



HAL
open science

Un algorithme pour battre le record du SwissTrainChallenge :

Emmanuel Clédat, Dirk Lauinger

► **To cite this version:**

Emmanuel Clédat, Dirk Lauinger. Un algorithme pour battre le record du SwissTrainChallenge :.
Revue XYZ, 2018, 157, pp.30 - 36. hal-04424473

HAL Id: hal-04424473

<https://hal.science/hal-04424473>

Submitted on 29 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Un algorithme pour battre le record du SwissTrainChallenge : poser le pied dans chacun des 26 cantons le plus rapidement possible en utilisant uniquement des transports publics

■ Emmanuel CLEDAT - Dirk LAUINGER

Le réseau ferré suisse est mondialement connu pour sa fiabilité et sa ponctualité. Il a aussi une troisième particularité très importante : sa connectivité. En effet, le moindre petit village de la Confédération est desservi par les transports publics. Deux doctorants ont voulu mettre ces particularités à l'épreuve, le vendredi 7 septembre 2018, à travers un défi un peu particulier.

MOTS-CLÉS

Réseau de transports, optimisation, connectivité, graphe, Chemins de fer fédéraux, SwissTrainChallenge, système d'information géographique, système d'information ferroviaire, horaire, problème relaxé

Le but est de poser le pied dans chacun des 26 cantons et demi-cantons suisses en utilisant uniquement les transports publics, et ce le plus rapidement possible. Cette idée a été proposée pour la première fois par Nicolas Rossé, journaliste de la RTS : Radio et Télévision Suisse. À l'époque, les CFF (Chemins de fer fédéraux) lui ont dit que c'était impossible^[1]. Pourtant, après de longues recherches autant géographiques que dans les horaires de trains, Philip Morf, spécialiste de la planification des trains aux CFF, a finalement trouvé un itinéraire pour relier tous les cantons pour le prix d'une carte journalière. Ce défi a beaucoup inspiré Raphael Zufferey, Chef scout de la troupe de Nyon, qui s'est dit qu'il serait possible de faire plus rapidement^[2]. Après l'inauguration du tunnel de base du Saint-Gothard, c'est au tour d'un groupe tessinois de battre le record^[3]. Chaque édition a nécessité sa dose d'esprit d'aventure, de courage et de chance, pour aller toujours plus vite (Figure 1). Pourtant, la bonne réussite de ce défi repose sur une préparation très importante : la planification des horaires pour que les correspondances se suivent au mieux, qui pour les trois premières

fois, a été établie à la main. Pour battre le temps record de la dernière édition, il devenait nécessaire de déployer l'artillerie lourde : l'emploi d'un algorithme d'optimisation couplé à un Système d'information Géographique (SIG). Cet article présente la construction de ce SIG, le principe de fonctionnement de l'algorithme d'optimisation et relate les péripéties lors du voyage.

Mise en forme des données dans le logiciel de système d'information géographique (SIG)

La première étape pour résoudre le problème du *SwissTrainChallenge* a été de télécharger les données géographiques. Celles-ci se trouvent sur le site open-data des CFF^[4]. Les premières entités chargées dans le logiciel de SIG ont été les 24 200 gares et arrêts de bus, représentés par des entités ponctuelles (Figure 2). En revanche, le canton des gares était inconnu.

Pour déterminer le canton auquel appartiennent les gares, nous avons téléchargé les frontières administratives des cantons^[5], puis effectué une

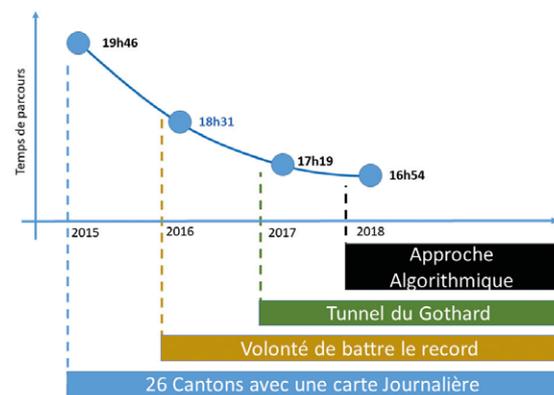


Figure 1. Records successifs du SwissTrainChallenge

jointure spatiale pour associer les gares dont les coordonnées géographiques se situaient à l'intérieur d'un canton au canton correspondant.

Les entités géographiques ponctuelles étaient alors toutes à disposition, en revanche, aucune topologie ne les liait les unes aux autres. En d'autres termes, il y avait les gares, mais pas les rails. Une solution aurait été d'importer la couche "rails" disponible sur les Swiss-Open-Geo-Datas^[6], puis d'associer les gares aux rails grâce à une jointure spatiale. Une information très importante aurait toujours manqué : l'heure de passage des trains dans les gares et si un train s'arrête dans une gare ou non. Pour pallier ce problème, nous avons téléchargé les horaires des trains, toujours sur les open-data des CFF^[4]. Ces horaires sont disponibles au format HRDF (HAFAS Raw Data Format), et au format GTFS (General Transit Feed Specification). Dans la mesure où il est plus facile d'obtenir les jours de circulation des trains avec le format de

