PROJECT WORK (Projektarbeit Master) FS 2015

Development of a GIS based optimization model to determine optimal work-zones for large scale infrastructure networks

Author(s): Marcel Burkhalter ETH-Nr. 10-914-976

Supervisor: Prof. Dr. Bryan Adey
Dr. Nam Lethanh

Zürich, 29. Mai 15

ETH Zürich
Institut für Bau- und Infrastrukturmanagement
Professur für Infrastrukturmanagement
Professor Dr. Bryan T. Adey

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Development of a GIS based optimization model to determine optimal work-zones for large scale infrastructure networks

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Burkhalter	Marcel

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date
Zürich, 27.5.2015

Signature(s)
M Burkhalter

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper

Abstract

Road managers have to deal with intervention on their network over and over again. The agency, user and society have to be considered in evaluating where, when and what for an intervention is executed. Beside the costs and benefits for every possible intervention on each object some constraints needs to be taken into account. Work-zones have some limitation in their own length and in the distance between two of them. Additionally, the agency has a limitation in its budget for one time period.

An optimization model is developed which determine the optimal work-zones on a road network considering all those constraints. It is GIS based what leads to a robust model for the application of large scale, real world networks. The model is proofed with an application to the road network of the canton of Valais containing almost 2'000 objects. Four different scenarios are examined and their results are discussed.

Acknowledgment

First of all I like to thank Dr. Nam Lethanh. For his kind and motivating supervision I would like to express my deep gratitude to him.

Second I like to thank Prof. Dr. Bryan Adey which gave me the basics to work out this master project work.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem statements

Every object in a network has a deterioration process which leads to decreasing performance of the object. In a road network those objects are plain roads, tunnels and bridges. To hold those on a required level of service a lot of interventions have to be performed over their entire life. Infrastructure manager have to decide where on the network intervention have to be executed. Answering this question needs a deeper view into the costs and benefits from the owner, user and society. However, Not only the interventions have to be defined also the used traffic configuration during the execution of an intervention and possible groupings of construction zones into one single work-zone should be considered. Exactly this grouping of interventions into work-zones can have a huge positive impact on the costs. To install two traffic configurations separately from each other on two close-by objects has higher costs than to install one single traffic configuration over the two objects. Nevertheless, there are some constraints which have to be considered. Due to the driver safety the size of those work-zones cannot be too long and two different work-zones cannot be too close to each other. On the other hand, the Budget for one time period is limited too.

A basic model has been developed which determine the optimal work-zones (Hajdin und Adey 2006). However, a database based model is required to have a simple set up for large scale, real networks. Further on, the models should be GIS based to get understandable and nice graphical output.

1.2 Purpose

In this work a GIS based optimization model is developed which determine the optimal work-zones on a road network while considering the budget and length constraints and the agency, user and society costs. The model should be inside a GIS based structure that allows a robust application of the model for large scale infrastructure network. This developed model is then demonstrated the with a real world example.

2 State of research

2.1 First models

In 2006 Hajdin and Adey presented an optimization model able to determine the optimal worksite on a highway network. This model considers the agency and user costs and finds the worksite which gives the highest benefit by undercutting the limited budget. Beside this budget constraint it includes a maximum worksite length constraint (Hajdin und Adey 2006). This first model has some limitations. It is only able to find the optimal worksite and cannot be used to optimize multiple work-zones. Nevertheless, this was the starting point for this project. In 2007 Hajdin and Lindemann extended the algorithm with a minimal distance between work-zones constraint, which allows using the model for multiple work-zones. The difference between a worksite and a work-zone can be seen in figure 2.1. A work-zone is defined as a zone with the same traffic configuration. It can contain multiple worksites and areas where no intervention is executed between those worksites.

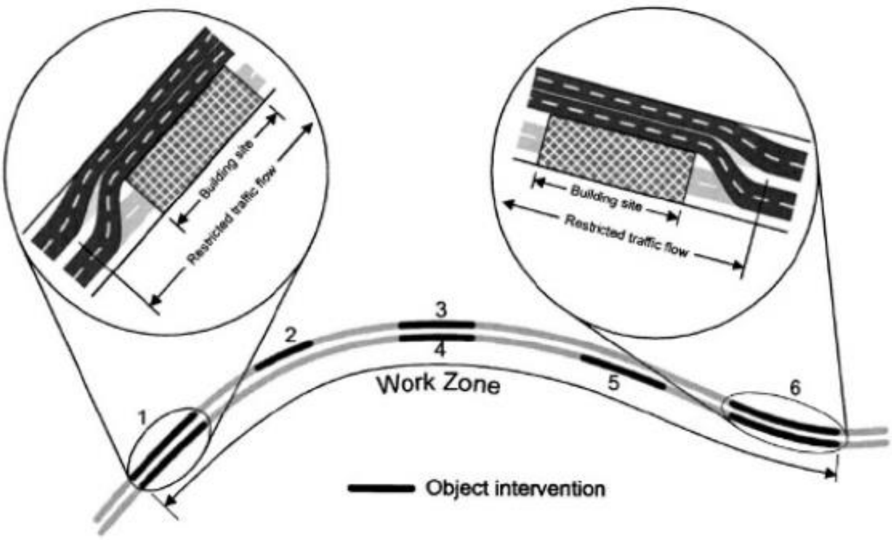


Figure 2.1: Systematics of worksites and work-zones

However, so fare the model could only be applied on linear networks and they are manually set up in Excel therefore they are not useful for large scale, real world networks. Within the framework of a master project C. Eicher expanded the model further (Eicher 2014). In his work he came up with a solution for networks which containing loops and he created an application to set up the optimization problem automatically.

3 The optimization model

3.1 Mathematical description

The present model was developed by the works showed in the last chapter. Here it is presented in detail.

3.1.1 Terminology

Objects: $n = 1,\ldots,N$ — N: total number of Objects

Interventions: $k_n = 1,\ldots,K_n$ — K_n: total number of different interventions per object n

3.1.2 Objective function

The objective function is the main function in the model. It maximizes the total positive impacts of the interventions.

$$Max\ Z = \sum_{n=1}^{N} \delta_{n,k} * (B_{n,k} - C_{n,k}) \tag{1}$$

Where $\delta_{n,k}$ is a binary variable, which is 1 if the intervention k is executed on object n. $B_{n,k}$ and $C_{n,k}$ are the benefits and costs of intervention k on object n.

3.1.3 Constraints

The optimization model has multiple constraints. Firstly, there is the continuity constraint, which makes sure that for each object n exactly one intervention k is selected.

$$\sum_{k=1}^{K} \delta_{n,k} = 1\ \forall n \tag{2}$$

Secondly, it must be possible to set a limitation of the intervention cost in total. Therefor the budged constraint is formulated.

$$\sum_{n=1}^{N}\sum_{k=1}^{K} \delta_{n,k} * C_{n,k} \leq \Omega \tag{3}$$

Where Ω is the maximum budget.

Further on, there exist boundaries in the areal extension of work-zones. One work-zone is limited by its length and two work-zones cannot be too close to each other. Therefore, the maximum work-zone length constraint and the minimal distance between work-zone constraint are established.

$$\sum_{l=a_l^w}^{e_l^w}\sum_{n=a_n^w}^{e_n^w} \lambda_{l,n} \leq \Lambda^{MAX}\ \forall w \tag{4}$$

Where Λ^{MAX} is the maximum work-zone length, $\lambda_{l,n}$ is the length of the object [l,n], a^w and e^w (with a_l^w = l and $a_n w = n$) are the first respectively the last object in the work-zone w.

$$\sum_{l=a_l^d}^{e_l^d} \sum_{n=a_n^d}^{e_n^d} \lambda_{l,n} \geq \Lambda^{MIN} \ \forall d \tag{5}$$

Where Λ^{MIN} is the minimal distance between two work-zones, a^d and e^d are the first respectively the last object in the default section d.

The last two constraints, the maximum work-zone length and the minimum distance between work-zones, are combined into a single constraint containing a matrix with all not possible combinations due to the length constraints.

$$\sum_{n=1}^{N} \sum_{k=1}^{K} \delta_{n,k} * \gamma_{n,k,i} \leq 1 \ \forall i \tag{6}$$

Where $\gamma_{n,k,i}$ is a I-J matrix, with I the total number of not possible combinations and J the sum product of objects and possible interventions per each object.

4 Program algorithm

4.1 Overview

The main goal of this work is to implement the optimization model on a GIS based network. Therefore, an algorithm is developed which is able to take the input network from a database, transform it in a way it contains all used information, set up the optimization, run the optimization and store the result back in the database. This structure can be applied on every GIS platform to do a precise graphical modelling of the optimization model on a network. Due to its connection with a database it can easily be expanded for multiple time periods by updating the condition state after executing an intervention.

For this work postgreSQL is used to perform database queries and MATLAB to run the optimization process. So far the optimization was run in Excel with the What's Best! solver. In the first models all used matrices have been constructed manually or with in-build Excel functions. C. Eicher developed a MATLAB program which sets up all used vectors and matrices for the optimization. This routing algorithm is reused in the presented work to set up a MATLAB based optimization. With the optimization done in MATLAB two memory consuming steps (import to Excel and export again) are eliminated and the manually arrangement of the What’s Best! solver can be omitted.

In the next sections an overview over the MATLAB and SQL program are given. The more important codes from this two programs are well commented in the appendix (B, C and D).

4.2 Transform data inside SQL

The input data can have any style of setup and the values can have all kind of characteristic. There does not have to be an object identification variable at all. Therefore, the data has to be set up in a way that it can be used. Two main steps can be separated in the setting up of the table. The developed SQL query is *sql_1_generate_input*. This already includes the cost calculation as a third step.

1. Producing a network with serial IDs and the start and end node of each object.
2. Add additional object information for later use.
3. Calculating the costs and benefits for every possible intervention of each object.

For the first step, a new table (the main table) is generated with the determination of a new serial ID for each object to get a clear and unique identification of the objects. After its creation an extension of postgreSQL is used on the table: **pgrouting**. Pgrouting contains a lot of powerful network functions. After it is inserted in postgreSQL the start and end nodes can be calculated by using the *pgr_createTopology(table, precision, geometric data, object id)* function. It uses the geometric network data *the_geom* to create all nodes in the network and to assign them to the right objects as start and end nodes. Therefore, the table needs some predefined source and target columns.

For the second and third step some additional tables with more cost information are used in the database. Those are:

- **Conditional state (CS):** Contains all possible CSs an object can be in and the related costs to this CS.
- **Intervention (I):** Contains all I which could be executed on an object and its related costs.
- **Traffic configuration (TC):** Contains all TC which could be executed on an object and its related costs.
- **Cost variables:** All variables used for the cost calculation have to be stored in a table to be used later in the calculation step.

The second step adds all used information to the objects which are missing in the data so far. For example, it has to be defined which two traffic configuration could be applied on each object and the condition state of each object has to be included.
In the third and last step of the set up query the costs and benefits are calculated for every traffic configuration that can be executed on each object. This is done in an extra table where a detailed cost calculation is performed and only the final results for each traffic configuration are taken back into the main table. Now the table is set up to run the MATLAB process including the optimization.

4.3 MATLAB

In this section the MATLAB algorithm is explained which is used to first set up and later to run the optimization. In previous models the optimization was run in Excel with the *What's Best!* solver. However, if MATLAB is already used to set up the constraints it can also be used for the optimization itself.

4.3.1 Importing SQL data

Before the optimization can be set up the data has to be imported. Therefore, a connection to the SQL database is established inside MATLAB by using the *database()* function. To be able to connect MATLAB with a database in postgreSQL, the JDBC driver has to be installed. After the connection stands, an SQL query can be executed directly from MATLAB which imports the data.

4.3.2 Set up the optimization

To set up the optimization the routing algorithm is used developed by C. Eicher. The input values for this algorithm are:

- the network data with the object-ID, start-node, end-node and object length
- information about how many different interventions can be applied on each object
- the maximum work-zone length
- the minimum distance between work-zones

The Algorithm constructs the following output:

- **Continuity matrix:** An *N-M* matrix where *N* is the number of objects and *M* the sum product of possible interventions *k* on each object *n*. It contains ones if the object *n* is physically connected with the object considered in *m*.
- **RHS vector:** *N* long vector containing the numbers of neighbouring objects for each object *n*.
- **Combination matrix:** An *I-M* matrix where *I* is the number of not possible combinations of selected objects.

