard to solve. Even though it falls outside of the scope of the presented research, additional research into dealing with this type of deployment in an efficient way would be of great practical interest.

Finally, as mentioned in the contributions, the modelling framework can be utilised to

design other network types and topologies. This may include using the approach to design feeder networks where splitters are connected in a ring topology, potentially reducing optical fibre length while improving signal redundancy. Similarly, the model can easily be modified to design Active Optical Networks (AONs), where active splitters with dramatically lower insertion losses are used to increase the feasible network diameter.

## 8.4 Closure

With service providers under immense pressure to provide increasing bandwidth to consumers at lower cost, any improvement in designing future-proof networks with greater accuracy can benefit all stakeholders and end-users, providing reliable networks at competitive prices. Throughout this thesis, we illustrated how increasing computational performance allows us to incorporate complexity and, by extension, model accuracy, reducing overall deployment cost or increasing service availability. Additionally, accounting for uncertain service utilisation allows service providers to reduce expected expenditures, the savings of which can in turn be passed on to the consumer.

However, it is evident that at every step of the way, the design of automated planning systems for practical use is essentially a compromise between formulating the most accurate model and finding the highest quality solution in the available time, using the available resources.

# Bibliography


[1] INVOCOM. (2003) Attenuation of optical fibers. [Online]. Available: http://www.invocom.et.put.poznan.pl/~invocom/C/P1-9/swiatlowody_en/p1-1_2_2.htm

[2] ITU-T, *Gigabit-capable Passive Optical Networks (G-PON): General characteristics*, Mar 2003, ITU-T Rec. G.984.1.2003 (superceded). [Online]. Available: http://www.itu.int/rec/T-REC-G.984.1-200303-S/en

[3] R. Paschotta. (2015) Encyclopedia of laser physics and technology: Optical fiber communications. [Online]. Available: https://www.rp-photonics.com/optical_fiber_communications.html

[4] The Fiber Optic Association Inc. (2015) The FOA reference guide to fiber optics: Fiber optic network optical wavelength transmission bands. [Online]. Available: http://www.thefoa.org/tech/ref/basic/SMbands.html

[5] TeleGeography. (2015, April) Global network construction resurgence. [Online]. Available: https://www.telegeography.com/press/press-releases/2015/04/23/global-network-construction-resurgence/index.html

[6] ——. (2013, April) International bandwidth demand is decentralizing. [Online]. Available: http://www.telegeography.com/press/press-releases/2013/04/17/international-bandwidth-demand-is-decentralizing/index.html

[7] ——. (2013, October) Africa's international bandwidth demand to lead the world. [Online]. Avail-

able: http://www.telegeography.com/press/press-releases/2013/10/31/africas-international-bandwidth-demand-to-lead-the-world/index.html

[8] (2015, May) The Zettabyte Era – Trends and Analysis. Cisco. White paper. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.html

[9] F. Effenberger, D. Clearly, O. Haran, G. Kramer, R. D. Li, M. Oron, and T. Pfeiffer, "An introduction to PON technologies [Topics in Optical Communications]," *Communications Magazine, IEEE*, vol. 45, no. 3, pp. S17 –S25, March 2007.

[10] IEEE, *Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment: Media Access Control Parameters, Physical Layers, and Management Parameters for Subscriber Access Networks*, IEEE Std. 802.3ah, 2004.

[11] ITU-T, *Gigabit-capable passive optical networks (GPON): General characteristics*, Mar 2008, ITU-T Rec. G.984.1. [Online]. Available: http://www.itu.int/rec/T-REC-G.984.1/en

[12] ——, *Gigabit-capable passive optical networks (GPON): Physical Media Dependent (PMD) layer specification*, Mar 2008, ITU-T Rec. G.984.2. [Online]. Available: http://www.itu.int/rec/T-REC-G.984.2/en

[13] ——, *Gigabit-capable passive optical networks (GPON): Transmission Convergence Layer Specifications*, Mar 2008, ITU-T Rec. G.984.3. [Online]. Available: http://www.itu.int/rec/T-REC-G.984.3/en

