

ASD2 – Labo3

*Réseau ferroviaire*

Jean-Luc Blanc, Gwendoline Dössegger, Gabriel Roch

15/11/2019

Table des matières

[1 Introduction 1](#_Toc23348713)

[2 Questions / réponses 2](#_Toc23348714)

[3 Conclusion 4](#_Toc23348715)

# Introduction

Dans le cadre de ce laboratoire, nous avons appliqué les algorithmes d’arbres couvrants minimaux et de plus court chemin à un graphe représentant le réseau ferroviaire de la Suisse (cf. figure ci-dessous).

La figure suivante représente la carte des grandes lignes du réseau ferroviaire. Les sommets du graphe associé correspondent aux villes et les arêtes correspondent aux lignes de chemin de fer les reliant. Pour chaque ligne, trois poids y sont associés : la distance en kilomètres séparant deux villes, la durée moyenne du trajet en minutes ainsi que le nombre de voies sur la ligne.

L’objectif final était d’implémenter et d’appliquer des algorithmes, vus durant le cours, à un graphe correspondant à un cas concret. L’application des algorithmes nous a permis de répondre aux différentes questions présentent dans le chapitre suivant.



*Figure 1 - Carte du réseau ferroviaire Suisse*

# Questions / réponses

**Arbre recouvrant de poids minimum :**

Nous voulons faire des travaux d’entretien sur le réseau ferroviaire, mais dans un souci d’économie nous souhaitons réduire le coût des travaux au maximum. Toutes les villes devront être accessibles par une ligne rénovée. Chaque ligne possède un nombre de voies allant de 1 à 4. Le coût pour rénover 1 km de ligne de chemin de fer varie selon le nombre de voies :

* 15M CHF par km pour les lignes ayant 4 voies
* 10M CHF par km pour les lignes ayant 3 voies
* 6M CHF par km pour les lignes ayant 2 voies
* 3M CHF par km pour les lignes ayant 1 voie

Quelles lignes doivent être rénovées ?

Viege - Brigue : 54 MF

Romanshorn - Constance : 60 MF

Saint-Gall - Romanshorn : 60 MF

Lucerne - Arth-Goldau : 66 MF

Delemont - Bienne : 105 MF

Bienne - Berne : 108 MF

Soleure - Berne : 129 MF

Delemont - Bale : 135 MF

Neuchatel - Berne : 150 MF

Romont - Fribourg : 150 MF

Lugano - Chiasso : 150 MF

Montreux - Lausanne : 150 MF

Thoune - Interlaken : 168 MF

Lugano - Bellinzone : 174 MF

Zoug - Arth-Goldau : 180 MF

Fribourg - Berne : 192 MF

Olten - Aarau : 195 MF

Lucerne - Interlaken : 216 MF

Liestal - Bale : 225 MF

Soleure - Olten : 228 MF

Yverdon-les-Bains - Lausanne : 234 MF

Yverdon-les-Bains - Neuchatel : 240 MF

Zurich - Bale : 264 MF

Viege - Sion : 270 MF

Zurich - Schaffhouse : 282 MF

Thoune - Montreux : 285 MF

Winterthur - Constance : 324 MF

Zurich - Winterthur : 375 MF

Sion - Montreux : 402 MF

Winterthur - Wil : 405 MF

Coire - Brigue : 483 MF

Bellinzone - Arth-Goldau : 600 MF

Lausanne - Geneve : 900 MF

Quel sera le coût total de la rénovation de ces lignes ? 7959 MF

**Plus court chemin :**

Quel est le chemin le plus court entre Genève et Coire ? Quelles sont les villes traversées ?

Longueur : 367 km

Via Genève -> Lausanne -> Montreux -> Sion -> Viège -> Brigue -> Coire

Mêmes questions mais en supposant que la gare de Sion est en travaux et donc qu’aucun train

ne peut transiter par cette gare.

Longueur : 399 km

via Genève -> Lausanne -> Romont -> Fribourg -> Berne -> Olten -> Aarau -> Zürich -> Coire

Quel est le chemin le plus rapide entre Genève et Coire en passant par Brigue ? Quelles sont les villes traversées ?