4.3.3 Optimization

For the optimization some further inputs are necessary. The costs and benefits for every possible intervention on each object as well as the budget limitation are needed. With those values the MATLAB function *intlinprog(f, intcon, A, b, Aeq, beq, lb, ub)* can be used to calculate the binary values $\delta_{n,k}$ for the combinations of objects n and interventions k. The input variables have the flowing characteristics.

- **Objective function (*f*):** $\sum \delta_{n,k} * B_{n,k} * (-1)$
  Sums up the benefits of the combinations multiplied by their binary variables. Because the *intlinprog()* is minimizing the objective function, it has to be multiplied by (-1).
- **Smaller than constraints (*A, b*):** $A * \delta_{n,k} \leq b$
  These are the combination and budget constraints.
- **Equal than constraints (*Aeq, beq*):** $Aeq * \delta_{n,k} = beq$
  In *Aeq* and *beq* the continuity constraint is included.
- **Binary declaration (*intcon, lb, ub*):** *intcon* defines the variables $\delta_{n,k}$ as integers and $lb=0$ and $ub=1$ defines the lower and upper bound.

4.3.4 Exporting to SQL

Before the export to the SQL the binaries have to be transformed in a table that shows for every object which intervention is selected.

This table is exported to a new SQL table *matlab_export* by using the established connection between the MATLAB and the database.

4.4 Complete the graphical output in SQL

In the last part the results from the MATLAB optimization are inserted in the general SQL network table what leads to a single table containing all information about the network, the costs and benefits for every object and the executed interventions. This table can be graphical represented in QGIS to show where the optimal work-zones are inside the network.

5 GIS based application on the road network of Valais

5.1 Geographical field of study

To present the developed algorithm it is applied on the road network of the canton of Valais. It is an interesting example because of its different road types. The high mountain roads have a different characteristic than the roads in the relatively warm valley. The road network itself is quite linear due to the surrounding mountains. However, some areas inside the whole network have a more nested network type what makes it to a nonlinear network.

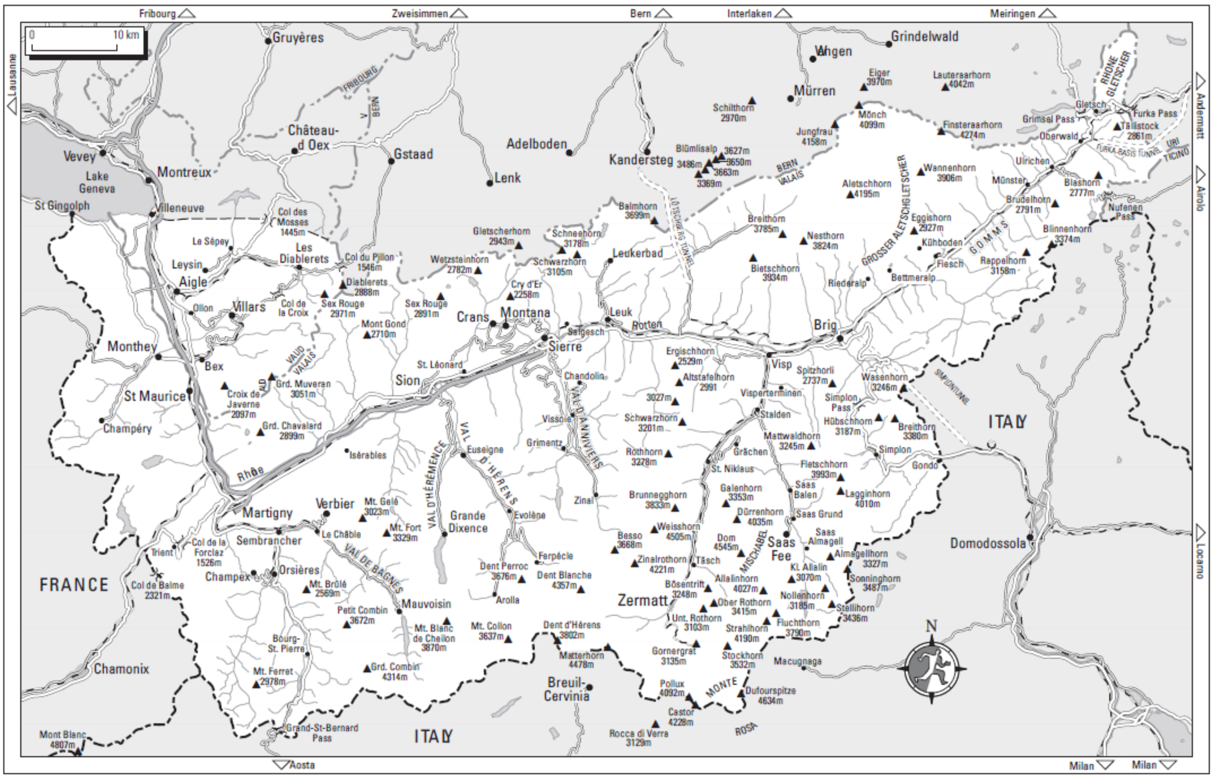


Figure 5.1: Map of the canton of Valais (www.Switzerland.isyours.com)

5.2 Examined scenarios

For this case study three different scenarios are compared with a reference scenario (Table 5-1). The constraints for the reference scenario are *2'000m* of maximal work-zone length, *3'000m* as a minimal distance between two work-zones and an unlimited budget. In each of the three scenarios one constraint is changed. *Scenario 1* has a budget limitation of *20Mio CHF*. In *Scenario 2* the maximal work zone length is limited to *1'000m*. For the *scenario 3* the minimal distance between two work-zones is reduced to *2'000m*.

Table 5-1: Evaluated scenarios.

Scenario	Budget [CHF]	Maximum work-zone length [m]	Minimal distance between work-zones [m]
Reference	Unlimited	2’000	3’000
Scenario 1	**20 Mio**	2’000	3’000
Scenario 2	Unlimited	**1’000**	3’000
Scenario 3	Unlimited	2’000	**2’000**

5.3 Interventions

5.3.1 Condition state

An object can be in five different condition state (CS). In CS 1 and 2 the object is in an excellent or good condition what provides an adequate level of service. CS 3 is barely adequate and maintenance work should be considered. CS 4 and 5 are inadequate level of service. In this case study the condition states are randomly selected with a steady distribution over all five condition states.

5.3.2 Intervention types

For objects with a condition state two or better no intervention is defined. For objects in condition state three a facing improvement of rough pavement is executed. Roads in condition state four get a resurfacing on top of the existing subgrade with milling (100mm). Is a section in condition state five a complete renewal of the surface, binder and base (200mm) course is executed.

5.3.3 Traffic configuration

For every object three traffic configurations (TC) are selected. Firstly, each object has its regular TC_0. The second and third TCs are based on the road type. Due to missing information about the number of lines on the objects, assumptions have to be made. It is assumed that highways have four lanes plus two emergency lanes where a “4-0” or a “H2-N2” TC can be applied. All other roads have two lanes where “1-0” and “N2” are considered as TC. Figure 5.2 and figure 5.3 shows the chosen traffic configurations.

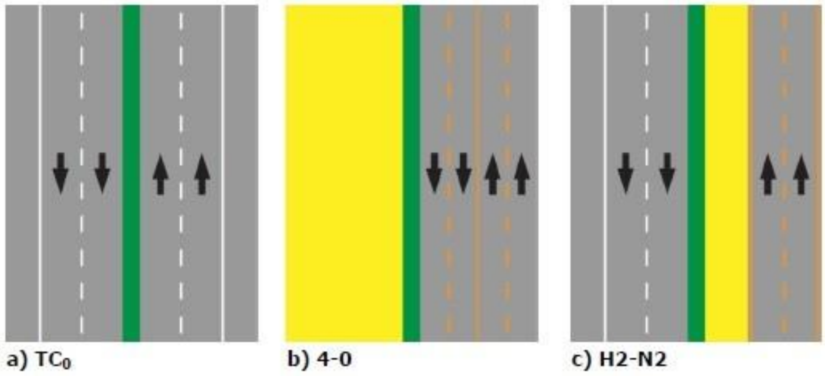


Figure 5.2: TC on highways. (a) TC_0, (b) TC_1 = “4-0”, (c) TC_2 = “H2-N2”

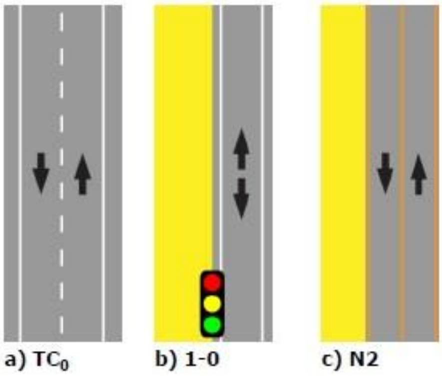


Figure 5.3: TC on other roads than highways. (a) TC_0, (b) TC_1 = “1-0”, (c) TC_2 = “N2””

If an object of CS 1 or 2 is surrounded by objects with CS ≥ 3 an optimal work-zone can be determined which over span the object in CS 1. Then the traffic configuration on these objects is changed too, but no intervention is executed on it.

5.4 Simplification of the network

The basic network table is given in a SQL database where the roads are split up into different objects. The two directions of a highway are modelled by two separated chains of objects. This means that every part of the highway has to parallel objects, each modelling one side of the highway. In the optimization model those parallel objects are considered as unrelated objects. For determine the optimal work-zones those parallel highway objects cannot be seen as unrelated, because an intervention is connected to both side of a highway. To find a GIS based solution in a way that those objects could be connected is going over the scope of this work. Therefore, the data is simplified manually so that the highways are also represented only by one chain of objects. Figure 5.4 shows the difference between the real QGIS data and the simplified data.

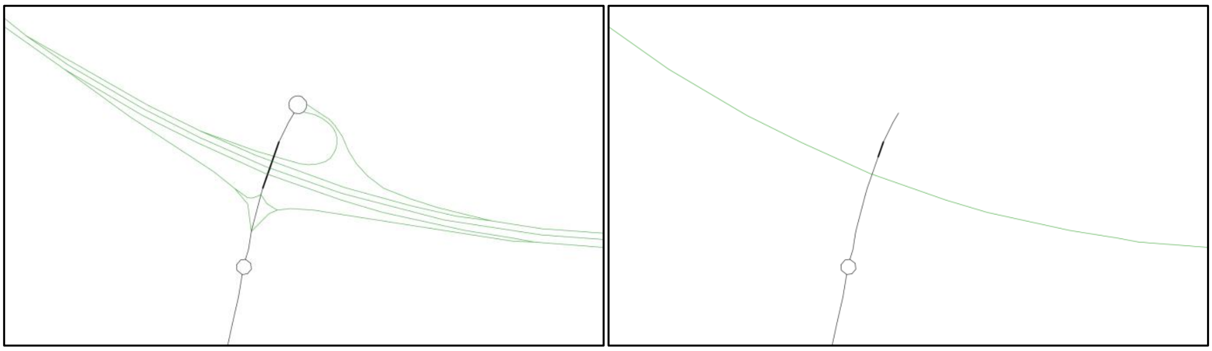


Figure 5.4: (left) Real GIS data with parallel string of objects for highways. (right) Simplified network.

5.5 Base table

The simplified input data contains *1'959 objects* with a total length of *627km*. Every object has an allocated road type (motorway, primary, secondary and mountain). Beside the road type the object type is divided too. Bridges and tunnels have a positive binary in the column Bridge / Tunnel. The traffic volume and the maximal speed on each object is given as additional information. The whole network contains *1'781 nodes* which are the starting and end nodes of the objects.