[14] ——, *Gigabit-capable passive optical networks (GPON): ONT Management and Control Interface Specifications*, Mar 2008, ITU-T Rec. G.984.4. [Online]. Available: http://www.itu.int/rec/T-REC-G.984.4/en

[15] (2015, Feb) FTTH/B Global Panorama end 2014 – Total subscribers. FTTH Council Global Alliance. [Online]. Available: http://www.ftthcouncil.info/map.html

[16] (2015) FTTH Council Africa – Annual Report 2015. FTTH Council Africa. [Online]. Available: http://www.ftthcouncilafrica.com/attachments/article/173/FTTH-COUNCIL-AFRICA-Annual-Report-2015.pdf

[17] (2014, Nov) Africa's International Bandwidth Passes 2 Tbps. Hamilton Research. [Online]. Available: http://www.africabandwidthmaps.com/?p=4269

[18] B. Tubbs. (2015, April) Home fibre 'land grab' heralds price war. ITWeb. [Online]. Available: http://www.itweb.co.za/index.php?option=com_content&view=article&id=142307:Home-fibre-land-grab-heralds-price-war&catid=260

[19] (2015, Sep) South Africa: Telkom Announces FTTH Rollout To Reach 1 Million Homes By 2018. Hamilton Research. [Online]. Available: http://www.africabandwidthmaps.com/?p=4726

[20] (2015, Sep) Africa bandwidth maps – FTTH. Hamilton Research. [Online]. Available: http://www.africabandwidthmaps.com/ftth/

[21] H. van de Groenendaal. (2015, June) Virtual panel discussion: Fibre to the home – as many questions as there are answers. EngineerIT. [Online]. Available: http://www.ee.co.za/article/virtual-panel-discussion-fibre-home-many-questions-answers.html

[22] T. H. Naylor and J. M. Finger, "Verification of computer simulation models," *Management Science*, vol. 14, no. 2, pp. B–92–B–101, 1967.

[23] J. P. Kleijnen, "Verification and validation of simulation models," *European Journal of Operational Research*, vol. 82, no. 1, pp. 145–162, 1995.

[24] J. Carson, "Model verification and validation," in *Simulation Conference, 2002. Proceedings of the Winter*, vol. 1. IEEE, 2002, pp. 52–58.

[25] R. G. Sargent, "Verification and validation of simulation models," in *Winter Simulation Conference, 2005 Proceedings of the 37th*, 2005, pp. 130–143.

[26] H. Bidgoli, *The Internet Encyclopedia, Volume 2*, ser. The Internet Encyclopedia. Wiley, 2004.

[27] ITU-T, *Information technology - Open Systems Interconnection - Basic Reference Model: The basic model*, ITU-T Recommendation Std. X200, Jul 1994. [Online]. Available: http://www.itu.int/rec/T-REC-X.200-199407-I/en

[28] ISO/IEC, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*, ISO/IEC Std. 7498-1, 1994. [Online]. Available: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

[29] R. Braden, "Requirements for internet hosts – communication layers," *RFC 1122*, Oct 1989.

[30] ISO/TC, *Optics and photonics – Spectral bands*, ISO/TC Std. 20 473, 2007. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=39482

[31] B. A. Forouzan, *Data Communications and Networking*, 4th ed. 1221 Avenue of the Americas, New York, NY, 10020: McGraw-Hill International Edition, 2007.

[32] The Fiber Optic Association Inc. (2015) The FOA reference guide to fiber optics: Optic fiber. [Online]. Available: http://www.thefoa.org/tech/ref/basic/fiber.html

[33] ITU-T, *Characteristics of a single-mode optical fibre and cable*, Nov 2009, ITU-T Rec. G.652. [Online]. Available: https://www.itu.int/rec/T-REC-G.652-200911-I/en

[34] ISO/IEC, *Information technology – Generic cabling for customer premises*, ISO/IEC Std. 11 801, 2002. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=36491

[35] ITU-T, *Characteristics of a 50/125 µm multimode graded index optical fibre cable for the optical access network*, July 2007, ITU-T Rec. G.651.1. [Online]. Available: https://www.itu.int/rec/T-REC-G.651.1-200707-I/en

