Master Thesis Applied Mathematics at Delft University of Technology

Optimizing Healthcare Accessibility through Flood Resiliency Improvements of Roads in a Network

A case study for Timor-Leste

Britt van Veggel

A collaboration between the World Bank Group and Analytics for a Better World

Optimizing Healthcare Accessibility through Flood Resiliency Improvements of Roads in a Network

by

Britt van Veggel

to acquire a Master of Science degree in Applied Mathematics
at the Delft University of Technology
to be defended publicly on 10th of December, 2021.

Committee

Prof. dr. ir. Karen Aardal
Prof. dr. ir. Dick den Hertog
Dr. ir. Tina Nane

You must be the change you wish to see in the world.

Mahatma Gandhi

Abstract

Access to healthcare is a requirement for human well-being that is partly dependent upon safe infrastructure. One of the UN Sustainable Development Goals regarding healthcare is to achieve universal healthcare coverage, which includes access to quality essential health-care services. In many developing nations, roads are often vulnerable to floods. Floods can cause roads (especially roads with a dirt or gravel surface type) to become inaccessible for a long period of time. This inaccessibility can cause many inhabitants to lose access to a healthcare facility within a crucial traveling time span. Upgrading flood prone roads on which many households are dependent in order to access a healthcare facility, could reduce this threat to healthcare accessibility for many inhabitants. This research aims to use optimization techniques to reduce the impact floods can have on healthcare accessibility, and apply a case study to the country Timor-Leste. We formulate an optimization model that maximises the number of households that can access a healthcare facility within a 5 kilometer traveling distance via a flood resilient route, given a specific budget. Alongside this formulation, we provide a (simple) flood and costs model for the road as well as different heuristics to find (near-)optimal solutions. Our research includes multiple tests to determine which heuristic works best and which parameters and other settings increase the computational performance of these heuristics for Timor-Leste. The heuristic that performs the best is a dynamic greedy heuristic. This algorithm is able to generate an optimal solution for all possible budgets within 4 hours.

Foreword

With a great sense of fulfillment I write this foreword. It is the last thing I want to do before I hand in my thesis. It is such a strange feeling that a project I have worked on for such a long time with so much passion and dedication has come to an end. This project has been a perfect match for me. For a long time now, I have been quite upset about the state of our earth and the inequality that exists within it. I have been searching for ways to be a part of the change I want to see happen and this project has been a beautiful way of doing so. I have been able to apply my passion for optimization, my eagerness to try new things and my love for programming into doing something that I really believe in.

Dear Dick, first and foremost, I want to thank you for this beautiful opportunity and the incredible dedication with which you have supported me during this project. This project has been a dream come true and would have never ended the way it has if it was not for how much time you have invested into supporting me. Every single Tuesday morning, I was able to voice my progress and concern and ask all questions I had on my mind. That has been a blessing. Your ambition is incredibly inspiring. And maybe even more inspiring, is the way you seem to know exactly how to reach your goals. I am so, so eager to continue contributing to the beautiful work that you are doing with Analytics for a Better World and I cannot wait to see what's next.

Dear Karen, I have enjoyed working with you so much. Whenever we had a meeting, we spent 30 minutes talking about the project, and 90 minutes about real complex issues like the climate crisis, the COVID pandemic or gender inequality. You also made me feel confident about myself as a mathematician, by encouraging me to try out my own ideas while supporting me in developing them. And let us not forget the last month, where you were sick as a dog and still spent you Saturday evenings correcting (among other things) my embarrassing spelling mistakes. I am really happy you are going to stay involved during my PhD and we can continue creating beautiful things together. Thank you very much for everything.

Furthermore, I want to thank Tina Nane for taking the time to be a part of my committee and Dion Gijswijt for jumping in to relieve Karen of her professional duties in case she is too sick to do so. I also want to thank Parvathy Krishnan and Kai Kaiser for providing the perspective of the World Bank within this project. And, another thank you goes out to Panchamy Krishnan, who has helped us find realistic boundaries within which we can develop flood and cost models.

Lastly, I want to thank my friends and family. First and foremost, I want to thank my parents. They have supported me from start to end through my studies. Even when I decided to quit my second master, they continued to have faith in me and helped me however they could. I feel how proud they are of me every single day and that idea makes me very happy. Secondly, I want to thank my old roommate Tom Valckx, who has been my *IT-guy* during this project and

functioned as my rubber duck whenever I just had to express my enthusiasm about this project or needed to write an important e-mail. I also want to thank Jasper Derikx and Tom Smid who have repeatedly taken the time to help me with programming issues and think out ways to formulate my thoughts. There are many more people from my personal life who deserve a huge thank you, especially for the amount of hours they have peacefully endured listening to me talk enthusiastically about my thesis. And at last, the army of spellcheckers over the past month deserve a big thank you.

I hope the reader is able to sense the passion and care I have put into this project through the upcoming pages.

Britt van Veggel,
Nijmegen, December 3rd, 2021.

Contents

1 Introduction

1.1 Context and relevance

Access to healthcare is a requirement for human well-being that is partly dependent upon safe infrastructure. One of the UN Sustainable Development Goals regarding healthcare is to achieve universal healthcare coverage, which includes access to quality essential health-care services. Ensuring that inhabitants of developing countries have access to a healthcare facility within 5 kilometer traveling distance, could achieve this goal [1].

The World Bank Group is a financial institution that aims to decrease extreme poverty worldwide by providing loans, grants and policy ideas to governments around the developing world. Analytics for a Better World (ABW) is a research collaboration between the University of Amsterdam and Massachusetts Institute for Technology led by Dick den Hertog (UvA) and Dimitris Bertsimas (MIT) that aims to stimulate research that applies analytics to societal issues. The goal of the Analytics for a Better World collaboration with the World Bank is to create models that allow policy makers to make more data driven choices within their projects.

The ABW collaboration with the World Bank started off creating a facility location model for placement of healthcare facilities in developing countries. This model took into account all current infrastructure, regardless of flooding risks on the roads. Floods play a large role in disrupting the accessibility of these healthcare facilities. As many roads in developing nations are dirt and gravel roads and these are vulnerable to floods and seasonal precipitation, these roads are most vulnerable to being inaccessible for a long period of time. This loss in infrastructure forces people in need of healthcare to travel much longer distances over unaffected roads towards a healthcare facility, or disrupts all traveling routes towards any healthcare facility all together. Thus, updating roads to be flood resilient while taking into account how this affects the healthcare accessibility of the population is an important addition to the healthcare facility location model.

1.2 Focus and scope

This research aims to formulate a model that identifies links that must be upgraded in order to increase civilian flood resilient access to healthcare facilities within a 5 kilometer traveling distance. Alongside this formulation, it aims to develop and test different algorithms that can find near-optimal solutions. The model will be applied to and tested on the country of Timor-Leste as a case study.

The optimization model that will be formulated aims to solve the following:

Maximize	Number of households that are connected to a healthcare facility
Subject to	Costs of upgrading the road segments does not exceed a given budget
	Paths of households to healthcare facilities are no longer than 5 kilometers
	Paths of households to healthcare facilities must be flood resilient

This research will touch on literature concerning road network optimization models and a background research regarding practical aspects of development aid and infrastructure modeling. It will propose different formulations and algorithmic approaches to find (near-)optimal solutions, and test which formulations and algorithms work best on the Timor-Leste dataset. We will also analyze of the Timor-Leste data, and a flood model and a cost model for the roads. It is important to mention that these flood and cost models are simplistic models and should be extended upon by an infrastructure expert in order to represent reality better.

What is important to note about this research is that the facilities we aim to improve the accessibility of are healthcare facilities. But these facilities could also be schools, markets, emergency aid posts, et cetera. Also, we have chosen a travel distance threshold of 5 kilometers because this is a threshold proposed by the World Bank, but our algorithms will also be applicable for different distances.

1.3 Research questions and objectives

The main research question of this project was

> *How to minimize the impact that flood prone roads have on healthcare accessibility in developing nations, using optimization techniques.*

The subquestions were the following:

- Is there any (related) literature on this problem?
- What formulation suits our problem?
- How do we prepare the data we need as input for our optimization model, cq.:
  - How to model flood risks on roads;
  - How to model upgrading costs;
  - How to identify the households that can not yet access a healthcare facility within 5 kilometers via a flood resilient road;
- How to generate a suitable path set for our optimization model?
- Which algorithms could be useful to find (near-)optimal solutions for our model, especially for large scale practices?

- How can we increase the computational performance of our heuristics?
- What heuristic performs best?

1.4 Structure of the thesis

This thesis contains nine chapters, a references section and an appendix. The structure of the chapters and their functions are as follows:

(1) **Introduction**: Introduces the context, focus, scope, objectives and structure of this thesis;

(2) **Literature review and background research**: Outlines the literature review and background research that substantiates this research. This contains a review of mathematical models surrounding road network design and a background research on the country of Timor-Leste, objectives and methods of the World Bank and infrastructure modeling in practice;

(3) **Our formulations**: Proposes an optimization objective and constraints. It introduces the terminology that will be used within this project and the demands from the World Bank that have to be taken into account. Furthermore, it proposes the two different formulations and compares them in order to choose the most suitable one;

(4) **Data preprocessing**: Outlines the Timor-Leste data we work with and explains the flood model, the cost model and the accessibility algorithm and analyzes their results;

(5) **Generating paths**: Proposes two algorithms that are able to generate a set of paths that can be used as input for the selected optimization model. It compares the performance of these algorithms in order to select the most suitable one;

(6) **Algorithms to find (near-) optimal solutions for Timor-Leste**: Discusses different heuristics that have been developed and compared to solve the optimization problem. Includes a proposed implementation of the branch and bound algorithm for small scale, local scenarios and multiple heuristics that can be applied to the whole nation of Timor-Leste. Two of which use this small scale implementation of the branch and bound algorithm, and the other is a dynamic greedy algorithm;

(7) **Considered configurations**: Many different configurations have been considered and tested for our different algorithms. The analyses about these considered configurations are quite extensive and can distort the narrative. Therefore, they are summoned in this chapter;

(8) **Performance results of the main heuristics**: Analyzes and evaluates the performance of the two proposed heuristics that can be applied to the nation of Timor-Leste;

(9) **Conclusions and discussion**: Summarizes research and the results it has brought forward, and lists recommendations for further research.

Lastly, it is advised to print this thesis in color because some images use a lot of color coding.

Meaning of variables

Variable	General meaning (might be specified alongside model in which it is used)
a_{lr}	Binary variable indicating if link l is part of route r
A	Set of healthcare facility areas in Timor-Leste
α	A factor of a term that is flexible or not yet decided
B	Budget
BS	Set of budget scenarios
c_l	Flow costs of link l (could be minimum, maximum, for a specific commodity, etc)
d_i	Demand between O-D pair i
D_k	Number of routes terminating in zone k (could be for certain commodity or demand)
e_l	Construction costs of link l
$end(p)$	Function returning the node at the end of path p (which is the node on the road to which a set of household clusters is connected)
E	Set of links / edges
f_l	Flow on link l (could be for a certain commodity)
$H_i(u)$	Function returning the demand for O-D pair i according to costs u
h_r, h_i	Number of trips made on path $r \in P$ or O-D pair i
K	Number of paths generated per O-D pair when generating relevant paths
κ	Set of commodities along with a demand for each demand
l_p	Length of path p (in kilometers)
N	Set of nodes
N_{hcf}	Subset of nodes that represent a healthcare facility
$N_{hcf,n}$	Subset of nodes that represent healthcare facilities that are within a 5 kilometer range of node n
N_r	Subset of nodes that represent the nodes on the road that connect at least one
O_j	Number of routes originating from zone j (could be for certain commodity or demand)
P	Path set
P_i	Paths between O-D pair i
ϕ	Generic objective value function
R_k	Required flow of demand k to be shipped
S_p, S_n	Number of households dependent upon path p, or number of households whom are closest to roadnode n
u_i	Minimum costs of O-D pair i
x_l	Decision variable indicating if link l has been upgraded
y_n	Decision variable indicating if household cluster n is connected via a path
z_p	Decision variable indication if path p is fully flood resilient (due to upgrades on the links that lie on it)

2 Literature review and background research

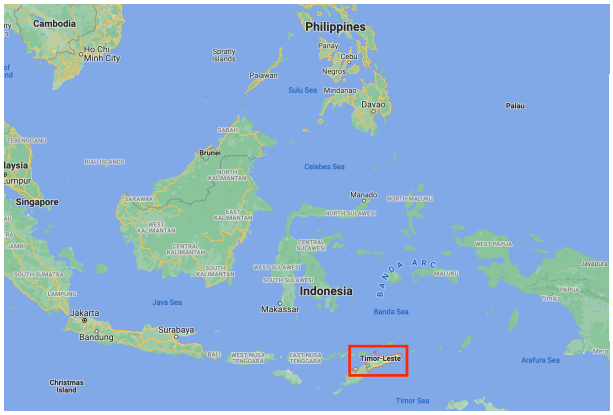
In this chapter the literature review and background research is discussed. The literature review and background research was essential for this project because it provided us with a better insight into the mathematical and practical aspects of problems such as this one, which allows us in provide the World Bank with a product that fit their needs.

First, we will elaborate on background information about Timor-Leste and activities of the World Bank. Next, we will discuss the literature research that was done in order to inspire the formulation for a model that is suitable for our purposes and to come up with an algorithmic approach that is able to find a (near-) optimal solution within an acceptable running time. The optimization that we looked into is the road network design problem. This problem differs from our problem, but does focus on connecting different parts of a network, which is a challenging aspect of our problem. Lastly, we will discus practical aspects of infrastructure modeling.

2.1 Background information about Timor-Leste and activities of the World Bank

Timor-Leste is a small South-East Asian country right below Indonesia. Its geographical location can be seen in Figure 1a. Timor-Leste gained the status of sovereign state on May 20^{th}, 2002, after a long colonial history with Portugal and a territorial history with Indonesia [2]. It has a population size of a little over 1.3 million inhabitants [3]. With an annual GDP of less than $1500 per head of the population, it has 42% of its inhabitants living in poverty [3, 4]. The capital city of Timor-Leste is Dili, which lies in the mid-north of the country. Furthermore, the country is divided into thirteen districts. The district of Oecussi does not lie attached to the peninsula and the district of Dili is includes an island named Ataúro. The distribution of these districts can be found in Figure 1b. The country has an area of 15007 squared kilometers, containing 347 healthcare facilities and 7638.8 kilometers of road. Timor-Leste is often affected by heavy floods and landslides [5]. In Chapter 4 the statistics regarding spread of inhabitants, healthcare facilities, roads and flooding data found in our data are described.

In November 2019, the World Bank Group approved the Country Partnership Framework for Timor-Leste [3]. This strategy guides the World Bank Group's program through the fiscal years 2020 and 2024. The framework aims to support the government of Timor-Leste to transform its natural wealth into improved human capital and sustainable infrastructure in three key focus areas. One of these goals is to improve access and quality of connective infrastructure in transport sectors. Infrastructure is a backbone for the economy, the food-supply chain and for healthcare accessibility. With the growing threat of climate change, natural hazards like floods, extreme heat and quick changes in temperature will disrupt and damage infrastructure systems more often [6]. In Timor-Leste, the largest threat is floods from rivers and heavy precipitation (especially cyclones) [5]. As the World Bank wants to take a much more data driven approach,



(a) Location of Timor-Leste in South-East Asia

(b) Districts of Timor-Leste (and the island of Ataúro)

Figure 1 Geography of Timor-Leste

modeling the flooding risks on the roads, the costs to make the roads flood resilient, the accessibility to healthcare facilities and the optimal road investments to increase this accessibility, is essential [7].

2.2 Network Design Problems

This section addresses the mathematical theory that has inspired our optimization model. Our problem, to the best of our knowledge, has not been studied before. Therefore, we began our project with finding literature that was somewhat like our problem. The literature we studied concerned the road network design problem (RNDP). This problem is similar because it takes into account the network structure to make sure that there exists a connection between different parts of the network. It is also similar because it decides whether a link must be accessible or not.

Until 1984, the mathematical field of optimization had not yet played a big role within the transportation sciences [8]. The pioneers that set optimization to be a standard within this field, are Boyce and Magnanti and Wong. In this section, we will formulate and explain the network design models they proposed. Alongside these models, we discuss algorithms that were discussed in the literature to find (near-)optimal solutions to these optimization problems. After we discuss the theory, algorithms that have been applied in order to find (near-)optimal solutions for the road network design models will be discussed.

2.2.1 The work of Boyce

Boyce [9] outlines three different road network optimization models. All of these models work with origin-destination pairs (O-D pairs), which refers to a commodity that has to be trans-

min	$\sum_{i \in I} c_i^+ d_i$	Total travel costs for all O-D pairs
s.t.	$\sum_{l \in E} e_l x_l \leq B$	Construction costs stay within budget
	$c_i^+ = \min_{r \in P_i} \sum_{l \in E} c_l a_{lr} x_l$	Costs of O-D i is the costs for a route $r \in P_i$
	$x_l \in \{0,1\} \qquad \forall l \in E$	Integrality constraints

Input

$$a_{lr} = \begin{cases} 1 & \text{If link } l \text{ is part of route } r \\ 0 & \text{else} \end{cases}$$

B	Budget for the project
c_l	Fixed travel costs of link l
c_i^+	Total minimum travel costs of O-D i
d_i	Fixed demand between O-D i
e_l	Construction costs of link l
E	Set of all possible links
I	Set of O-D pairs
P_i	Set of routes for O-D i

Decision variables

$$x_l = \begin{cases} 1 & \text{Link } l \text{ in } E \text{ is included in the network} \\ 0 & \text{else} \end{cases}$$

Model 1 Network design model of Boyce as formulated in [9].

ported from an origin to a destination in a certain quantity. In our case, such a commodity is a household traveling from their house (Origin) to a healthcare facility (Destination). All possible routes for an O-D-pair i are included in the paths sets P, and thus contain links that can be established or included. We will describe only the the road network design model because our final model does not take any inspiration from the other models that is not also included in the road network design model.

The network design problem

The network design problem aims to find the best set of links to construct, in order to minimize the road user costs. These costs can also be interpreted as distances. It aims to ensure that all O-D pairs are connected and also ensures that the construction costs stay within a set budget. The model is is stated in Model 1.

This problem differs from our problem because it ensures all O-D pairs are connected. This will likely not be feasible for our case, especially because we bound the traveling distances. Besides from that, the traveling costs are minimised in this formulation, while we need to maximise the number of connected households.

Remarks

What is important to mention about these models is that a set containing all the paths of the O-D pairs has to be part of the input. Boyce [9] does not mention how to attain this set, which makes the models quite abstract because generating a set of multiple possible paths between different O-D pairs is an optimization problem on its own.

Furthermore, the definition of the c_i^+ variable contains a min. This is not a proper way to formulate a linear optimization model (and minimizing the costs for each path individually probably does not yield an optimal overall solution). It is most likely that Boyce formulated this model to describe an idea, rather than an applicable optimization model. The fact that Boyce does not apply any algorithms to solve this model nor explains how to generate a set of paths, also insinuates this.

This models is centered around demand of commodities and their transportation (or in proper terms: flow) costs and benefits. Therefore, it does not apply exactly to our case. But, it does serve as inspiration for our models. For example, the idea of using a set of paths as input was very inspirational. As we want paths of at most 5 kilometers from a household (Origin) to a healthcare facility (Destination), we can create a set of paths containing paths of at most 5 kilometers long. This will be one of the crucial factors of the final model.

2.2.2 The model of Magnanti and Wong (1985)

Magnanti and Wong [10] describe a general model that can result in many different optimization problems. This general model sets a basis for many optimization problems on graphs. Among these problems are the renowned Minimum Spanning Tree Problem, Shortest Path Problem, Steiner Tree Problem, Traveling Salesman Problem, Budget Design Problem, Network Design Traffic Equilibrium Problem, et cetera. The objective function of the general model is not fixed, because these different problems have different objectives.

This model also works with O-D pairs, but the paths for these O-D pairs are not an input variable. The generation of these paths is part of the optimisation model. This makes the model very different from Boyce's model (Model 1) both conceptually and computationally. This is due to the fact that besides from generating the optimal set of links to upgrade, it will also have to generate the shortest paths, resulting in many more variables to optimize over.

First, the general model will be formulated and explained in Model 3. Afterwards, the application of this model to the road network design problem will be formulated. In the Appendix, Section A, an additional example of the general model applied to the minimum spanning tree can be found.

The general model of Magnani and Wong

The general model has a generic objective, but three fixed constraints. The first constraint regards the generation of a path going from an origin to a destination. So, for example, within the Network Design Problem, it allows the nodes that need to be connected to find a path within the optimization process. The second constraint bounds the flow that can travel over a link. This constraint does not apply to our case, because we assume our roads can take any number of travelers. The third constraint restricts the construction costs by a certain budget. The fourth constraint is a generic constraint that is added to leave room for any other needed constraints. The fifth constraint ensures the variables are integer.

Road Network Design Problem based upon the general Magnanti & Wong model

The application of this model to the network design problem changes the general model in the following way. First of all, the objective function is set. It aims to minimize the total flow costs over all O-D pairs. Furthermore, the second constraint, regarding the capacity of the flow now has a slightly different goal. It does not bound the capacity of a link anymore, but it ensures a path only travels over a link that has been established. The constraints that have not changed are the construction cost constraint and path generation constraint.

min $\phi(f, x)$ Generic objective function (0)

s.t. $\sum_{j \in N} f_{ij}^k - \sum_{l \in N} f_{li}^k = \begin{cases} R_k & \text{if } i = O(k) \\ -R_k & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases}$ The flow of commodity k on its accompanying O-D pair must start with R_k flow at its origin $O(k)$ and end with R_k flow at its destination $D(k)$, and for all the remaining nodes in the network the flow must go in and out of said node, or non must have gone in nor out. (1)

$$\forall k \in \kappa, i \in N$$

$f_{ij} = \sum_{k \in \kappa} f_{ij}^k \leq K_{ij} x_{ij} \qquad \forall (i, j) \in E$ Costs of O-D i is the minimal costs for a route $r \in P_i$ (2)

$\sum_{(i,j) \in E} e_{ij} x_{ij} \leq B$ Construction costs must be less or equal to the budget (3)

$(f, x) \in S$ Side constraint S (4)

$f_{ij}^k \geq 0, x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in E, k \in \kappa$ Integer constraints (5)

Input

B	Budget for the project
c_{ij}^k	Per unit arc travel costs
$D(k)$	Destination nodes of demand of commodity k
E	(Potential) edge set
e_{ij}	Construction costs of arc (i, j)
κ	Set of commodity demands for different commodities
$O(k)$	Origin nodes of demand of commodity k
R_k	Required flow of demand of commodity k to be shipped
N	Node set

Decision variables

$f_{ij}^k =$ Amount of demand of k that flows via edge (i, j)

$x_{ij} = \begin{cases} 1 & \text{Link } (i, j) \text{ in } E \text{ is included in the edge set} \\ 0 & \text{else} \end{cases}$

Model 2 General formulation Magnanti & Wong.

$$\textbf{min} \quad \sum_{ij \in E} \sum_{k \in \kappa} c_{ij}^k f_{ij}^k \qquad \text{Minimize road user costs} \tag{0}$$

$$\textbf{s.t.} \quad \sum_{j \in N} f_{ij}^k - \sum_{l \in N} f_{li}^k = \begin{cases} 1 & \text{if } i = O(k) \\ -1 & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \qquad \text{For any O-D pair } k\text{, the path should start at the origin } (O(k),) \text{ and end at the destination } (D(k)) \text{ and either pass through a node on the way or not at all.} \tag{1}$$

$$\forall k \in \kappa, i \in N$$

$$f_{ij}^k \leq x_{ij} \qquad \forall (i,j) \in E \qquad \text{Costs of O-D } i \text{ is the minimal costs for a route } r \in P_i \tag{2}$$

$$\sum_{(i,j) \in E} e_{ij} x_{ij} \leq B \qquad \text{Construction costs must be less or equal to the budget} \tag{3}$$

$$f_{ij}^k, x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E, k \in \kappa \qquad \text{Integer constraints} \tag{4}$$

Input

B	Budget for the project
c_{ij}^k	Per unit arc travel costs
$D(k)$	Destination nodes of demand k
E	(Potential) edge set
e_{ij}	Construction costs of arc (i, j)
κ	Set of commodity demands for different commodities
$O(k)$	Origin nodes of demand k
N	Node set

Decision variables

$$f_{ij}^k = \begin{cases} 1 & \text{link } (i,j) \text{ is part of the path for O-D pair } k \\ 0 & \text{else} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{link } (i,j) \text{ in } A \text{ is included in the edge set} \\ 0 & \text{else} \end{cases}$$

Model 3 Road network design problem formulation based upon to Magnanti & Wong.