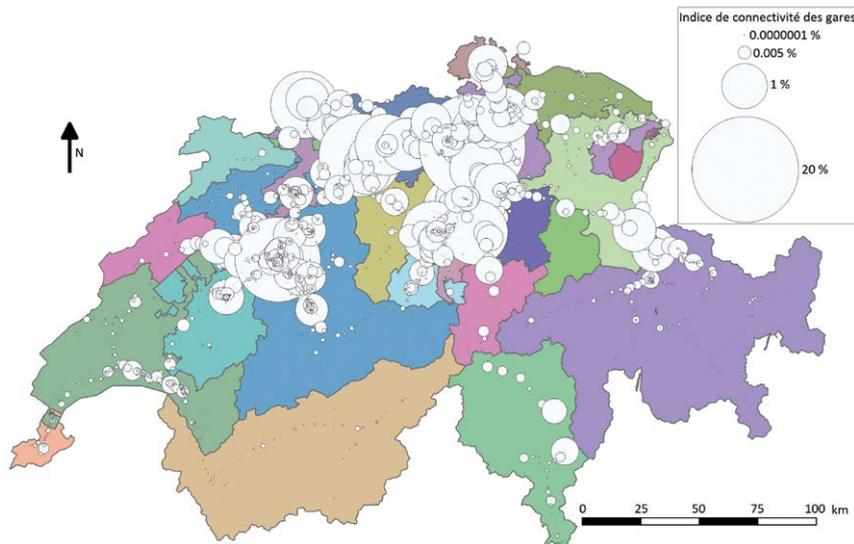


Figure 2. Affichage des gares dans un SIG (Système d'Information Géographique)

données GTFS, nous avons opté pour cette solution. Cette base de données est composée de plusieurs tables, dont la table 'Stop_Times' donnant les heures de passage des trains dans les gares, et la table 'Calendar' indiquant le jour de passage du train (Figure 3).

D'autre part, il existe des gares très proches qui peuvent être considérées comme une seule gare (comme par exemple la gare d'une ville, et l'arrêt de bus devant cette gare). Ces gares proches ont pu être agrégées grâce à une seconde jointure spatiale.

Pour réduire le nombre de gares à considérer, nous avons choisi des gares remarquables, c'est-à-dire, des gares où il nous semblait pertinent de changer de train. Pour cela, nous avons construit la matrice d'adjacence M de l'ensemble des gares de notre base de données, et calculé un indice de connectivité des gares : le Score d'Analyse Spectrale (décrit dans^[7] pour de l'analyse de réseau et utilisé dans^[8] le cadre d'analyse de réseaux topographiques). Ce score se définit en calculant le vecteur propre \vec{v} associé à la valeur propre maximale λ_{max} de M .

$$M \vec{v} = \lambda_{max} \vec{v}$$

La valeur de \vec{v} associée à la gare étudiée

est un indice de la connectivité de cette gare à l'ensemble du réseau. Le vecteur \vec{v} étant normalisé (sa norme euclidienne est unitaire), il est plus pratique de considérer le carré de ses valeurs, de manière que la somme des indices de connectivités des gares soit de 100 %. Cette valeur peut être affichée dans notre SIG. La gare ayant le meilleur score est sans surprise Zürich Hauptbahnhof, avec un score de 19.7 %. La première idée a été de sélectionner, dans chaque canton, les 5 gares les mieux connectées au sens de notre indicateur. Cependant, ni cet indicateur, ni aucun autre, plus empirique que nous avons essayé, n'a donné des résultats qui nous satisfaisaient. En effet, pour le canton de Genève par exemple, la gare de "Genève Aéroport" arrivait dans les premières, juste derrière "Genève Cornavin" (gare principale de Genève). Or, il était évident qu'il est inutile de s'enfoncer aussi loin dans le canton de Genève : tous les trains partant de "Genève Aéroport" s'arrêtent en gare de Genève avant de rejoindre d'autres cantons.

L'intelligence humaine était ici nécessaire pour choisir les gares potentielles de passage. Le SIG, représentant les gares en fonction de leur connectivité, se révélait comme un outil parfait

d'aide à la décision pour cette tâche (Figure 2). Les 81 gares retenues sont soit des nœuds ferroviaires (c'est-à-dire à l'intersection de grandes lignes), soit la première gare à l'entrée d'un canton. Le choix de ces gares étant manuel, l'oubli d'une gare importante pour effectuer une connexion était possible. Pour cela, un algorithme a été implémenté pour vérifier la présence de connexions rapides entre les gares choisies. Dans le cas contraire, cet algorithme a ajouté des points de transfert entre deux gares remarquables pour les connecter ou accélérer une connexion existante. Cet algorithme a ajouté 44 nouvelles gares d'intérêt aux 81 sélectionnées manuellement.

Une fois ces gares d'intérêt choisies, les horaires de trains ont pu être reformatsés, impliquant des simplifications. Les horaires des trains qui passent par plus de deux gares remarquables ont été séparés en morceaux distincts. Ainsi, le train faisant St-Maurice Lausanne Genève a été coupé en deux trains : St-Maurice Lausanne, puis Lausanne Genève. Les seules informations judicieuses à introduire dans l'algorithme d'optimisation présenté dans la section suivante sont la gare de départ, l'horaire de départ, la gare d'arrivée, l'horaire d'arrivée et la liste des cantons dans lesquels il est possible de poser le pied en prenant ce train. Cette liste de cantons peut être plus longue que simplement le canton de la gare de départ et celui de la gare d'arrivée. Par exemple, le train qui va de Biel-Bienne à Basel s'arrête dans le canton de Soleure, du Jura et de Bâle-Campagne. Il permet de poser le pied dans 5 cantons mais ne fait intervenir que deux gares remarquables. Par abus de langage, nous appellerons un trajet entre deux gares remarquables 'train' (Figure 4).