[36] FTTH Council Global Alliance (FCGA). (2015, Feb) FTTH Council - definition of terms. [Online]. Available: http://www.ftthcouncilmena.org/documents/TechnologyDocs/FCGA-DefinitionOfTerms-V.4-Revision2015.pdf

[37] Fiberstore. (2014, Nov) Fiber optic couplers and splitters tutorial. [Online]. Available: http://www.fs.com/Fiber-Optic-Couplers-and-Splitters-Tutorial-aid-405.html

[38] Multicom Inc. (2015) FBT vs. PLC Fiber Optic Splitters - What's The Difference? [Online]. Available: http://www.multicominc.com/training/technical-resources/fbt-vs-plc-fiber-optic-splitters-whats-the-difference/

[39] ITU-T, *Interfaces for the optical transport network*, Feb 2012, ITU-T Rec. G.709. [Online]. Available: https://www.itu.int/rec/T-REC-G.709-201202-I/en

[40] IEEE, *Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment 1: Physical Layer Specifications and Management Parameters for 10 Gb/s Passive Optical Networks*, IEEE Std. 802.3av, 2009.

[41] ITU-T, *10-Gigabit-capable passive optical networks (XG-PON): General requirements*, ITU-T Recommendation Recommendation G.987.1, Jan 2010. [Online]. Available: http://www.itu.int/rec/T-REC-G.987.1/en

[42] ——, *10-Gigabit-capable passive optical networks (XG-PON): Physical media dependent (PMD) layer specification*, Jan 2010, ITU-T Rec. G.987.2. [Online]. Available: http://www.itu.int/rec/T-REC-G.987.2/en

[43] ——, *10-Gigabit-capable passive optical networks (XG-PON): Transmission convergence (TC) layer specification*, Jan 2010, ITU-T Rec. G.987.3. [Online]. Available: http://www.itu.int/rec/T-REC-G.987.3/en

[44] mybroadband.co.za. (2015, March) Huawei GPON launched in South Africa. [Online]. Available: http://mybroadband.co.za/news/fibre/121174-huawei-gpon-launched-in-south-africa.html

[45] T. Weise, *Global Optimization Algorithms - Theory and Application*. it-weise.de (self-published): Germany, 2009. [Online]. Available: http://www.it-weise.de/projects/book.pdf

[46] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.

[47] M. Slater, "Lagrange multipliers revisited," Cowles Commision, Discussion Paper 403, November 1950.

[48] M. Roos and J. Rothe. (2012, Mar. 24) Mathematical programming glossary supplement: Introduction to computational complexity. [Online]. Available: http://glossary.computing.society.informs.org/notes/complexity.pdf

[49] CPLEX Optimization Studio. IBM ILOG. 12.6. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/

[50] Gurobi Optimizer. Gurobi Optimization. [Online]. Available: http://www.gurobi.com/products/gurobi-optimizer/gurobi-overview

[51] CLP solver. COIN-OR Foundation Inc. [Online]. Available: http://www.coin-or.org/Clp/

[52] G. B. Dantzig, *Origins of the simplex method*. ACM, 1990.

[53] K. Murty, *Linear programming*, 1st ed. Wiley, Oct. 1983.

[54] A. Land and A. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, Jul. 1960.

[55] R. Gomory, "An algorithm for the mixed integer problem," The Rand Corporation, Tech. Rep. RM-2597, 1960.

[56] J. E. Mitchell, *Handbook of applied optimization*. Oxford University Press, 2002, ch. Branch-and-cut algorithms for combinatorial optimization problems, pp. 65–77.

[57] G. Cornuéjols, "Valid inequalities for mixed integer linear programs," *Mathematical Programming*, vol. 112, no. 1, pp. 3–44, 2008.

[58] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations research*, vol. 8, no. 1, pp. 101–111, 1960.

[59] J. F. Benders, "Partitioning procedures for solving mixed integer variables programming problems," *Numerische Methematik*, vol. 4, pp. 238–252, 1962.

[60] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management science*, vol. 50, no. 12 supplement, pp. 1861–1871, 2004.