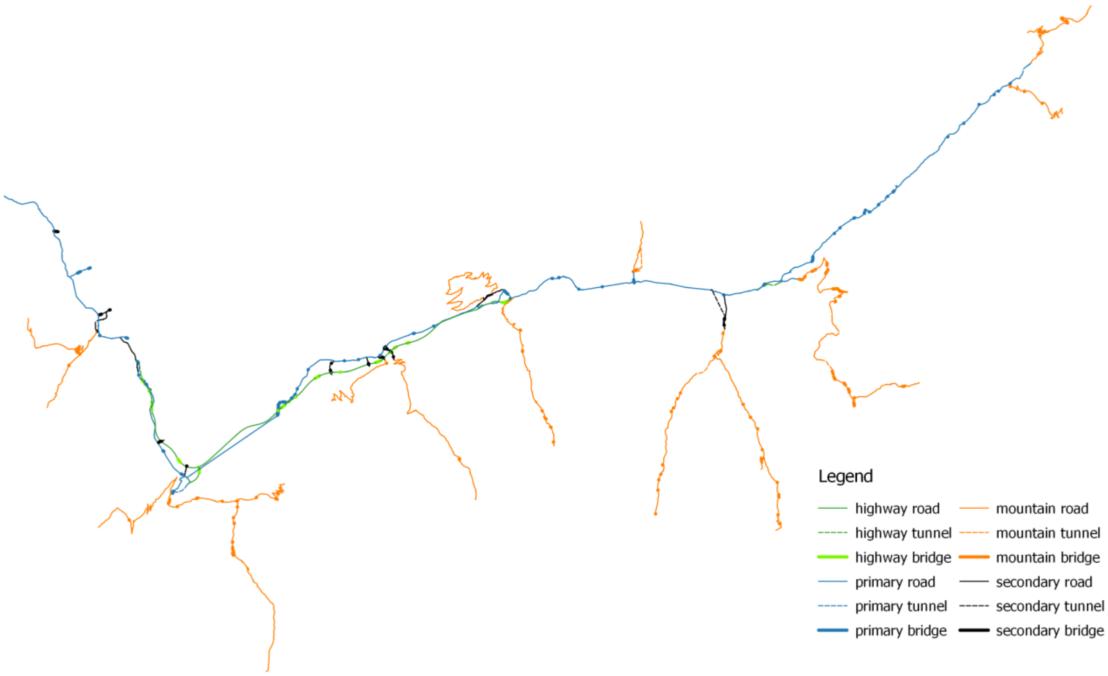


Figure 5.5: General road network of the canton of Valais with four different road types.

5.6 Cost and benefit calculation

After the whole network data is defined the costs and benefits can be calculated. In a first step the costs and benefits are estimated for different impacts. Beside the general calculation an example for one single object of the whole network is given. This object is a *primary road* with a length of *457m* and a width of *11m*. The speed limitation is *80km/h* and the daily traffic volume is *9'400cars/day* respectively *600trucks/day*. Its condition state is *5* and the two traffic configuration adjusted to it are "1-0" for TC_1 and "N2" for TC_2.

5.6.1 Intervention cost

The first costs are related directly to the intervention itself. They are split up into two different types of direct intervention costs (fix and variable) and in additional costs due to the traffic configuration *TC*. The first are the fixed intervention costs $CI_{n,I}^{f}$. The second are variable and size dependent intervention costs $CI_{n,I}^{v}$. Those two together with the additional costs ($CI_{n,TC}^{add}$) gives the total intervention costs for object *n* in traffic configuration *TC* for intervention *I*.

$$CI_{n,I,TC} = CI_{n,I}^{f} + CI_{n,I}^{v} + CI_{n,TC}^{add} \tag{7}$$

Where $CI_{n,I,TC}$ are the total intervention costs.

All costs that are not directly connected to the length of the object are part from the fixed intervention costs. This includes construction site equipment such as electricity, access roads, sanitary equipment, etc.. The fixed intervention costs can be calculated with the following therm.

$$CI_{n,I}^{f} = (1 - k_n^B - k_n^T) * CIF_I^R + k_n^B * CIF_I^B + k_n^T * CIF_I^T \tag{8}$$

Where CIF_I^R, CIF_I^B and CIF_I^T are the fix intervention costs for road sections, bridges and tunnels, k_n^B and k_n^T are binary values for the object *n* to be a bridge or a tunnels.

The variable costs include all costs which depend on the size of the object. For example: the amount of used material or fuel for the machines. Also the hours of manpower is depending on the size of the object. To calculate those, we can use equation (9).

$$CI_{n,I}^{v} = \left((ci_I^R * (1 - k_n^B - k_n^T) + k_n^B * ci_I^B) * w_n + k_n^T * ci_I^T\right) * L_n \tag{9}$$

Where ci_I^R, ci_I^B and ci_I^T are the variable costs for a road section, bridge or tunnel depending on the intervention type *I*. L_n and w_n are the length and the width of object *n*.

Table 5-2 shows estimated fix and variable costs for road sections. These costs come from a report published by the "Bundesamt für Strassen" (ASTRA 2008). For bridges and tunnels does not exist a similar report. However, in table 5-3 and table 5-4 are estimated values for Bridges and Tunnels which are roughly approximated by members of the institute for geotechnical engineering (IGT) and of the institute of construction and infrastructure management (IBI) at the ETH Zurich.

Table 5-2: Intervention costs and producing capacity for road sections.

CS	Intervention	Fix costs [CHF]	Variable costs [CHF/m2]	Production capacity [m2/day]
CS 3	FI	3'500	8.00	3'000
CS 4	RT	4'100	52.00	6'000
CS 5	CR	9'600	108.80	1'000

Table 5-3: Intervention costs for bridges.

CS	Fix costs [CHF]	Variable costs [CHF/m2]
CS 3	20'000	2'100
CS 4	30'000	2'800
CS 5	40'000	3'500

Table 5-4: Intervention costs for tunnels.

Condition state	Fix costs [CHF]	Variable costs [CHF/m']
CS 3	100'000	20'000
CS 4	150'000	35'000
CS 5	200'000	50'000

The costs so far do not depend on the traffic configuration. Nevertheless, some traffic configurations have a significant impact on the intervention costs. The following points are considered in this paper.

- An additional cost of 20 % of the total fix and variable costs is taken into account for traffic configuration where the work is under traffic.
- Additional 10 % are added if more than one shift is used per day (night work).
- If the traffic configuration needs a traffic light, an additional 15'000 CHF are added.

Example:

$$CI_I^f = (1-0-0) * 9'600\, CHF/I = 9'600\, CHF/I$$

$$CI_I^v = (109\, CHF/m^2 * (1-0-0)) * 11m * 457m = 547'943\, CHF/I$$

$$CI_{TC=1}^{add} = 0.2 * (9'600 + 547'943) + 15'000\, CHF/I = 126'509\, CHF/I$$

$$CI_{TC=2}^{add} = 0.2 * (9'600 + 547'943) = 111'509\, CHF/I$$

$$\boldsymbol{CI}_{\boldsymbol{TC=1}} = (9'600 + 547'943 + 126'509) = \boldsymbol{684'052\, CHF/I}$$

$$\boldsymbol{CI}_{\boldsymbol{TC=2}} = (9'600 + 547'943 + 111'509) = \boldsymbol{669'052\, CHF/I}$$

5.6.2 Costs for travel time loss

Costs due to travel time loss are indirect impacts from interventions. Those can be calculated with the travel speed, which is related to the traffic configuration.

Axhausen investigated in his paper values for opportunity costs for people driving cars (Axhausen, et al. 2006). On average the value for one hour of travel is *18.2 CHF* and for the special purpose of business it is *32.5 CHF*. Different studies deal with the values for trucks but they vary in a huge way. The variety lies between *38 US-Dollars* (De Jong 2008) and *371 US-Dollars* (Small, et al. 1999). For this paper we assume $TTS^C = 18.2\ CHF/h$ for cars and

TTS^T = *132.5 CHF/h* for trucks, which contains the *32.5 CHF* for business trips and an additional *100 CHF* for the big variety.
The costs for one day can be calculated with the following equation.

$$CLT_{n,TC}^{d} = \left(\frac{L_n}{v_{TC}} - \frac{L_n}{v_0}\right) * (DTV_n^T * TTS^T + DTV_n^C * TTS^C) \qquad (10)$$

Where L_n is the length of object n, v_0 is the original travel speed, v_{TC} is the reduced travel speed due to the changed TC, DTV_n^T is the mean daily truck traffic on n and DTV_n^C is the mean daily car traffic.
Since this costs only occur during the intervention time it must be multiplied with the duration of the changed traffic configuration (= duration of intervention).

$$CLT_{n,TC} = CLT_{n,TC}^{d} * \ \varphi_{n,I,TC} \qquad (11)$$

Where $CLT_{n,TC}{}^{d}$ are the costs for one day and $\varphi_{n,I,TC}$ is the duration of the intervention I and the traffic configuration TC.

Example:

Duration of intervention:

$\varphi \qquad = {(457m * 11m)}\big/{(1'000\, m^2/day)} = 5.03\ days$

Travel time loss costs:

$CLT_{TC=1}^{d} \qquad = \left({0.457km}\big/{(0.5 * 80\ km/h)} - {0.457km}\big/{(80\ km/h)}\right) * (600\ trucks/day * 132.5\ CHF/h + 9'400\ cars/day * 18.2\ CHF/h) = 1'431\ CHF/day$

$\boldsymbol{CLT_{\ TC=1}} \qquad = 1'431\ CHF/day * 5.03 days = \mathbf{7'198}\ \boldsymbol{CHF/I}$

$CLT_{TC=2}^{d} \qquad = \left({0.457km}\big/{(80\ km/h - 20\ km/h)} - {0.457km}\big/{(80\ km/h)}\right) * (600\ trucks/day * 132.5\ CHF/h + 9'400\ cars/day * 18.2\ CHF/h) = 477\ CHF/day$

$\boldsymbol{CLT_{\ TC=2}} \qquad = 477\ CHF/day * 5.03 days = \mathbf{2'400}\ \boldsymbol{CHF/I}$

5.6.3 Accident cost

There are two kinds of accident costs which have to be considered. Firstly, there are the regular accident costs. They are generated by accidents in normal traffic configuration. The best way to take those costs into account is by evaluating the generated costs per vehicle kilometer. Sigrist suggested a value of *0.23 CHF per vehicle kilometer*, which is a good enough estimation for this work (Sigrist 2012).

$$CA_n = L_n * (DTV_n^T + DTV_n^C) * C_A * 365 \qquad (12)$$

Where CA_n is the regular accident cost on object n and C_A is the value for accident costs per vehicle kilometer (0.23 CHF/km).
Secondly, there are increased accident costs due to changed traffic configuration ($CA_{n,TC}$). They occur because narrower roads at work-zones increase traffic jams what leads to more accidents. A German studies estimated this additional costs by *56%* of the regular accident costs (Bakaba, et al. 2012). *56%* of *0.23 CHF/km* makes *0.13 CHF per kilometer*.

$$CA_{n,TC} = L_n * (DTV_n^T + DTV_n^C) * C_{A,TC} * \varphi_{n,I,TC} \qquad (13)$$

Where $CA_{n,I}$ is the value for additional accident costs per vehicle kilometer (0.13 CHF/km) due to a changed traffic configuration and $\varphi_{n,l,TC}$ is the duration of the changed TC.

Example:

$$CA = 0.457km * (9'400\, cars/day + 600\, trucks/day) * 0.23\, CHF/vehkm * 365days$$
$$= 383'652\, CHF/year$$
$$\boldsymbol{CA_{TC}} = 0.457km * (9'400\, cars/day + 600\, trucks/day) * 0.13\, CHF/vehkm * 5.03days$$
$$= \mathbf{2'988}\, \boldsymbol{CHF/I}$$

5.6.4 Vehicle cost

The vehicle costs are calculated using fuel consumption. For cars and trucks the costs are calculated separately, because of the big difference in fuel consumption between the two vehicle types. A third part are the maintenance costs. All three together give the total vehicle costs CV_n on object n.

$$CV_n = (CV_n^C + CV_n^T + CV_n^M) * 365 \tag{14}$$

Where CV_n^C are the operation costs for cars, CV_n^T are the operation costs for trucks and CV_n^M are the vehicle maintenance costs on object n.

The mean fuel consumption for cars lais around 6.7 liters per 100km (VCA & DFT 2011). On mountain roads cars need more fuel than on normal flat roads. Because there is no value for cars on mountain roads in the literature, we assume that the increase is similar to the one for trucks (+18%). The fuel consumption is also depending on the condition state CS. With decreasing condition state the fuel consumption is increasing. It is assumed that the increase in CS_3 is 5%, in CS_4 12% and in CS_5 20%. In 2014 the mean petrol and diesel price in Switzerland was *1.88 CHF/l* (BFS 2015).

$$CV_n^C = F^C * (1 + f_{n,MR}) * L_n * DTV_n^C * (1 + f_{n,CS}) * P_{fuel} \tag{15}$$

Where F^C is the mean fuel consumption of cars, $f_{n,MR}$ the fuel consumption increase for cars on mountain roads, L_n the object length, $f_{n,CS}$ the fuel consumption increase for cars due to the CS and P_{fuel} the mean fuel price.