Un problème d'optimisation

La méthode classique pour trouver le trajet le plus court entre deux nœuds d'un graphe est l'algorithme A* (prononcer "A étoile")^[9]. Appliqué à notre problème, cet algorithme trouve un chemin reliant un grand nombre de cantons en peu d'itérations, mais il n'arrive pas à trouver un



Figure 3. Organisation simplifiée des bases de données des horaires des trains au format GTFS



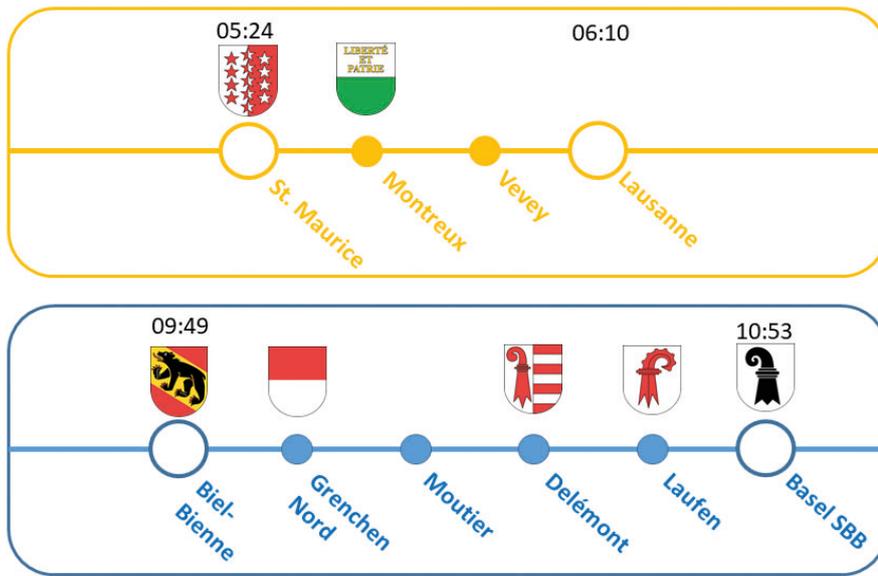


Figure 4. Extraction des "trains" : trajets de gare remarquable à gare remarquable, ayant comme attribut la liste des cantons d'arrêt.



chemin passant par tous les cantons. L'approche qui s'est relevée gagnante est celle de l'optimisation linéaire en nombres entiers^[10]. Cette méthode permet de traiter des problèmes combinatoires ayant une fonction de coût et des contraintes linéaires. Un problème d'optimisation combinatoire a déjà fait une apparition dans la revue XYZ^[11]. Nous allons ici introduire cette méthode en la comparant à la multilatération^[12] un problème connu des topographes, et présenter une application originale à notre problème d'optimisation des trains.

■ **L'exemple de la multilatération**

Soient trois bornes A, B, et C connues en coordonnées planimétriques x et y. On veut déterminer les coordonnées d'un point M en x et y. Pour cela, on

mesure les distances d_A , d_B et d_C entre M et respectivement les points A, B et C (Figure 5).

Sauf dans des cas très particuliers, les trois cercles de centre A, B et C et de rayon d_A , d_B et d_C devraient s'intersecter en un point unique : M. Cependant, due aux erreurs de mesures, ces trois cercles ne se coupent pas en un point unique. Il faut alors corriger les mesures d_A , d_B et d_C en y ajoutant les résidus v_A , v_B et v_C pour obtenir les mesures compensées \widehat{d}_a , \widehat{d}_b et \widehat{d}_c telles que les cercles de rayon \widehat{d}_a , \widehat{d}_b et \widehat{d}_c se coupent en un point unique.

Fréquemment, on répartit les résidus v_A , v_B et v_C en utilisant le critère des moindres carrés^[13, 14], c'est-à-dire que l'on va minimiser la fonction de coût $f(v_A, v_B, v_C) = v_A^2 + v_B^2 + v_C^2$.

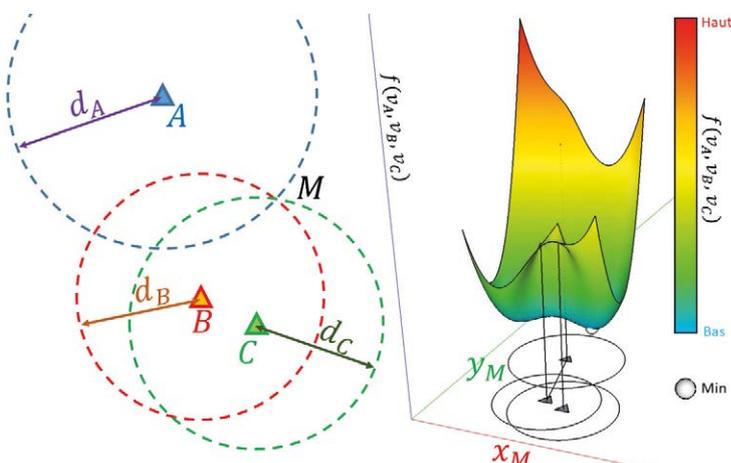


Figure 5. La multilatération (gauche) et sa fonction de coût (droite)

La meilleure position du point M au sens des moindres carrés correspond au point (x_M, y_M) et aux résidus v_A , v_B , v_C qui minimisent la fonction de coût $f(v_A, v_B, v_C)$ tel que :

$$\sqrt{(x_M - x_k)^2 + (y_M - y_k)^2} = d_k + v_k,$$

$$\forall k \in \{A, B, C\}$$

c'est-à-dire au point le plus bas de la courbe représentée sur la figure 5. L'ajustement de mesures de distance lors d'une multilatération se traduit alors par la résolution de ce problème d'optimisation. L'exemple de la multilatération permet d'introduire l'optimisation de l'itinéraire de nos trains.