[61] J. Li and G. Shen, "Cost minimization planning for greenfield passive optical networks," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 1, no. 1, pp. 17 –29, June 2009.

[62] E. Danna, E. Rothberg, and C. Le Pape, "Exploring relaxation induced neighborhoods to improve mip solutions," *Mathematical Programming*, vol. 102, no. 1, pp. 71–90, 2005.

[63] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[64] I. Boussad, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82 – 117, 2013, prediction, Control and Diagnosis using Advanced Neural Computations.

[65] A. Shapiro, D. Dentcheva, and A. Ruszczyski, *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, 2009.

[66] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.

[67] G. B. Dantzig, "Linear programming under uncertainty," *Management science*, vol. 1, no. 3-4, pp. 197–206, 1955.

[68] A. L. Soyster, "Convex programming with set-inclusive constraints and applications to inexact linear programming," *Operations research*, vol. 21, no. 5, pp. 1154–1157, 1973.

[69] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Operations research letters*, vol. 25, no. 1, pp. 1–13, 1999.

[70] G. B. Dantzig and A. Madansky, "On the solution of two-stage linear programs under uncertainty," in *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, vol. 1. University of California Press Berkeley, CA, 1961, pp. 165–176.

[71] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable robust solutions of uncertain linear programs," *Mathematical Programming*, vol. 99, no. 2, pp. 351–376, 2004.

[72] P. McGregor and D. Shen, "Network design: An algorithm for the access facility location problem," *Communications, IEEE Transactions on*, vol. 25, no. 1, pp. 61 – 73, Jan. 1977.

[73] R. Galvao and L. Raggi, "A method for solving to optimality uncapacitated location problems," *Annals of Operations Research*, vol. 18, pp. 225–244, 1989.

[74] G. Cornujols, G. L. Nemhauser, and L. A. Wolsey, "The uncapacitated facility location problem," Carnegie-Mellon University Pittsburgh PA Management Sciences Research Group, Tech. Rep. MSRR-493, 1983.

[75] N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on Computing*, vol. 13, no. 1, pp. 182–196, 1984.

[76] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian journal of Mathematics*, vol. 8, no. 3, pp. 399–404, 1956.

[77] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum flow problem," in *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, ser. STOC '86. New York, NY, USA: ACM, 1986, pp. 136–146.

[78] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*. Springer Science & Business Media, 2012, vol. 2.

[79] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multicommodity flow problems," in *Foundations of Computer Science, 1975., 16th Annual Symposium on*. IEEE, 1975, pp. 184–193.

[80] X. Cheng and D.-Z. Du, *Steiner trees in industry*. Springer Science & Business Media, 2013, vol. 11.

[81] M. X. Goemans and Y.-S. Myung, "A catalog of Steiner tree formulations," *Networks*, vol. 23, no. 1, pp. 19–28, 1993.

[82] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, ser. The IBM Research Symposia Series, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, Eds. Springer US, 1972, pp. 85–103.

[83] S. Guha, A. Meyerson, and K. Munagala, "Hierarchical placement and network design problems," in *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*. IEEE, 2000, pp. 603–612.

[84] D. Karget and M. Minkoff, "Building Steiner trees with incomplete global knowledge," in *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, 2000, pp. 613–623.

[85] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, "Provisioning a virtual private network: a network design problem for multicommodity flow," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, ser. STOC '01. New York, NY, USA: ACM, 2001, pp. 389–398.

[86] D. Cieslik, *Steiner Minimal Trees*, ser. Nonconvex Optimization and Its Applications. Kluwer Academic, 1998.

[87] S. Khan, "Heuristics-based PON deployment," *Communications Letters, IEEE*, vol. 9, no. 9, pp. 847 – 849, Sep. 2005.

[88] ——, "Passive optical network layout in Manhattan," *Photonics Technology Letters, IEEE*, vol. 15, no. 10, pp. 1488 –1490, Oct. 2003.

[89] A. Mitcsenkov, G. Paksy, and T. Cinkler, "Topology design and capex estimation for passive optical networks," in *Broadband Communications, Networks, and Systems, 2009. BROADNETS 2009. Sixth International Conference on*, Sep. 2009, pp. 1 –8.