2.2.3 Advised algorithmic approaches to find (near-)optimal solutions

For both Road Network Design models, two algorithmic methods to find (near-)optimal solutions were repeatedly advised. These methods are decomposition methods and the branch and bound method [9–11].

For the decomposition methods, the advised methods are Danzig-Wolfe decomposition [9] (column generation) and Benders decomposition [11] (row generation). The Danzig-Wolfe decomposition was advised for the model of Boyce (Model 1), but an implementation was not provided. For the Magnanti and Wong model (Model 3), an implementation of the Benders decomposition [10] was provided. It was advised to iteratively generate routes for variables $f_{ij}^{(k)}$, and using these to generate an optimal network configuration.

For small scale cases, a branch and bound algorithm is advised for the Magnanti and Wong model (Model 3) [11]. Hoang [12] argues that when applying branch and bound to Model 3, choosing a different lower bound function would decrease computations immensely. This lower bound is defined as follows:

$$\phi(x^P) + \sum_{(i,j)\in\bar{A}_F} \sum_{k\in\kappa} (1 - x_{ij}) I_{ij}^k(x^P) \tag{2.1}$$

Where A_F is the set of edges that have been fixed in the branch and bound enumeration tree, and $\bar{A}_F$ is the set of edges that has not yet been fixed. The vector x^P is the vector where all $x_l = 1$ for all links $l \in \bar{A}_F$ and $I_{ij}^k(x^P)$ is the increase in travel costs for O-D commodity k if it travels over the network defined by x^P without link (i, j). The idea is that if the link (i, j) is removed from the solution, then the costs from i to j must be at least $I_{ij}^k(x^P)$.

In addition to this lower bound, Magnanti and Wong [10] also advise to relax the x variable in Model 3 to speed up calculations.

We also aimed to find literature that applied these models to large scale cases and see if they proposed different heuristics. Heng et al. [13] applied a branch and bound algorithm on a combined facility location and network design model formulated like the Magnanti and Wong model (Model 2) to a small in Cambodia. The branch and bound algorithm (as implemented in CPLEX) has a set maximum running time of 20minutes. They conclude that this model is applicable to smaller districts with 130.000 O-D pairs and an area of 2000 squared kilometers (thus too small for the case of the entire of Timor-Leste). The formulation of this model can be found in [13].

2.2.4 Infrastructure modeling in practice

In order to ensure that this model is realistic, it is important to get a grasp on the practical aspects of the infrastructure modeling. This regards modeling the flood risk on the roads, interventions to make roads flood resilient, and modeling the costs of these interventions. It is also important to understand what data was necessary to be able to model these aspects.

flood risk on infrastructure modeling

Flood risks on road infrastructure can be modeled as realistically and detailed as is needed for a project [14]. The basis for every flood model would be the road data and flooding risk statistics. Incorporating surface type would be a next step. Factors like altitude, year of construction, current state of the road, et cetera, add more detail to such a model. However, the interaction of these aspects on the road quickly becomes more complex and could best be left to a more specialized researcher. Therefore, it was most advisable to create a very simple model that could be expanded upon by an expert. Our focus should be mostly on the optimisation.

Possible flood resiliency interventions

There are many interventions that can be made to make a road more resilient to floods. Some interventions are installing drainage systems, asphalting a road and raising the level of the road [6]. But, as stated in the section above, this is very hard to model and our models can best be simple dummy models that leave room for expansions from an expert.

Cost modeling

The construction costs of different road interventions are impossible to model, especially this much in advance, because the smallest unpredictable factors could be of great influence: the weather on the day of construction, the value exchange rate for the currency in Timor-Leste, the state of their economy, and so on [14]. Besides from that, there is such a broad range of possible interventions, that modeling their flood effect would need much more data than solely the data provided by the World Bank.

3 Our formulations

In this chapter the final model and an alternative to that model are presented. Before we discuss these models, it is important to define some terms and explain the demands we received from the World Bank that we would need to give shape to.

Terminology

- When a road segment is considered to be *at risk*, this means that it is at risk of being flooded to such an extent that it can heavily disrupt healthcare accessibility for the households that depend upon that road segment.
- An *upgraded* road segment is a road segment that was at risk, but has now undergone construction through which it can be considered to no longer be at risk.
- When we mention a household being *connected*, this household has access to a healthcare facility via a flood resilient path within 5 kilometer traveling distance.
- A *healthcare facility area* is the area of 5 kilometers around a specific healthcare facility. (Examples of healthcare facility areas can be found in Figure 16).

Demands of the World Bank

Together with the World Bank, the following demands were established.

(1) The model should find a construction plan for the roads such that less households are affected by flood disruptions when traveling towards a healthcare facility;
(2) There should be a fixed budget for the construction costs, and if possible a Pareto curve analysis (which is a graph that shows the benefit of an optimal solution for different budgets);
(3) The paths from household (clusters) to healthcare facilities should abide traveling thresholds as proposed in the UN Sustainable Development Goals. These distance thresholds are 2, 5 and 10 kilometer distance. The 5 kilometer traveling distance should be the main focus;
(4) Connecting more households to healthcare facilities should be more important than connecting one household via multiple roads.

Another wish of the World Bank was to create a Python tool that would be as computationally efficient as possible such that it could still be interactive. It is always difficult to predict what is feasible and what is not, but as a target we chose to aim for a computing time maximum of at most half a day (12 hours).

3.1 Facility Accessibility Road Network Upgrading Problem Models

The demands of the World Bank gave flexibility for different models. For the final model, the Facility Accessibility Road Network Upgrading Model (FARNUM), the idea is as follows:

Maximize Number of households that are connected to a healthcare facility

Subject to Costs of upgrading the road segments does not surpass a set budget

Paths of households to healthcare facilities are no longer than 5 kilometers

Paths of households to healthcare facilities must be flood resilient

Two models for this problem have been designed, each with the same objective functions and in essence the same constraints. The difference between these two models is that one model takes a set of paths as input (like Boyce's model (Model 1)) and the other model generates the paths for the O-D pairs during the optimization (like the models of Magnanti and Wong (Subsection 2.2.2)). This difference in how the paths are generated can make a big difference in performance and computational results. This will be analyzed later on.

For both models it is important to note that the updating costs of an at risk link will be the cheapest update that ensures (near) flood resilient access (eg. if a link will be considered to be flood resilient when updated to gravel, the updating costs for asphalt will not be considered). How this is modeled, will be explained in Chapter 4. Lastly, these models use a maximum distance of 5 kilometers, but this distance can be changed.

We will first discuss the model that takes the set of paths as input. Afterwards, we discuss the model that generates the paths within the optimization.

3.1.1 Path pre-generated model, inspired by Boyces model

The Boyce-inspired model aims t maximize the number of connected households. This model takes a set of paths going from household clusters to healthcare facilities of at most 5 kilometer distance as input, and then finds an optimal combination of these path, such that if the at risk road segments that lie on these paths would be upgraded, a maximum number of households would become connected. What makes this model extra efficient, is that it only takes into account the road segments that are at risk of being flooded, which heavily decreases the number of road segment variables to optimize over. The formulation of this model can be found in Model 4.

max	$\sum_{p \in P} S_p z_p$	Maximize the number of connected households	(0)
s.t.	$\sum_{l \in E} e_l x_l \leq B$	Budgetary restrictions	(1)
	$z_p \leq x_l \quad \forall l \in p \cap E^*, \forall p \in P$	A path can only exist if all edges on the path are flood resilient.	(2)
	$\sum_{p \in P_{end(n)}} z_p \leq 1 \quad \forall n \in N_r$	Each household cluster has at most one flood resilient path leading towards it, in order to not count households double.	(3)
	$z_p, x_l \in \{0,1\} \quad \forall p \in P, \forall l \in E^*$	Integer constraints	(4)

Input

B	Budget
e_l	Costs of updating edge l to be flood resilient
E^*	Edge set containing only the links that are at risk
N	Node set
$l, (i,j) \in E$	Link l or (i,j) in edge set E, containing only the at risk roads
$N_r \subset N$	Subsets of nodes that connect at least one household to the road
$p \in P$	A path in the set of all possible paths P
$P_{\text{end}(n)} \subset P$	The set of all possible paths of a distance of at most 5 kilometers long ending up in node n
S_p	Number of households that can access a healthcare facility during all seasons within 5km via path p

Decision variables

$$x_l = \begin{cases} 1 & \text{If edge } l \text{ is upgraded to be flood resilient} \\ 0 & \text{else} \end{cases}$$

$$z_p = \begin{cases} 1 & \text{If path } p \text{ is flood resilient} \\ 0 & \text{else} \end{cases}$$

Model 4 The FARNUP where the path must be pre-generated. This model was the final model for the problem.

3.1.2 Path generating model, inspired by Magnanti and Wongs model

The alternative to the former model, is a model that generates the paths during the optimization. This idea stems from the model formulated by Magnanti and Wong (Model 2). The model we have formulated, uses the formulation of Magnanti and Wong to generate paths for O-D pairs within the optimization process. The generating of these paths is captured in the decision variables $f_{ij}^{(n)}$. How this works exactly will be explained after introducing the model.

In Model 5 the formulation is presented. Each formula will be accompanied by an explanation of its meaning. The definition of the input and decision variables can be found below.

Constraint 1.1-1.3 ensure a path for any O-D-pair is made up of connected edges. They correspond to constraint 1 of Model 3, which is why they are bundled as a part of constraint 1. This works as follows: any node $k \in N$ either lies on a path or it does not. If it does not, it has no edges of the path entering nor leaving it. If node k does lie on a path, but is not the household cluster n nor a healthcare facility in HCF_h, it will have one edge entering and one edge leaving. Say edge (j, k) is entering, and (k, l) is leaving. Then $f_{jk}^{(n)} = 1$ and $f_{kl}^{(n)} = 1$. Therefore, $f_{jk}^{(n)} - f_{kl}^{(n)} = 0$.

Now, if the household cluster we are analyzing is n, there will be no edge of the path entering n, and only one leaving. In mathematical terms, $f_{nk}^{(n)} = 1$ for some $k \in N$ and $f_{jn}^{(n)} = 0$ for all $j \in N$. Therefore, $f_{jh}^{(n)} - f_{hk}^{(n)} = -1$. This argument works analogously for the case where the node $k \in N$ is an element of HCF_h.

Lastly, we can formulate constraint 1.3 in the way we do because we are in a fortunate situation. All the healthcare facilities only have one edge connected to them: the edge that connects them to the road. Therefore, a path can not pas through a healthcare facility node. This is why we can simply exclude the set of healthcare nodes from the set of nodes that a path can pass through.

Constraint 2 bounds the path distance to 5 kilometers and constraint 3 bounds the construction costs. Constraint 4 could alternatively have been written as $f_{ij}^{(n))} \leq x_{ij}$. But writing it as above, reduces the number of constraints of the model from $|N_r||E|$ to $|E|$. This works equally well when formulated as $\sum_{n \in N_r} f_{ij}^{(n)} \leq |N_r|$, which also ensures a route cannot contain edge (i, j) if this edge has is not upgraded (in model terms, if $x_{ij} = 1$).

Another remark will be on constraint 5. This constraint ensures that a household cluster can be counted in the objective function. Because we are maximising over these y_n's, it is not necessary to ensure that if a path exists, y_n must equal 1.

The last remark will be on the fact that this model does not ensure that all household that are considered to be connected, actually are connected. Because this model can not take into account the distance from each household to the node on the road. Therefore, the number of households counted per path are the number of households that access the road at a specific node, not the households that have a traveling distance of 5 kilometers in total. This is also a weak aspect of this model.

$$\textbf{max} \sum_{n \in N} S_n y_n \qquad \text{Maximize the number of connected households.} \tag{0}$$

$$\textbf{s.t.} \quad \sum_{j \in N} f_{jn}^{(n)} - \sum_{i \in N} f_{ni}^{(n)} \geq -1, \quad \forall n \in N_r \qquad \text{The path for household cluster } n \text{ can only leave the cluster } n \text{ once and at most once.} \tag{1.1}$$

$$\sum_{n_{hcf} \in N_{hcf,n}} \left(\sum_{j \in N} f_{jn_{hcf}}^{(n)} - \sum_{i \in N} f_{n_{hcf}i}^{(n)} \right) \leq 1, \quad \forall n \in N_r \qquad \text{The path for household cluster } n \text{ can only enter a healthcare facility once, at most once, and can not leave it. It can also only end at one healthcare facility.} \tag{1.2}$$

$$\sum_{j \in N} f_{jk}^{(n)} - \sum_{i \in N} f_{ki}^{(n)} = 0, \quad \forall k \in N \setminus \{N_{hcf,n}, n\}, n \in N_r \qquad \text{The path for household cluster } n \text{ must leave any node it enters, if this node is not } n \text{ or a healthcare facility in } N_{hcf,n} \tag{1.3}$$

$$\sum_{(i,j) \in E} f_{ij}^{(n)} l_{ij} \leq 5\text{km} \qquad \forall n \in N_r \qquad \text{All paths must be at most of 5 kilometer length.} \tag{2}$$

$$\sum_{(i,j) \in E} e_{ij} x_{ij} \leq B \qquad \text{Upgrade construction costs must be less or equal to the budget} \tag{3}$$

$$\sum_{n \in N_r} f_{ij}^{(n)} \leq |N_r| x_{ij} \qquad \forall (i,j) \in E \qquad \text{A path can only be on an edge that has been upgraded} \tag{4}$$

$$y_n \leq \sum_{j \in N} f_{jn}^{(n)} \qquad \forall n \in N_r \qquad \text{A household cluster is connected to a healthcare facility if there is a path going out of the node } n \tag{5}$$

$$f_{ij}^{(n)}, y_n, x_l \in \{0,1\} \forall (i,j), l \in E, n \in N \qquad \text{Integer constraints} \tag{6}$$

Input

B	Budget
e_{ij}	Costs of updating edge (i, j) to be all-seasons
E	Edge set
l_{ij}	Kilometer length of edge (i, j)
$n \in N$	Node in node set N
$N_{hcf} \subset N$	Subset of nodes containing all the nodes that represent a healthcare facility
$N_{hcf,n} \subset N_{hcf}$	Subset of nodes representing healthcare facilities that are within 5 kilometer range of node n
$N_r \subset N$	Subsets of nodes that connect at least one household to the road
S_n	Number of households connected to road-node n

Decision variables

$$f_{ij}^{(n)} = \begin{cases} 1 & \text{if edge } (i,j) \text{ is a part of the flood resilient path of household cluster } n \\ 0 & \text{else} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if edge } (i,j) \text{ is (upgraded to) a flood resilient road} \\ 0 & \text{else} \end{cases}$$

$$y_n = \begin{cases} 1 & \text{if household cluster } n \in N_r \text{ is connected} \\ 0 & \text{else} \end{cases}$$

Model 5 Magnanti and Wong inspired formulation for the Road Network Upgrading Problem.

3.1.3 Additional and alternative formulations

Alternative objective functions

- **Maximize the number of connected households while minimizing of construction costs**

Formulation for Model 4	$\sum_{p \in P} S_p z_p - a\alpha \sum_{l \in E} e_l x_l$
Formulation for Model 5	$\sum_{n \in N_r} y_n - \alpha \sum_{l \in E} e_l x_l$

  This objective function penalizes expensive links and thus makes a stronger trade-off between how many households are connected and how much this costs. This objective function is very useful when the budget does not need to be spent entirely, but the relevance of the solution needs to be very high. The factor α weights the costs, which can help balance out the importance of the number of households versus minimizing the costs. A smaller α will enforce more prioritization on connecting as many households as possible, while a larger α will weigh the connecting a new household alongside how much it costs. One issue to look out for when using this objective function is that it might not add routes to household clusters that can be afforded, because they are relatively very expensive.

- **Maximize the number of connected households while minimizing the traveling distances of the paths**

Formulation for Model 4	$\sum_{p \in P} z_p (S_p - \alpha l_p)$
Formulation for Model 5	$\sum_{n \in N_r} y_n - \alpha \sum_{n \in N_r} \sum_{(i,j) \in E} f_{ij}^{(n)} l_{ij}$

  This objective function not only maximizes the number of connected households, but also tries to minimize the distance of each included path (the length of a path is what variable l_p stands for). Routes will still be no longer than 5 kilometers, but the solution with the shortest total traveling distance will be selected. As is also the case for the above objective function, the factor α serves to balance these two objectives. A smaller constant α will prioritize maximizing the number of connected households and a larger constant α will prioritize the shortening of the traveling distances. One negative effect of this objective function could be that might prioritize to connect households that are closer to the healthcare facility because it penalizes longer links and routes, resulting in households that already have difficulty with accessibility. Or, even worse, not include any paths because this results in a smaller total traveling distance.

Alternative objective functions

- **Setting a minimum percentage of households that must be connected**

  Formulation for Model 4 $\frac{1}{\sum_{n \in N_r} S_n} \sum_{p \in P} S_p z_p \geq PC$
  Formulation for Model 5 $\frac{1}{\sum_{n \in N_r} S_n} \sum_{n \in N_r} S_n y_n \geq PC$

  This constraint will ensure that at least $PC\%$ of the households are connected. This constraint does not add anything if the objective function only maximizes the number of connected households, because either the model already connects at least $PC\%$ of the households, or the program is infeasible. Only when incorporating a distance minimization or cost minimization to the model, could this constraint be of use (if the percentage is feasible).

3.2 What model to use

The decision whether to generate the paths during or before the model is optimized can have a big effect on the computational results. It could affect the running time, the quality of the solution and the memory use. The hypothesis is that generating the paths during the optimization takes a long time to run and uses a lot of memory.

For both models, the branch and bound algorithm (as implemented by Gurobi) was applied on a small scale scenario. When the branch and bound algorithm was applied to Model 5, it was not able to properly run and crashed, because there was not enough storage on the laptop available. This was not an issue for Model 4 (regardless of how the paths were generated). Proving the hypothesis that generating the paths before hand would be computationally much more favorable. This could be explained because there is a much larger set of variables and constraints to optimise over. In some cases this leads to a much shorter running time (these scenarios are called extended formulation [15]), but this is not the case with our models.

The differences in the number of variables are the following. For Model 5 there are

$$\underbrace{2|N_r||E|}_{\text{number of } f \text{ variables}} + \underbrace{|E|}_{\text{number of } x \text{ variables}} + \underbrace{|N_r|}_{\text{number of } y \text{ variables}}$$

variables.

While the number of variables for x and y are quite trivial, the number of variables for f could use some explanation. The number of variables for f is $2|N_r||E|$ because every household cluster will have $|E|$ links to find a path over. But, because the direction of these edges matters for the path, resulting in a total of $2|N_r||E|$ decision variables

For Model 4, the model that takes a set of paths as input variable, we only need

$$\underbrace{|P|}_{\text{number of } z \text{ variables}} + \underbrace{|E^*|}_{\text{number of } x \text{ variables}}$$

variables.

We can assume that $|P| \leq 2|N_r||E| + |N_r|$ because, in our case, we know that for every household cluster there are less than $|E - 1|$ paths of at most 5 kilometers long.

Besides, Model 4 has less constraints than Model 5. For Model 4 we have

$$\underbrace{1}_{\text{constraint 1}} + \underbrace{\sum_{p \in P} |p \cap E^*|}_{\text{constraint 2}} + \underbrace{|N_r|}_{\text{constraint 3}}$$

constraints. While for Model 5 we have

$$\underbrace{|N_r|}_{\text{constr 1.1}} + \underbrace{|N_r|}_{\text{constr 1.2}} + \underbrace{|N_r|(|N| - 1 - |N_{hcf}|)}_{\text{constr 1.3}} + \underbrace{|N_r|}_{\text{constr 2}} + \underbrace{1}_{\text{constr 3}} + \underbrace{|E|}_{\text{constr 4}} + \underbrace{|N_r|}_{\text{constr 5}}$$
$$= |N_r|(|N| - |N_{hcf}| + 3) + |E| + 1$$

constraints. Provided our tests, and the possible explanation as discussed above, we can conclude that Model 4 is a more efficient choice.

It must be noted that generating a set of paths of at most 5 kilometer long for each household beforehand can cost a lot of memory as it must save the all the edges on the path and generating the paths will cost some computational time as well. How much computational time this saves is dependent on how these algorithms work. This will be discussed in Chapter 5.

4 Data preprocessing

This chapter will elaborate on how the data was prepared in order to be able to apply the FAR-NUP model (Model 4) to Timor-Leste. The model needs a broad set of input data. This data is:

- A road set, expressed as a graph with nodes and edges;
- A flood analysis (with binary indications) for each road segment;
- An upgrading cost indication for each road segment;
- A set of possible paths of at most 5 kilometers long each;
- An indication of how many households can be connected for each path.

In order to obtain the needed input variables, we need to model the flood risk and the upgrading costs. In order to generate paths, we need to identify the households that are affected by floods. Therefore, we need an accessibility analysis for the entire nation. This chapter will focus on the flood risk model, the cost model and the accessibility analysis. The generation of paths will be discussed in Chapter 5 because this was a difficult aspect of the research that deserves a thorough explanation.

We will first present the data we have worked with, what the sources are of this data and some observations we have made of our data. Afterwards, we will discuss the flood model and the cost model. Afterwards, we describe how we analyzed the healthcare accessibility.

What is important to emphasize about the flood and the cost model is that modeling flood risks and upgrading costs is a very difficult task that requires thorough, specialised research [14]. Simple dummy models have been be developed in order to test the optimization models, because we are a team of mathematicians with no background in infrastructure and because it is essential in order to apply and test our optimization models. The optimization models have been designed to be generic such that the flood and cost models can be expanded upon or replaced.