■ **Optimisation de l'itinéraire des trains**

Pour montrer la structure de ce problème, imaginons un pays dans lequel il n'y a que deux trains avec les heures de départ et d'arrivée décrites dans le tableau 1.

Train	t^d : heure de départ	t^a : heure d'arrivée
A	0	1
B	1	2

Tableau 1. Horaire de départ et d'arrivée.

Pour l'instant nous ne considérons ni les gares de départ, ni celles d'arrivées, ni s'il est possible de prendre le train A ou B, ni même si prendre le train A ou B a un quelconque intérêt pour la réussite de notre défi. Pourtant, il est déjà possible de poser le problème sous la forme d'un problème d'optimisation.

Dans le problème de la multilatération, les variables d'intérêt à déterminer sont les coordonnées (x_M, y_M) du point M. Dans le problème des trains, le couple de variable à déterminer est $\{\epsilon_A, \epsilon_B\}$ où $\epsilon_A = Vrai$ si on prend le train A et $\epsilon_A = Faux$ si on ne prend pas le train A (de même pour B). Sachant que notre but est d'arriver le plus tôt possible, notre fonction de coût f à minimiser est le temps d'arrivée, calculée pour toute combinaison de ϵ_A et ϵ_B dans le tableau 2. Le début de la journée est représenté par l'heure 0.

Ce problème est fondamentalement différent du problème de la multilatération. En effet, si le problème de la multilatération vise à déterminer des nombres réels (les coordonnées x et



$f(\epsilon_A, \epsilon_B)$		ϵ_A	
		Faux	Vrais
ϵ_B	Faux	0	$t_A^a = 1$
	Vrais	$t_B^a = 2$	$\max\{t_A^a, t_B^a\} = 2$

Tableau 2. Fonction de coût pour le problème des trains avec les horaires du tableau 1.

y du point M), le problème de l'itinéraire des trains utilise des variables booléennes : *Vrai* ou *Faux*.

Pour obtenir un problème qui ressemble plus au problème de la multilatération, nous effectuons une opération appelée relaxation. La valeur *Faux* sera représentée par 0, la valeur *Vrai* par 1, et au lieu de ne considérer que ces deux valeurs, nous allons considérer l'ensemble des valeurs réelles entre 0 et 1. On n'a donc plus $\epsilon_A, \epsilon_B \in \{Faux, Vrai\}$ mais $\epsilon_A, \epsilon_B \in [0,1]$ où $[0,1]$ représente l'intervalle des nombres réels entre 0 et 1. La fonction de coût exprimant l'heure d'arrivée de l'itinéraire est exprimée ci-dessous et représentée par la *figure 6*. $f(\epsilon_A, \epsilon_B) = \max\{\epsilon_A t_A^a, \epsilon_B t_B^a\}$

Pour l'instant, le minimum de la fonction de coût f est trivial. Sachant qu'il faut arriver le plus tôt possible, le mieux est de ne pas prendre de train. Il faut alors ajouter deux types de contraintes : (1) la contrainte de poser le pied dans chacun des 26 cantons, et (2) les contraintes temporelles lors des correspondances. Ces contraintes vont s'exprimer sous forme de contrainte d'inégalités. À nouveau, illustrons ce concept mathématique avec l'exemple de la multilatération.

■ L'exemple de la multilatération

Sachant que le point M se trouve à l'Est de la ligne AB, ses coordonnées doivent satisfaire l'inégalité :

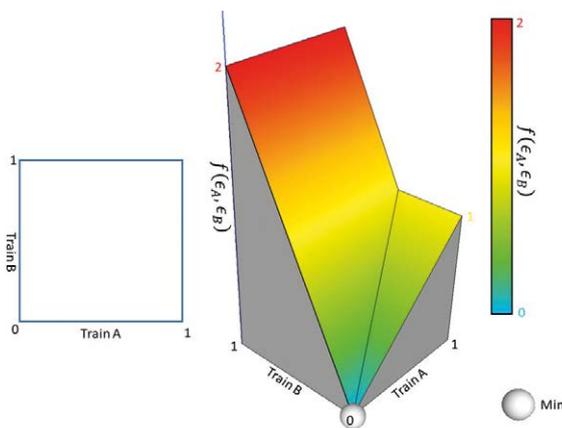


Figure 6. Fonction de coût pour le problème des deux trains

$$\chi x + \psi y + \zeta \geq 0$$

où les coefficients χ, ψ et ζ satisfont les trois conditions :

- (1) $\chi x_A + \psi y_A + \zeta = 0$
- (2) $\chi x_B + \psi y_B + \zeta = 0$
- (3) $\chi x + \psi y + \zeta \geq 0 \forall (x, y)$ à l'est de (AB).