The operation costs for trucks are calculated in a similar way. Instead of 6.7 liters per 100 km trucks need in average 33 liters per 100 km. The increase on mountain roads is taken into consideration by an additional 6 liters per 100 km (VBA 2008). The same values as for cars are taken for the increase due to the condition state of the road.

$$CV_n^T = (F^T * f_{n,MR}) * L_n * DTV_n^T * (1 + f_{n,CS}) * P_{fuel} \tag{16}$$

Where F^T is the mean fuel consumption of trucks, $f_{n,MR}$ the fuel consumption increase for trucks on mountain roads, L_n the object length, $f_{n,CS}$ the fuel consumption increase for trucks due to the CS and P_{fuel} the mean fuel price.

The maintenance costs include regular service on the vehicles such as tire replacement, oil change or car cleaning. A report from the touring club of Switzerland shows that 67% of the fuel costs are additionally spent for car maintenance in 2011 (TCS 2011). However, this value does not include the depreciation in value. The same report suggests taking another 51% of the fuel cost into account for the depreciation in value. Furthermore, it is assumed that these parameters are the same for both vehicle types.

$$CV_n^M = (\lambda^{SR} + \lambda^{DV}) * (CV_n^C + CV_n^T) \tag{17}$$

Where λ^{SR} is a factor for increased service and repair and λ^{DV} is a factor for depreciation in value.

Example:

With executing an intervention:

$CV_{I\neq 0}^C \quad = 6.7l/100km * (1+0) * 0.457km * 9'400\, cars/day * 1.88\, CHF/l = 541\, CHF/day$

$CV_{I\neq 0}^T \quad = (33l/100km + 0) * 0.457km * 600\, trucks/day * 1.88\, CHF/l = 170\, CHF/day$

$CV_{I\neq 0}^M \quad = (0.67 + 0.51) * (541\, CHF/day + 170\, CHF/day) = 839\, CHF/day$

$\boldsymbol{CV_{I\neq 0}} \quad = (541\, CHF/day + 170\, CHF/day + 839\, CHF/day) * 365 days = \mathbf{565'743\, CHF/year}$

Without executing an intervention:

$\boldsymbol{CV_{I=0}} \quad = 565'743\, CHF/year * (1 + 0.2) = \mathbf{678'891\, CHF/year}$

5.6.5 Emission cost

While driving, vehicles produce emissions which are bad for the environment. The biggest impacts are the emission of carbon dioxide (CO_2), carbon monoxide (CO), nitrogen oxide (NO_X), volatile organic compounds (VOC) and particulate matter (PM). Table 5-5 shows the composition of the exhaust fumes for petrol and diesel cars (Geringer und Tober 2015).

Table 5-5: Emission composition.

emission	petrol cars	diesel cars & trucks
CO2	20 %	20 %
CO	0.258 %	0.025 %
NOX	0.020 %	0.061 %
VOC	0.018 %	0.005 %
PM	0.000 %	0.005 %

In a media message from the Bundesamt für Statistik is stated that 25.6% of all cars in Switzerland are diesel cars. The others (74.4%) are assumed to be petrol cars, because only 1% are electric or hybrid cars (BFS 2015). For trucks we assume that all are diesel operated. As seen by the vehicle costs, a car uses on average 6.7l / 100km and is emitting 164.8 grams of CO_2 per kilometer (VCA & DFT 2011). For trucks we assume that the emissions are proportionally to the fuel consumption. That means 811.7 grams per kilometer.

For calculating the costs and estimation of the consequences has to be done, which is vague because the social an indirect impacts are hard to measure. Sigrist did an assumption for the economic impacts, which is used here (Sigrist 2012). Table 5-6 shows the amount of emissions by cars and trucks for each pollution type *pt* in gram per vehicle kilometer and also the costs in CHF per ton.

Table 5-6: Emission consumption and costs

Emission	w^C_{pt} [g/veh.km]	w^T_{pt} [g/veh.km]	Costs [CHF/ton]
CO2	164.8	811.7	22.05
CO	1.807	1.015	10'669.35
NOX	0.221	2.476	3'200.81
VOC	0.007	0.203	34.85
PM	0.131	0.203	9'033.39

With equation (18) the emission costs for one year can be calculated. Just as for the vehicle costs there is an increase in the emission costs due to both the condition state and road type (in this case on mountain roads). The same factors are assumed as for the vehicle costs due to the calculation of the emission costs based on the fuel consumption. This means 18% increase on mountain roads and 5% in CS_3, 12% in CS_4 and 20% in CS_5.

$$CE_n = (1 + f_{n,MR}) * (1 + f_{n,CS}) * \sum (w^C_{pt} * DTV^C_n + w^T_{pt} * DTV^T_n) * C_{pt} * 10^{-6} * 365 \qquad (18)$$

Where CE_n are the emission costs for object n, $f_{n,MR}$ and $f_{n,CS}$ are the increase factors for higher emission on mountain roads and due to the condition state, $w_{pt}{}^C$ and $w_{pt}{}^T$ are the amounts of emissions of pollution type *pt* for cars and trucks and C_{pt} are the costs for emission type *pt*.

Example:

With executing an intervention:

$\boldsymbol{CE_{I\neq 0}}$ $= (1 + 0) * 0.457km * 256\,CHF/day * 365days = \mathbf{93'354}\,\boldsymbol{CHF/year}$

Without executing an intervention:

$\boldsymbol{CV_{I=0}}$ $= 93'534\,CHF/year * (1 + 0.2) = \mathbf{112'241}\,\boldsymbol{CHF/year}$

5.6.6 Total cost and benefit

To calculate the total impacts from one intervention I on one object n we have to take two different situations into account. Firstly, the road is in a certain traffic configuration (not TC_0) because an intervention is performed. Secondly, the object can be in a non-normal traffic configuration, but no intervention is performed. The last situation can occur if on object n no intervention is executed but interventions are performed on the two adjacent objects (*n-1* and *n+1*). Another cas is if the total benefit from an intervention is lower than the total benefit of no intervention is perfromed even though the road is in a not-normal traffic configuration.

For the first situation (performing an intervention) equation (19) can be used. In it, the total costs for the intervention are subtracted from the generated benefits.

$$CB_{n,I,TC} = t * (\Delta CV_n + \Delta CE_n) - (CI_{n,I,TC} + CLT_{n,TC} + CA_{n,TC}) \qquad (19)$$

Where ΔCV_n and ΔCE_n are the difference between the vehicle resp. emission costs before and after an intervention and t is the time span for which the benefits are considered.

In the second situation is no intervention performed. That means all costs and benefits based on the intervention getting eliminated and only the costs due to a changed traffic configuration stay in the equation (20).

$$CB_{n,noI,TC} = -(CLT_{n,TC} + CA_{n,TC}) \qquad (20)$$

For the optimization problem, we use only the maximum benefit for each object ($CB_{max,n}$ = $Max(CB_{n,I,TC}$, $CB_{n,noI,TC})$). If the maximum occur from the second situation, the intervention costs are set zero otherwise the real intervention costs are taken to the optimization program.

Example:

$$\Delta CV_{I\neq 0} = (CV_{I\neq 0} - CV_{I\neq 0}) = (112'241\,CHF/year - 93'534\,CHF/year) = 205'775\,CHF/year$$

$$\Delta CE_{I\neq 0} = (CE_{I\neq 0} - CE_{I\neq =0}) = (678'891\,CHF/year - 565'743\,CHF/year) = 1'244'634\,CHF/year$$

Benefit with doing intervention and TC[1]:

$$\boldsymbol{CB_{I\neq 0,TC=1}} = t * (\Delta CV_{I\neq 0} + \Delta CE_{I\neq 0}) - (CI_{TC=1} + CLT_{TC=1} + CA_{TC})$$
$$= 15 years * (205'775\,CHF/year + 1'244'634\,CHF/year) - (684'052\,CHF/I + 7'198\,CHF/I + 2'988\,CHF/I) = \mathbf{1'283'587\,\boldsymbol{CHF}}$$

Benefit without doing intervention and TC[1]:

$$\boldsymbol{CB_{I=0,TC=1}} = -(CLT_{TC=1} + CA_{TC})$$
$$= -(7'198\,CHF/I + 2'988\,CHF/I) = \mathbf{-10'186\,\boldsymbol{CHF}}$$

For TC[1] the intervention is executed because $CB_{I\neq 0,TC=1} > CB_{I=0,TC=1}$.

Benefit for TC[1]: $\boldsymbol{MAX}(\boldsymbol{CB_{TC=1}}) = \mathbf{1'283'587\boldsymbol{CHF}}$

Benefit with doing intervention and TC[2]:

$$\boldsymbol{CB_{I\neq 0,TC=2}} = t * (\Delta CV_{I\neq 0} + \Delta CE_{I\neq 0}) - (CI_{TC=2} + CLT_{TC=2} + CA_{TC})$$
$$= 15 years * (205'775\,CHF/year + 1'244'634\,CHF/year) - (669'052\,CHF/I + 2'400\,CHF/I + 2'988\,CHF/I) = \mathbf{1'203'385\,\boldsymbol{CHF}}$$

Benefit without doing intervention and TC[2]:

$$\boldsymbol{CB_{I=0,TC=2}} = -(CLT_{TC=2} + CA_{TC})$$
$$= -(2'400\;\,CHF/I + 2'988\,CHF/I) = \mathbf{-5'388\,\boldsymbol{CHF}}$$

For TC[2] the intervention is executed because $CB_{I\neq 0,TC=2} > CB_{I=0,TC=2}$.

Benefit for TC[2]: $\boldsymbol{MAX}(\boldsymbol{CB_{TC=2}}) = \mathbf{1'203'385\,\boldsymbol{CHF}}$

5.7 Results

The graphical representations of the four scenarios are given in the appendix. In table 5-7 the selected objects are compared. In the reference scenario 682 objects are selected of which 10% are bridges and 1.6% are tunnels. The two scenarios with higher restrictions (Scenario 1 and 2) have a smaller proportion of selected bridges and tunnels. This can be explained by the higher intervention costs for those objects. If there are more restrictions less expensive objects are preferred which have a high benefit. Another effect of less selected object is that those who are selected have a higher condition state.

Table 5-7: Comparison of the selected objects.

Scenario	Object selected			Mean CS
	Total	% Bridge	% Tunnel	
Ref.	682	10.6%	1.6%	3.2
1 (budget)	159	2.5%	0.6%	3.6
2 (mdbw)	258	3.9%	0.8%	3.7
3 (mwzl)	949	9.5%	2.0%	3.2

Table 5-8 shows the comparison of the costs and benefits. In the reference scenario the costs are 95Mio CHF and the benefit 265Mio CHF. In this scenario the costs of an intervention does not matters due to the unlimited budget. Is the budget limited to 20Mio CHF the costs play a significant role. Objects with a high benefit/cost ratio are more likely to be chosen because they give a great benefit for low costs. Therefore,the B/C ratio is higher in scenario 1 than in the reference scenario. Shortening the maximal work-zone length reduces the number of selected objects. The work-zones from the reference scenario are reduced in a way that the new length is inside its boundaries. With doing this the costs and benefits are decreasing in the same way and the B/C ratio stays at its level. If the minimum distance between work-zones is shortened the optimization is less restrictive and more work-zones can be arranged. Due to more freedom the constellation of work-zones can be better optimized and the benefit is increasing more than the costs.

Table 5-8: Comparison of the costs and benefits.

Scenario	Cost		Benefit		B/C
	CHF	% of reference	CHF	% of reference	
Ref.	95 Mio.	-	265 Mio.	-	2.8
1 (budget)	19 Mio.	19.6 %	88 Mio.	33.2%	4.7
2 (mwzl)	71 Mio.	74.9 %	197 Mio.	74.2%	2.8
3 (mdbw)	118 Mio.	123.8 %	355 Mio.	134.1%	3.0

6 Conclusion and outlook

6.1 Conclusion

A GIS based optimization model to determine optimal work-zones in a network has been developed and applied on a real world, large scale network. To compare it with previous models, this model can be connected to any database of a network and allows user to represent the optimized solution on a graphical way on any GIS platform. With the input and output of the model having a relational database structure the model can easily be expanded for multiple time periods. Compared with previous models the optimization in MATLAB is more robust and no manually set up is required.