4.1 The used data

In order to apply the optimization model and all the other models to Timor-Leste, we needed data from this area. This data was provided by The World Bank. These were 5 sets of data:

- Topographic data for Timor-Leste.
  Source: The World Bank;
- Geospatial road data for Timor-Leste.
  Sources: eStrada and OSM, combined via an algorithm created by Valentijn Stienen. The paper has yet to be published;

- Geospatial household data from Timor-Leste.
  Source: Census;
- Geospatial healthcare data from Timor-Leste.
  Source: World Health Organisation;
- Geospatial flood hazard data for Timor-Leste.
  Source: Fathom Flood Risk Intelligence.

4.1.1 Observations of the Timor-Leste data

This subsection will get into some oberservations we have made about our data. For the road, household and healthcare facility dataset we will show the spread per district, and visualise the data. Afterwards some details about the flooding data is shown.

In Figure 2 we see the different districts of Timor-Leste.

Figure 2 Districts of Timor-Leste (and the island of Atáuro that is a part of the district of Dili).

There were two sources of road data: eStrada and OSM. We used a dataset that mapmatched these two data sets, because the symmetric differences between these roadsets was very large. The paper on how this mapmatch algorithm worked exactly is still being written by Valentijn Stienen from Tilburg University.

The road data has been split into segments of 50 meters. This is done because it makes it easier to identify the flooding risks on parts of a road and in order to be able to connect households more realistically to the road.

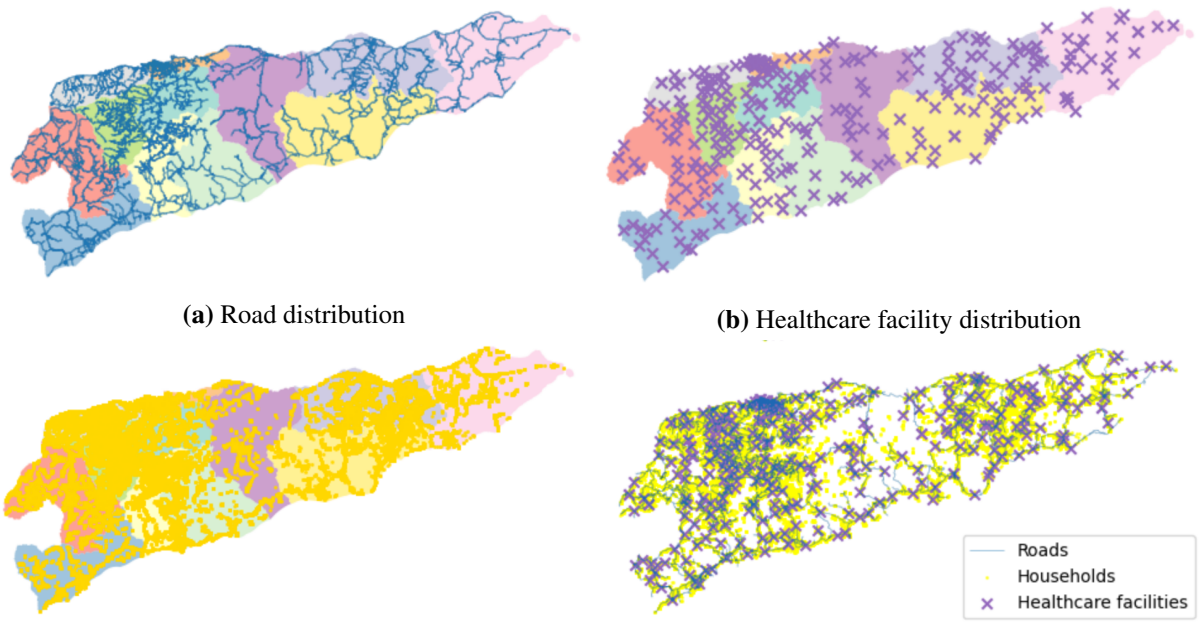
Because the district of Oescussi and the island of Ataúro are not present in the Fathom data, and are not areas of interest for the World Bank at the moment, they are excluded from all the other

data sets as well. But if these areas were to be included, the models can easily be applied to these areas separately because their road networks are isolated from the rest of the Timor-Leste data.

The data of the healthcare facilities comes from the World Health Organization and the data of the household distribution comes from Census Bureau. In Table 1 the most important statistics of in our data are shown for each district separately. In Figure 3, the distribution of these datasets is visualised.

Province	**Kilometers of road**	**Number of healthcare facilities**	**Percentage of total population**
Aileu	12194km	23	4.2%
Ainaro	13030km	19	5.77%
Baucau	16363km	48	12.37%
Bobonaro	18109km	30	10.12%
Covalima	15323km	21	6.83%
Dili	13465km	22	18.14%
Ermera	16088km	31	12.01%
Lautém	12071km	26	6.73%
Liquiça	13074km	28	6.49%
Manatuto	10665km	24	4.16%
Manufahi	9205km	22	5.06%
Viqueque	12437km	27	8.13%
Total	162031km	321	100%

Table 1 Distribution of roads (in kilometers), healthcare facilities (in numbers) and population distribution (in percentage of total population).



(a) Road distribution

(b) Healthcare facility distribution

(c) Household distribution

(d) All together

Figure 3 Visualisation of the road data, healthcare facility data and household data. Shown separately and all together.

4.1.2 Flooding data

The flooding data was provided by Fathom [16]. The Fathom flood-hazard model is a global gridded dataset of flood hazard produced at the global scale. It provides pluvial and fluvial hazard scenarios, expressed in *return periods*, which indicates the probability of occurrence (i.e. once in 5, 10, 20, 50, 75, 100, 200, 250, 500, 750 and 1000 years).

Each country set includes three subsets:

- *Fluvial Undefended* (FU): fluvial floods (floods from rivers, lakes or streams) hazard data, without defence estimation;
- *Fluvial Defended* (FD): fluvial flood hazard data, with defence estimation;
- *Pluvial* (P): pluvial flood (floods from percipitation) hazard data.

The defended version of the fluvial hazard maps accounts for the effect of flood defense measures in lowering the hazard intensity; Fathom notes that this is based on a statistical estimate of flood protection standards (FloPros) and does not account for the presence of physical structures (e.g. dikes, barriers). The undefended version is recommended for general risk assessment purpose.

For our purposes we have used the fluvial defended and the pluvial layer with a 1 in 500 years return period. We chose to use the defended layer rather than the undefended layer, because this is not a general risk assessment. A visualisation of these layers can be found in Figure 4.

(a) Fluvial Defended

(b) Pluvial

Figure 4 Visualisation of 1 in 500 flooding risk data. Data has been partly rounded in order to make the visualisation clearer.

4.2 Flood model

The flood model aims to identify whether a road segment is at risk of being inaccessible due to floods. In order to identify this, the model takes two aspects of a road segment into account:

(1) Flooding risk on segment (sum of fluvial and pluvial both with a return period of 1 in 500 years, the highest value on the segment);
(2) Surface type of the road segment;

The model assumes that there are three types of road surfaces: dirt, gravel and asphalt. The assumption is that dirt is less flood resilient than gravel, and that asphalt is always flood resilient. Every road surface type has a threshold under which a road segment is assumed not to be at risk of heavy flooding. These thresholds can be found in Table 2. These values are selected

randomly. Thus, for example, a gravel road with a flooding risk of 0.75 is at risk, but a dirt road with a flood risk of 0.12 is not.

Road type	Flood risk threshold
Dirt	0.15
Gravel	0.25
Asphalt	Always flood resilient

Table 2 The chosen flood risk thresholds for a road surface type. If the flood risk quantity on a road segment (which is a sum of the Fathom fluvial and pluvial quantities) surpasses this threshold, a segment is considered to be at risk, and thus in need of an upgrade in order to be all-seasons accessible.

4.2.1 Constructing road surface data

It was necessary to construct road surface data because only 10% of the mapmatched road data contained an indication of a road surface type. Because there were 11 types of road surfaces among this labeled dataset, they were divided into three groups and assigned to be either asphalt, gravel or dirt. For the remaining 90% of the unmarked data, 78% had an highway indication. This indication was used to generate a surface type. The remaining 22% was generated randomly. For this generation, it was ensured that the whole road was assigned the same surface type, rather than that different segments on a road had different surface types. This was done because roads are most likely to be entirely of the same surface type, rather than segments of different surface types.

The enumeration below shows how different indications were mapped to the three used road surface types.

(1) Simplify the 10% of the data that does have a surface type assigned. The mapping is as follows:

final surface type	OSM surface type classification
asphalt	asphalt, paved, concrete
gravel	gravel
dirt	dirt, compacted, Travessa De Ai-Kakau, sand, mud, ground, unpaved

(2) Map the data that has not yet been labeled, that has an indication for the column *highway* as follows:

final surface type	OSM highway classification
asphalt	primary, primary_link, motorway, motorway_link
gravel	service, secondary, secondary_link, steps, tertiary
dirt	unclassified, residential, footway, construction, tertiary_link, living_street

(3) For the remaining data that has not yet been labeled by the past two steps, assigning a surface type was just done randomly.

4.2.2 Results flood model

When this model was applied to Timor-Leste, 22% of the roads appeared to be at risk of being flooded. The results are visualised in Figure 5, the quantitative results for each district can be found in Table 3. What is important to keep in mind about this model is that it is a dummy model and thus the result does not portray the actual flooding risks on the road well.

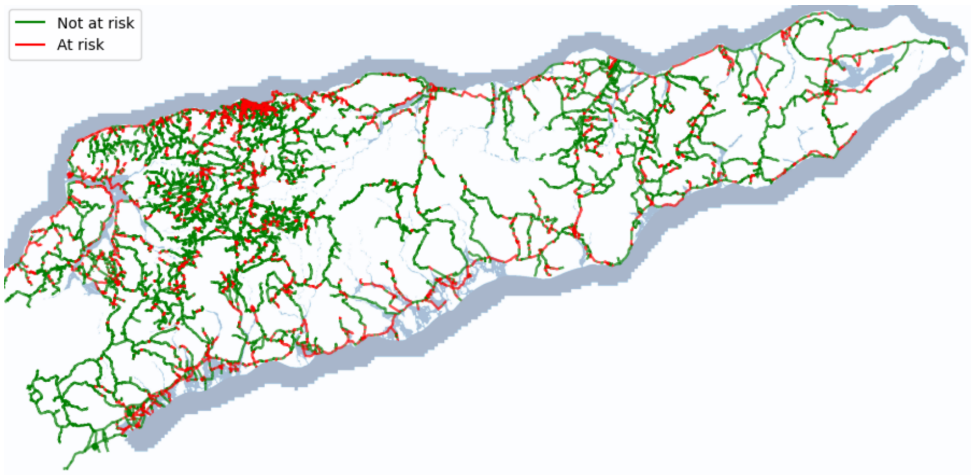


Figure 5 Visualisation of the results of the flood model

Region	Kilometers at risk	Percentage of district
Ainaro	94.3km	15.29%
Aileu	76.86km	13.49%
Baucau	119.11km	15.47%
Bobonaro	184.51km	21.62%
Covalima	152.04km	21.8%
Dili	245.46km	41.88%
Ermera	111.76km	14.44%
Liquiça	171.39km	28.25%
Lautém	135.76km	23.23%
Manufahi	146.8km	32.71%
Manatuto	132.91km	25.53%
Viqueque	143.12km	23.6%
Total or average	7570.25km	22.46%

Table 3 The results of the flood model per region. For each district, this table sets out how many kilometers of road are at risk and the percentage of all the roads in this region are at risk.

4.3 Cost model

The cost model calculates per road segment how much it will cost to make the segment flood resilient. This is dependent upon the flood model, because the flood model shows what is needed for a link to become flood resilient. The costs of upgrading a segment are assumed to be dependent upon the length of the segment and the upgrade that is needed. Each type of upgrade has a cost per kilometer. The costs of this upgrade per kilometer are then multiplied with the length. There are three types of upgrades possible, they are listed in Table 4.

Upgrade type	Cost per km
Dirt to gravel	2
Dirt to asphalt	15
Gravel to asphalt	12

Table 4 The different types of upgrades for a road segment and the costs per kilometer.

The formula for the costs is as follows:

$$cost_e = upgrade\_type \times length_e$$

4.3.1 Results cost model

A visualisation of the costs can be found Figure 6. The results per province are shown in Table 5.

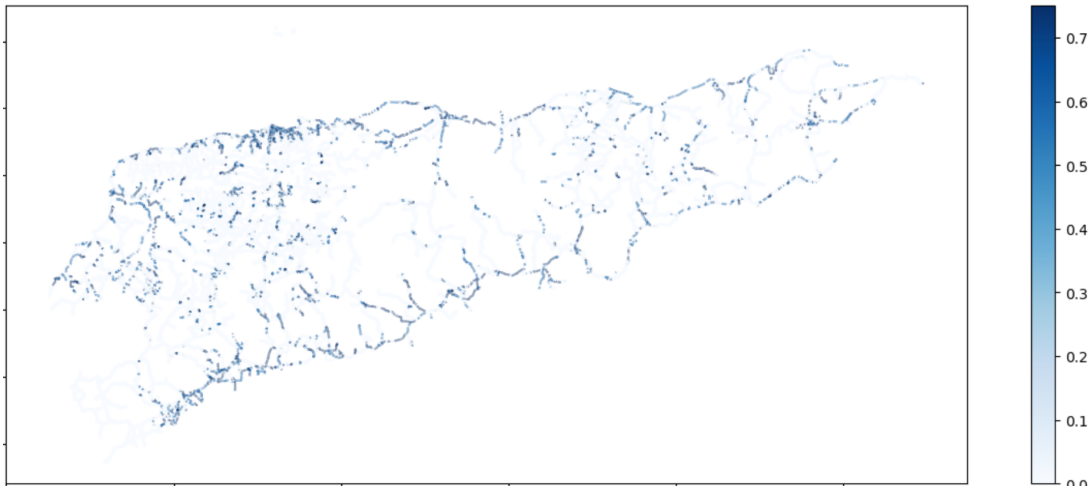


Figure 6 Visual result of upgrading costs on edges

Region	Km road	Upgrading costs	Percentage of total costs
Aileu	567.56	976.61	4.54%
Ainaro	614.47	1200.06	5.58%
Baucau	767.43	1366.11	6.35%
Bobonaro	852.49	2253.05	10.48%
Covalima	697.37	1895.09	8.81%
Dili	576.74	3155.94	14.68%
Ermera	773.02	1375.76	6.4%
Lautém	584.33	1694.61	7.88%
Liquiça	606.69	2192.91	10.2%
Manatuto	520.26	1697.08	7.89%
Manufahi	448.61	1908.63	8.88%
Viqueque	606.44	1784.75	8.3%
Total	7615.71	21502.4	100.0%

Table 5 Results of upgrading costs per area

4.4 Accessibility model

The accessibility analysis aims to analyze which households are connected to a healthcare facility and which are not. Our model can do this with and without taking into account the flooding risk. For the convenience of the reader we will repeat the following definition: a household is

considered to be *connected* if there is a path from a household to a healthcare facility via a flood resilient route that is no longer than 5 kilometers.

In order to efficiently compute the accessibility statistics, households are clustered according to where they access the road. Because the road has been split into segments, households can be attached to the ends of these segments. If a healthcare facility is closer to a household than a road, the household is mapped to the healthcare facility. An example of how these connections and clusters are made can be seen in Figure 9. This clustering is done because otherwise the same route would be computed for every household in the cluster separately, which costs a lot of computational time. The distance from a household to the road can be bounded, which means that if a household does not live within an x kilometer range of any node on a road, it will not be attached to the road. In our case, we have set the maximum distance from a household to the road of at most 5 kilometers.

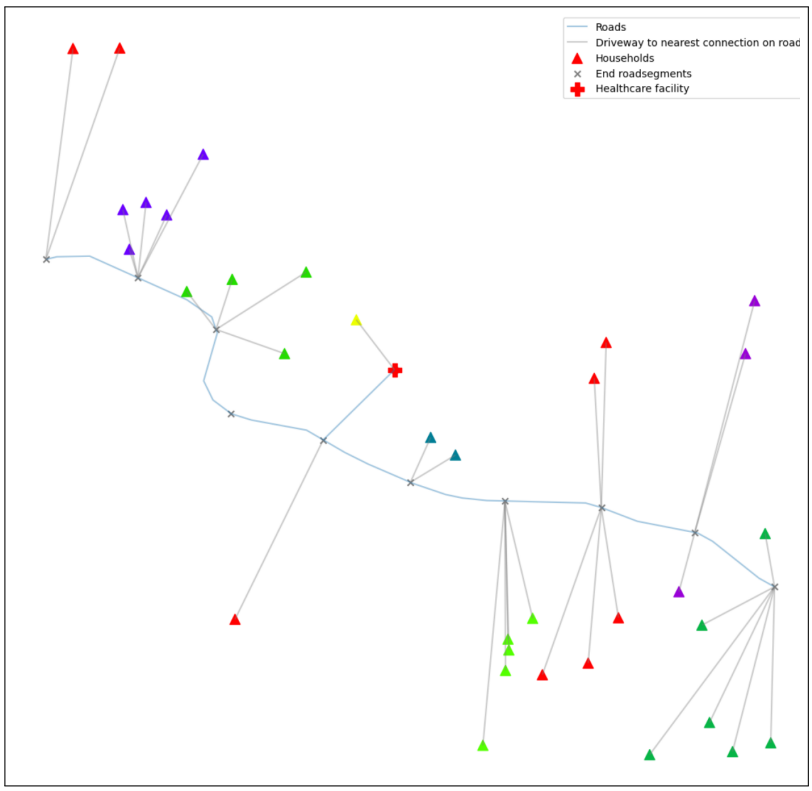


Figure 7 An example of how households cluster according to how they access the road.

The shortest path could only be calculated from one node to one other node. Therefore, we could not just calculate one shortest path from a household cluster to set of healthcare facilities. It was necessary to calculate the shortest distance to each individual healthcare facility, and then save the shortest distance of the found distances. In order to speed up these computations, the shortest paths for a household cluster are only calculated towards the healthcare facilities that are within a 5 kilometer radius. The shortest path from a cluster to any of the healthcare facilities is then saved.

After the shortest traveling distance from a household cluster to any healthcare facility is found, the shortest traveling distance can be attributed to the individual households within the household cluster. For this, the distance from the household to the node on the road is added to the found shortest distance of the road node to the healthcare facility.

The distinction between the analysis that takes into account the flood risk and the analysis that does not, is that the at risk road segments are dropped from the road data after the households have been clustered according to where they access the road. When it comes to analyzing the flood affected accessibility, the households are first assigned to the nearest road node (or healthcare facility), no matter if the road is at risk or not. Once these households have been assigned, the road segments that are at risk of being flooded are dropped from the edge set. Over the remaining edges the shortest paths are being computed.

Once these shortest distances to a healthcare facility have been calculated, the households are split into a group having traveling distance of at most 5 kilometers, and a group having a traveling distance longer than 5 kilometers.

The accessibility analysis has been applied with and without taking into account the flooding risks. First we elaborate on the accessibility analysis regardless of flooding risks. Afterwards we discuss the accessibility taking into account the flooding risk.

4.4.1 Results accessibility analysis regardless of flooding risks

We find that a total of 79% of the households of Timor-Leste is able to travel to healthcare facility within 5 kilometers. The 5 kilometer accessibility potential is especially high in Dili, where 98% of the households is able to access a healthcare facility within 5 kilometer traveling distance. The areas where the accessibility is the lowest are Ainaro, Bobonaro and Ermera, where at most 67% is able to access a healthcare facility. The results can be found in Figure 8a.

4.4.2 Results accessibility analysis taking into account flooding risks

The accessibility analysis that takes into account the flooding risks shows that only 36% of all households in Timor-Leste can access a healthcare facility within 5 kilometer traveling distance via a flood resilient route. This is 45% of the households that are actually able to access a healthcare facility within 5 kilometer traveling distance (which 79% of all households).

We see that especially Dili is heavily affected by floods, as only 16.8% of the households that would be able to access a healthcare facility can access a healthcare facility during floods. The district that seems least affected by floods is Ermera, where 71% of the households that would be able to be connected, is. The results are shown in Figure 8b.

Region	Households	Households with access	Percentage with access
Ainaro	10064	6547	65.05%
Aileu	7339	5701	77.68%
Baucau	21584	17225	79.8%
Bobonaro	17670	11514	65.16%
Covalima	11917	9284	77.91%
Dili	31662	31085	98.18%
Ermera	20959	13936	66.49%
Liquiça	11320	8681	76.69%
Lautém	11739	9685	82.5%
Manufahi	8836	6762	76.53%
Manatuto	7256	5844	80.54%
Oecussi	0	0	0.0%
Viqueque	14198	10910	76.84%
Total	174545	137174	78.59%

(a) Accessibility analysis of households that can access a healthcare facility within 5 kilometer traveling distance when taking into account all edges, regardless of flood risk on edge.

Region	Households	Households connected	Percentage connected	Percentage connectable
Ainaro	10064	4181	41.54%	63.86%
Aileu	7339	3425	46.67%	60.1%
Baucau	21584	10079	46.7%	58.51%
Bobonaro	17670	5332	30.18%	46.31%
Covalima	11917	3666	30.76%	39.49%
Dili	31662	5208	16.45%	16.75%
Ermera	20959	9945	47.45%	71.36%
Liquiça	11320	3580	31.63%	41.24%
Lautém	11739	4957	42.23%	51.18%
Manufahi	8836	2498	28.27%	36.94%
Manatuto	7256	3544	48.84%	60.64%
Viqueque	14198	5804	40.88%	53.2%
Total	174545	62219	35.65%	45.36%

(b) Accessibility analysis of households that are connected when taking into account flooding risks.

Figure 8 Results of the accessibility analyses.

5 Generating paths

One of the most important input of the FARNUP model (Model 4) is the set of paths. The paths must be no longer than 5 kilometers long (but this parameter can be changed if desired). The paths should originate at a healthcare facility and terminate at a household cluster node (which is a node on the road). To remind the reader, a *household cluster* is a cluster of households that is clustered according to where they access the road. As stated before, the road dataset is split into road segments of at most 50 meters. A household can access the roads at to the ends of these segments. When households access the road at the same ends (or nodes) of a road segment, they are clustered. A visual example of this can be seen in Figure 9.