Une fois cette contrainte appliquée à la fonction de coût $f(v_A, v_B, v_C)$ la *figure 5* devient la *figure 7*.

■ Optimisation de l'itinéraire des trains

Dans ce qui suit, nous appliquons les contraintes d'inégalité à notre problème des deux trains. Bien que les contraintes que nous souhaitons imposer à ce problème soient fondamentalement différentes de celles de la multilatération, la formulation mathématique leur ressemblera beaucoup.

Nous avons deux types de contraintes à exprimer. La première est de poser le pied dans chacun des 26 cantons. La seconde est issue des horaires de trains. Commençons avec la contrainte du nombre des cantons et considérons un pays avec deux trains. Il y a une seule gare remarquable et deux cantons (*Figure 8*). Au début de la journée, on se situe dans la gare a dans le canton α . Nous voulons poser le pied dans le canton β . Pour cela, il faut soit prendre le train A qui part de a , fait une boucle dans le canton β et revient dans la gare

a , soit prendre le train B qui fait un trajet similaire. Cette contrainte se traduit avec des opérations logiques :

$$\epsilon_A \vee \epsilon_B = Vrai$$

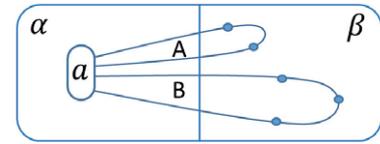


Figure 8. Carte du problème de deux trains

Pour le problème relaxé, le symbole logique \vee signifiant "ou" devient une addition.

$$\epsilon_A + \epsilon_B \geq 1$$

La *figure 9* présente l'effet de cette contrainte sur le domaine de la fonction de coût.

Le résultat de ce problème d'optimisation est évident. Il faut prendre le train A pour arriver à l'heure 1. Pour comprendre les contraintes liées aux changements de train, considérons un problème plus complexe dans lequel il y a deux gares b et c dans le canton β . Le train A va de a à b et le train B va de b à c (*Figure 10*). La première contrainte : il faut prendre au moins le train A ou le train B pour poser le pied dans le canton β , reste valide. Pour les correspondances, nous ajoutons une seconde contrainte : on ne peut pas prendre le train B si l'on n'a pas pris le train A. D'un point de vue logique, cette contrainte s'exprime par une implication.

$$\epsilon_B \Rightarrow \epsilon_A$$

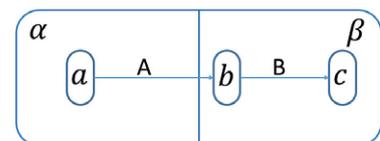


Figure 10. Carte du problème de deux trains avec une correspondance

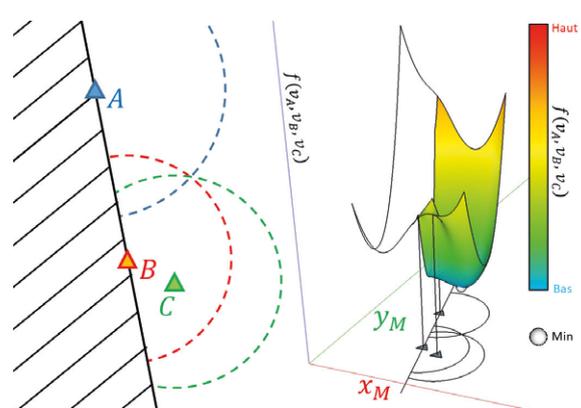


Figure 7. Fonction de coût pour la multilatération après réduction par la contrainte d'inégalité.

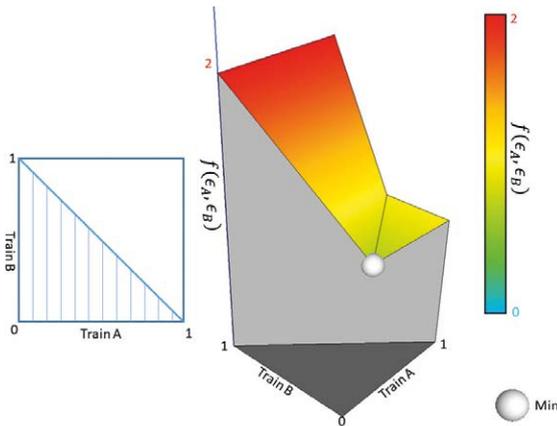


Figure 9. Contrainte d'inégalité (les hachures représentent la zone d'impossibilité) et fonction de coût f

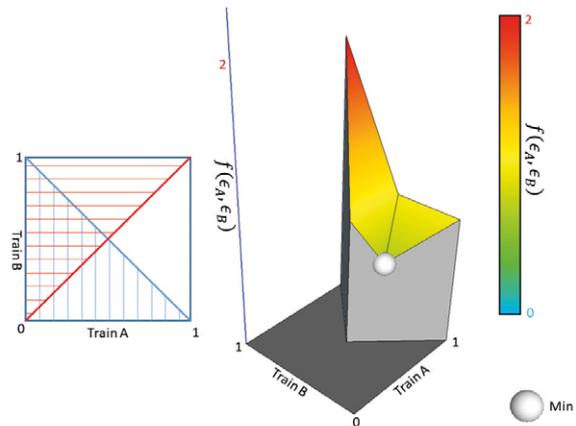


Figure 11. Deux contraintes d'inégalité (les hachures représentent la zone d'impossibilité) et fonction de coût f



Pour le problème relaxé, l'implication se traduit par une inégalité (Figure 11).

$$\epsilon_B \leq \epsilon_A$$

Bien qu'il soit évident que la solution optimale est de prendre le train A et de ne pas prendre le train B, ce problème permet d'introduire les outils mathématiques nécessaires à la résolution du problème complet.