The application to the road network of the canton of Valais considered almost $2'000$ objects and more than $1'780$ nodes. Four different road types have been included and for each object three different traffic configurations have been examined. It was shown that the presented model can be applied on such a large scale network including the owner, user and society costs. Further on, it has been shown that the model can be applied on a GIS-based network. Therefore,some limitations occur with lane separated roads.

6.2 Outlook

To be able to apply the model on any GIS-based network further development has to be done in the area of roads which are split up into multiple parallel object-chains. Either the model has to be extended with a further constraint or the algorithm to set up the SQL database needs to include a part where those parallel objects are grouped together.

Further the routing algorithm to set up the optimization has to be improved that a faster routing performance is possible and the system gets more user friendly.

In a next step, an updating function has to be developed so that the model can be used for multiple time periods. The presented model with its GIS based structure gives a good base for this expansion.

Bibliographie

ASTRA. *Forschungspaket Massnahmenplanung im EM von Fahrbahnen. Standartisierte Erhaltungsmassnahmen.* Switzerland: Bundesamt für Strassen, 2008.

Axhausen, et al. *Swiss value of travel time savings.* Switzerland: ETH Zürich, 2006.

Bakaba, J. E., et al. «Untersuchung der Verkehrssicherheit im Bereich von Baustellen auf Bundesautobahnen.» Germany, 2012.

BFS. «Bundesamt für Statistik: LIK, Durchschnittspreise für Benzin und Diesel, Jahreswerte 2014.» mar. 2015. http://www.bfs.admin.ch/bfs/portal/de/index/themen/05/02/blank/key/durchschnittspreise.html.

BFS. *Medienmitteilung: Strassenfahrzeuge 2014.* Switzerland: Bundesamt für Statistik, 2015.

De Jong, G. «Value of Freight Travel-Time Savings.» *Handbook of Transport Modelling*, 2008: 553-564.

Eicher, C. «Master Project FS 2014, Optimal grouping of interventions on road networks.» Switzerland: ETH Zürich, 2014.

Geringer, B., und Tober. *Abgasemissionen - Mobilität.* mar 2015. http://www.auto-umwelt.at/_emissionen/em_abg.htm.

Hajdin, R., und B.T Adey. «Optimal worksites on highway networks subject to constraints.» Lake Louise, Canada: Second IFED Forum, 2006.

Sigrist, M. «Optimal grouping of interventions on road networks (OPINRONET).» Switzerland: ETH Zürich, 2012.

Small, K. A., R. Noland, X. Chu, und D Lewis. «Valuation of Travel-Time Savings and Predictability in Congested Conditions for Highway User-Cost Estimation.» Washington, D.C., 1999.

TCS. «Kosten eines Musterautos.» 2011. from http://www.tcs.ch/de/auto-mobilitaet/autokosten/kosten-eines-musterautos.php.

VBA. *Das Nutzfahrzeug - Umweltfreundlich und effizient.* Germany: Verband der Automobilindustrie, 2008.

VCA & DFT. *Car fuel and emissions information.* United Kingdom: Vehicle Certification Agency & United Kingdom Department of Transportation, 2011.

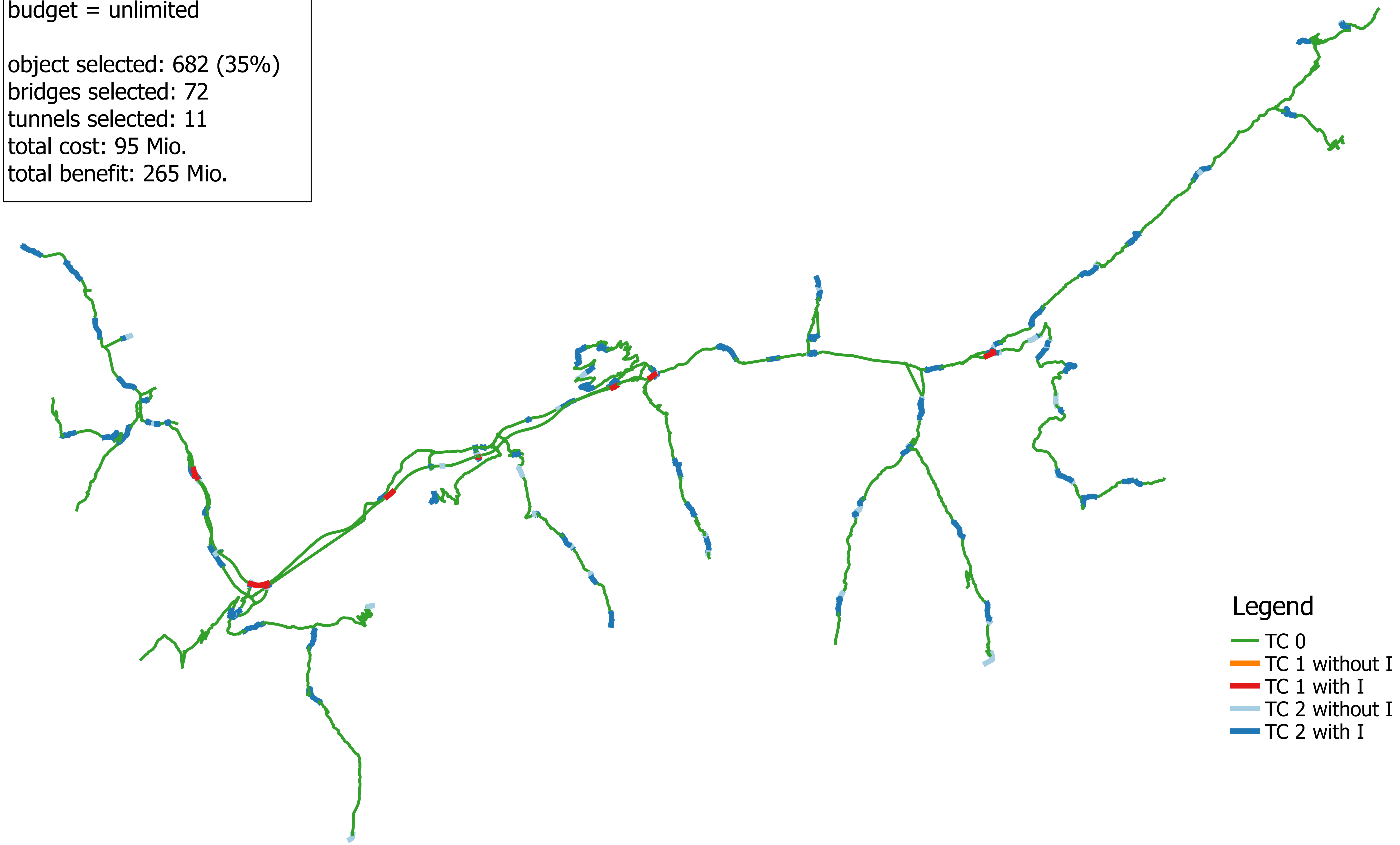
Appendix A:

Graphical representation of the case study

Reference scenario

mwzl = 2'000 m
mdbw = 3'000m
budget = unlimited

object selected: 682 (35%)
bridges selected: 72
tunnels selected: 11
total cost: 95 Mio.
total benefit: 265 Mio.

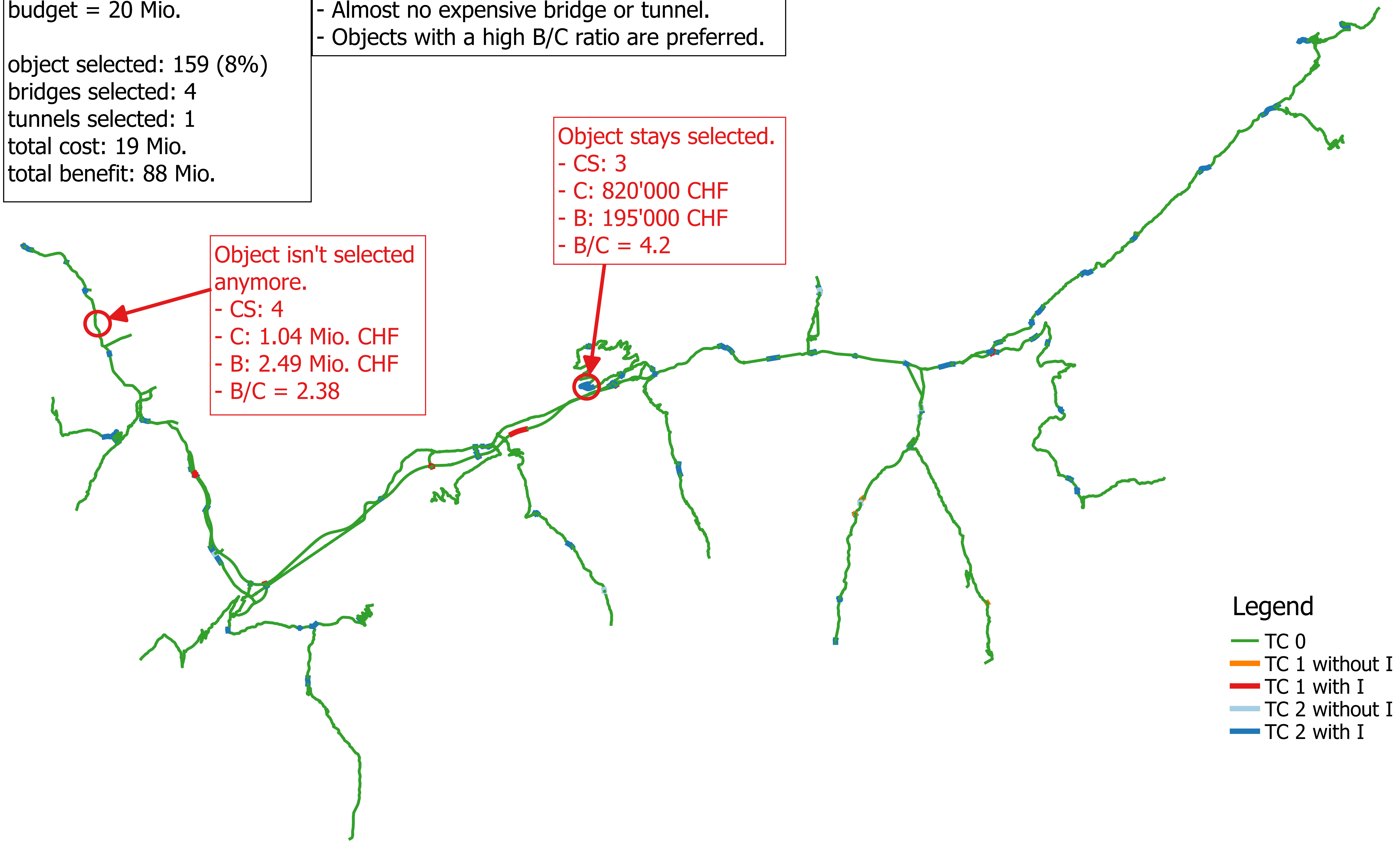


Scenario 1

mwzl = 2'000 m
mdbw = 3'000m
budget = 20 Mio.

object selected: 159 (8%)
bridges selected: 4
tunnels selected: 1
total cost: 19 Mio.
total benefit: 88 Mio.

General:
- Less object selected.
- Almost no expensive bridge or tunnel.
- Objects with a high B/C ratio are preferred.