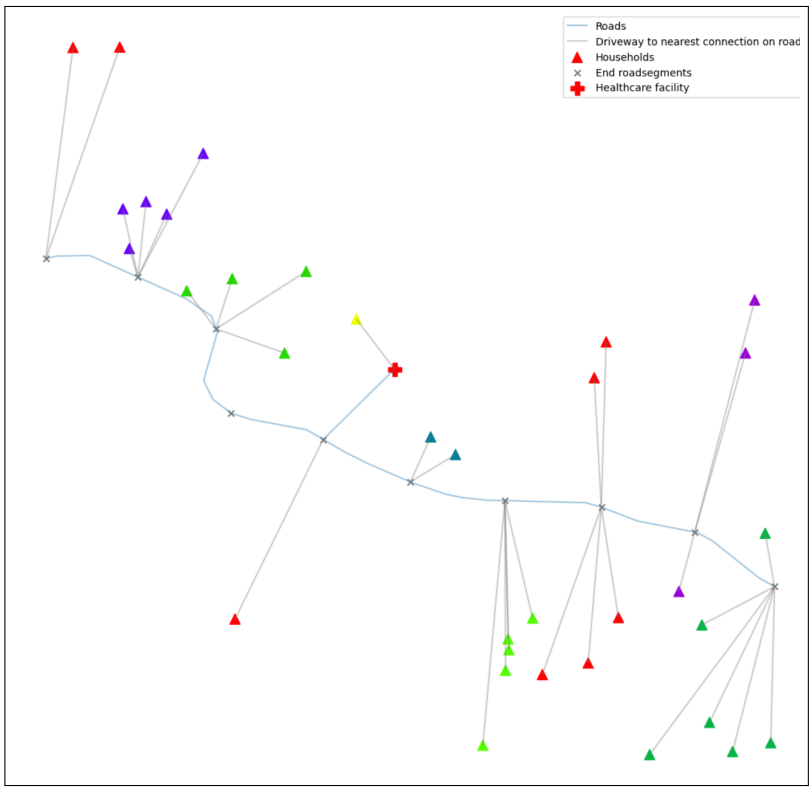


Figure 9 An example of how households cluster according to how they connect to the road. All households that connect to the same node on the road are clustered together. Each cluster in this image has its own color.

Two different algorithms were developed to generate a set of paths. Initially, an algorithm was created that would generate all possible paths using a state space search method. This algorithm is explained in Section 5.1. It works well for healthcare facility areas where the infrastructure is sparse, but for more complex healthcare facility areas it could take very long and will sometimes not terminate. Therefore, a new algorithm was created. This algorithm would generate only *relevant seeming paths*, which means that the path is a balance between short or cheap to upgrade. This algorithm will be explained in Section 5.2. In Section 5.3 the two methods are compared and the results are elaborated upon.

5.1 Algorithm to generate all paths

The algorithm that generates all paths is a *state space search algorithm* [17]. Such an algorithm makes use of an search tree and a set of active nodes that are are processed at every state in order to attain new (partial) solutions from earlier found (partial) solutions. One example of a famous state space search algorithm is the branch and bound algorithm.

In the case of our path generating algorithm, an active node is paths that is shorter than 5 kilometers that can still be expanded with road segments to create a longer path. The initial path starts off at the main healthcare facility. At every state, the algorithm chooses one active node (thus: a path) and seeks all the edges that are connected to the end node of the path. Every edge that is connected to the last visited node (whose end node is not yet visited via the path), generates a new path if it is added to the path that is being processed. The algorithm saves the newly found paths that are shorter than 5 kilometers and can still be expanded to the set of active nodes. If a path ends at a household cluster node, it is saved to the final set of paths. Thus, all final paths start at the central healthcare facility, end at a road node that has a household cluster attached to it and are no longer than 5 kilometers.

The algorithm uses a *depth first search*. In this context, that means that the newest path in the set of active paths will be the first path that will be expended in the next iteration(s). This method was chosen because it uses less memory because the set of active nodes will not expand to the maximum amount possible, before cutting off branches.

The pseudocode of this algorithm can be found in Listing 1.

A visual example of this algorithm can be found in Figure 10. Here we see how, at iteration 1, we generate all paths that start at the healthcare facility (which in our case are two paths). At the next iteration, we expand the southern path. The expansion of this path continues on to iteration 4, after which the path is 5 kilometer, and can not be expanded. It then continues on with the path to the south eastern side, because this was the last found path. In iteration 9, all 5 kilometer paths in this area have been found. Note that this example is not representative when it comes to segment length, segments are made longer in order to portray the example better.

Results

This algorithm works well for healthcare facility areas with a sparse infrastructure, but for more infrastructure dense healthcare facility areas it could take very long and would sometimes not terminate. Examples of a sparse and a dense scenario of a healthcare facility area can be found in Figure 11. In Section 5.3, the computational results of this algorithm will be elaborated on more extensively.

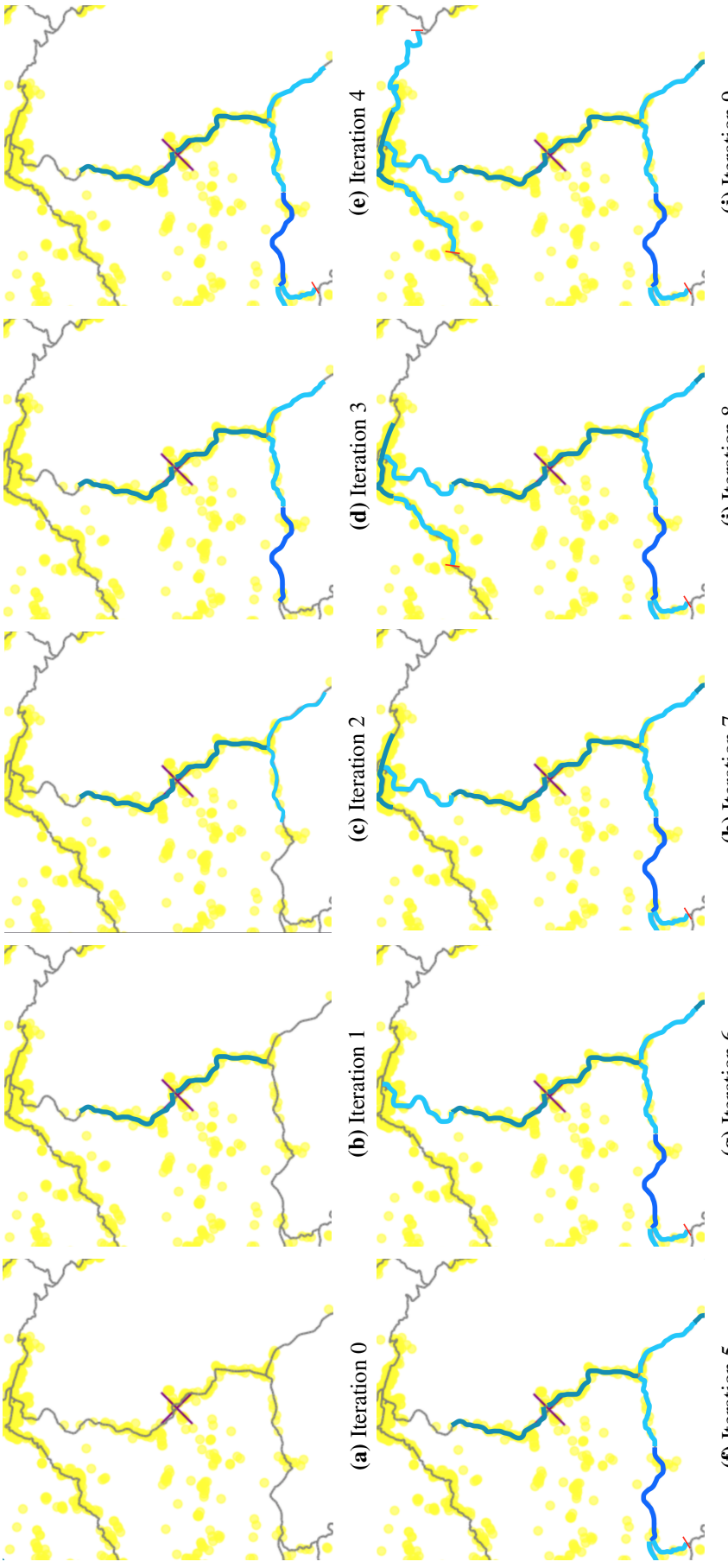
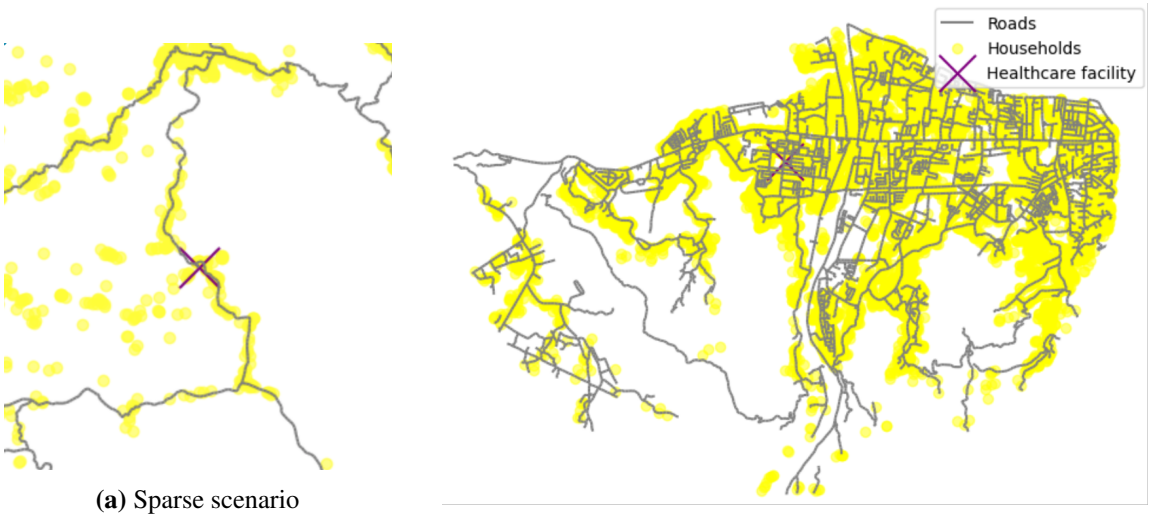


Figure 10 Visual example of how the path generation algorithm generates paths, applied to Figure 11a. The length of the segments is increased in order to make the amount of frames reasonable. The × in the figures symbolizes the healthcare facility.

```
final_paths = []
for hcf in healthcare_facilities:
    first_path = [main_hcf]
    active_paths = [first_path]
    while active_paths not empty:
        current_path = newest(active_paths)
        last_visited_node = last_visited_node(current_path)
        for all nodes n connected to last_visited_node:
            if n not in current_path and path_dist(current_path + n) <= 5:
                add (path + n) to active_paths
                if n household cluster node :
                    add (path + n) to final_paths
        remove current_path from active_paths
```

Listing 1 Pseudocode of algorithm that generates all possible 5 kilometer paths from all the healthcare facilities to all surrounding household clusters.



(a) Sparse scenario

(b) Dense scenario

Figure 11 Examples of a thin and a dense healthcare facility scenario

5.2 Algorithm to generate relevant paths

An algorithm that is less computationally demanding is the algorithm that generates only relevant seeming paths. A path is considered to be relevant when it is cheap to upgrade or has a short distance, or is a combination of both. Therefore, this algorithm generates K paths for each healthcare facility and any healthcare facility within a 5 kilometer radius (removing the duplicates). The edge weights are some convex combination of the length of the edge and the upgrading costs.

The algorithm calculates the shortest path using the Pandana shortest path package [18]. This package uses Dijkstra's algorithm [19] to find the shortest path and combines this with contraction hierarchies. *Contraction hierarchy methods* are a form of pre-processing a network such

that a distance matrix can be computed where junctions are ordered according to their cruciality in the network [20]. It then computes the shortest distance between the important junctions in order to create shortcuts such that it does not have to compute these distances every time it tries to find an individual shortest path.

The pseudocode of this algorithm can be found in Listing 2.

```
for k in K:
    λ = k/(K-1)
    edges.weights = λ * edges.upgrade_costs + (1-λ) * edges.km_length
    find shortest paths with above weights for all unconnected households
    calculate km_length for every path
    drop all paths longer than 5km
    add new paths to set of paths
drop all duplicate paths
```

Listing 2 Pseudocode of algorithm that potentially generates a smaller set of relevant paths from any node to the healthcare facility.

In order to establish which value for K would yield an effective balance between optimal results and low running time, the algorithm was ran for different values of K. From our empirical test, we concluded that the best choice is $K = 4$. The full analysis can be found in Section 7.4.

5.3 Comparing these algorithms

One algorithm has to be chosen for our final implementation. Therefore, we ran an empirical test to conclude which algorithm to chose. When comparing these two algorithms with each other, two performance aspects were taken into account: the computational time and the quality of the solution that was obtained from inputting these paths into the branch and bound optimization algorithm. It was applied to $n = 75$ arbitrary healthcare facility areas. The hypothesis is that the algorithm that generated all paths would generate better solutions (if it was able to terminate) and that the algorithm that generates the relevant paths would have a much shorter running time, both in terms of path generation running time and the optimization algorithm running time. The running time of the optimization algorithm would be shorter because there were less variables to optimize over.

The algorithms were compared using the branch and bound implementation. For $n = 75$ different healthcare facility areas both algorithms were applied, that each generated a path set for the area. The branch and bound algorithm was then ran twice, once for every path set. The running time and results were then saved. We set a MIP gap of 5% in order to make the total running time of this test shorter. Because the algorithm that generates all paths could sometimes run endlessly long, we bounded the running time of this algorithm to be at most 5 minutes. If it had not finished within 5 minutes, the algorithm would be cut off and another healthcare facility area would be selected. We used $K = 4$ for the number of paths to generate between each O-D pair.

The budget for each healthcare facility area was set to be 10% of the total cost of upgrading the entire area.

The hypothesis about how the algorithms would perform compared to each other appeared to be true, as we can see in Table 6. But there is still a good argument to be made in favor of the algorithm that generates the relevant paths. We see that the differences in computational times is much larger than the differences in objective values. The algorithm that generated all paths was on average 19 times slower than the relevant path algorithm, while there were barely any differences in the objective value.

	All paths	**Relevant paths**
Mean objective value	110.79	109.87
Mean path generation running time (sec)	20.46	2.08
Mean optimization running time (sec)	6.06	0.086
Mean number of paths found	444	63
Cases with same objective value		75.81 %
Cases where all path generation surpassed 5min		14.67%

Table 6 Quantitative results of path generation comparison test applied to $n = 75$ healthcare facility areas. The averages are taken over $n = 56$ cases that were able to generate all paths within 5 minutes. The number of path generated per O-D pair is $K = 4$.

IN Figure 12 and Figure 13 we have plotted the results of every single test (ordered according to the results of the all path method). We can see in Figure 12 that there are barely any differences between the objective values.

In Figure 13, we see that the running time (both to generate the paths and to optimize over them) of the all path generating algorithm is generally much larger than the relevant path generating algorithm.

Therefore, we can conclude that the relevant path heuristic is more useful than the heuristic that generates all paths.

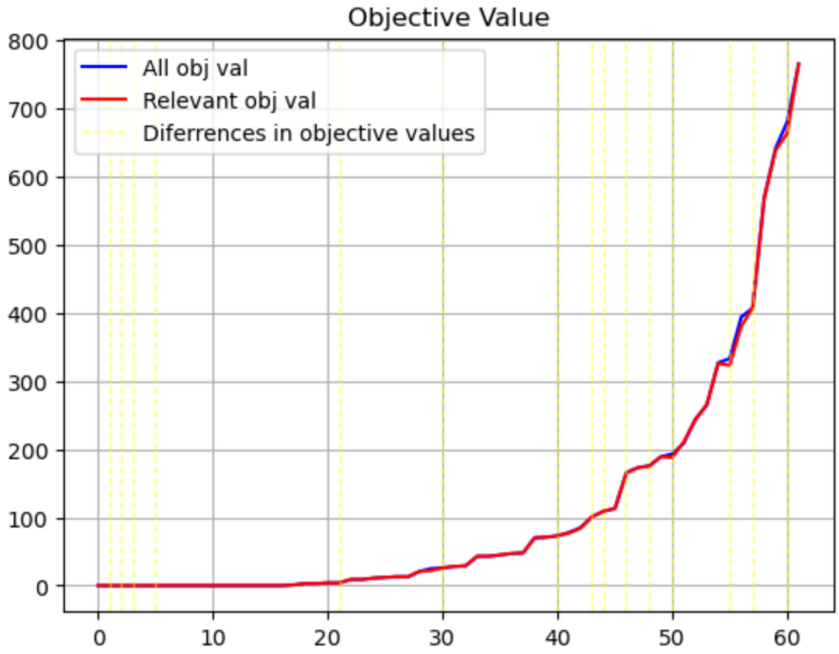
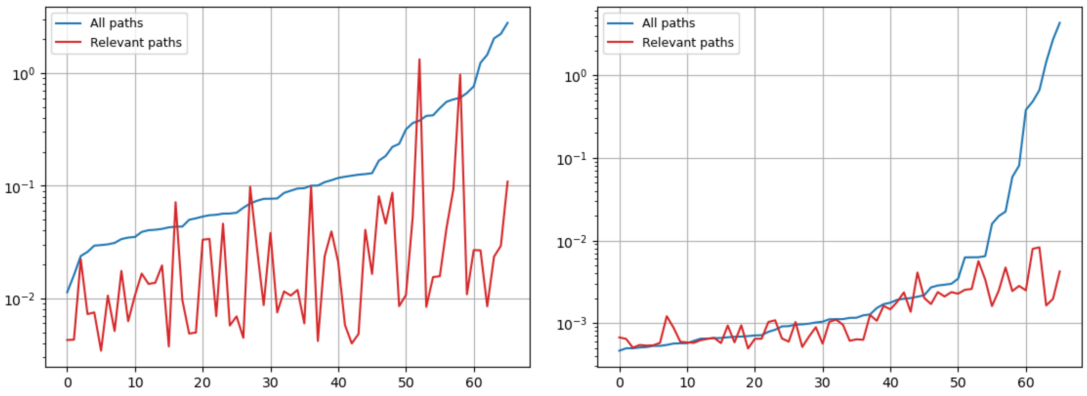


Figure 12 The (ordered) objective values for the two different path generation algorithms.



(a) Generating paths. **(b)** Optimizing the model (5% gap).

Figure 13 Running times between the two path generation algorithms (ordered).

6 Algorithms to find (near-)optimal solutions for Timor-Leste

Multiple algorithms have been developed in order to find a (near-)optimal solution for the Facility Accessiblity Road Network Upgrading Model (as formulated in Model 4). The test case for these algorithms was the country of Timor-Leste. This affected the development process, because many healthcare facilities in Timor-Leste have an overlapping healthcare facility area. Therefore, this is a factor that has been taken into account in all algorithms. We will start off with the analysis about this overlap.

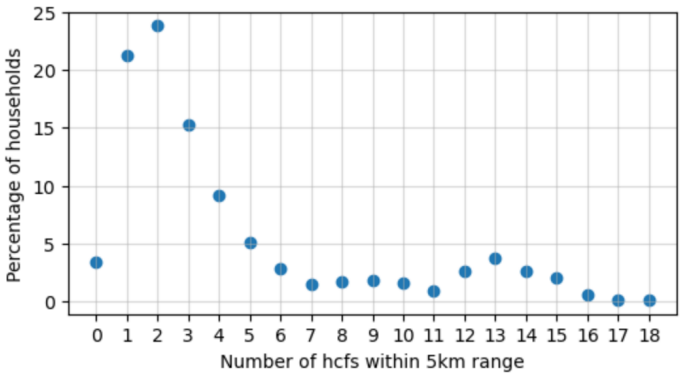
Afterwards, we will discuss the different algorithms that have been developed and tested. Due to the vast majority of papers recommending branch and bound, we started off applying and implementing that (using the Gurobi software). We were able to apply it to a small scale instances, these were the healthcare facility areas. The branch and bound algorithm was not applicable to the whole nation of Timor-Leste. Therefore, we needed to create heuristics that would be able to find a national solution for Timor-Leste. We started off creating heuristics that used the branch and bound algorithm applied to healthcare facility areas and combine them. We came up with two heuristics that did this, of which only one was able to produce a solution. Sadly, the algorithm that could produce a solution, did not satisfy our running time demands (a maximum of 12 hours). Therefore, a third method was developed that did not use the branch and bound method. This method is called the dynamic greedy method and it produces good solutions and satisfies our running time demands.

Taking into account overlap between healthcare facility areas

What is important to take into account when finding a national solution for the country of Timor-Leste, is that 75.4% of the households in Timor-Leste fall within multiple healthcare facility areas. Approximately 5% of the households even fall within 5 kilometer radius of more than 14 healthcare facilities. This data was acquired through an analysis of the Timor-Leste data. The quantitative results can be found in Figure 14.

Figure 15 shows the different healthcare facilities with the 5 kilometer radii around them. This figure also shows the households, and colors them according to how frequent they appear in healthcare facility area.

From this analysis we can conclude that it is important to take this overlap into account when finding local and national solutions.



(a) Plot frequency households live within number of healthcare facilities range

0	3.4%		
1	21.2%	**10**	1.58%
2	23.83%	**11**	0.92%
3	15.24%	**12**	2.58%
4	9.21%	**13**	3.76%
5	5.05%	**14**	2.56%
6	2.8%	**15**	2.08%
7	1.51%	**16**	0.62%
8	1.66%	**17**	0.12%
9	1.79%	**18**	0.09%

(b) Statistics

Figure 14 Results for overlap research of healthcare facility 5 kilometer radii

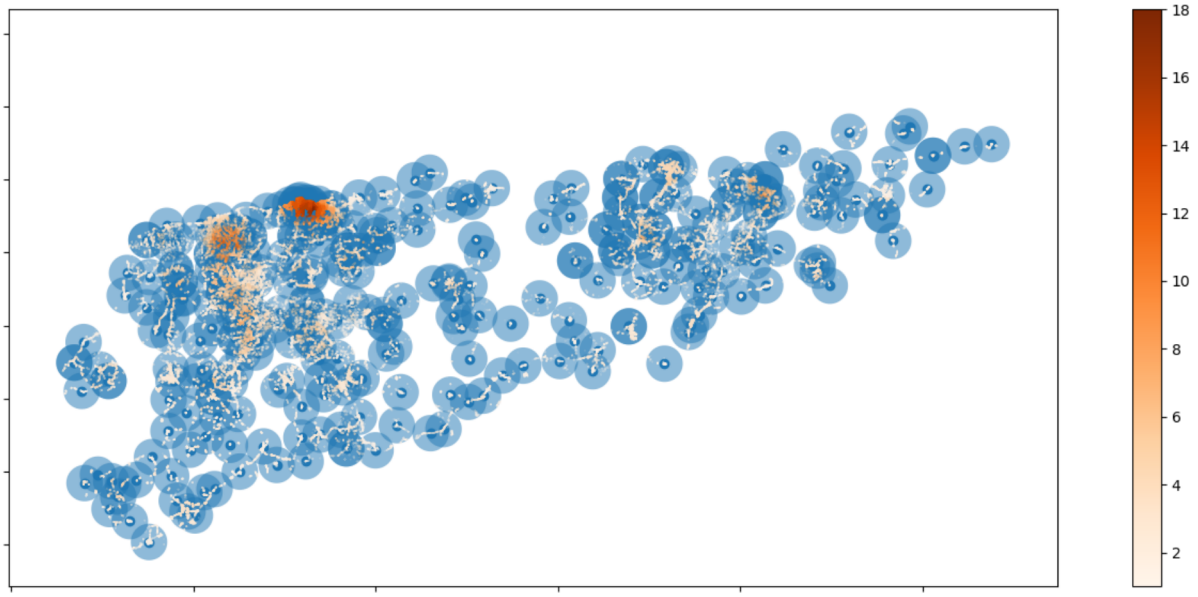


Figure 15 Visualisation of the healthcare facility areas. This image shows the overlap of these areas as well as the household distribution. The households are colored according to the frequency of which they fall within a healthcare facility area.