Temps : 321 minutes

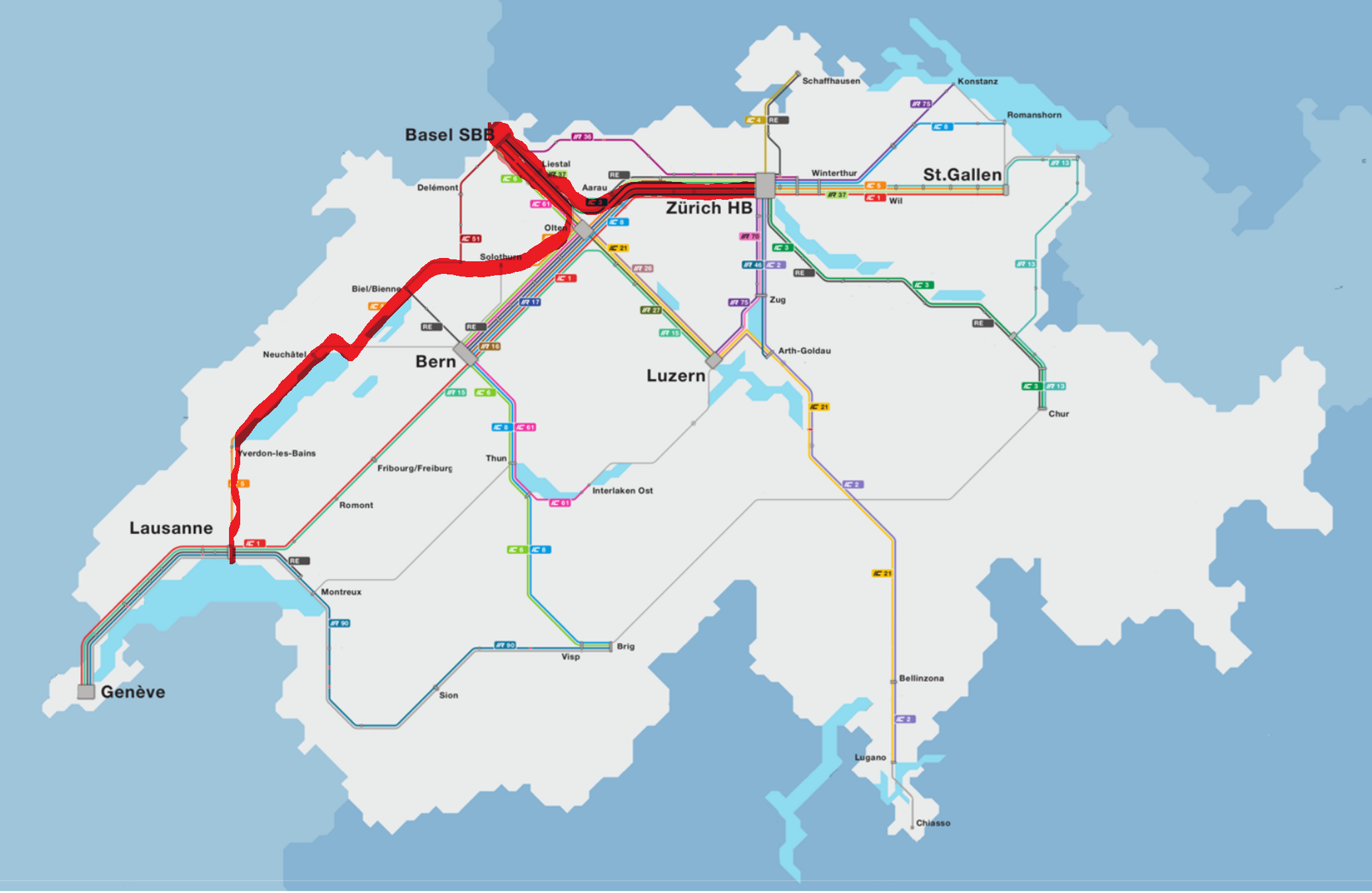
via Genève -> Lausanne -> Montreux -> Sion -> Viège -> Brigue -> Viège -> Thoune -> Berne -> Olten -> Aarau -> Zurich -> Coire

Mêmes questions mais entre Lausanne et Zurich, en passant par Bâle.

Temps : 174 minutes

via Lausanne -> Yverdon-les-Bains -> Neuchâtel -> Bienne -> Soleure -> Olten -> Liestal -> Bale -> Liestal -> Aarau -> Zurich

Pour la dernière question, vous mettrez en évidence le réseau ferroviaire réduit sur la carte (surligné en rouge) menant de Lausanne à Zürich passant par Bale.



# Conclusion

Au terme de ce laboratoire, nous estimons avoir rempli l’ensemble des exigences du cahier des charges avec notre implémentation des algorithmes.

Grâce à la prolongation de la date de rendu, nous avons pu améliorer notre première version comme suit. Au départ, nous avons fait 4 Wrapper (un pour chaque question) ainsi qu’une copie du TrainNetwork lorsque l’on souhaitait mettre une gare en travaux. Cela nous a permis de répondre rapidement aux questions et ainsi avoir un programme fonctionnel. Suite à cela, nous avons factorisé le code en 2 classes Wrappers, une pour les graphes orientés et l’autre pour les graphes non-orientés. De plus, nous passons au constructeur une référence en argument qui pointe vers la fonction calculant le poids des arêtes.

Concernant les gares en travaux, notre fonction de calcul du poids retourne numeric\_limits<int>::max() ce qui permet de ne pas prendre en compte la gare en travaux. Malheureusement, avec cette valeur, il y a un débordement de tampons et les calculs ne se faisaient pas correctement. Le code des Wrappers a donc été modifié pour détecter quand la fonction de poids retourne cette valeur et ainsi d’ignorer les arrêtes correspondants.

Nous avons rencontré une difficulté durant ce laboratoire. Celle-ci concerne l’implémentation de Dijkstra. Comme nous avions commencé le Laboratoire avant d’avoir vu l’algorithme de Dijkstra, ce dernier nous a pris beaucoup de temps avant de réussir à l’implémenter correctement.