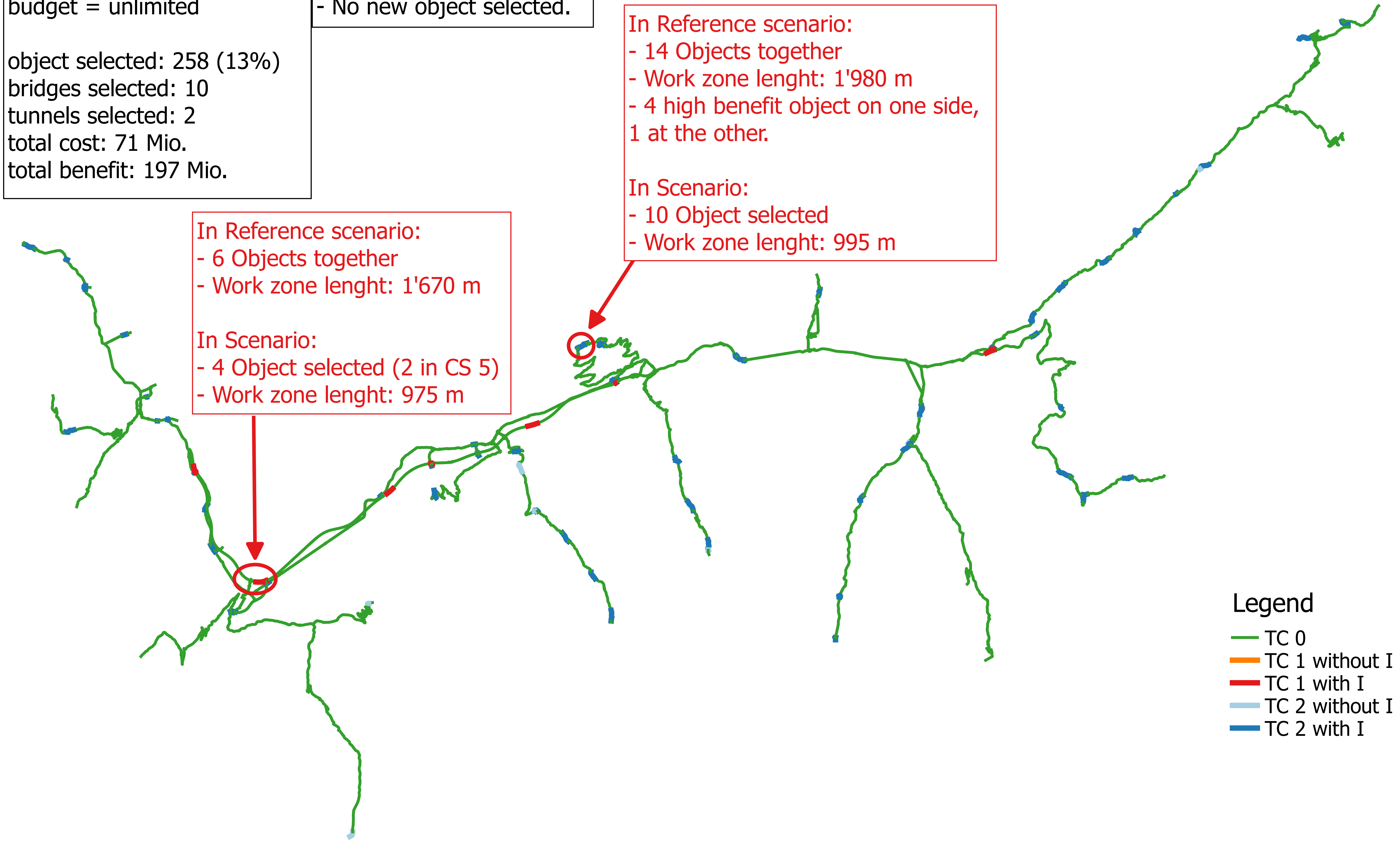


Scenario 2

mwzl = 1'000 m
mdbw = 3'000m
budget = unlimited

object selected: 258 (13%)
bridges selected: 10
tunnels selected: 2
total cost: 71 Mio.
total benefit: 197 Mio.

General:
- Less object selected.
- No new object selected.

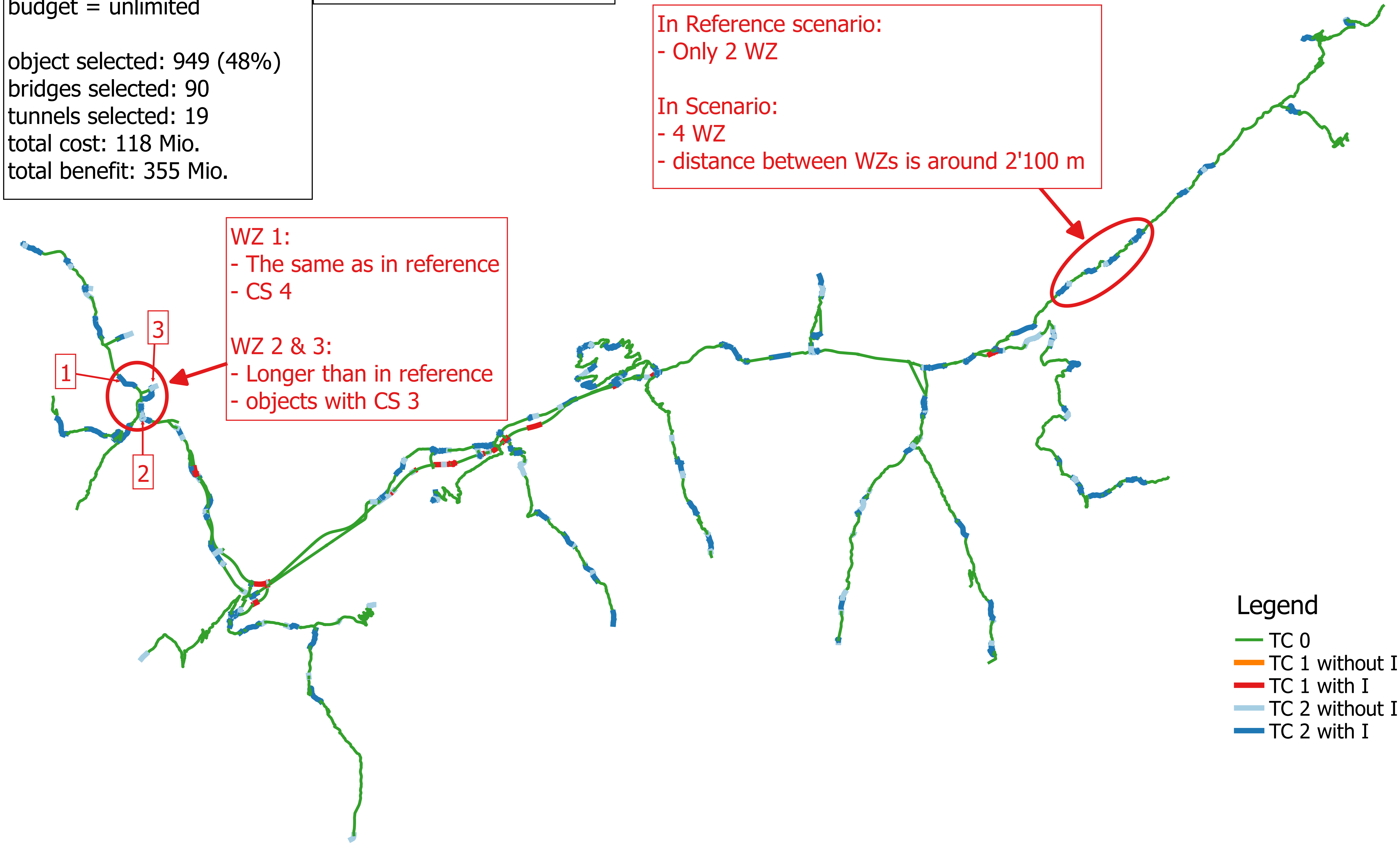


Scenario 3

mwzl = 2'000 m
mdbw = 2'000m
budget = unlimited

object selected: 949 (48%)
bridges selected: 90
tunnels selected: 19
total cost: 118 Mio.
total benefit: 355 Mio.

General:
- Much more object selected.



Appendix B:

SQL query to set up the network

Set up the main table

```
/*
Drop all tables if they already exist.
*/

DROP TABLE if exists "wallisroadXY_3_intervention";
DROP TABLE if exists cost_calculation;
DROP TABLE if exists "wallisroadXY_3_intervention_vertices_pgr";

/*
Create table main table.
Here the main table is created based on the reduced network table.
All information about the network are in this table. The cost calculation
will be safed in it and afert the optimization the results are stored
inside it.
*/
-- Creating the table
CREATE TABLE "wallisroadXY_3_intervention"
(
  id serial PRIMARY KEY,
  source integer,
  target integer,
  type character varying(16),
  bridge integer,
  tunnel integer,
  maxspeed integer,
  length double precision,
  "DTV" integer,
  the_geom geometry(LineString,21781),
  "cost_I0" integer default '0',
  "benefit_I0" integer default '0',
  "cost_I1" double precision,
  "benefit_I1" double precision,
  "intervention_1" boolean,
  "cost_I2" double precision,
  "benefit_I2" double precision,
  "intervention_2" boolean,
  traffic_configuration integer,
  intervention boolean
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "wallisroadXY_3_intervention"
  OWNER TO postgres;
COMMENT ON TABLE "wallisroadXY_3_intervention"
  IS 'Network with calculated costs and benefit for 3 different traffic configuration.';
```

```
-- Inserting basic network data
INSERT INTO "wallisroadXY_3_intervention"(
            source, target, type, bridge, tunnel, maxspeed, length,
            "DTV", the_geom)
SELECT
        cast(source as integer), -- cast convert char value into integer
        cast(target as integer),
        type,
        bridge,
        tunnel,
        maxspeed,
        length,
        "TV",
        the_geom
FROM "wallisroadXY_2_reduced_network";

/*
Generating unique source and target node starting with 1 till the last.
This is done by pg routing.
*/
-- Insert extension if it doas not exist.
CREATE EXTENSION if not exists pgrouting;
SELECT pgr_version();
-- Generates nodes
select pgr_createTopology('wallisroadXY_3_intervention', 0.001, 'the_geom', 'id');
```

Creating additional table for cost calculation

```
/*
In this part a secound table is created and used to calculate the cost.
In this way the detailed cost calculation can be looked at in a seperate
table and the main table has a good overview.
*/
-- Creating cost calculation table
CREATE TABLE cost_calculation (
  id integer PRIMARY KEY,
  type character varying(16),
  road integer,
  bridge integer,
  tunnel integer,
  maxspeed double precision,
  length double precision,
  width integer,
  "DTV" integer,
  "DTV_cars" integer,
  "DTV_trucks" integer,
  cs integer,
  duration_I double precision default '0',
  TC_1 character varying(5),
  TC_2 character varying(5),
  IC_fix double precision default '0',
  IC_variable double precision default '0',
  IC_add_1 double precision default '0',
  IC_add_2 double precision default '0',
  IC_1 double precision default '0',
  IC_2 double precision default '0',
  speed_1_abs double precision default '0',
  speed_1_rel double precision default '0',
  speed_1_fact double precision default '0',
  speed_2_abs double precision default '0',
  speed_2_rel double precision default '0',
  speed_2_fact double precision default '0',
  speed_1 double precision default '0',
  speed_2 double precision default '0',
  TTLC_1_d double precision default '0',
  TTLC_1 double precision default '0',
  TTLC_2_d double precision default '0',
  TTLC_2 double precision default '0',
  AC_year double precision default '0',
  AC_add double precision default '0',
  mr_factor double precision default '1',
  cs_factor double precision default '1',

  VC_cars_wI double precision default '0',
  VC_trucks_wI double precision default '0',
  VC_cars_noI double precision default '0',
  VC_trucks_noI double precision default '0',
  VC_maint_wI double precision default '0',
  VC_maint_noI double precision default '0',
  VC_wI double precision default '0',
  VC_noI double precision default '0',
  EC_wI double precision default '0',
  EC_noI double precision default '0',
  benefit_wI_1 double precision default '0',
  benefit_noI_1 double precision default '0',
  benefit_wI_2 double precision default '0',
  benefit_noI_2 double precision default '0',
  cost_I0 double precision default '0',
  benefit_I0 double precision default '0',
  cost_I1 double precision default '0',
  benefit_I1 double precision default '0',
  intervention_1 boolean,
  cost_I2 double precision default '0',
  benefit_I2 double precision default '0',
  intervention_2 boolean
)
WITH (
  OIDS=FALSE
);
ALTER TABLE cost_calculation
  OWNER TO postgres;
```

```sql
-- Inserting the basic network data
INSERT INTO cost_calculation(
            id, type, bridge, tunnel,
            maxspeed, length, "DTV")
SELECT
  "wallisroadXY_3_intervention".id,
  "wallisroadXY_3_intervention".type,
  "wallisroadXY_3_intervention".bridge,
  "wallisroadXY_3_intervention".tunnel,
  "wallisroadXY_3_intervention".maxspeed,
  "wallisroadXY_3_intervention".length,
  "wallisroadXY_3_intervention"."DTV"
FROM
  public."wallisroadXY_3_intervention";
```