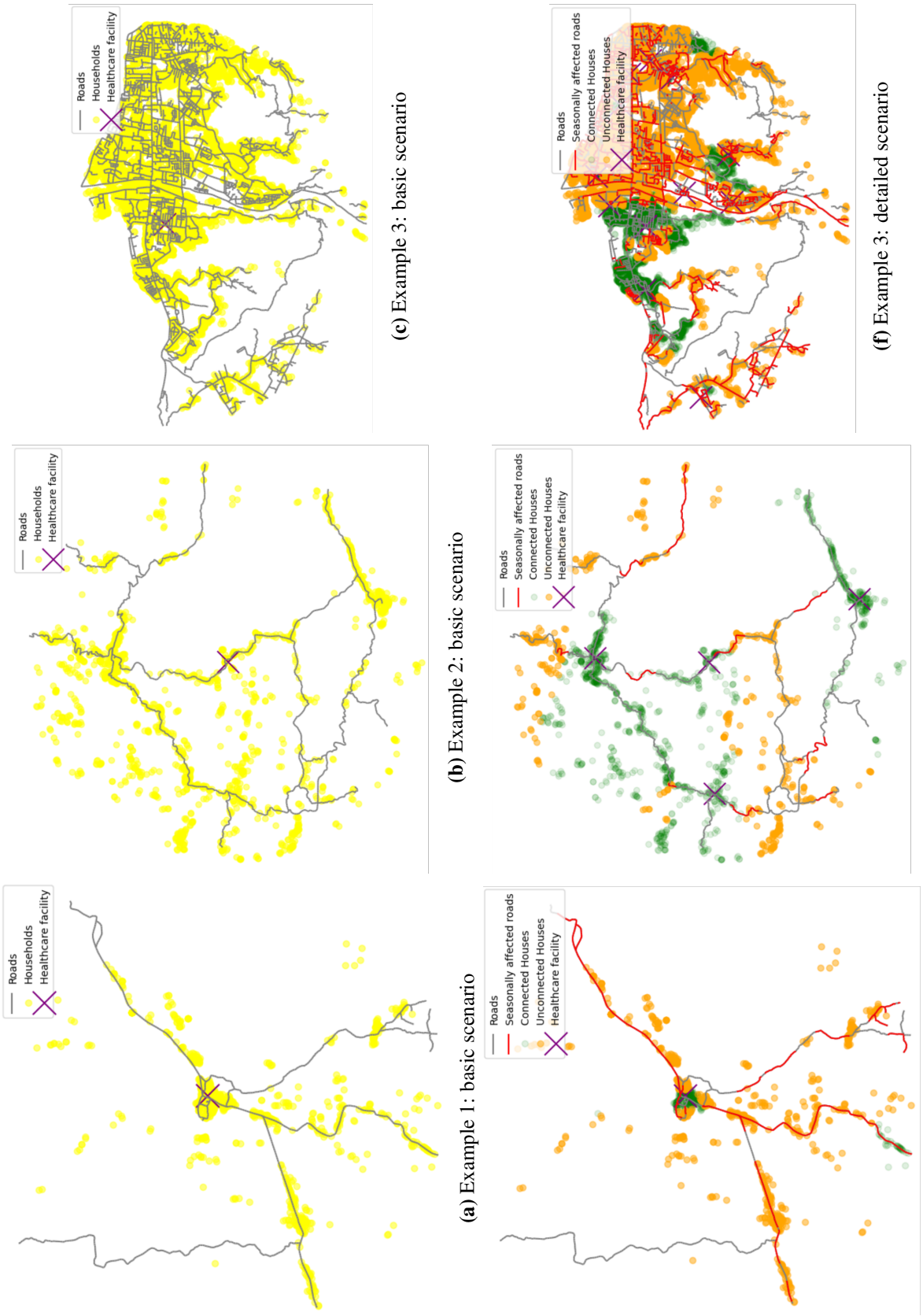
6.1 Branch and bound algorithm for small scale instances (healthcare facility areas)

We were able to apply the branch and bound algorithm to all healthcare facility areas. To remind the reader once more, this means a healthcare facility and the data that lies within a 5 kilometer radius around the healthcare facility. In order to provide the reader with some intuition for these healthcare facility areas, some examples can be found in Figure 16. This figure shows three healthcare facility areas. Each image is shown twice, above we see the areas with only the roads, households and the central healthcare facility and below we see this data alongside an indication of whether the households are connected and which segments of the road are at risk.

The branch and bound algorithm [15] is a algorithm that can find the optimal solution of an mixed integer optimization problem. It is a state space search method that creates a rooted tree in order to find the optimal solution. Suppose the problem is a maximization problem. At every iteration, the branch and bound algorithm solves an LP relaxation to find local upper bound and keeps account of the lowest bounds for the found integer solutions. Once this is done, the algorithm chooses a variable from the LP solution (usually according to some sort of prioritization of the variables) that is non integer, and splits the LP program into two new LP programs. To one program it adds the constraint that the selected variable must be larger or equal to the found decimal value, and to the other program it adds the constraint that the selected variable must be smaller or equal to the found decimal value. It saves the new LP programs to the set of active nodes, and updates the global optimum if the solution to the LP is integer. The branches of the tree are pruned due to three criteria: if the LP subproblem is infeasible, if the LP subproblem is integral or if the the lower bound obtained from the LP solution in a subproblem is less than or equal to the global lower bound. Therefore, the branch will not be further explored, because the optimal solution will not lie within this branch. The tree can be explored until an optimal solution is found, but, it can also be stopped according to some stopping criterion. Such a stopping criterion is generally either a certain gap percentage between the upper and lower bounds (the MIP gap) or a maximum running time.

We applied the branch and bound method using the Gurobi solver. Many choices could be made in order to optimize this implementation. The parameters and settings that were tested are summed up below. The full analysis can be found in Chapter 7.

- Can the LP solution with some postprocessing provide an optimal solution? *Conclusion*: no. Analysis can be found in Section 7.1;
- Can we relax the x or z variable by allowing them to take on real values and still attain a MIP answer and will it speed up the optimization process? *Conclusion*: no. Analysis can be found in Section 7.2;



(a) Example 1: basic scenario **(b)** Example 2: basic scenario **(c)** Example 3: basic scenario

(d) Example 1: detailed scenario **(e)** Example 2: detailed scenario **(f)** Example 3: detailed scenario

Figure 16 Three different examples of a healthcare facility areas. The first row shows the initial spatial data: the selected healthcare facility and all households and roads within 5 kilometer range of the healthcare facility. The second layer (d, e and f) adds to this the indication if a road is at risk and the distribution of connected and unconnected households.

- Which MIP gap yields an efficient trade-off between (near-)optimality and fast running time? *Conclusion*: 1%. Analysis can be found in Section 7.3;
- Could optimizing separately on grids be usefull? *Conclusion*: no. Analysis can be found in Section 7.6.

The frequency of certain cutting planes has also been examined, but it has not contributed to the final branch and bound implementation. The findings can be found in the Appendix, Section D.

6.1.1 Locally taking into account the overlap between healthcare facility areas

In order to find a more representative optimal solution on healthcare facility level, it was concluded that it was important to also take other healthcare facilities in the area into consideration. As explained above (Figure 14), 75,4% of the households live within a 5 kilometer radius of more than one healthcare facility. When looking more closely into this, it appeared that a solution for a healthcare facility level will not be representative for the area if we do not take into account the other healthcare facilities within the area.

This is best explained using an example. If we look at the healthcare facility area in Figure 17, we can see that there are three more healthcare facilities within this area. Now, if we were to only take into account at the healthcare facility in the centre, the most productive investment would be in road R1 (above the healthcare facility) and road R2 because this road segment connects the many households along R2. If the northern healthcare facility is taken into account, only R2 is updated in order to connect this group. Updating R1 to connect the 4 households alongside that road is one of the least interesting investments in this area. This example shows why taking into account the surrounding healthcare facilities in a healthcare facility area is important to get a more representative solution.

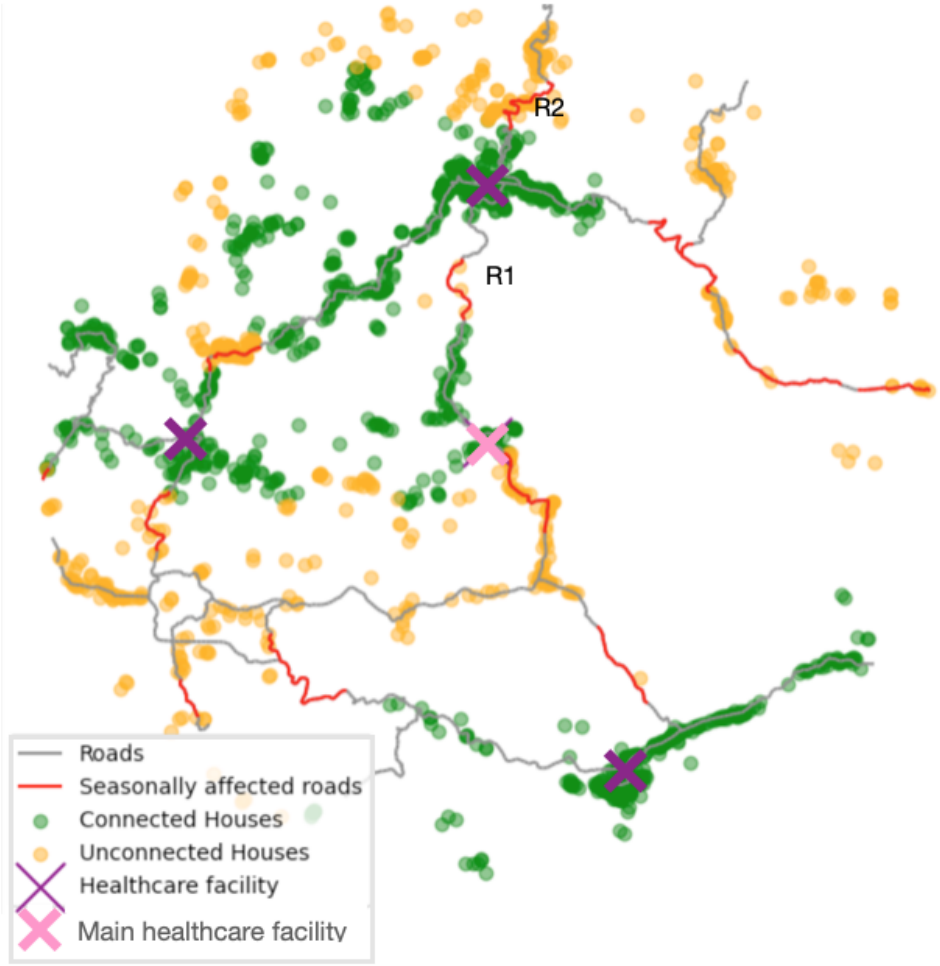
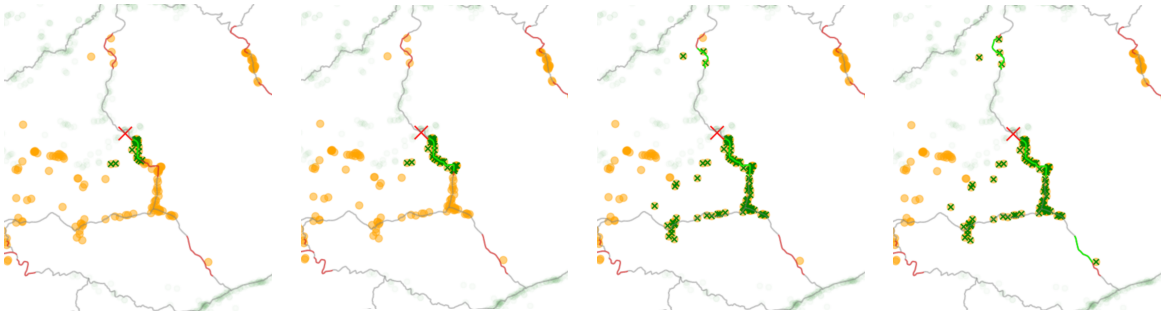


Figure 17 Example of healthcare facility level area where taking into account surrounding healthcare facilities provides a better solution.

6.1.2 Example of a local solution found by branch and bound

An example of different solutions for different budgets for a certain healthcare facility area are shown in Figure 18 (this is the same area as example 2 from Figure 16). We chose to include this to provide the reader with some intuition as to how these optimal investments changed as the budget increased. The three different budgets are 2.5%, 5%, 7.5% and 15% of the costs of upgrading the entire area. What we can clearly see in this figure is that the most viable area to invest in, is towards the south of the healthcare facility area. This holds because that is the area where the smallest budget makes the largest difference. We can see that as the budget increases, investing in the southern area stays prioritized, because at that spot the most households are dependent upon the cheapest upgrading costs. Only when that area has been upgraded, will the four households near the north of the healthcare facility to gain flood resilient access.



(a) 2,5% of area costs, connects 23 households more

(b) 5% of area costs, connects 38 households more

(c) 7,5% of area costs, connects 90 households more

(d) 15% of area costs, connects 92 households more

Figure 18 An example of optimal solutions for one healthcare facility area found by branch and bound. Three budget scenarios are shown, these budgets are a certain percentage of the total updating costs of the entire healthcare facility level.

6.2 Large scale heuristics that use branch and bound

This section will address how the model was applied to find a near-optimal solution for the entire nation of Timor-Leste using the branch and bound implementation for healthcare facility areas. The branch and bound algorithm could not be applied to the entire nation of Timor-Leste at once because this model would optimize over

$$\underbrace{36530}_{\textit{number of at risk road segments}} + \underbrace{103368}_{\textit{number of paths}} = 139898$$

variables. Therefore, a heuristic had to be invented that could efficiently find a near-optimal solution. We knew we could apply the model on a healthcare facility level. Therefore, two heuristics were developed to find a large scale solution that used this implementation for the healthcare facility areas of the branch and bound algorithm. The initial solution was the multiple budget scenarios knapsack method. Sadly, this method was infeasible due to the extremely long running time. Because of this, the pre-assign budget method was developed. This method was surely an improvement on the running time, but could not satisfy our desires. First, we will explain the multiple budget scenarios knapsack method and then we will discuss its successor, the pre-assign budget method.

6.2.1 The multiple budget scenario knapsack method

The idea behind the multiple budget scenario knapsack method is that it combines different local solutions to find a national solution. For each healthcare facility area it calculates the optimal solution for a few different budget scenarios. Afterwards, these solutions are combined using a multiple-choice knapsack model [21] as formulated in Model 7. This would be done using the branch and bound method again. The pseudo-code for this algorithm can be found in

Listing 3. One example of a possible list for $BUDGET\_PERCENTAGES$ is [50%, 30%, 20%, 15%, 10%, 7.5%, 5%].

```
#find a solution for every healthcare facility
for every healthcare facility h:
    demarcate roads, unconnected households and healthcare facilities
       within 5km radius of h
    cluster unconnected households based on where they access the road
    find all relevant paths between the healthcare facilities and the
       household clusters
    for budget_percentage in BUDGET_PERCENTAGES:
        budget = budget_percentage * total costs of updating all at risk
           roads in area
        find optimal solution for area given budget
        save:
         - budget
         - connected households (number and index of connected households)
         - upgraded roads (indexes)

#combine found solutions using a multi knapsack problem
combine solutions using a Gurobi implementation

*Apply local search technique to enhance global solution (not yet thought
   out)*
```

Listing 3 Pseudocode for multiple budget scenario knapsack method.

This method would need some postprocessing because, as we have seen before, the overlap of households and roads between healthcare facility areas is high. Because of this, a road segment could be upgraded multiple times in different solutions, and included multiple times. Since this would imply that certain investment are accounted for multiple times, the budget projections could be off and the multiple-choice knapsack solution could also be far off from the actual solution. Thus, a postprocessing heuristic would need to be invented.

Before we developed a local search technique, it was already clear that calculating all scenarios was far beyond our computational restrictions. The algorithm was ran with only one budget percentage for each healthcare facility area (so only running the Gurobi optimization once for every healthcare facility with one budget). It took more than 8 days for only about a third (121 or the 347) of all the healthcare facility scenarios to just find one solution, afterwards the computer crashed. Which means that calculating for multiple budgets could easily take 9 times longer. This does not satisfy our running time goals. We could have restricted the running time of the optimization, but because we wanted to develop a better method rather than allow this method to produce solutions that would be far from optimal, we chose to invest time in other algorithms.

One of the aspects that causes this calculation to take so long is the optimization over the infrastructure dense areas (as shown in Figure 11b). These areas have a huge amount of road

max	$\sum_{a \in A} \sum_{s \in BS} S_{as} y_{as}$		Maximize the number of connected households	(0)
s.t.	$\sum_{a \in A} \sum_{s \in BS} e_{as} y_{as} \leq B$		Budgetary restrictions for upgrading costs	(1)
	$\sum_{s \in BS} y_{as} \leq 1$	$\forall a \in A$	Every area can only have one scenario included in the solution.	(2)
	$y_{as} \in \{0, 1\}$	$\forall a \in A, s \in BS$	integer constraints	(3)

Input variables

A	Set of different healthcare facility level areas
B	Budget
BS	Set of different budget scenarios
e_{as}	The costs of including scenario s or area a
S_{as}	The number of households that are connected due to the solution of scenario s for area a

Decision variables

$$y_{as} = \begin{cases} 1 & \text{if secnario } s \text{ of area } a \text{ is included in the final solution} \\ 0 & \text{else} \end{cases}$$

Model 6 The formulation for the knapsack model used in the multiple budget scenario knapsack method. This will be used to combine local solutions in order to find a national solution.

segments and most nodes in these local networks have a high degree, which leads to an immense amount of possible paths. This results in a very large amount of variables to optimize over. This number could be reduced if the overlap between areas was taken into account. The problem that comes with this is that multiple budget scenarios per healthcare facility cannot be calculated, and thus a multiple knapsack model can not be applied to the situation.

6.2.2 Pre-assing budget method

From the research shown in the last section (Subsection 6.2.1), calculating multiple solutions for all the healthcare facilities is computationally demanding and needs a local search heuristic and improves the overlap between areas in the found solution. The idea that could solve this issue is incorporating all previously found solutions into the healthcare facility area that is currently being optimized over. This can be done by making a clear order in relevance between areas, assigning them a certain budget and finding an optimal solution for them while taking into account earlier found solutions. Due to this, roads that are updated in one solution will not be optimized over again in later solutions. The same holds for households that have already been connected. This decreases the number of variables in of the models of healthcare facilities that are optimized over later on in the algorithm.

In order to apply this idea, we can only calculate an optimal solution for one budget for every healthcare facility. We would also need to order the healthcare facilities in a way that the areas that could be invested into most effectively would be upgraded first, and the areas with the least beneficial investments at the end.

Therefore, two things needed to be established:

(1) How to order the households from (likely) most effective investment area to least?
(2) How to assign the budgets?

1. How to order the households from (likely) most effective investment area to least
The ordering of the households is done according to the relevance ratio of the healthcare facility area. To remind the reader of the concept relevance ratio, the relevance ratio is the ratio between the number of households that (can) benefit from an intervention and the costs of the intervention. Because we did not know the exact optimal intervention, this was calculated by counting all the unconnected households within the area and summing up the costs of upgrading all the roads in the area.

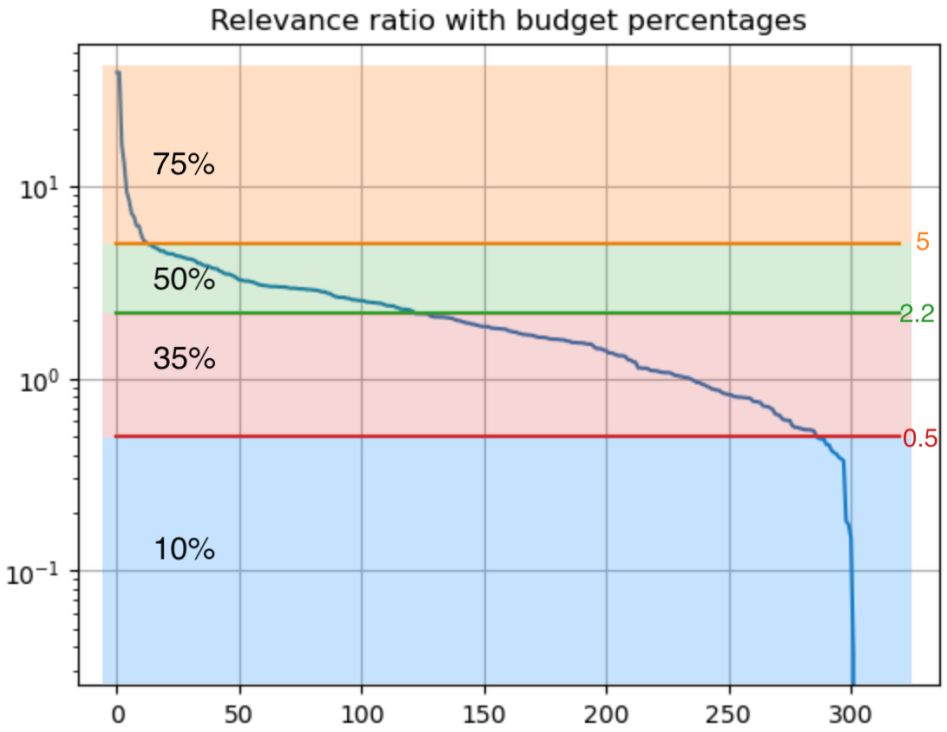
2. How to assign budgets
Two methods were tested to assign budgets. The one that worked the best was the method that assigns budgets according to the relevance ratio of a healthcare facility area. If the relevance ratio falls within a certain interval, the budget would be a certain percentage of the total costs to upgrade a certain area.

The other method is via a knapsack problem based on the unconnected households in the area and the costs of upgrading all the at risk roads within the area. The knapsack solution turns out not to be useful because it would just assign a value of 1 to the areas with the highest relevance ratio, and 0 to the lower ones. More details about this can be found in Section 7.5.

The budget assigning according to the relevance ratio is done on the basis of intervals. Each interval is assigned a percentage, and the final budget of the healthcare facility will then be that percentage of the total costs to upgrade the area. The intervals and their corresponding percentages are shown in Figure 19 alongside the distribution of the relevance ratio. This figure also contains the exact intervals and their assigned percentage.

In order to be able to constraint the total budget for a national budget, these local budgets are being compensated by a factor that ensures that the sum of all the local budgets sum up to the set national budget. This factor is calculated as follows:

$$factor = \frac{national\_budget}{\sum_{h \in HCF} ratio\_percentage * total\ updating\ costs\ for\ area\ h}$$



≥	<	%
5	∞	75%
2.2	5	50%
0.5	2.2	35%
0	0.5	10%

Figure 19 How the budget percentages are distributed according to relevance ratio of an area.

From which logically follows

$$national\_budget = factor * \sum_{h \in HCF} ratio\_percentage * total\ updating\ costs\ for\ area\ h$$

The pseudocode for the algorithm can be found in Listing 4. The results of this method will be discussed in Chapter 8.