■ Application au problème complet

La condition de visiter le canton c , s'exprime par la condition logique suivante, où Ω_c est l'ensemble des trains qui partent, passent ou arrivent dans le canton c .

$$\bigvee_{i \in \Omega_c} \epsilon_i = \text{Vrai}$$

Cette contrainte doit être validée pour tout canton c de l'ensemble des cantons. En remplaçant le "ou" logique (noté "V") par une somme et l'égalité par une inégalité, on obtient la formulation algébrique suivante.

$$\sum_{i \in \Omega_c} \epsilon_i \geq 1$$

La prise en compte des horaires est beaucoup plus compliquée. On étudie nos déplacements avant le départ du train n : il faut être arrivé dans la gare de départ de n plus de fois que l'on en est partis. On note Ω_n^a l'ensemble des trains qui arrivent à la gare de départ du train n avant que n ne parte. Similairement, Ω_n^d est l'ensemble des trains qui partent de la gare de départ du train n avant que n ne parte. Il faut encore considérer le cas particulier de la gare de départ du premier train pour laquelle le nombre de départs peut être égal au nombre d'arri-

vées plus 1. On introduit la variable ζ_n égale à 1 si le train n part de la gare de départ, 0 si non.

$$\zeta_n + \sum_{j \in \Omega_n^a} \epsilon_j - \sum_{k \in \Omega_n^d} \epsilon_k \geq \epsilon_n$$

Comme pour toutes les autres relaxations, le problème exact est obtenu quand $\epsilon_n \in \{0,1\}$. Pour des raisons de simplicité, la formulation logique a été omise.

Finalement, la relaxation du problème minimisant le temps d'arrivée du parcours fait intervenir N trains, les ensembles Ω_c , Ω_n^a et Ω_n^d étant des sous-ensembles de $\{1, \dots, N\}$.

Ceci est un problème d'optimisation linéaire. Il peut être résolu par des méthodes telles que la méthode du simplexe^[15] ou des points intérieurs^[16]. Une fois ce problème résolu pour des variables ϵ appartenant à l'intervalle $[0,1]$, il faut fixer les valeurs ϵ soit à 0, soit à 1 (il est évidemment impossible de prendre un train "à 75 %"). Cette opération s'apparente à un autre problème connu des topographes : la résolution des ambiguïtés entières dans le cadre d'un calcul d'une ligne

de base entre deux antennes GNSS. Ce dernier problème fait intervenir deux types d'inconnues à déterminer. Des nombres réels (coordonnées des points) et des entiers naturels (nombre de cycles entiers de l'onde entre le satellite et le récepteur GNSS, la phase restante étant mesurée par le récepteur^[17]). Notre problème de trains est finalement plus simple puisqu'il ne fait intervenir que des valeurs booléennes. La méthode de résolution est similaire. Après avoir résolu le problème relaxé, il faut récursivement résoudre des sous-problèmes avec les contraintes d'intégralité actives. Ceci peut se faire par exemple par un algorithme de séparation et évaluation^[18-19]. Dans le pire des cas, il faut tester toutes les 2^N combinaisons des N trains possibles. Il est donc très important de choisir un nombre limité de gares remarquables pour que N soit petit et le problème soit assez simple pour qu'on arrive à le résoudre tout en étant assez complexe pour trouver un bon trajet. Pour la résolution, nous avons utilisé le solveur Gurobi^[20] dialoguant avec le langage Matlab via l'interface/solveur yalmip^[21].

$$\begin{aligned} \min_{\epsilon \in \mathbb{R}^N} \quad & \max_{n \in \{1, \dots, N\}} \epsilon_n t_n^a \\ \text{tel que} \quad & \sum_{i \in \Omega_c} \epsilon_i \geq 1 \quad \forall c \in \mathcal{C} \\ & \zeta_n + \sum_{j \in \Omega_n^a} \epsilon_j - \sum_{k \in \Omega_n^d} \epsilon_k \geq \epsilon_n \quad \forall n \in \{1, \dots, N\} \\ & 0 \leq \epsilon_n \leq 1 \quad \forall n \in \{1, \dots, N\} \end{aligned}$$



Résultats : calcul de l'itinéraire et vérification sur le terrain

L'algorithme implémenté a été exécuté sur un ordinateur de bureau avec un processeur : i7-6700 CPU, 3.4GHz, 64GB RAM. Après 10 jours de calcul d'optimisation, l'itinéraire proposé permettait un temps de parcours de 16 h 54 (Tableau 3). A noter qu'on n'a pas de garantie sur l'optimalité de cette solution qui a été trouvée pour une gare de départ fixe. C'est-à-dire qu'en changeant la gare de départ ou en laissant l'algorithme tourner plus longtemps, il peut être possible de trouver un itinéraire encore plus rapide.

Les règles traditionnelles d'écriture d'une publication scientifique exigent la présentation de la validation expérimentale des méthodes proposées. Notre expérimentation scientifique s'apparente davantage à une aventure digne de Phileas Fogg et de Jean Passepartout^[22].