[90] S. van Loggerenberg, "Optimization of passive optical network planning for fiber-to-the-home applications," Master's thesis, North West University Potchefstroom Campus, April 2013.

[91] A. Mitcsenkov, G. Paksy, and T. Cinkler, "Geography- and infrastructure-aware topology design methodology for broadband access networks (FTTx)," *Photonic Network Communications*, vol. 21, no. 3, pp. 253–266, 2011.

[92] K. Papaefthimiou, Y. Tefera, D. Mihaylov, J. Gutierrez, and M. Jensen, "Algorithmic PON/P2P FTTH access network design for CAPEX minimization," in *Telecommunications Forum (TELFOR), 2013 21st*, Nov 2013, pp. 168–171.

[93] B. Kantarci and M. Hussein, "Availability and cost constrained fast planning of passive optical networks under various survivability policies," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, Oct. 2010, pp. 260 –263.

[94] B. Kantarci and H. Mouftah, "Availability and cost-constrained long-reach passive optical network planning," *Reliability, IEEE Transactions on*, vol. 61, no. 1, pp. 113 –124, March 2012.

[95] L. Shi, S.-S. Lee, H. Song, and B. Mukherjee, "Energy-efficient long-reach passive optical network: A network planning approach based on user behaviors," *Systems Journal, IEEE*, vol. 4, no. 4, pp. 449 –457, Dec. 2010.

[96] Y. Kim, Y. Lee, and J. Han, "A splitter location-allocation problem in designing fiber optic access networks," *European Journal of Operational Research*, vol. 210, no. 2, pp. 425–435, 2011.

[97] A. Eira, J. Pedro, and J. Pires, "Optimized design of multistage passive optical networks," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 4, no. 5, pp. 402–411, 2012.

[98] A. Agata and K. Nishimura, "Suboptimal PON network designing algorithm for minimizing deployment cost of optical fiber cables," in *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*, 2012, pp. 1–6.

[99] C. Swamy and A. Kumar, "Primal-dual algorithms for connected facility location problems," *Algorithmica*, vol. 40, no. 4, pp. 245–269, 2004.

[100] A. Gupta, A. Kumar, and T. Roughgarden, "Simpler and better approximation algorithms for network design," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. ACM, 2003, pp. 365–372.

[101] E. Bonsma, N. Karunatillake, R. Shipman, M. Shackleton, and D. Mortimore, "Evolving greenfield passive optical networks," *BT Technology Journal*, vol. 21, pp. 44–49, 2003.

[102] K. Poon, D. Mortimore, and J. Mellis, "Designing optimal FTTH and PON networks using new automatic methods," in *Access Technologies, 2006. The 2nd Institution of Engineering and Technology International Conference on*, June 2006, pp. 49 – 52.

[103] M. Lv and X. Chen, "Heuristic based multi-hierarchy passive optical network planning," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, Sep. 2009, pp. 1 –4.

[104] A. Kokangul and A. Ari, "Optimization of passive optical network planning," *Applied Mathematical Modelling*, vol. 35, no. 7, pp. 3345 – 3354, 2011.

[105] B. Lakic and M. Hajduczenia, "On optimized passive optical network (PON) deployment," in *Access Networks Workshops, 2007. AccessNets '07. Second International Conference on*, Aug. 2007, pp. 1 –8.

[106] T. y Villalba, S. Rossi, M. Mokarzel, M. Salvador, H. Neto, A. Cesar, M. Romero, and M. de L. Rocha, "Design of passive optical networks using genetic algorithm," in *Microwave and Optoelectronics Conference (IMOC), 2009 SBMO/IEEE MTT-S International*, Nov. 2009, pp. 682 –686.

[107] A. Chu, K. F. Poon, and A. Ouali, "Using ant colony optimization to design GPON-FTTH networks with aggregating equipment," in *Computational Intelligence for Communication Systems and Networks (CIComms), 2013 IEEE Symposium on*, April 2013, pp. 10–17.