Remarks

What is most important to remark about this method is that there has not been a thorough analysis of how these percentages corresponding to the intervals are best established. This is due to shortage in time. We preferred to invest in the heuristic we will discuss next at that point in the research. Also, this method only provides one solution for one budget scenario. In order to calculate a Pareto curve (the graph that sets out how many households can be connected for different budgets), many runs must be made which will cost a much larger amount of time.

```
#calculate relevance ratios for all hcf areas, assign budget percentages
   and order the hcfs
for h in hcfs:
    relevance ratio of h = total number of unconnected households within 5km radius of h / cost of updating all at risk roads within 5km radius of h
    if relevance ratio ≥ 5: budget for h = 0.75 * costs of updating all
       roads within area
    if relevance ratio ≥ 2.2 and < 5: budget for h = 0.5 * costs of
       updating all roads within area
    if relevance ratio ≥ 0.5 and < 2.2: budget for h = 0.35 * costs of
       updating all roads within area
    if relevance ratio < 0.5: budget for h = 0.1 * costs of updating all
       roads within area
order hcf areas according to relevance ratio

#reset budgets to not surpass national budget
factor = national budget / sum of all assigned budgets for hcfs
reset budget for hcfs to factor * initial budget

for every healthcare facility h in descending order:
    demarcate roads, unconnected households and healthcare facilitieswithin
        5km radius of h
    find all 5km paths healthcare facilities and households within area
    find optimal solution for 5km range area using Gurobi
    save:
        - connected households index
        - number of connected households
        - updated healthcare facilities index
        - upgrading costs
    update the road dataset with the newly updated and no longer at risk
       roads
    mark all households that are now served as connected
```

Listing 4 Pseudocode for global solution algorithm where budgets are assigned before the optimization, refered to as the budget asign method.

6.3 Dynamic greedy heuristic

The last heuristic that was developed and tested is the dynamic greedy heuristic. A greedy heuristic is a heuristic that assembles a solution by adding the locally optimal addition at each stage of the algorithm. The heuristic we have developed adds new paths to the solution according to relevance ratios. Because the paths are so interdependent upon each other, it is important to take the results of the formerly added paths into consideration. Therefore, after every new path has been added, the number of dependent households, at risk edges, and the upgrading costs of the remaining paths are updated and the relevance ratio is recalculated. We chose to refer to this heuristic as dynamic because of the interdependence and constant need for updating makes it a bit more complex than a general greedy heuristic. The psuedo-code can be found in Listing 5.

```
#prepare the paths data set
cluster households according to where they enter the road
find for each cluster all healthcare facilities within 5km radius
find all relevant paths between the household clusters and nearby hcfs

for every path:
    calculate the costs of upgrading the at risk roads on the path
    calculate how many unconnected households are in the dependent clusters

#apply dynamic greedy algorithm
spent = 0
while there are still unconnected households and paths left and budget >
   spent:
    for every path calculate the relevance ratio: number dependent, unconnected households / costs of upgrading at risk links
    find path with highest relevance ratio whose costs ≤ budget - spent

    spent = spent + costs of this path
    save:
    - added path
    - households connected by adding of this path
    - costs of adding path
    - spent
    - computational time so far

    mark all at risk links on this path as no longer at risk
    mark the households dependent on this path as connected
    for all remaining paths:
        recalculate upgrading costs of all still at risk links on path
        recalculate number of dependent, unconnected households
        if costs of adding path are 0:
            add path, update road and household dataset and save
               information
```

Listing 5 Pseudo code for the dynamic greedy heuristic that finds (near-)optimal local solution for a healthcare facility

The idea of the algorithm is that it iteratively adds the path with the highest relevance ratio. But because the paths are so interdependent upon each other, it is important to take the results of the formerly added paths into consideration as well. If there is a set budget, the algorithm checks for every path it is considering to add, if its addition does not surpass the total budget spent.

This algorithm can be applied to the entire nation of Timor-Leste at once. It is able to produce a Pareto curve on its findings. A Pareto curve is a graph that sets out how beneficial an investment is alongside the costs of the investment. This allows all possible budget scenarios to be calculated and compared.

The results of this algorithm applied to Timor-Leste can be found in Chapter 8.

We would like to note that this algorithm fits the wishes of the World Bank very well. From a policy perspective, a Pareto curve provides a lot of insight in to how to chose a budget for a

project. This would not have been possible with the multiple budget scenario knapsack method nor with the pre-assing budget method.

7 Considered configurations

When developing our algorithms, there were often many implementation options and parameters needed to be used. Therefore, theoretical and empirical research was needed to substantiate these choice. The analyses of these tests could disrupt the narrative of this thesis, because they can be quite extensive. Therefore, some of these analyses are discussed in this separate chapter.

The first two sections (Section 7.1 and Section 7.2) of this chapter regards analyses regard relaxing variables of the FARNUP model (Model 4). Afterwards, two empirical parameter tests are discussed. These parameters are the MIP gap (Section 7.3) and how many relevant paths to generate between each O-D pair (K) (Section 7.4). The section after that, discusses why using a knapsack-like approach to assign budgets to healthcare facility area does not work (Section 7.5). The last section (Section 7.6) explains why working with grids would not work for this problem.

7.1 Can solving the LP relaxation with some postprocessing yield an integer solution faster than the branch and bound algorithm?

One hypothesis that was formed during the implementation of the branch and bound algorithm, was the hypothesis that the LP relaxation could yield solutions that would easily convert to (near-)optimal integer solutions. This could hold because the problem is somewhat like a knapsack problem: the most relevant links and paths will be fully included and the least beneficiary links and paths will not. Therefore, the LP relaxation with some postprocessing could work.

We were sadly not able to prove that this problem could yield solutions like a knapsack problem would. Therefore, we chose to approach this more empirically. We ran $n = 48$ test cases where the LP solution and the MIP solution were computed using the branch and bound implementation applied to healthcare facility areas, and compared the performance measures. We bounded the running time of the branch and bound algorithm such that the running time would not take too long. This is why we only have results for 48 cases rather than the initial 50 random samples we selected.

The average results of these 48 cases can be found in Table 7. We see that the objective value of the LP solution is on average 134% larger than the MIP solution and that the objective values were only equal in only 32% of the cases. Based on this, we can already conclude that the LP relaxation does not yield a solution that is very alike the MIP solution. This can also be seen very clearly in Figure 20, where all the different objective value results of all the 48 cases are plotted alongside each other (ordered according to the MIP objective value).

To finalize our understanding of the untranslatableness of the LP solution to a MIP solution, we have a look at the solutions for one healthcare facility area. In Figure 21 we see the solution

Average	LP case	MIP case
Objective value	1016,71	757,72
Optimization time (seconds)	0,56	28,5
LP solutions generates MIP solution	32%	100%

Table 7 Quantitative results LP relaxation test

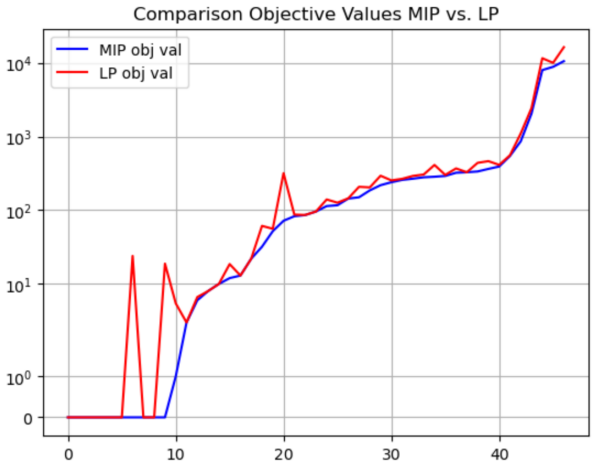


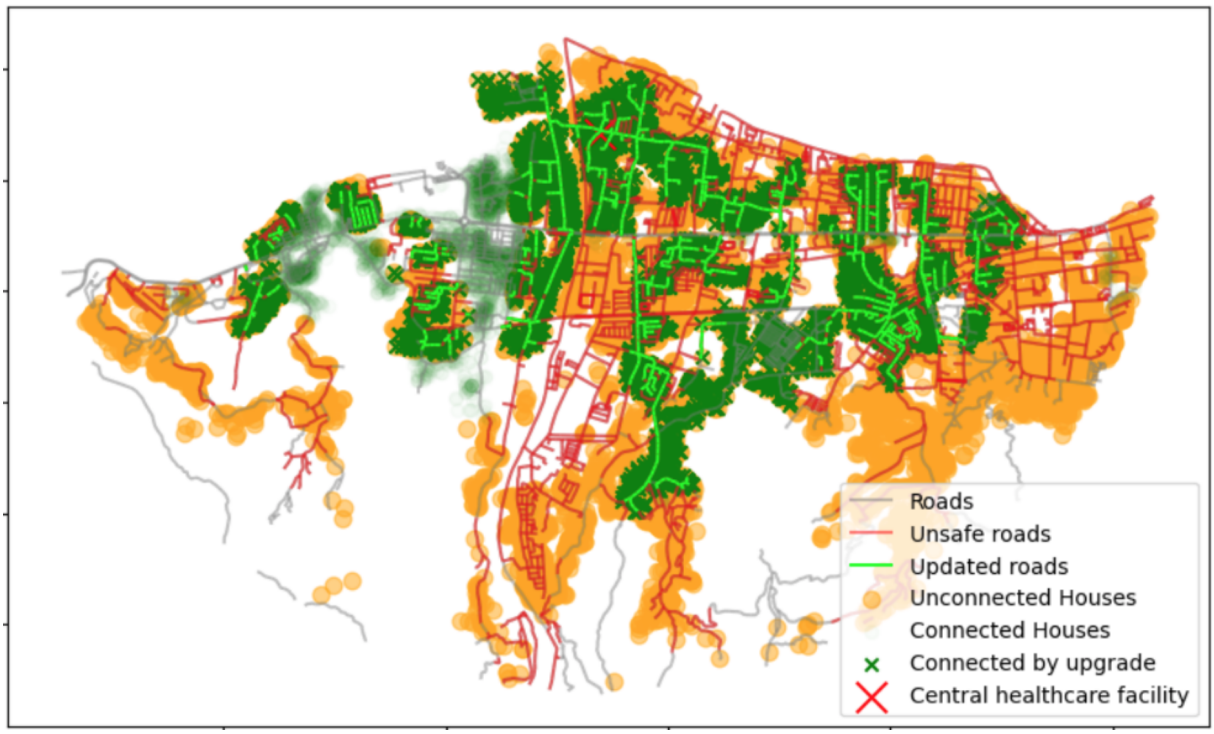
Figure 20 Objective value results of the MIP results versus the LP results (logarithmic scale, ordered according to MIP value)

for the same healthcare facility area, one portraying the MIP solution, the other the LP solution. Two big differences that can be see in Figure 21 are:

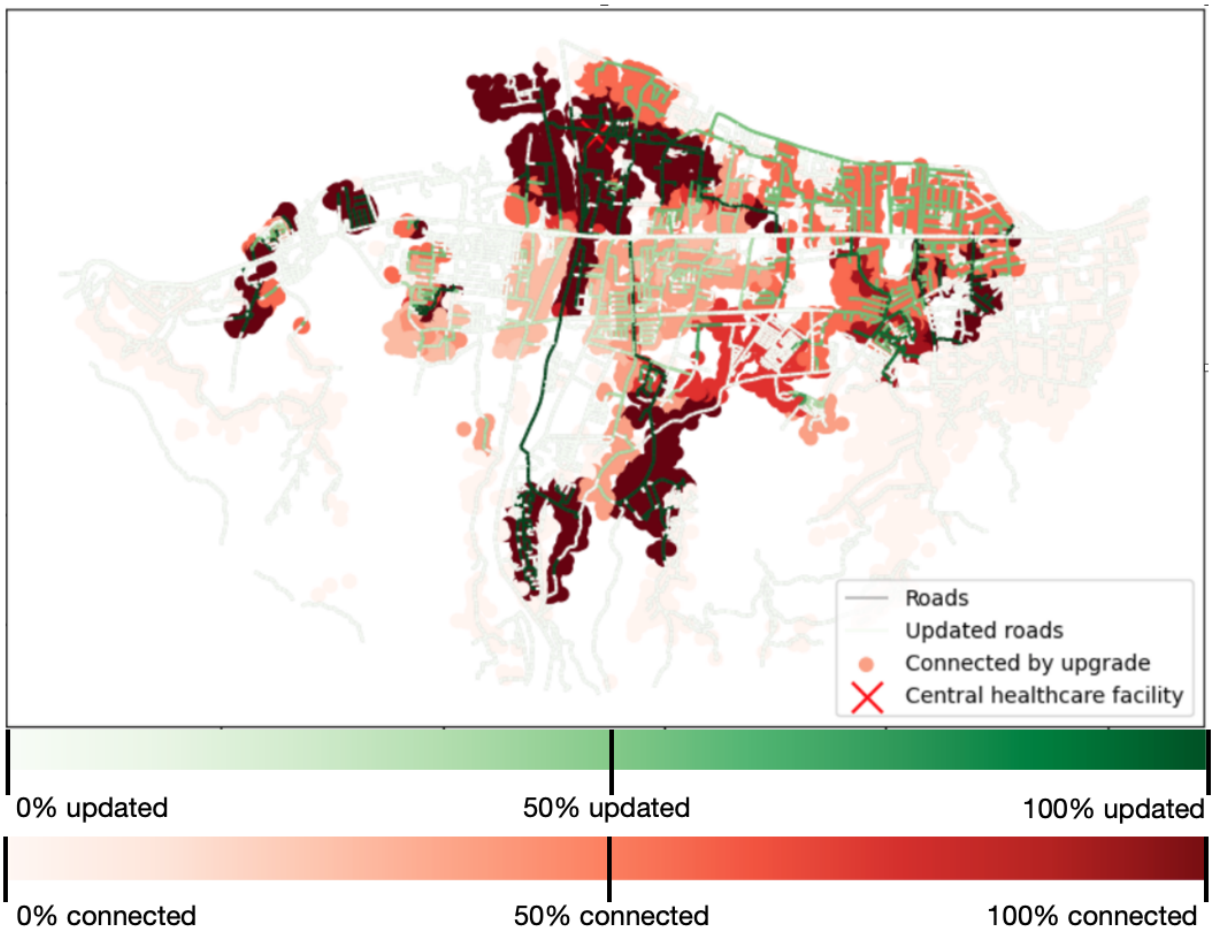
- different areas are invested in;
- a large share of variables of the LP solution are fractional.

We have come up with an postprocessing heuristic, and applied it to the case shown in Figure 21 to see if it could get us anywhere near an answer. When we applied the heuristic the budget was heavily surpassed by (187%). Therefore, this heuristic is not useful. The algorithm and the idea behind it can be found in the Appendix, Section B.

Therefore, we can conclude that the LP relaxation does not provide a solution that can be (easily) converted to an integer solution.



(a) MIP solution for example scenario



(b) LP solution for example scenario

Figure 21 Visual example of the MIP solution and the LP solution for the same healthcare facility area.

7.2 Does relaxing exactly one variable speed up computations while providing proper MIP solutions?

We have also executed a test that tested if we could relax just one of the two variables, and if this provided us with a useful solution faster. We can show that this results in useful solutions by means of mathematical proof. We have also carried out an empirical test. We will first show the mathematical proofs, and afterwards discuss the results of the empirical test.

Theorem 1 *Relaxing the x variables in the FARNUP model (Model 4) yields a MIP solution.*

Proof
As we can see in constraint (2) of Model 4: if the x variables are relaxed, they are still forced to take on the value 1 if a path it is a part of has taken on the value 1 (because a path must still be binary). Also, an x value will not be fractional in the solution if all paths it is a part of are 0, because this will only drive up the budget, but does not increase the objective value. Therefore, the x values will be minimized (and thus be 0) if they are not included in activated paths.

Theorem 2 *Relaxing the z variables in the FARNUP model (Model 4) yields a solution where households dependent upon a path with a non-zero decision value are connected.*

Proof
Assume that in the optimal solution there is a z_p with a fractional value and assume z_p is the only active path ending at $end(p)$. Then, there exists a better solution. This solution is the same solution we have assumed to be optimal, except this specific z_p is now 1. Because z_p can be set to be 1 and still abide the second and third constraint due to the integrality of the x variables, yet the objective value will be higher. So, if z_p is fractional, there must be at least one other path p' ending at $end(p)$. If this path p' has exactly the same number of households dependent upon it, than z_p can be set to 1 and $z_{p'}$ to 0 (or the other way around) and the objective value would be the same. If we assume that path p' has less households dependent upon it, this solution is also not optimal. Because, for all paths ending in $end(p)$, we can take the path (or a path) with the highest S_p value, and set this path to 1, and the other to 0. This way we still satisfy all constraints, especially constraint 3, yet our objective value because we know that $\sum_{i \in I} \lambda_i S_{p_i} \leq \max\{S_{p_i} \mid i \in I$ if $\sum_{i \in I} \lambda_i = 1$. Therefore, we can assume that all z_p will also be binary.

7.2.1 Running tests

A test was ran that compared $n = 48$ random cases. This test showed that relaxing the x or the z variables does not provide us with a faster optimization algorithm.

In Table 8, we can see that the objective values are near the same, and the computational time is as well. The results for the relaxed z variables are the fastest, but the objective value is the lowest. Therefore, we can conclude that relaxing a variable does not provide us a faster method to find a MIP solution.

Average	X relaxed	Z relaxed	MIP
Objective value	1085,2	1085	1085,2
Optimization time (seconds)	1,61	1,12	1,72

Table 8 Averages of the results for the test where different variables were relaxed.

7.3 What MIP gap to choose?

In order to make a well educated decision on the choice of the MIP gap termination criteria, different MIP gaps must be tested. For this test, we wanted to find the MIP gap that provided us with the proper balance between running time and optimality of the solution. In order to do this, the branch and bound algorithm was applied with four different MIP gap stopping criteria for $n = 80$ different healthcare facility areas. These MIP gaps were 5%, 1%, 0.5% and 0.1%. All scenarios were run with a budget that is 10% of the costs to upgrade the entire healthcare facility area.

We compared these results on the base of running times, objective values and upgrading costs. The first two were the most important. In Table 9 we can see the averages of these factors. We can conclude from these results that the 1% gap has the best balance between speediness and near-optimality. It is the fastest and its objective value lies very close to the highest objective values compared to 5%.

Average	5%	1%	0.5%	0.1%
Running time (sec)	1.278	1.212	1.228	1.362
Objective value	469.775	475.75	476.088	476.3
Upgrade costs	24	23.91	23.9	23.92

Table 9 Average results for the test runs with different MIP gaps. Tested on $n = 80$

7.4 How many paths to generate per O-D pair (K) when generating the relevant paths?

In Section 5.2, we have established that the algorithm that generates the relevant paths is a more suitable method to generate paths than the method that generates all paths (Section 5.1). When applying the relevant path generating algorithm, the choice for the number of paths that

is generated per O-D pair (K) could affect the quality of the solution and the running time of the algorithm. Not only will it affect the running time of the path generation algorithm, but it can also affect the running time of the optimization algorithm because it affects the number of decision variables. Therefore, an empirical test was carried out to test how different values of K affect the optimization algorithm.

We applied our empirical test to the dynamic greedy algorithm because this algorithm performed best out of all the large scale heuristics (which will be discussed in Chapter 8), and would therefore give a better understanding of how different K-values affect the running time and performance.

The hypothesis is that a higher value of K would attain better optimal values because there are more paths to choose from. But a higher K-value would theoretically also run for a longer time because it would take longer to generate and process these paths, and there would also be more variables to iterate over.

The dynamic greedy algorithm is run for eight different values of K, these values are 2, 3, 4, 5, 7, 9, 11 and 13. For every K-value, a set of Pareto curve coordinates is saved. These coordinates are selected according to how much of the maximum possible budget had been spent. These budget percentages are 1%, 2.5%, 5%, 7.5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%, 55%, 60%, 70%, 80% and 100% of the total costs to upgrade all at risk roads. The exact budgets for those different solutions are also saved because they differed slightly, and will give a better view of the optimality of a solution.

The running time analysis alongside the number of generated paths is shown in Figure 22. This figure shows how the running time is made up. This is the time it takes to generate the paths (dark blue) and the time it takes to optimize over this set of paths (light blue). Here we see that the running times increase linearly as K increases. This motivates a preference for a lower K-value rather than a higher one.

In Figure 23 we see the different Pareto curves for the different K-values. These curves do not provide an obvious best choice because the curves are very much alike. Therefore, a better quantification of performances is needed. The following two methods are applied:

(1) The sum of the relevance ratios of the coordinate samples. Thus for each coordinate the relevance ratio, $\frac{number\ of\ households}{budget}$, has been calculated. This quantifies how many households can be connected per unit of currency, which makes the value of an investment comparable. Because every test run for each K has the same number of samples, the sums of these relevance ratios can be compared, because they express the benefit of the currency units.

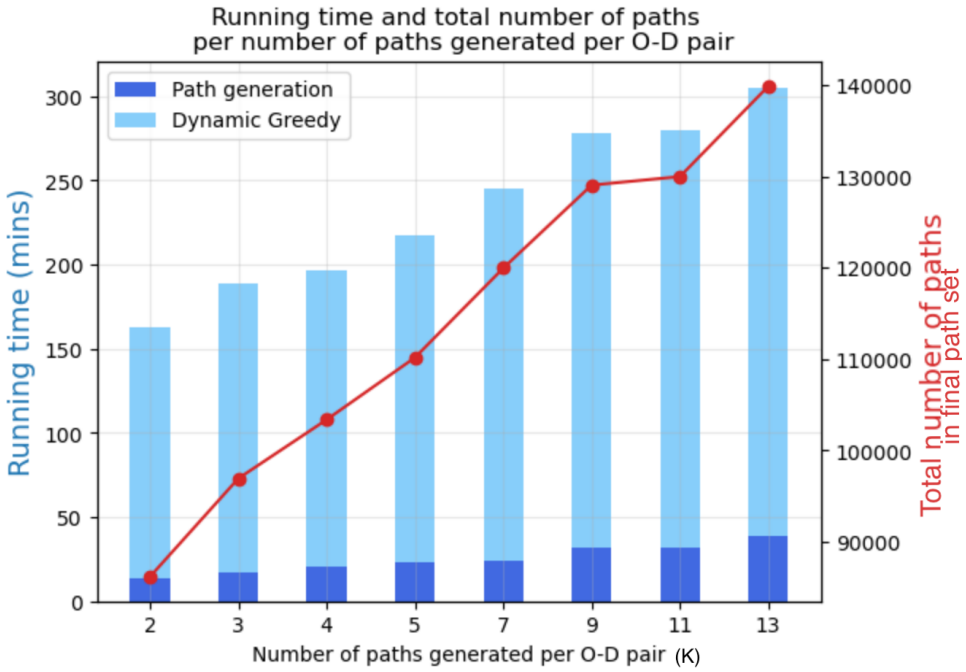


Figure 22 Running times and total number of paths generated for each value of K. Running times are split into running time to generate paths (dark blue) and to apply the greedy algorithm (light blue).