Départ à 5h24 de Saint-Maurice, en Valais. 6h48 changement à Genève. 8h32, Neuchâtel, les difficultés ont commencé. 9 min de retard annoncé sur le trajet Neuchâtel-Berne. Un tel retard pouvait toujours être rattrapé, et ne posait pas de problèmes majeurs puisque le temps de correspondance à Berne était assez large. Malheureusement, ce retard s'est empiré, ce qui a causé un retard de 20 min à l'arrivée à Berne. Sachant que le record ne serait pas pour cette fois, nous avons alors choisi de continuer notre itinéraire, avec une heure de retard. Le but n'était plus de battre le record, mais de prouver que – en l'absence de problèmes de correspondances – notre méthode permet de battre ce record. Pour assurer la réussite de notre défi, nous avons choisi de séparer les tâches : Dirk Lauinger dans le rôle de Phileas Fogg pour poser le pied dans chacun des 26 cantons, et Emmanuel Cledat dans le rôle de Jean Passepartout pour assurer l'intendance de l'aventure. Ce choix a été renforcé par les problèmes rencontrés pour atteindre le canton de Thurgovie. L'option proposée par notre algorithme était d'aller poser le pied à Stein-am-Rhein, petite enclave du canton de Schaffhouse entre l'Alle-

Gare de départ	départ	Gare arrivée	arrivée		
St-Maurice				VS	
St-Maurice	5:24	Lausanne	6:10	VD	
		Genève	6:48	GE	
Genève	7:15	Neuchâtel	8:24	NE	
Neuchâtel	8:32	Kerzers	8:48	FR	
		Bern	9:07	BE	
Bern	9:13	Biel/Bienne	9:38		
Biel/Bienne	9:49	Grenchen Nord	9:59	SO	
		Delémont	10:18	JU	
		Laufen	10:34	BL	
		Basel	10:53	BS	
Basel	11:17	Olten	11:48	SO	
		Zofingen	11:56	AR	
		Luzern	12:30	LU	
Luzern	12:42	Alpnachstad	12:59	OW	
Alpnachstad	12:59	Hergiswil	13:03	NW	
		Luzern	13:17		
Luzern	13:18	Arth-Goldau	13:46	SZ	
Arth-Goldau	13:50	Bellinzona	14:47	TI	
Bellinzona	15:07	S. Vittore, Zona industrielle	15:18	GR	
S. Vittore, Zona industrielle	15:31	Bellinzona	15:50		
Bellinzona	16:13	Flüelen	16:48	UR	
		Arth-Goldau	17:09		
Arth-Goldau	17:13	Zug	17:27	ZG	
		Zürich HB	17:51	ZH	
Zürich HB	18:05	Schaffhausen	18:43	SH	
13 Schlatt bei Diessenhofen, Thurgau				En Publibike	TG
Schaffhausen	19:16	Zürich HB	19:55		
Zürich HB	20:12	Ziegelbrücke	20:58	SG	
Traverser la rivière qui fait la frontière par le pont				GL	
Ziegelbrücke	21:02	Uznach	21:14		
Uznach	21:16	Herisau	21:46	AR	
Herisau	21:58	Jakobsbad	22:18	AI	

Tableau 3. Horaire proposé par notre algorithme, réajusté manuellement.

magne et la Thurgovie. Cependant, des erreurs dans l'utilisation des données ont conduit à des horaires de départ erronés pour le train entre Winterthur et Stein am Rhein, ce qui a conduit à devoir modifier manuellement l'itinéraire. La solution choisie a alors été d'emprunter des PubliBikes : vélos en libre-service à la gare de Zürich, les emmener à Schaffhouse avec nous

dans le train, pédaler pendant 3.7 km pour rejoindre le canton de Thurgovie, puis faire le trajet inverse avant de poursuivre le trajet calculé par notre algorithme. Malheureusement, des actes de vandalisme sur les PubliBikes Zürichoïses ont rendu cette option inexploitable. La solution retenue a alors été de nous séparer à Bellinzona. Pendant que Dirk allait rejoindre le canton des





Grisons avec le car postal, Emmanuel partait à Schaffhouse pour louer des vélos proposés par les CFF. La suite de l'histoire est commune : retrouvailles à Schaffhouse, Thurgovie à vélo et retour à Schaffhouse pour finir l'aventure sur les rails.

Lors de ce voyage, deux types de données ont été acquises. Des données GNSS, avec un récepteur dans le sac à dos porté par chacun des aventuriers, et des photos de chaque canton (Figure 2). Cet événement a été médiatisé par 13 reportages ou articles de journaux qui relatent plus en détails les aspects pratiques et culturels du voyage [23, 25].

Conclusion

Cette aventure est la preuve qu'une approche algorithmique permet de trouver l'itinéraire permettant de battre le record du *SwissTrainChallenge* qui consiste à poser le pied dans chacun des 26 cantons de la confédération helvétique le plus rapidement possible, le tout, en n'empruntant que des transports publics.

La particularité de notre méthode est le couplage de l'intuition humaine (dans la sélection des gares remarquables) à la puissance de calcul d'un ordinateur (pour la résolution du problème d'optimisation).