[108] W. Xiong, C. Wu, L. Wu, X. Guo, Y. Chen, and M. Xie, "Ant colony optimization for PON network design," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, May 2011, pp. 380 –383.

[109] A. Ouali and K. F. Poon, "Optimal design of GPON/FTTH networks using mixed integer linear programming," in *Networks and Optical Communications (NOC), 2011 16th European Conference on*, July 2011, pp. 137 –140.

[110] K. F. Poon and A. Ouali, "A MILP based design tool for FTTH access networks with consideration of demand growth," in *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, Dec. 2011, pp. 544 –549.

[111] R. Chowdhury and B. Jaumard, "A cross layer optimization scheme for WDM PON network design and dimensioning," in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 3110–3115.

[112] M. Grötschel, C. Raack, and A. Werner, "Towards optimizing the deployment of optical access networks," ZIB, Takustr.7, 14195 Berlin, Tech. Rep. 13-11, 2013.

[113] C. Hervet and M. Chardy, "Passive Optical Network design under Operations Administration and Maintenance considerations," *Journal of Applied Operational Research*, vol. 4, no. 3, pp. 152–172, 2012.

[114] L. Gouveia, M. J. Lopes, and A. de Sousa, "Single PON network design with unconstrained splitting stages," *European Journal of Operational Research*, vol. 240, no. 2, pp. 361 – 371, 2015.

[115] B. Kantarci and H. Mouftah, "Optimization models for reliable long-reach PON deployment," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, July 2011, pp. 506 –511.

[116] C. Hervet, A. Faye, M.-C. Costa, M. Chardy, and S. Francfort, "Solving the two-stage robust FTTH network design problem under demand uncertainty," *Electronic Notes in Discrete Mathematics*, vol. 41, no. 0, pp. 335 – 342, 2013.

[117] A. Bley, I. Ljubić, and O. Maurer, "Lagrangian decompositions for the two-level FTTx network design problem," *EURO Journal on Computational Optimization*, vol. 1, no. 3-4, pp. 221–252, 2013.

[118] S. Gollowitzer and I. Ljubić, "MIP models for connected facility location: A theoretical and computational study," *Comput. Oper. Res.*, vol. 38, no. 2, pp. 435–449, Feb. 2011.

[119] I. Ljubić, "A Hybrid VNS for Connected Facility Location," in *Hybrid Metaheuristics*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4771, pp. 157–169.

[120] A. Arulselvan, A. Bley, S. Gollowitzer, I. Ljubić, and O. Maurer, "MIP modeling of incremental connected facility location," *Network Optimization*, pp. 490–502, 2011.

[121] S. Gollowitzer, B. Gendron, and I. Ljubić, "Capacitated network design with facility location," 2012. [Online]. Available: http://homepage.univie.ac.at/ivana.ljubic/research/publications/CapConFL.pdf

[122] I. Ljubić and S. Gollowitzer, "Layered graph approaches to the hop constrained connected facility location problem," *INFORMS J. on Computing*, vol. 25, no. 2, pp. 256–270, Apr. 2013.

[123] M. Leitner, I. Ljubic, M. Sinnl, and A. Werner, "On the two-architecture connected facility location problem," ZIB, Takustr.7, 14195 Berlin, Tech. Rep. 13-29, 2013.

[124] A. Bley, S. Hashemi, and M. Rezapour, "IP modeling of the survivable hop constrained connected facility location problem," *Electronic Notes in Discrete Mathematics*, vol. 41, no. 0, pp. 463 – 470, 2013.

[125] M. Chimani, M. Kandyba, and M. Martens, "2-interconnected facility location: Specification, complexity, and exact solutions," *Electronic Notes in Discrete Mathematics*, vol. 41, no. 0, pp. 21 – 28, 2013.

[126] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. pp. 712–716, 1971.

[127] Q. Botton, B. Fortz, L. Gouveia, and M. Poss, "Benders decomposition for the hop-constrained survivable network design problem," *INFORMS journal on computing*, vol. 25, no. 1, pp. 13–26, 2013.

[128] S. Terblanche, "Contributions towards survivable network design with uncertain traffic requirements," Ph.D. dissertation, North West University Potchefstroom Campus, April 2008.