(2) The (logarithmic) area under the curve (AUC). The area under the curve quantifies beautifully how high the overall investment value is. When we applied the regular AUC, all values were still very much alike. This is due to the fact that in the AUC, the higher values weight more heavily. Because all the roads in Timor-Leste will not be upgraded, we do not want the results to become heavily influenced by the higher budget values. Therefore, we chose to use the logarithmic values of both the budget and the number of households. This ensured that the highest budgets do not entirely make up the AUC. Therefore, we chose to apply the log-AUC. The log-AUC is calculated as follows

$$\sum_{i=1}^{n-1} \frac{1}{2} \times (\log(budget[i+1]) - \log(budget[i])) \times (\log(hhds[i+1]) + \log(hhds[i]))$$

.

The results of the test statistics for every K value can be seen in Figure 24. A table with the results of our two test statistics and the regular AUC can be found in Section C of the Appendix. We chose to visualise our results because the table seemed very chaotic. We see that $K = 4$ scores high for both the sum of the relevance ratios and (log)-AUC, while still having a very low running time. Therefore, $K = 4$ is the advised parameter.

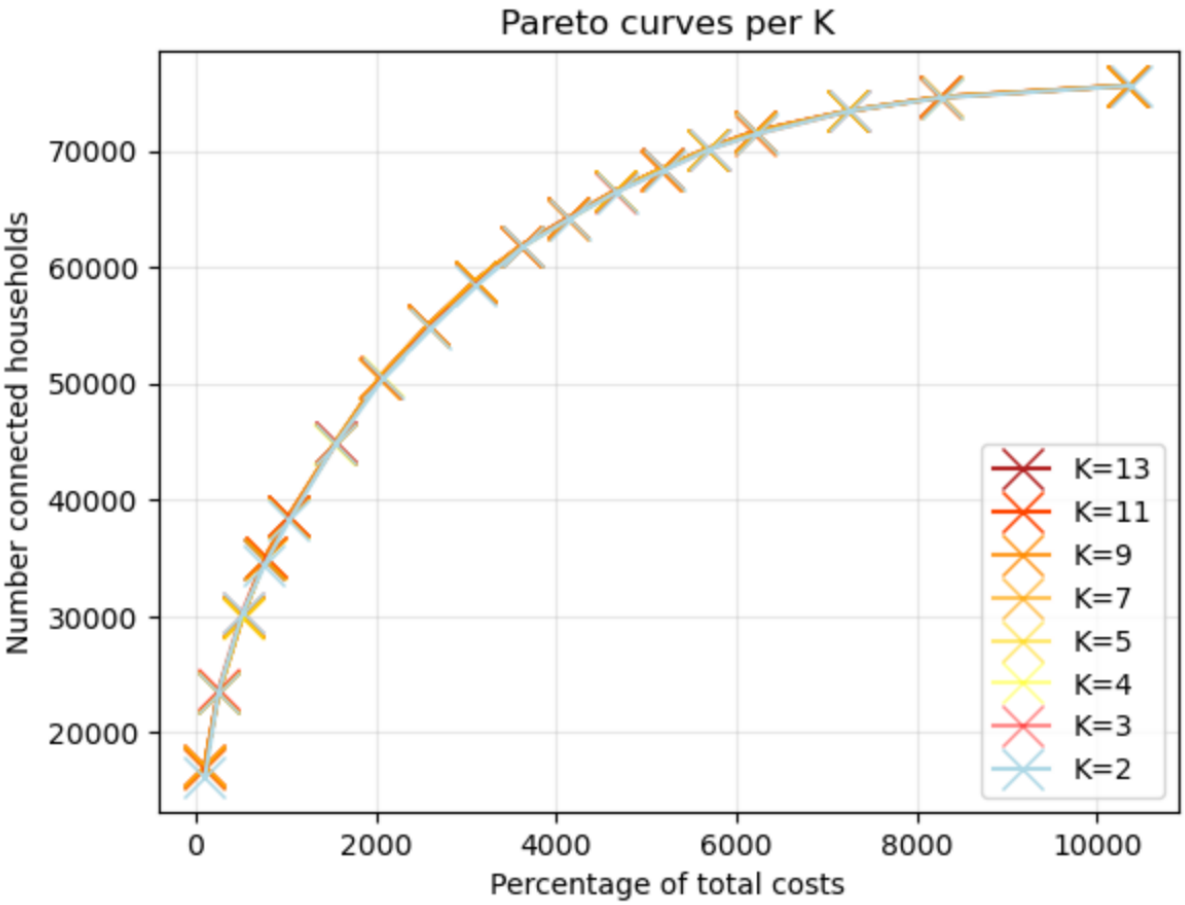


Figure 23 Pareto curve for each K.

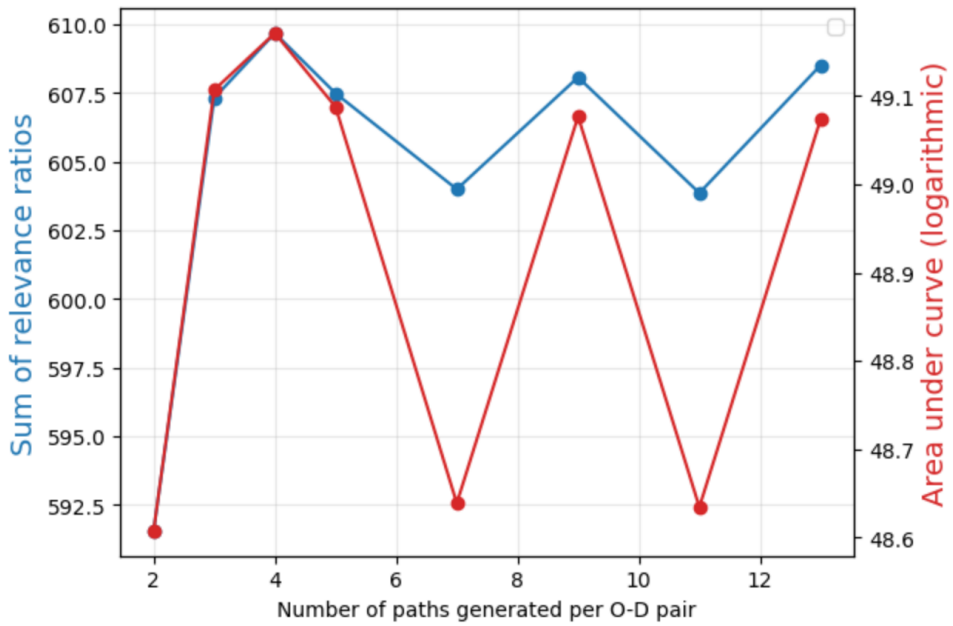


Figure 24 Plot of the sum of the relevance ratios and the area under the curve for every K.

7.5 How to pre-assigning budgets?

Two methods have been developed and compared in order to establish how to assign budgets to healthcare facility areas. These two methods are:

(1) On the basis of relevance ratios of the area (so not the calculated cost-benefit, but just on the number of unconnected households and costs to upgrade all roads within a healthcare facility area). This method has been thoroughly elaborated on in Subsection 6.2.2;

(2) By means of a knapsack-like problem.

The idea of the knapsack-like problem is to assign budgets (or budget percentages, to be a percentage of the costs to upgrade all at risk roads) based on the number of unconnected households in the area, and the costs of updating all the roads in the area. The issue with this is that the algorithm assigns the areas with the highest relevance ratio to have 100% of the budget (or the maximum possible budget percentage). Furthermore, one area is assigned a fractional value (because it is unable to assign 100% of the budget without surpassing the budget constraint) and the rest 0. This is exactly what you expect from an LP relaxation of a knapsack problem.

We have experimented with different settings in order to avoid this. We have tried setting a lower and an upper bound to these values and we have tested adding a binary variable v that would address if a healthcare facility area had been assigned a budget(percentage) larger than 0. This would be added to the objective value

$$\textbf{max} \sum_{a \in A} \sum_{s \in BS} S_{as} y_{as} + \alpha \sum_{h \in HCF} v_h$$

resulting in the incentive to assign as many healthcare facility areas a budget percentage that is larger than 0.

This resulted in the healthcare facilities being assigned either the highest possible value, or the lowest possible value. Sadly, we could not attain a smart method to use the knapsack LP to assign these budgets.

max	$\sum_{a \in A} \sum_{s \in BS} S_{as} y_{as}$		Maximise the number of connected people	(0)
s.t.	$\sum_{a \in A} \sum_{s \in BS} E_{as} y_{as} \leq B$		Budgetary restrictions	(1)
	$\sum_{s \in BS} y_{as} \leq 1$	$\forall a \in A$	Every area can only have one scenario included in the solution.	(2)
	$y_{as} \in \{0, 1\}$	$\forall a \in A, s \in BS$	integer constraints	(3)

Input variables

A	Set of different healthcare facility level areas
B	Budget
BS	Set of different budget scenarios
E_{as}	The costs of including scenario s or area a
S_{as}	The number of households that are connected due to the solution of scenario s for area a

Decision variables

$$y_{as} = \begin{cases} 1 & \text{if secnario } s \text{ of area } a \text{ is included in the final solution} \\ 0 & \text{else} \end{cases}$$

Model 7 The formulation for the knapsack model used in the multiple budget scenario knapsack method.

7.6 Why not optimize on grids?

The idea to apply the branch and bound algorithm on grids has been considered. The idea was to split the country up in grids of 5 kilometer by 5 kilometer. This way, the overlap between healthcare facility areas could be avoided. But, it would lead to problems when it comes to including all households within a 5 kilometer reach of the healthcare facilities. Because a healthcare facility could be part of one grid cell, but the households that can travel to this healthcare facility within 5 kilometers could be part of another grid. An example of this can be seen in Figure 25.

A few methods had been considered to work around this:

- Add a margin around a grid that overlaps with other grids. The problem with this is that you would need to have a margin of 5 kilometers around every grid cell, which means that an area is optimized upon multiple times and that an postprocessing heuristic is needed. This would be computationally very demanding and thus not useful;
- Add to the data within the grid the data outside the grid that corresponds to the paths of a household cluster that travels to one of the healthcare facilities within the grid. This would also mean that the number of variables that need to be optimized over increases heavily, and that a lot of postprocessing is needed.

Therefore, the conclusion was drawn that working with grids is not useful.

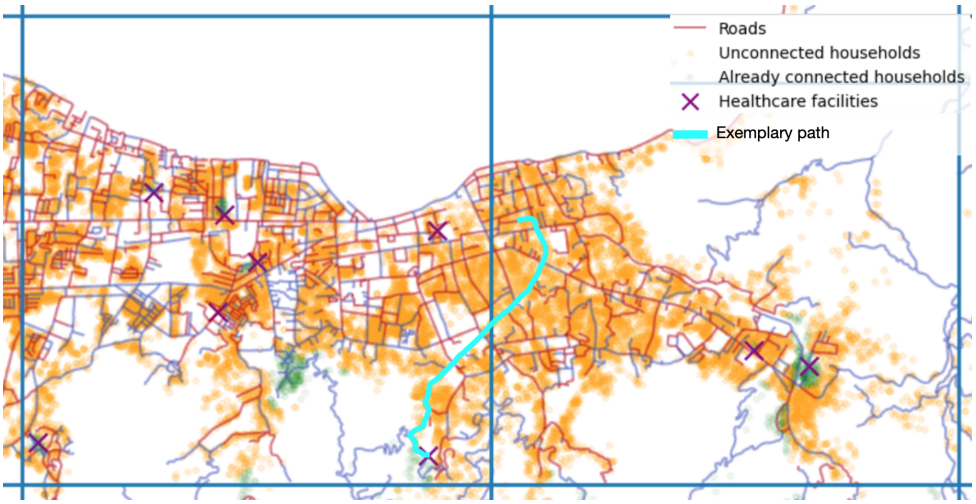


Figure 25 An example of how grids could break up an O-D pair.

8 Performance results of main heuristic

In this section, the performances of the two large scale algorithmic approaches proposed in Chapter 6 will be analyzed. The performances are measured in two different ways. The first performance aspect is the quality of the found solution. This is measured in terms of the number of households that can be connect and at what price. Another performance aspect that will be considered is the computational time that is needed to run the algorithm. To remind the reader, we have chosen a maximum running time of 12 hours. We will not review the multiple budget scenario algorithm because we were unable to run it nor have we invested time in developing a postprocessing heuristic to attain a proper solution.

First, the results of the pre-assigned budget method are discussed and afterwards we discuss the results of the dynamic greedy heuristic. For both heuristics, we list the parameters and settings we have chosen. Afterwards, we discuss the quality of the solution and finally we show the results that regard the running time.

8.1 Pre-assigned budget method

Parameters and other settings

The parameters and settings that we have chosen when running the pre-assigned budget method are as follows:

Budget (B)	4879.49
Maximum running time for branch and bound	3 hours
MIP gap	2.5%
Number of paths per O-D pair (K)	4

When we were developing this algorithm, it was initially able to meet our running time demands. When we ran it later on to finalize the results for this thesis, it suddenly ran for days. This could be due to the changes in datasets, but is still a strange phenomenon. This could also indicate that the algorithm is unstable.

In order to still be able to evaluate the algorithm, we have chosen to bound the running time and increase the MIP gap to 2.5% rather than 1%, in order to be able to compare the algorithm. Therefore, these parameter values that differ from the advised parameter values are chosen.

Quality of the solution

The results for the solution can be found in Table 10. We see that, for a budget of 4879.49, the algorithm is able to increase the percentage of connected households from 35% to 70%. As we have seen in Section 4.4, for only 79% of the households there exists a path of at most 5

kilometers traveling distance from a household cluster towards a healthcare facility. Therefore, 70% of the households is 89% of all the households that could be connected.

	Before	After	Gain
Amount of households connected	62219.0	122390.0	60171.0
Percentage of all households	35.65%	70.12%	34.47%
Percentage of connectable households	45.36%	89.22%	43.86%
Amount of households unconnected	74955	14784	14784

Table 10 Results of the found solution for the pre-assign budget with a budget of 4879.49.

Running time performance

The running time performance is shown in Table 11. We can see that the total calculation time for this heuristic is 3845.85 minutes, which is approximately 64 hours. This heavily surpasses our time limit. Therefore, this algorithm does not satisfy our demands.

	Calculation time (min)
Generating paths	22.82
Optimization	3823.03
Total calculation time	3845.85

Table 11 Running time of budget pre-assign method for a set budget of 4879.49

8.2 Dynamic greedy

The dynamic greedy algorithm is evaluated in two ways. We ran it in such a way that we can compare it to the pre-assign budget method. But, we also want to compare it to more accurate solutions. Therefore, we compare it to the branch and bound algorithm on small scale areas. First, we discuss the statistics that we have attained in order to compare it to the pre-assign budget method. Afterwards, we discuss the performance of the dynamic greedy heuristic when compared to the branch and bound method.

Parameters and other settings

Budget (B)	≤ 4879.49
Number of paths per O-D pair	4

Quality of the solution

The statistics of the solution that the dynamic greedy was able to find for a budget of 4879.49 are shown in Table 12. For this budget, the heuristic is able to increase the number of connected

households from 36% to 74%. This is a very good result, because from the accessibility analysis in Section 4.4 we know that for only 79% of the households there exists a path towards a healthcare facility of at most 5 kilometers. Therefore, we can conclude that this algorithm connects 94% of all the households that could be connected. The Pareto curve for this heuristic can be seen in Figure 26.

	Before	After	Gain
Number connected	62219.0	129612.0	67399.0
Percentage of all households	35.65%	74.26%	38.61%
Percentage of connectable households	45.36%	94.49%	49.13%
Nr unconnected	74955.0	7562.0	7562.0

Table 12 Performance of dynamic greedy for budget of 4876.92.

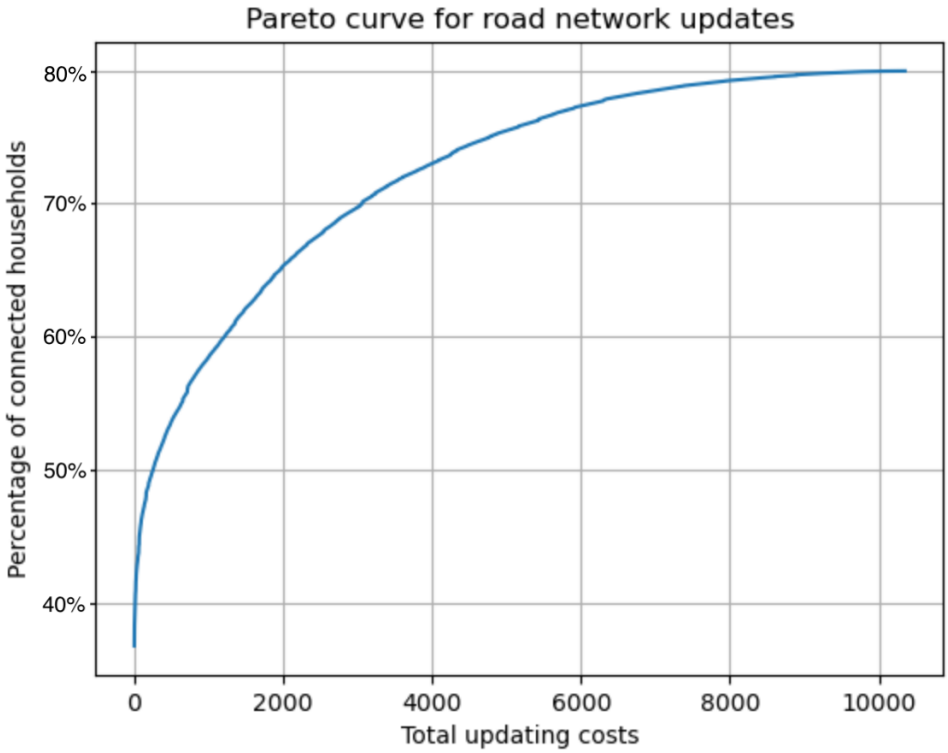


Figure 26 The Pareto curve of the solutions found by the dynamic greedy heuristic for Timor-Leste

Running time performance

The running time performances of the dynamic greedy algorithm are shown in Table 13. Because we want to compare this algorithm to the pre-assigned budget method but also want to show the computational results when we do not bound budget and allow the algorithm to terminate, we include the computational results for both. This algorithm satisfies our maximum running time of 12 hours demand.

Calculation time (min)	**for specific budget**	**without budget constraint**
Generating paths	20.59	20.59
Dynamic Greedy	140.78	186.11
Total calculation time	161.37	206.7

Table 13 Running time of Dynamic Greedy algorithm for a set budget of 4879.49 and to find the entire Pareto curve.

8.2.1 Comparing the dynamic greedy algorithm to the pre-assign budget method

Both in quality of the solution and in running time, the dynamic greedy algorithm performs much better than the pre-assign budget method. We see that it is able to connect 4% more of the households, in a much shorter running time. Therefore, this algorithm is much better to find a solution for large scale scenarios, such as the entire nation of Timor-Leste.

8.2.2 Comparing the dynamic greedy to branch and bound on healthcare facility areas

In order to establish the accuracy of the dynamic greedy algorithm, comparing it to the branch and bound algorithm would be an interesting measure. Because we could only apply the branch and bound algorithm on a healthcare facility level, we applied the dynamic greedy to a healthcare facility as well. We applied the branch and bound and the dynamic greedy on $n = 86$ healthcare facility areas.

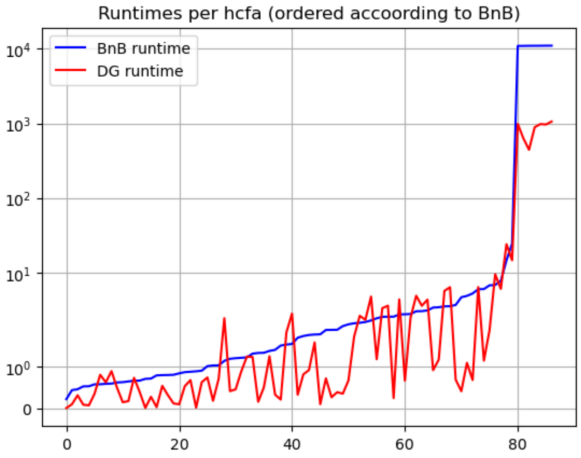
The branch and bound algorithm had a set time limit, which was a time limit of 3 hours optimization time per healthcare facility area. This was done because otherwise it could take much too long. The MIP gap was set to 1% (as argued for in Section 7.3).

From the results we can conclude that the dynamic greedy algorithm preforms very well. As we can see in Table 14, the dynamic greedy algorithm runs much faster than the branch and bound algorithm (more than 12x faster) yet the optimality of the solution of the dynamic greedy algorithm is slightly better. This can be explained because the running time of the branch and bound algorithm was capped at 3 hours, which could mean that a (near-)optimal solution was not yet found for certain areas.

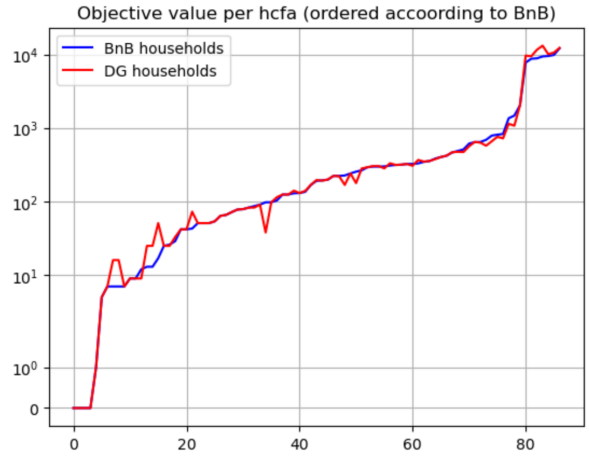
The results per instance are visualised in Figure 27.

Average	Branch and Bound	Dynamic Greedy
Run time (seconds)	884.4	70.4
Connected households	1006.6	1116.6
Budget Spent	50.5	50.4

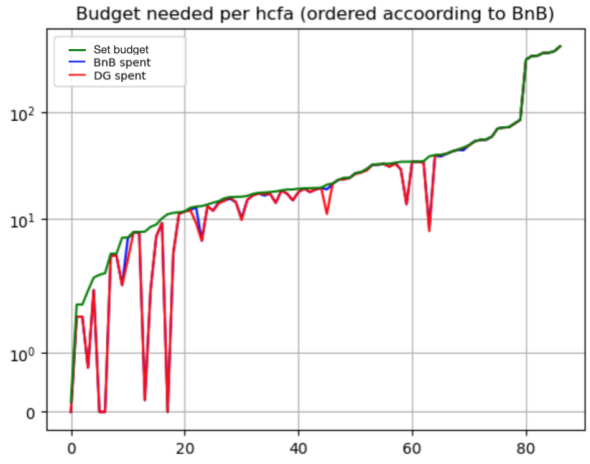
Table 14 Performance results of comparative tests for branch and bound and dynamic greedy applied to $n = 86$ healthcare facility areas.



(a) Running times



(b) Objective values



(c) Budgets

Figure 27 Performance results of the comparative test for the dynamic greedy algorithm and the branch and bound algorithm for all $n = 86$ cases. The orderings are done for the branch and bound algorithm for each test case individually. Performance measures are the running time, the objective value, and the needed budget.