Additional network information

```sql
-- Add random CS
/*
The following code can be used to have random CS. in this case
we import it from an other table, where it was created randomly
too. But then it doas not change every time.

UPDATE cost_calculation
   SET cs = random()*4+1;
*/
UPDATE cost_calculation
SET cs = random_cs.cs
FROM random_cs
WHERE cost_calculation.id = random_cs.id;

/*
Some other network data is used to calculate the cost.
Here they are defined.
*/
-- Add width of objects (highway 22m, other 11m)
UPDATE cost_calculation
   SET width=11
   WHERE type = 'primary' or type = 'secondary' or
   type = 'mountain';
UPDATE "cost_calculation"
   SET width=22
   WHERE type = 'motorway';

-- Add 2 different traffic configuariations
UPDATE cost_calculation
   SET TC_1='4-0',
   TC_2='H2-N2'
   WHERE type = 'motorway';
UPDATE cost_calculation
   SET TC_1='1-0',
   TC_2='N2'
   WHERE type = 'primary' or type = 'secondary' or
   type = 'mountain';

-- Traffic volume for cars and trucks (assume 6% trucks)
UPDATE cost_calculation
   SET ("DTV_cars","DTV_trucks") = (0.94 * "DTV", 0.06 * "DTV");

-- Binary column for road
UPDATE cost_calculation
   SET road = (1-bridge-tunnel);
```

```sql
-- Mountain factor (increase factor for mountain road)
UPDATE cost_calculation
   SET mr_factor = 1 + (SELECT mr_fuel_increase FROM variables_fix)
   where type = 'mountain';

-- Condition state factor (increase factor due to the condition state)
UPDATE cost_calculation
   SET cs_factor = (1+ (SELECT increase FROM condition_state
        where cs = cost_calculation.cs)/100);
```

Cost calculation

Intervention costs

```sql
/*
The intervention duration and intervention cost are calculated.
*/
-- Duration of intervention
UPDATE cost_calculation
   SET duration_I = (length*width)/(road* (SELECT producing_capacity
        FROM interventions where cs = cost_calculation.cs and road)
        + bridge*(SELECT producing_capacity  FROM interventions
        where cs = cost_calculation.cs and bridge)
        + tunnel*(SELECT producing_capacity  FROM interventions
        where cs = cost_calculation.cs and tunnel))
   where cs>2; -- smaller than cs = 3, we don't have an intervention.

-- Intervention cost
UPDATE cost_calculation
   SET (IC_fix, IC_variable) = (
   -- fix intervention costs
   (road* (SELECT fix_cost  FROM interventions
        where cs = cost_calculation.cs and road)
        + bridge*(SELECT fix_cost  FROM interventions
        where cs = cost_calculation.cs and bridge)
        + tunnel*(SELECT fix_cost  FROM interventions
        where cs = cost_calculation.cs and tunnel)),
   -- variable intervention costs
   (((road * (SELECT variable_cost_area  FROM interventions
        where cs = cost_calculation.cs and road)
        + bridge*(SELECT variable_cost_area  FROM interventions
        where cs = cost_calculation.cs and bridge)) * width
        + tunnel*(SELECT variable_cost_line  FROM interventions
        where cs = cost_calculation.cs and tunnel)) * length)
        )
    where cs>2; -- smaller than cs = 3 we don't have an intervention.
   -- additional intervention costs
UPDATE cost_calculation
   SET (IC_add_1, IC_add_2) = (
   ((IC_fix+IC_variable)*
        (SELECT add_under_traffic+add_shift  FROM traffic_configuration
        where tc = cost_calculation.TC_1)/100
        + (SELECT add_traffic_light FROM traffic_configuration
        where tc = cost_calculation.TC_1)),
   ((IC_fix+IC_variable)*
        (SELECT (add_under_traffic+add_shift)  FROM traffic_configuration
        where tc = cost_calculation.TC_2)/100)
        + (SELECT add_traffic_light FROM traffic_configuration
        where tc = cost_calculation.TC_2));
   -- total intervention cost
UPDATE cost_calculation
   SET (IC_1, IC_2) = (
   IC_fix + IC_variable + IC_add_1,
   IC_fix + IC_variable + IC_add_2);
```

Travel time loss costs

```
/*
For the travel time loss cost the reduced speed due to the traffic
configuration has to be calculated first. Then the travel time loss
costs can be calculated.
*/
-- Reduced speed
UPDATE cost_calculation
   SET (speed_1_abs, speed_1_rel, speed_1_fact,
        speed_2_abs, speed_2_rel, speed_2_fact) = (
   (SELECT speed_change_abs  FROM traffic_configuration
        where tc = cost_calculation.TC_1),
   maxspeed - (SELECT speed_change_rel  FROM traffic_configuration
        where tc = cost_calculation.TC_1),
   (SELECT speed_change_fact  FROM traffic_configuration
        where tc = cost_calculation.TC_1) * maxspeed,
   (SELECT speed_change_abs  FROM traffic_configuration
        where tc = cost_calculation.TC_2),
   maxspeed - (SELECT speed_change_rel  FROM traffic_configuration
        where tc = cost_calculation.TC_2),
   (SELECT speed_change_fact  FROM traffic_configuration
        where tc = cost_calculation.TC_2) * maxspeed
        );
UPDATE cost_calculation
   SET (speed_1, speed_2) = (speed_1_rel, speed_2_rel);
UPDATE cost_calculation
   SET speed_1 = speed_1_abs + speed_1_fact where speed_1_rel = maxspeed;
UPDATE cost_calculation
   SET speed_2 = speed_2_abs + speed_2_fact where speed_2_rel = maxspeed;
UPDATE cost_calculation
   SET speed_1 = 1 where speed_1 = 0; -- to make sure no divisation by 0
UPDATE cost_calculation
   SET speed_2 = 1 where speed_2 = 0; -- to make sure no divisation by 0

-- Cost for travel time loss per day
UPDATE cost_calculation
   SET (TTLC_1_d, TTLC_2_d) = (
   (length/1000 *
        case when (1/speed_1 - 1/maxspeed) < 0 then 0
          else (1/speed_1 - 1/maxspeed) end
        * ("DTV_cars" * (SELECT tts_cars  FROM variables_fix)
        + "DTV_trucks" * (SELECT tts_trucks  FROM variables_fix))),
   (length/1000 *
        case when (1/speed_2 - 1/maxspeed) < 0 then 0
          else (1/speed_2 - 1/maxspeed) end
        * ("DTV_cars" * (SELECT tts_cars  FROM variables_fix)
        + "DTV_trucks" * (SELECT tts_trucks  FROM variables_fix))));

-- Cost for travel time loss per intervention duration
UPDATE cost_calculation
   SET (TTLC_1, TTLC_2) = (TTLC_1_d * duration_I, TTLC_2_d * duration_I);
```

Accident costs

```
/*
Here the  Accident costs are calculated
*/
-- Accident cost
UPDATE cost_calculation
   SET (AC_year, AC_add) = (
   length * "DTV" * (SELECT vehicle_cost  FROM variables_fix) / 1000 * 365,
   length * "DTV" * (SELECT add_vehicle_cost  FROM variables_fix) / 1000
        * duration_I);
```

Vehicle costs

```
/*
The vehicle costs are calculated for the case an intervention is performed
(CS = 1 after it) and for the case that no intervention is performed.
*/
-- Vehicle costs with Intervention (CS = 1)
UPDATE cost_calculation
   SET (VC_cars_wI, VC_trucks_wI) = (
   -- Cost for cars
   ((SELECT mean_fuel_cars FROM variables_fix) * mr_factor * length / (10^5)
        * "DTV_cars" * (SELECT fuel_price FROM variables_fix)),
   -- Cost for trucks
   ((SELECT mean_fuel_trucks FROM variables_fix) * mr_factor * length / (10^5)
        * "DTV_trucks" * (SELECT fuel_price FROM variables_fix)));

-- Vehicle costs with no Intervention (real CS)
UPDATE cost_calculation
   SET (VC_cars_noI, VC_trucks_noI) = (
   -- Cost for cars
   (VC_cars_wI * cs_factor),
   -- Cost for trucks
   (VC_trucks_wI * cs_factor));

-- Vehicle maintenance cost
UPDATE cost_calculation
   SET (VC_maint_wI, VC_maint_noI) = (
   ((VC_cars_wI + VC_trucks_wI)
        * ((SELECT service_repair_factor FROM variables_fix)
        + (SELECT depreciation_factor FROM variables_fix))),
   ((VC_cars_noI + VC_trucks_noI)
        * ((SELECT service_repair_factor FROM variables_fix)
        + (SELECT depreciation_factor FROM variables_fix))));

-- Total vehicle cost
UPDATE cost_calculation
   SET (VC_wI, VC_noI) = (
   (VC_cars_wI + VC_trucks_wI + VC_maint_wI)*365,
   (VC_cars_noI + VC_trucks_noI + VC_maint_noI)*365);
```

Emission costs

```
/*
Like the vehicle cost also the emission cost are calculated for the 2
cases intervention performed and no intervention.
*/
-- Emission cost
UPDATE cost_calculation
   SET EC_wI = (mr_factor
        * ((SELECT emission_cars * emission_cost FROM variables_emission
        where emission_type = 'CO2') * "DTV_cars"
        + (SELECT emission_trucks * emission_cost FROM variables_emission
        where emission_type = 'CO2') * "DTV_trucks"
        + (SELECT emission_cars * emission_cost FROM variables_emission
        where emission_type = 'CO') * "DTV_cars"
        + (SELECT emission_trucks * emission_cost FROM variables_emission
        where emission_type = 'CO') * "DTV_trucks"
        + (SELECT emission_cars * emission_cost FROM variables_emission
        where emission_type = 'NOx') * "DTV_cars"
        + (SELECT emission_trucks * emission_cost FROM variables_emission
        where emission_type = 'NOx') * "DTV_trucks"
        + (SELECT emission_cars * emission_cost FROM variables_emission
        where emission_type = 'PM') * "DTV_cars"
        + (SELECT emission_trucks * emission_cost FROM variables_emission
        where emission_type = 'PM') * "DTV_trucks"
        + (SELECT emission_cars * emission_cost FROM variables_emission
        where emission_type = 'VOC') * "DTV_cars"
        + (SELECT emission_trucks * emission_cost FROM variables_emission
        where emission_type = 'VOC') * "DTV_trucks") / (10^6) * 365);
UPDATE cost_calculation
   SET EC_noI = EC_wI * cs_factor;
```

Benefit calculation

```
/*
Now the benefits and costs are calculated.
For the benefits two states are compared for each traffic condition.
First the intervention has a higher benefit than only the traffic
condition without an intervention or the other way round.
*/
-- Benefit calculation
UPDATE cost_calculation
   SET (benefit_wI_1, benefit_wI_2, benefit_noI_1, benefit_noI_2) = (
   ((SELECT time_periode FROM variables_fix) * ((VC_noI - VC_wI)
        + (EC_noI - EC_wI)) - (IC_1 + TTLC_1 + AC_add)),
   ((SELECT time_periode FROM variables_fix) * ((VC_noI - VC_wI)
        + (EC_noI - EC_wI)) - (IC_2 + TTLC_2 + AC_add)),
   ((TTLC_1 + AC_add) * (-1)),
   ((TTLC_2 + AC_add) * (-1)));

-- Total cost and benefit (the intervention_1 and _2 boolean are to
-- see if an intervention is actualy executed.
UPDATE cost_calculation
   SET (cost_I1, cost_I2, benefit_I1, benefit_I2,
   intervention_1, intervention_2) = (
   case when benefit_wI_1 > benefit_noI_1 then IC_1
        else 0 end,
   case when benefit_wI_2 > benefit_noI_2 then IC_2
        else 0 end,
   case when benefit_wI_1 > benefit_noI_1 then benefit_wI_1
        else benefit_noI_1 end,
   case when benefit_wI_2 > benefit_noI_2 then benefit_wI_2
        else benefit_noI_2 end,
   case when benefit_wI_1 > benefit_noI_1 then true
        else false end,
   case when benefit_wI_1 > benefit_noI_1 then true
        else false end);
```

Updating main table

```
/*
The main table can be updated with the cost and benefit values.
*/
-- Update the main table
UPDATE "wallisroadXY_3_intervention"
SET
        "cost_I1" = cost_calculation.cost_I1,
        "benefit_I1" = cost_calculation.benefit_I1,
        "intervention_1" = cost_calculation.intervention_1,
        "cost_I2" = cost_calculation.cost_I2,
        "benefit_I2" = cost_calculation.benefit_I2,
        "intervention_2" = cost_calculation.intervention_2
FROM
        cost_calculation
WHERE
        "wallisroadXY_3_intervention".id = cost_calculation.id;
```