[129] A. M. Costa, J.-F. Cordeau, and B. Gendron, "Benders, metric and cutset inequalities for multicommodity capacitated network design," *Computational Optimization and Applications*, vol. 42, no. 3, pp. 371–392, 2009.

[130] M. Iri, "On an extension of the maximum-flow minimum-cut theorem to multicommodity flows," *Journal of the Operations Research Society of Japan*, vol. 13, pp. 129–135, 1971.

[131] K. Onaga and O. Kakusho, "On feasibility conditions of multicommodity flows in networks," *Circuit Theory, IEEE Transactions on*, vol. 18, no. 4, pp. 425–429, Jul 1971.

[132] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[133] C. D'Ambrosio, A. Frangioni, L. Liberti, and A. Lodi, "On interval-subgradient and no-good cuts," *Operations Research Letters*, vol. 38, no. 5, pp. 341–345, 2010.

[134] atesio GmbH. Bundesallee 89, Berlin, Germany. [Online]. Available: http://www.atesio.de

[135] S. van Loggerenberg, M. Grobler, and S. Terblanche, "Optimization of PON planning for FTTH deployment with fiber duct sharing," in *Southern African Telecommunications and Networks Access Conference (SATNAC), 2013 Proceedings of*, Sep. 2013.

[136] M. L. Fredman and R. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," in *Foundations of Computer Science, 1984. 25th Annual Symposium on*, Oct 1984, pp. 338–346.

[137] Qt 5.2.1 framework. The Qt Company. Digia Plc., Valimotie 21, 00380, Helsinki. [Online]. Available: http://www.qt.io/

[138] T. Koch and A. Martin, "Solving Steiner tree problems in graphs to optimality," *Networks*, vol. 32, no. 3, pp. 207–232, 1998.

[139] S. Chopra, E. R. Gorres, and M. Rao, "Solving the Steiner tree problem on a graph using branch and cut," *ORSA Journal on Computing*, vol. 4, no. 3, pp. 320–335, 1992.

[140] W. Römisch, "Scenario generation," *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

[141] M. Kaut and S. W. Wallace, "Evaluation of scenario-generation methods for stochastic programming," *Stochastic Programming E-Print Series*, vol. 14, 2003. [Online]. Available: http://edoc.hu-berlin.de/browsing/speps/

[142] N. N. Taleb, *The Black Swan: The Impact of the Highly Improbable*. Random House, 2007.

[143] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE network*, vol. 14, no. 6, pp. 16–23, 2000.

[144] J. Q. Hu, "Diverse routing in optical mesh networks," *Communications, IEEE Transactions on*, vol. 51, no. 3, pp. 489–494, March 2003.

# Appendix A

# Formulation reference

Table A.1: Formulation constituents and descriptions

| Name | Description | Constituents | Reference |
|---|---|---|---|
| $BENDERS_C$ | Basic Benders formulation with column generation | $PON_M$, $PON_F$, $PON_D$ | § 4.3.2 |
| $BENDERS_S$ | Two-stage stochastic formulation | $PON_M$, $PON_F$, $PON_{DS}$ | § 6.2.1 |
| $BENDERS_R$ | Robust formulation | $PON_M$, $PON_F$, $PON_{DARC}$ | § 6.3.1 |
| $BENDERS^{rev}$ | Deterministic revenue formulation | $PON_M^{rev}$, $PON_F$, $PON_D^{rev}$ | § 6.4.1 |
| $BENDERS_S^{rev}$ | Two-stage stochastic revenue formulation | $PON_M^{rev}$, $PON_F$, $PON_{DS}^{rev}$ | § 6.4.1 |
| $BENDERS_R^{rev}$ | Robust revenue formulation | $PON_M^{rev}$, $PON_F$, $PON_{DARC}^{rev}$ | § 6.4.1 |
| $BENDERS_T$ | Multiple splitter type formulation | $PON_M$ with (7.1)–(7.3), $PON_F$, $PON_D$, | § 7.2 |
| $BENDERS_{full}^{surv}$ | Full edge-disjoint survivable formulation | $PON_M^{surv}$ with (7.10)–(7.13), $PON_F^{surv}$, $PON_D$ | § 7.3.1 |
| $BENDERS_{\lambda\text{-}edge}^{surv}$ | $\lambda$-edge disjoint survivable formulation | $PON_M^{surv}$ with (7.10)–(7.12), (7.19), (7.20), (7.24), (7.27), $PON_F^{surv}$, $PON_D$ | § 7.3.2 |
| $BENDERS_{\lambda\text{-}length}^{surv}$ | $\lambda$-length disjoint survivable formulation | $PON_M^{surv}$ with (7.10)–(7.12), (7.19), (7.20), (7.27), (7.28), $PON_F^{surv}$, $PON_D$ | § 7.3.2 |
| $BENDERS_{ML}$ | Multi-level formulation | $PON_{ML}$, $PON_{FL}$, $PON_D$, $PON_I^{\ell^*}$, with $\ell^* \in \mathrm{L}$, $\ell^* \neq 0$ | § 7.4.2 |