9 Conclusions and discussion

9.1 Conclusions

This research has aimed to answer the question

> *How to minimize the impact that flood prone roads have on healthcare accessibility in developing nations, using optimization techniques.*

During the literature review, we found that the road network design problem is somewhat alike our problem because this problem also aims to design a network that ensures there is a path between different O-D pairs. The road network design problem is different from our problem because it aims to construct new edges, while we aim to upgrade existing edges that are not flood resilient. Furthermore, the road network design problem has the a fixed destination for every origin. By this we mean, if we would apply it within our context of household clusters and healthcare facilities, that every household cluster is assigned to one healthcare facility. For our problem, we have the flexibility to connect a household cluster to one healthcare facility within a set of healthcare facilities. This is a large difference. Lastly, the two problems have a different objective.

Two different road network design models with fundamentally different ideas have been reviewed: the model of Boyce [9] and the model of Magnanti and Wong [10]. The difference between the model of Boyce and the model of Magnanti and Wong is that Boyce his model takes a set of paths between O-D pairs as input, while Magnanti and Wong generate these paths within the model. Both methods could be applicable to our model. We decided that taking a set of paths as input is more suitable for our purposes, because it was easier to apply to a large scale scenario such as Timor-Leste. Therefore, we have decided to model our problem as can be found Model 4. We named this model the Facility Accessibility Road Network Upgrading Model.

In order to apply the Facility Accessibility Road Network Upgrading Model, we had to prepare the input data. One of the most important aspects of this was to create a flood model for the roads and a cost model for the at risk road segments. Since this topic is out of the scope of our research, these models are simplistic. We also needed to construct data, because our data was incomplete. Lastly, we also developed an algorithm that can establish which household have access to a healthcare facility within 5 kilometers traveling distance (both when taking into account the flood vulnerable roads and not). From this analysis we concluded that 78.59% of the households would be able to access a healthcare facility within 5 kilometer traveling distance, but that only 35.65% of the households is able to do so via a flood resilient path.

Because our model takes a set of possible paths between household clusters and healthcare facilities of at most 5 kilometers long as input, a method needed to be developed that generates this set of paths. Two methods have been developed in order to do so. One method generates all paths and the other method generates the relevant paths. With relevant paths we mean paths that are a combination between short in distance or have low updating costs. These two methods have been compared via empirical tests based on quality of the solution and running time. Based upon this empirical test, the method that generates the relevant paths seemed to suit our purposes better, because the running time of both the path generation and optimization running time is much shorter while the quality of the solution only slightly differs.

Multiple algorithms have been developed to find (near-)optimal solutions for the FARNUP model. The branch and bound method is applicable on small scale scenarios, but it was computationally too heavy to find a solution for the entire nation of Timor-Leste. In order to find solutions for this large scale scenario, we had to develop other algorithms. While developing these algorithms, we concluded that it is very important to take into account the overlap between healthcare facility areas, because approximately 75.4% of the households in Timor-Leste live within multiple healthcare facility areas.

Two algorithms were able to produce solutions: the budget pre-assign algorithm and the dynamic greedy algorithm. The budget pre-assign algorithm makes use of the branch and bound method for small scale scenarios. It assigns budgets to different healthcare facility areas, orders them, and then finds an (near-)optimal solution for each different healthcare facility area using branch and bound while taking into account the solutions of the formerly optimized areas. This algorithm is outperformed by the dynamic greedy algorithm. The dynamic greedy algorithm adds paths according to relevance ratio of the paths while taking into account the previous solutions within approximately 3 and a half hours. This algorithm is able to find a (near-)optimal solution for a provided budget. But, this algorithm can also be ran until all connectable households have been connected and there are no more paths left and save the order in which the paths have been added. This creates a prioritization of investments which can give a very clear overview of the Pareto curve (the number of households that can be connected for every different budget). This fits the wishes of the World Bank very well because such an analysis is very useful for policy makers. When comparing these algorithms, we could see that the dynamic greedy algorithm was able to find a better solution in a much shorter running time.

9.2 Recommendations

This research had a time frame of about nine months. This forced us to make choices in what we researched and withheld us from digger deeper into different topics. Therefore, many recommendations can be made for further research.

First of all, in order to obtain realistic solutions from our model, the flood and cost model must be improved by a transportation scientist with recommendations from local governments.

Furthermore, because the dynamic greedy algorithm is the most promising heuristic, further research on this heuristic could make big differences as well. Many possible improvements can be explored. First of all, examining the possible submodularity of the model, could provide mathematical proof of why the dynamic greedy algorithm is so successful. The quality of the solution could also be improved by the following things:

- An aftermath with a local search heuristic or a genetic algorithm for the dynamic greedy algorithm. The local search heuristic could generate new solutions by swapping an activated path of the found solution with a non-activated path that is not part of the solution. The activation and dis-activation of these two paths, would also indicate which edges no longer need to be upgraded and which edges must now be upgraded;
- Experimentation with the starting situation to which the dynamic greedy method is applied;
- Besides from only taking into account sub-paths in the dynamic greedy algorithm, also take into account the paths that have exactly the same at risk road segments in common and cluster these as one investment. This could also improve the running time, because there are less paths to iterate over.

Even though we have concluded that the dynamic greedy algorithm outperforms the heuristics that make use of the small scale branch and bound solutions, there are some recommended improvements to these heuristics as well. For the multiple budget scenario knapsack method, some experimentation could be done with bounding the running time of the branch and bound algorithm. This has not been done because this idea only occurred later on in the research, when other, more promising methods were already developed. But, in order to yield proper solutions for this model, a postprocessing heuristic must also be developed. For the pre-assigned budget method, the most valuable improvement that could be made is a better method to assign budgets. This is the aspect where most improvement could be made. Lastly, examining the stability of the algorithm is also advised, because its running times have differed immensely during this research.

When it comes to generating paths, we have found that the algorithm that generates relevant paths performs best for our purposes. This is due to the fact that the algorithm makes use of contraction hierarchies. It could be interesting to implement contraction hierarchies in to the algorithm that generates all paths. Then, this path generating process could be much speedier, and thus obtain better solutions using the dynamic greedy algorithm (even though we have seen

in Section 7.4 that more paths do not necessarily improve the solution nor the running time of the algorithm).

When it comes to the Python software that has been developed, a lot of improvements could be made on the implementation that could speed up the algorithms as well. One improvement could be to replace the use of Pandas DataFrames with Python Lists.

References

1. Weiss, D. *et al.* Global maps of travel time to healthcare facilities. *Nature Medicine* **26,** 1835–1838 (2020).
2. Molnar, A. K. *Timor Leste: politics, history, and culture* (Routledge, 2009).
3. WorldBank, T. *The World Bank in Timor-Leste* https://www.worldbank.org/en/country/timor-leste/overview#1 (2021).
4. WorldBank, T. *The World Bank in Timor-Leste: Data* https://data.worldbank.org/country/timor-leste?display=graph.
5. Cook, A. D., Suresh, V., Nair, T. & Foo, Y. N. Integrating disaster governance in Timor-Leste: Opportunities and challenges. *International Journal of Disaster Risk Reduction* **35,** 101051 (2019).
6. Rozenberg, J. *et al.* From A Rocky Road to Smooth Sailing (2019).
7. Espinet Alegre, X., Rozenberg, J., Rao, K. S. & Ogita, S. Piloting the use of network analysis and decision-making under uncertainty in transport operations: preparation and appraisal of a rural roads project in Mozambique under changing flood risk and other deep uncertainties. *World Bank Policy Research Working Paper* (2018).
8. Yang, H. & H. Bell, M. G. Models and algorithms for road network design: a review and some new developments. *Transport Reviews* **18,** 257–278 (1998).
9. Boyce, D. E. Urban transportation network-equilibrium and design models: recent achievements and future prospects. *Environment and Planning A* **16,** 1445–1474 (1984).
10. Magnanti, T. L. & Wong, R. T. Network design and transportation planning: Models and algorithms. *Transportation science* **18,** 1–55 (1984).
11. Friesz, T. L. Transportation network equilibrium, design and aggregation: key developments and research opportunities. *Transportation Research Part A: General* **19,** 413–427 (1985).
12. Hoc, H. Topological optimization of networks: A nonlinear mixed integer model employing generalized Benders decomposition. *IEEE Transactions on Automatic Control* **27,** 164–169 (1982).
13. Heng, S., Hirobata, Y. & Nakanishi, H. *An integrated model of rural infrastructure design in developing countries* **in** *Proceedings of the Eastern Asia Society for Transportation Studies Vol. 6 (The 7th International Conference of Eastern Asia Society for Transportation Studies, 2007)* (2007), 41–41.
14. Krishnakumari, P. K. *Private communication* Private communication. 26 **april** 2016.
15. Wolsey, L. A. Integer programming john wiley & sons. *New York, NY* **4** (1998).
16. Smith, A. *et al.* New estimates of flood exposure in developing countries using high-resolution population data. *Nature communications* **10,** 1–7 (2019).
17. Zhang, W. *State-space search: Algorithms, complexity, extensions, and applications* (Springer Science & Business Media, 1999).
18. Foti, F., Waddell, P. & Luxen, D. *A generalized computational framework for accessibility: from the pedestrian to the metropolitan scale* **in** *Proceedings of the 4th TRB Conference on Innovations in Travel Modeling. Transportation Research Board* (2012).

19. Cunningham, W. J. C. W. H. & Schrijver, W. R. P. A. *Combinatorial optimization* 1997.

20. Geisberger, R., Sanders, P., Schultes, D. & Delling, D. *Contraction hierarchies: Faster and simpler hierarchical routing in road networks* **in** *International Workshop on Experimental and Efficient Algorithms* (2008), 319–333.

21. Martello, S. & Toth, P. A bound and bound algorithm for the zero-one multiple knapsack problem. *Discrete Applied Mathematics* **3,** 275–288 (1981).

Appendices

A Other example of applied Magnanti & Wong model: minimum spanning tree problem

A.1 Minimum Spanning Tree Problem

The minimum spanning tree problem [19] aims to find a spanning tree on an undirected, complete graph where the minimal number of included edges are used. A *spanning tree* is a tree in which every pair of vertices are connected by exactly one path. The objective of the minimum spanning tree problem is to include the least amount of needed edges. Therefore, this problem has an objective value of min $\sum_{(i,j)\in E} y_{ij}$.

In light of the general model of Magnanti and Wong (Model 2), the objective function ϕ is written as min $\sum_{(i,j)\in E} c_{ij}y_{ij}$. This holds because the cost of including one edge is one (because the value of a solution is measured in number of edges). Therefore, we can also formulate this objective functions as min $\sum_{(i,j)\in E} y_{ij}$. The origin-destination-pairs will be any pair of nodes, as any two nodes must have a path. Thus there exist $V \times V$ commodities, each one corresponding to a path between two nodes. There is no budget constraint.

The following variable settings holds for any $k \in \kappa$ and $(i, j) \in E$.

$$
\begin{aligned}
R_k &= 1 \\
c_{ij}^k &= 1 \\
K_{ij} &= |\kappa| \\
e_{ij} &= 0 \\
B &= \infty
\end{aligned}
$$

Resulting in the following formulation

$$
\begin{aligned}
\textbf{min} \quad & \sum_{(i,j)\in E} y_{ij} \\
\textbf{s.t.} \quad & \sum_{j\in V} f_{ij}^k - \sum_{l\in V} f_{li}^k = \begin{cases} 1 & \text{if } i = O(k) \\ -1 & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \\
& \sum_{k\in\kappa} f_{ij}^k \leq |\kappa| y_{ij} \\
& y_{ij}, f_{ij}^k \in \{0, 1\}
\end{aligned}
\tag{.1}
$$

The minimum spanning tree is relevant for our problem, as connectivity is very important for our model, and road networks are often a trade-off between a minimum spanning tree (as this

minimizes the costs) and a complete graph (as this minimizes the travel distances between nodes).

B Postprocessing heuristic for the LP relaxation

The idea of the postprocessing heuristic is that if a path variable takes a non-integer value, the household cluster dependent upon this path is either (1) connected by multiple fractional paths, whose variables sum up to 1, or (2) is not fully connected and is therefore part of the remainder set of solutions that could not be fully included.

For the first group of fractional solutions, the idea is that they are interchangeable. As their costs and number of households that benefit are equal. For example, say three paths are all valued at $\frac{1}{3}$, then we can choose one of these paths to value at 1 and the others at 0. To repeat, this is a hypothesis and will not be not proven.

In Listing 6, the pseudocode of the algorithm is given.

```
to_address_links = []
for all p in paths with p ≠ 1 and p ≠ 0:

    if ∃ set of paths P′ such that:
     - S_p′ = S_p for all p′ ∈ P′ and
     - ∑_{p′∈P′} z_p′ = 1 :
        for all p" in P′ with ∑_{p∈P_end(n)} z_p = 0, choose (one of) the p" with the
            highest value for z_p"
        set z_p" = 1
        for all p'∈ P' where p' ≠ p":
            z_p′ = 0
    else:
        z_p′ = 0

for all links l in edgeset E:
    if there exists a path p such that l ∈ p and z_p = 1:
        set x_l = 1
    else:
        set x_l = 0
```

Listing 6 Pseudocode of postprocessing heuristic that would create an (near-)optimal IP solution given an optimal LP solution.

C Choosing how many paths to generate per O-D pair (K)

C.1 Statistics of Pareto curve samples

K=2

% Budget	Budget	# connected	% connected
1.0%	103.83	16199.0	9.28%
2.5%	259.57	23505.0	13.47%
5.0%	519.14	30246.0	17.33%
7.5%	778.7	34415.0	19.72%
10.0%	1038.27	38240.0	21.91%
15.0%	1557.41	44862.0	25.7%
20.0%	2076.54	50406.0	28.88%
25.0%	2595.68	54697.0	31.34%
30.0%	3114.82	58412.0	33.47%
35.0%	3633.95	61776.0	35.39%
40.0%	4153.09	64177.0	36.77%
45.0%	4672.22	66471.0	38.08%
50.0%	5191.36	68301.0	39.13%
55.0%	5710.5	70159.0	40.2%
60.0%	6229.63	71459.0	40.94%
70.0%	7267.91	73443.0	42.08%
80.0%	8306.18	74567.0	42.72%
100%	10382.72	75606.0	43.32%

	Calculation time (min)
Generating paths	13.91
Dynamic Greedy	148.56
Total calculation time	164.26

Table 15 Statistics for K=2

K=3

% Budget	Budget	# connected	% connected
1.0%	98.73	16844.0	9.65%
2.5%	258.88	23777.0	13.62%
5.0%	518.03	30382.0	17.41%
7.5%	777.0	35086.0	20.1%
10.0%	1035.98	38582.0	22.1%
15.0%	1552.82	44930.0	25.74%
20.0%	2065.73	50539.0	28.95%
25.0%	2581.77	54841.0	31.42%
30.0%	3108.37	58618.0	33.58%
35.0%	3614.94	61746.0	35.38%
40.0%	4125.59	64145.0	36.75%
45.0%	4662.37	66478.0	38.09%
50.0%	5180.57	68297.0	39.13%
55.0%	5699.28	70148.0	40.19%
60.0%	6212.25	71431.0	40.92%
70.0%	7248.08	73401.0	42.05%
80.0%	8266.75	74573.0	42.72%
100%	10362.4	75606.0	43.32%

	Calculation time (min)
Generating paths	16.79
Dynamic Greedy	172.44
Total calculation time	191.09

Table 16 Statistics for K=3

K=4

% Budget	Budget	# connected	% connected
1.0%	97.86	16941.0	9.71%
2.5%	257.63	23506.0	13.47%
5.0%	517.44	30001.0	17.19%
7.5%	765.22	34895.0	19.99%
10.0%	1027.39	38468.0	22.04%
15.0%	1527.25	44768.0	25.65%
20.0%	2066.77	50587.0	28.98%
25.0%	2587.06	54941.0	31.48%
30.0%	3104.86	58673.0	33.61%
35.0%	3614.83	61718.0	35.36%
40.0%	4136.78	64106.0	36.73%
45.0%	4657.66	66491.0	38.09%
50.0%	5174.89	68394.0	39.18%
55.0%	5688.17	70148.0	40.19%
60.0%	6205.96	71438.0	40.93%
70.0%	7238.51	73403.0	42.05%
80.0%	8257.03	74575.0	42.73%
100%	10350.59	75606.0	43.32%

	Calculation time (min)
Generating paths	20.59
Dynamic Greedy	176.35
Total calculation time	256.76

Table 17 Statistics for K=4

K=5

% Budget	Budget	# connected	% connected
1.0%	98.55	16982.0	9.73%
2.5%	258.55	23399.0	13.41%
5.0%	517.11	29870.0	17.11%
7.5%	774.68	34997.0	20.05%
10.0%	1029.02	38458.0	22.03%
15.0%	1547.87	44829.0	25.68%
20.0%	2044.64	50477.0	28.92%
25.0%	2583.58	54996.0	31.51%
30.0%	3102.52	58758.0	33.66%
35.0%	3617.38	61810.0	35.41%
40.0%	4135.68	64174.0	36.77%
45.0%	4653.75	66504.0	38.1%
50.0%	5169.85	68406.0	39.19%
55.0%	5687.7	70168.0	40.2%
60.0%	6203.04	71544.0	40.99%
70.0%	7233.31	73403.0	42.05%
80.0%	8274.3	74621.0	42.75%
100%	10342.9	75606.0	43.32%

	Calculation time (min)
Generating paths	23.11
Dynamic Greedy	194.23
Total calculation time	219.54

Table 18 Statistics for K=5

K=7

% Budget	Budget	# connected	% connected
1.0%	103.48	17243.0	9.88%
2.5%	258.71	23556.0	13.5%
5.0%	517.42	29950.0	17.16%
7.5%	776.13	35101.0	20.11%
10.0%	1034.85	38562.0	22.09%
15.0%	1552.27	44948.0	25.75%
20.0%	2069.69	50720.0	29.06%
25.0%	2587.11	55054.0	31.54%
30.0%	3104.54	58812.0	33.69%
35.0%	3621.96	61847.0	35.43%
40.0%	4139.38	64216.0	36.79%
45.0%	4656.8	66531.0	38.12%
50.0%	5174.23	68434.0	39.21%
55.0%	5691.65	70191.0	40.21%
60.0%	6209.07	71642.0	41.05%
70.0%	7243.92	73427.0	42.07%
80.0%	8278.76	74629.0	42.76%
100%	10348.45	75609.0	43.32%

	Calculation time (min)
Generating paths	24.31
Dynamic Greedy	221.02
Total calculation time	247.38

Table 19 Statistics for K=7

K=9

% Budget	Budget	# connected	% connected
1.0%	98.72	17007.0	9.74%
2.5%	258.55	23465.0	13.44%
5.0%	515.91	29908.0	17.13%
7.5%	775.43	35068.0	20.09%
10.0%	1034.01	38583.0	22.1%
15.0%	1551.01	44916.0	25.73%
20.0%	2042.13	50514.0	28.94%
25.0%	2581.72	55042.0	31.53%
30.0%	3102.07	58814.0	33.7%
35.0%	3615.79	61864.0	35.44%
40.0%	4136.83	64241.0	36.8%
45.0%	4653.52	66546.0	38.13%
50.0%	5170.8	68445.0	39.21%
55.0%	5686.14	70197.0	40.22%
60.0%	6204.56	71651.0	41.05%
70.0%	7236.49	73425.0	42.07%
80.0%	8269.29	74627.0	42.76%
100%	10342.18	75609.0	43.32%

	Calculation time (min)
Generating paths	32.16
Dynamic Greedy	246.67
Total calculation time	281.18

Table 20 Statistics for K=9

K=11

% Budget	Budget	# connected	% connected
1.0%	103.29	17271.0	9.89%
2.5%	258.29	23614.0	13.53%
5.0%	515.66	29935.0	17.15%
7.5%	774.67	35080.0	20.1%
10.0%	1031.36	38564.0	22.09%
15.0%	1550.13	44783.0	25.66%
20.0%	2049.01	50576.0	28.98%
25.0%	2583.58	55056.0	31.54%
30.0%	3100.53	58808.0	33.69%
35.0%	3616.39	61872.0	35.45%
40.0%	4127.0	64195.0	36.78%
45.0%	4650.11	66540.0	38.12%
50.0%	5164.26	68434.0	39.21%
55.0%	5679.05	70187.0	40.21%
60.0%	6200.34	71638.0	41.04%
70.0%	7228.09	73407.0	42.06%
80.0%	8265.81	74623.0	42.75%
100%	10335.64	75609.0	43.32%

	Calculation time (min)
Generating paths	32.02
Dynamic Greedy	247.63
Total calculation time	282.02

Table 21 Statistics for K=11

K=13

% Budget	Budget	# connected	% connected
1.0%	98.72	16988.0	9.73%
2.5%	257.35	23419.0	13.42%
5.0%	516.56	29900.0	17.13%
7.5%	764.54	34893.0	19.99%
10.0%	1032.27	38559.0	22.09%
15.0%	1550.57	44934.0	25.74%
20.0%	2043.74	50524.0	28.95%
25.0%	2583.78	55054.0	31.54%
30.0%	3100.38	58806.0	33.69%
35.0%	3618.38	61882.0	35.45%
40.0%	4135.66	64244.0	36.81%
45.0%	4652.82	66545.0	38.12%
50.0%	5169.31	68441.0	39.21%
55.0%	5682.94	70200.0	40.22%
60.0%	6201.51	71660.0	41.06%
70.0%	7236.11	73430.0	42.07%
80.0%	8268.31	74632.0	42.76%
100%	10340.36	75609.0	43.32%

	Calculation time (min)
Generating paths	39.0
Dynamic Greedy	266.29
Total calculation time	307.45

Table 22 Statistics K=13

C.2 Results of logarithmic area under curve and sum of relevance ratios

	Relevance ratio sum	AUC	Deviation	Log AUC	Deviation
2	591.55	644444547.5	1132230.5	48.61	-0.32
3	607.32	643982716.64	670399.63	49.11	0.18
4	609.67	643217801.98	-94515.02	49.17	0.25
5	607.48	642867400.39	-444916.62	49.09	0.16
7	603.99	642929682.92	-382634.08	48.64	-0.29
9	608.07	643239379.21	-72937.79	49.08	0.15
11	603.85	642672406.4	-639910.6	48.63	-0.29
13	608.51	643144600.99	-167716.01	49.07	0.15

Table 23 Statistical results for the different Pareto measures. This includes the sum of the relevance ratios of al samples, the AUC (and the deviation from the mean) and the logarithmic AUC (and the deviation from that mean).

D Frequency of cutting planes in optimization

Cutting plane	Total times applied	Times present in scenarios
Gomory	173	13
Cover	7	3
Clique	20147	11
MIR	108	9
StrongCG	10	3
GUB cover	14	4
Zero half	5586	10
RLT	1960	14
Learned	0	0
Implied bound	0	0
Flow cover	0	0

Table 24 Results for cutting plane analysis for $n = 96$ heatlchare facility area scenarios. First column displays the total number of times a cutting plane has been applied over all the scenarios. The second column portrays how often the cutting plane was present at least once in a scenario (this was done because some cutting planes are applied very often once they are applied).