Appendix C:

MATLAB code

```
%% LOAD THE DATA
% Turn the profiler on
profile on

disp('read SQL');

% Define the SQL table
sql_table = '"wallisroadXY_3_intervention"';

% Import SQL data
network_table = import_sql(sql_table);

% Generate network data
network_data = network_table(:,1:4);

% Generate cost matrix
costs_matrix = network_table(:,5:10);

% Budget limitation
budget = 10^10;

% Maximum workzone length
mwzl = 500; %m

% Minimum distance between workzones
% IMPORTANT REMARK: mdbw >= mwzl
mdbw = 600; %m

%% Set up the optimization
% First part: seting up the optimization program without cost and benefit.

disp('Intervention combinations');

%Calculate how many intervention / traffic configuration can be differented
%for each object. Set up the intervention combination matrix
int_types = ones(length(network_data(:,1)))*3;
intervention_combinations = zeros(2,length(network_data(:,1))*3);
for i = 1:length(network_data(:,1))
    for c=1:3
        intervention_combinations(1,(i-1)*3+c) = i;
        intervention_combinations(2,(i-1)*3+c) = c;
    end
end

% ------------------Start programm from C. Eicher------------------
% Establish the adjacency matrix
[adjacency_matrix,network_nodes] =...
    establish_adjacency_matrix(network_data);

% Establish the continuity matrix
[continuity_matrix] =...
    establish_continuity_matrix(adjacency_matrix,int_types(:,1));

%{
Marecl: This is the old Eicher algorithm. This three steps are put together
```

```
in one function: combination_constrain

% Calculate the paths for the maximum workzone length constraint
[all_worksites_1,c,a] =...
    complete_network_1(network_data,adjacency_matrix,mwzl);

% Calculate the paths for the minimum distance between workzones
[all_worksites_2,b] =...
    complete_network_2(network_data,adjacency_matrix,mdbw);

% Calculate the object combinations that cannot be chosen simultaneously
[combinations] = find_combinations...
    (all_worksites_1,all_worksites_2,network_data);
%}

% New combination constrain calculation. This put the three upper parts
% together.
[combinations] = combination_constrain...
    (network_data,adjacency_matrix,mwzl,mdbw);

% Establish the combinations matrix
[pairs,combinations_matrix] =...
    establish_combinations_matrix(combinations,int_types(:,1));

% Create additional vectors needed for the optimazation
[RHS_continuity,object_matrix] = create_vectors...
    (adjacency_matrix,int_types(:,1));
% -----------------End programm from C. Eicher------------------


%% Set up cost-benefit
disp('Cost & Benefit');
% Second part: calculating the costs
cost_benefit = cost_per_combination(intervention_combinations, costs_matrix);


%% Optimization
disp('Optimization');
[binary, total_benefit] = optimization(cost_benefit, combinations_matrix,...
    continuity_matrix, RHS_continuity, budget);
disp(total_benefit);

% Intervention per object
intervention_for_object = zeros(length(network_data(:,1)),3);
%intervention_for_object(:,3) = zeros(length(network_data(:,1)),1);
for i = 1:length(intervention_combinations(1,:))
    if binary(i) == 1
        intervention_for_object(intervention_combinations(1,i),1) =...
            cost_benefit(1,i);
        intervention_for_object(intervention_combinations(1,i),2) =...
            cost_benefit(2,i);
        intervention_for_object(intervention_combinations(1,i),3) =...
            intervention_combinations(2,i)-1;
    end
end
```

```
%% Export to SQL
disp('Export');
export_sql(network_data(:,1), intervention_for_object(:,3));

% View the profiler
profile viewer

disp('End');
```

Function: import_sql


```
function network_table = import_sql(sql_import_table)
%% INPUT ARGUMENTS
%{
'sql_import_table': name of the table of which the data should be imported.
%}

%% Import Data
%{
A connection to the database is generated.
explanation of the connection: database(name of the database, username of db, password↙
for this user,
used driver, path to the db).
%}
conn = database('workzones','postgres','marcel',...
    'org.postgresql.Driver','jdbc:postgresql://localhost:5432/workzones');

% An SQL query is created and executed, which selects the table we want.
% This data has to be fetched and later the connection is closed.
file = exec(conn,  strcat('SELECT id, source, target, length, "cost_I0",↙
"benefit_I0",',...
       '"cost_I1", "benefit_I1", "cost_I2", "benefit_I2" FROM ',sql_import_table,'↙
ORDER BY id'));
input =fetch(file);
close(conn);

% Input isn't only the data we want. It has a lot of informations about the
% connection. So only the data is stored in table.
table = input.Data;

%% Prepear the networktable for giving back
% network_table is a n x 10 matrix which has all used information about
% each of the n objects.
network_table = zeros(length(table(:,1)),10);

% store the data from the input in network_table. cell2mat is used because
% the input is a stored as cell but we us later double values.
network_table(:,1) = cell2mat(table(:,1)); % id
network_table(:,2) = cell2mat(table(:,4)); % length
network_table(:,3) = cell2mat(table(:,2)); % source
network_table(:,4) = cell2mat(table(:,3)); % target
network_table(:,5) = cell2mat(table(:,5)); % cost I1
network_table(:,6) = cell2mat(table(:,6)); % benefit I1
network_table(:,7) = cell2mat(table(:,7)); % cost I2
network_table(:,8) = cell2mat(table(:,8)); % benefit I2
network_table(:,9) = cell2mat(table(:,9)); % cost I3
network_table(:,10) = cell2mat(table(:,10)); % benefit I3
end
```

```
function cost_benefit = cost_per_combination(intervention_combinations, costs_matrix)
% programm to create a cost-benefit matrix for all posible combinations of
% objects and interventions
%% INPUT ARGUMENTS
%{
'intervention_combinations': 3 x c matrix
    - c: number of object-intervention-traffic configuration combinations
    - row 1: object ID
    - row 2: intervention ID
'costs_matrix': n x 6 matrix
    - n: number of objects in the network
    - column 1: costs for intervention 1
    - column 2: benefit from intervention 1
    - column 3: costs for intervention 2
    - column 4: benefit from intervention 2
    - column 5: costs for intervention 3
    - column 6: benefit from intervention 3
%}

%% INITIALIZING
%  Initalize cost benefit tables.
cost_benefit = zeros(size(intervention_combinations));


%% unforming the costs_matrix
for i = 1:length(intervention_combinations(1,:))
    cost_benefit(1,i) = costs_matrix(intervention_combinations(1,i),...
        intervention_combinations(2,i)*2-1);
    cost_benefit(2,i) = costs_matrix(intervention_combinations(1,i),...
        intervention_combinations(2,i)*2);
end


end
```

```
function [binary, total_benefit] = optimization(cost_benefit, combinations_matrix,...
    continuity_matrix, RHS_continuity, budget)
% This function is doing the optimization. The inbuild matlab solver
% intlinprog is used.
%% INPUT ARGUMENTS
%{
'cost_benefit': 2 x m matrix
    - m: number of posible combinations
    - raw 1: cost for combination
    - raw 2: benefit for combination
'combinations_matrix': n x m matrix containing the binaries for the
combinations. n: number of objects.
'continuity_matrix': n x m matrix containing the binaries for the
continuity.
'RHS_continuity': n x 1 matrix containing the number of objects connected
to this object.
'budget': the maximum budget.
%}

%% Set up
% the constaint_matrix is a matrix containing all constaint which have a
% smaller than relationship. The combination_matrix and the costs have to
% be smaller or equally the constraint values. The cost are divided by the
% budget to make sure, that the values aren't going to be too big.
constraint_matrix = [combinations_matrix(:,:);cost_benefit(1,:)/budget];

% The constraint_vector is holding the constraint for the
% constraint_matrix. The maximum budget.
constraint_vector = [ones(length(combinations_matrix(:,1)),1);1];

% The equal_constraint_matrix containing the constraint which has to be
% equally to the values in the equal_constraint vector. The
% continuity_matrix has to be equally.
equal_constraint_matrix = continuity_matrix;
equal_constraint_vector = RHS_continuity;

% The objection_function are the benefits. In the intlinprog it has to be
% minimized, so it is multiplied by (-1).
objection_function = transpose(cost_benefit(2,:))*(-1)/budget;

% The variables has to be binaries.
lower_bounds = zeros(length(constraint_matrix(1,:)),1);
upper_bound = ones(length(constraint_matrix(1,:)),1);
intcon = zeros(length(objection_function),1);
for i = 1:length(intcon)
    intcon(i,1) = i;
end

%% Optimization
[binary,total_benefit,exitflag,output] = intlinprog(objection_function,intcon,...
    constraint_matrix,constraint_vector,equal_constraint_matrix,
equal_constraint_vector,...
    lower_bounds,upper_bound);
```

```
binary = binary;
total_benefit = total_benefit * budget * (-1);
end
```

```
function export_sql(network_data_id, intervention_for_object)
%% INPUT ARGUMENTS
%{
'network_data_id': n vector which contains the object id's
'intervention_for_object': n vector which contains the intervention
calculated by the optimization.
%}

%% Export Data
%{
A connection to the database is generated.
explanation of the connection: database(name of the database, username of db, password
for this user,
used driver, path to the db).
%}
conn = database('workzones','postgres','marcel',...
    'org.postgresql.Driver','jdbc:postgresql://localhost:5432/workzones');

% Export table name
sql_export_table = 'export_matlab';

% The SQL query to create a new export table is defined and executed.
sqlquery = strcat('DROP TABLE IF EXISTS export_matlab;',...
    'CREATE TABLE ',sql_export_table,' (id integer, intervention integer )',...
    'WITH ( OIDS=FALSE );');
exec(conn, sqlquery);

% The opbjec id's and the interventions per objects are saved in the new
% export table.
colnames = {'id', 'intervention'};
data = num2cell([network_data_id, intervention_for_object]);
insert(conn,sql_export_table,colnames,data);

% Connection to databese close.
close(conn);

end
```

Appendix D:

SQL query to insert optimization results

Insert optimization results in main table

```
/*
Drop the table if it already exists.
*/
drop table if exists "wallisroadXY_4_scenario_0";

/*
Create a new scenario table.
It looks the same as the "wallisroadXY_3_intervention" table, which is
created by the "sql_1_generate_input" query.
*/
CREATE TABLE "wallisroadXY_4_scenario_0"
        (like "wallisroadXY_3_intervention");

-- Inserting the values from the "wallisroadXY_3_intervention" table.
INSERT INTO "wallisroadXY_4_scenario_0"(
            id, source, target, type, bridge, tunnel, maxspeed, length, "DTV",
            the_geom, "cost_I0", "benefit_I0", "cost_I1", "benefit_I1", intervention_1,
            "cost_I2", "benefit_I2", intervention_2)
  SELECT id, source, target, type, bridge, tunnel, maxspeed, length, "DTV",
       the_geom, "cost_I0", "benefit_I0", "cost_I1", "benefit_I1", intervention_1,
       "cost_I2", "benefit_I2", intervention_2
    FROM "wallisroadXY_3_intervention";

/*
The solution from the optimization is inserted. The Matlab created a new
table (export_matlab) in which the optimal traffic configuration for each
object is stored.
*/
update "wallisroadXY_4_scenario_0"
  set traffic_configuration = export_matlab.intervention
    FROM export_matlab
    where export_matlab.id = "wallisroadXY_4_scenario_0".id;

/*
The optimal traffic configuration is compared with the binary value of
this TC to see if an intervention is really executed and then stored
in intervention (true or false).
*/
update "wallisroadXY_4_scenario_0"
  set intervention =
    case when
        traffic_configuration = 1 and intervention_1 = true or
        traffic_configuration = 2 and intervention_2 = true
        then true
    else false
    end;
```

Adjust a layer style

```
/*
This last part is only for representation in QGIS. We copy the standard
syle and save it in the table layer_styles with the table name of the
new created table. With this, the layout is automatically copied.
*/
INSERT INTO layer_styles(
            f_table_catalog, f_table_schema, f_table_name, f_geometry_column,
            stylename, styleqml, stylesld, useasdefault, description, owner,
            ui, update_time)
  SELECT f_table_catalog, f_table_schema, 'wallisroadXY_4_scenario_0',
       f_geometry_column, stylename, styleqml, stylesld, useasdefault,
       description, owner, ui, update_time
  FROM layer_styles
  where id=2;
```