Table A.2: Benders problem descriptions and reference

| Name | Description | Defined in |
|---|---|---|
| $PON_M$ | Standard Benders master problem | (4.21)–(4.24) |
| $PON_F$ | Feeder sub-problem | (4.34)–(4.36) |
| $PON_D$ | Distribution sub-problem | (4.25)–(4.27) |
| $PON_{DS}$ | Two-stage stochastic distribution sub-problem | (6.1)–(6.3) |
| $PON_{DARC}$ | Robust distribution sub-problem | (6.23)–(6.27) |
| $PON_M^{rev}$ | Revenue master problem | (6.34)–(6.38) |
| $PON_D^{rev}$ | Revenue distribution sub-problem | (6.39)–(6.41) |
| $PON_{DS}^{rev}$ | Two-stage stochastic revenue distribution sub-problem | (6.44)–(6.46) |
| $PON_{DARC}^{rev}$ | Robust revenue distribution sub-problem | (6.49)–(6.53) |
| $PON_M^{surv}$ | Full edge-disjoint master problem | (7.6)–(7.9) |
| $PON_F^{surv}$ | Full edge-disjoint feeder sub-problem | (7.14)–(7.16) |
| $PON_{ML}$ | Multi-level master problem | (7.34)–(7.46) |
| $PON_{FL}$ | Multi-level feeder sub-problem | (7.47)–(7.49) |
| $PON_I^{\ell^*}$ | Level $\ell^*$ intermediate sub-problem | (7.52)–(7.54) |

# Appendix B

# Conference and paper contributions from thesis

- S.P. van Loggerenberg, M. Ferreira, M.J. Grobler and S.E. Terblanche, "Benders Decomposition of the Passive Optical Network Design Problem Under Demand Uncertainty", *Networks*, Wiley, Submitted for publication, Nov. 2015.
- S.P. van Loggerenberg, M. Ferreira, M.J. Grobler and S.E. Terblanche, "PON-OPT: A Passive Optical Network Design Optimisation Tool", in *Southern African Telecommunications and Networks Access Conference (SATNAC), 2015 Proceedings of*, Hermanus, South Africa, Sep. 2015.
- S.P. van Loggerenberg, M. Ferreira, M.J. Grobler and S.E. Terblanche, "Benders Decomposition of the Passive Optical Network Design Problem", in *International Network Optimization Conference, 2015 Proceedings of 7th*, Warsaw, Poland, May 2015.
- S.P. van Loggerenberg, M. Ferreira, M.J. Grobler and S.E. Terblanche, "Optimal Passive Optical Network Planning Under Demand Uncertainty", in *Southern African Telecommunications and Networks Access Conference (SATNAC), 2014 Proceedings of*, Port Elizabeth, South Africa, Sep. 2014.

- S.P. van Loggerenberg, M.J. Grobler and S.E. Terblanche, "Optimization of PON Planning for FTTH Deployment with Fiber Duct Sharing", in *Southern African Telecommunications and Networks Access Conference (SATNAC), 2013 Proceedings of*, Stellenbosch, South Africa, Sep. 2013.