Nous avons aussi montré la qualité du réseau ferré suisse. En effet, excepté le train Neuchâtel-Berne, l'ensemble des trains a montré une ponctualité exemplaire ! Les points difficiles du parcours ont été réussis avec plus de facilité que prévu. Notre arrivée à 23 h 18 à Jakobsbad nous a prouvé que battre le record du *SwissTrainChallenge* en 16h54 est possible. Personnellement, nous serions très heureux de voir une autre équipe battre le record du *SwissTrainChallenge* en utilisant notre tableau d'horaire ou notre algorithme. ●

Contacts

Emmanuel CLEDAT

Doctorant au laboratoire de Topométrie de l'EPFL - emmanuel.cledat@epfl.ch

Dirk LAUINGER

Doctorant à l'EPFL à la Chaire d'analyse de risque et d'optimisation et au Centre d'Énergie - dirk.lauinger@epfl.ch

Remerciements

Nous souhaiterions remercier l'ensemble des journalistes qui ont médiatisé l'événement du *SwissTrainChallenge*. En particulier, Anne-Muriel Brouet, Laura Cipriano et Alain Herzog, travaillant pour l'EPFL, ainsi que Nicolas Rossé et François Isoard pour leur reportage pour la RTS (Radio et Télévision Suisse).

Bibliographie

- Rossé, N. *Les 26 cantons en un jour de train, c'est possible !* RTS (2015) <https://www.rts.ch/play/tv/19h30/video/nicolas-rosse-teste-le-reseau-ferroviaire-de-la-suisse-en-faisant-un-tour-des-26-cantons-en-20h?id=7008456&station=a9e7621504c6959e35c3ecbe7f6bed0446cdf8da>
- Swiss Train Challenge : 26 cantons en 18 h 31. <https://www.facebook.com/swisstrainchallenge/>
- Record-breaking friends visit all 26 Swiss cantons in 17 hours. *The Local* (2017).
- Plate-forme Open Data des transports publics suisses. <https://opentransportdata.swiss/fr/>
- Ruedin, D. *A Modified Shapefile for Plotting Swiss Cantons*. (2015). <https://druedin.com/2015/11/21/a-modified-shapefile-for-plotting-swiss-cantons/>
- Réseau ferré <https://opendata.swiss/fr/dataset/schienennetz>
- Gleyze, J.-F. *Réseaux, territoires et accessibilité*. (Institut Géographique National, 2001).
- Hullo, J.-F. *Fine registration of kilo-station networks - a modern procedure for terrestrial laser scanning data sets*. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* (2016).
- Hart, P. E., Nilsson, N. J. & Raphael, B. *A formal basis for the heuristic determination of minimum cost paths*. *IEEE Trans. Syst. Sci. Cybern.* 4, 100-107 (1968).
- Nemhauser, G. L. & Wolsey, L. A. *Integer and combinatorial optimization*. (Wiley, 1983).
- Jospin, L. & Ortolo, M. *Essaim de drones pour la cartographie de grandes zones, comment optimiser le plan de vol*. *XYZ* n°149 pp 25-31 (2016)
- Egels, Y. & Laroze, E. *La topo au disto, bon marché, tient dans la poche*. *XYZ* n°149 pp 49-52 (2016).
- Legendre, A.-M. *Nouvelles méthodes pour la détermination des orbites des comètes*. (1805).
- Gauß, C. F. *Theoria Motus Corporum Coelestium in sectionibus conicis solem ambientium*. (1809).
- Murty, K. G. *Linear programming*. (Wiley, 1983).
- Karmakar, N. *A new polynomial-time algorithm for linear programming*. in Sixteenth annual ACM symposium on the theory of computing (1984).
- Teunissen, P. J. G. & Verhagen, S. *GNSS Ambiguity Resolution: When and How to Fix or not to Fix?* in IV Hotine-Marussi Symposium on Theoretical and Computational Geodesy (eds. Xu, P., Liu, J. & Dermanis, A.) 143-148 (2008)
- Land, A. H. & Doig, A. G. *An automatic method of solving discrete programming problems*. *Econometrica* 497-520 (1960).
- Little, J. D., Murty, K. G., Sweeney, D. W. & Karel, C. *An algorithm for the traveling salesman problem*. *Operations Research* 972 (1963).
- Gurobi Optimization <http://www.gurobi.com/>
- Lofberg, J. *YALMIP : a toolbox for modeling and optimization in MATLAB*. in 2004 IEEE International Conference on Robotics and Automation pp. 284-289 (2004).
- Verne, J. *Le tour du monde en quatre-vingts jours*. (J. Hetzel, 1873).
- Students use algorithm in bid to visit all 26 Swiss cantons in less than 24 hours...by train*. *The Local* (2018).
- Late train derails bid to set new time record for visiting all 26 Swiss cantons by public transport*. *The Local* (2018).
- Faire le tour des 26 cantons en un jour, c'est le défi des transports publics suisses*. RTS <https://www.rts.ch/play/tv/19h30/video/faire-le-tour-des-26-cantons-en-un-jour-cest-le-defi-destransports-publics-suisses?id=9827914>

ABSTRACT

The Swiss Train Challenge is to set foot in all 26 cantons of Switzerland in as little time as possible, using only public transportation. Relying on human intuition informed by a geographical information system to select the relevant train stations, and on computational power to solve the resulting mixed-integer linear optimization problem, we find a solution that beats the current record of 17 hours and 19 minutes, by 25 minutes. When testing our itinerary in practice, we made all connections but one for which the arriving train was 20 minutes delayed - a rare event in Switzerland. This is the first time that an algorithm has been used to calculate the Swiss Train Challenge itinerary.