

# Combinaison d'APIs pour le calcul d'itinéraire multimodal

Sean Shorten<sup>1,2</sup>, Dominique Feillet<sup>1</sup>, Alexandre Iglesias<sup>2</sup>

<sup>1</sup> École nationale supérieure des mines de Saint-Étienne, SFL, Gardanne, France

`{sean.shorten,feillet}@emse.fr`

<sup>2</sup> Cityway, Aix-en-Provence, France

**Mots-clés :** *recherche opérationnelle, calcul d'itinéraire, API*

## 1 Introduction

Ces dernières années, des recherches sur le calcul d'itinéraire ont produit des algorithmes performants, à la fois pour le calcul routier et le calcul en transport en commun [1]. Ces améliorations ont permis aux opérateurs et éditeurs de logiciels de créer des calculateurs d'itinéraire efficaces (Geovelo, SNCF, Moovit, Google). Cependant, ces calculateurs coexistent et ont défini leurs propres APIs (Application Programming Interface) qui ne prennent pas en compte les autres APIs existantes. Des solutions telles que EU-Spirit ou DELFI, qui se base sur des travaux précédents[2], proposent de combiner ces APIs pour obtenir une solution. Cependant, les solutions imposent des contraintes fortes sur les APIs. Notre objectif est d'améliorer les méthodes de combinaison d'APIs pour permettre de couvrir une plus grande offre de transport.

## 2 Recherche de caractéristiques

L'étude a pour but de trouver des méthodes génériques permettant de combiner plusieurs APIs. Le premier objectif fût de chercher les caractéristiques des APIs ayant un impact sur le calcul d'itinéraire. Trois caractéristiques principales ont été répertoriées :

- La dépendance au temps. Cette caractéristique dépend du mode de transport envisagé. Ainsi, le transport en commun sera toujours dépendant du temps. Mais elle dépend aussi des données accessibles par l'API. Un trajet en voiture peut être dépendant du temps si l'information sur les accidents est prise en compte par l'API.
- La possibilité de faire des requêtes multiples. Certaines APIs proposent de calculer des itinéraires avec plusieurs points de départ et d'arrivée. Cette capacité permet de réduire le nombre d'appels nécessaires. Cependant, cette fonctionnalité n'est à ce jour proposée que par un faible nombre d'APIs. Si cette option n'est pas supportée, une autre API proposant cette fonctionnalité est utilisée en tant qu'estimateur.
- La définition des points de transition. Ces points permettent de définir les transitions potentielles lors de la combinaison de deux modes. Les modes tels que le transport en commun, le vélo en libre-service ou l'autopartage ont des points de transition bien définis. D'autres modes ne possèdent pas de points de transition. Les points de transition de l'API précédente et suivante selon l'ordre des enchainements des APIs sont alors utilisés. Cela est notamment le cas pour les modes privés tels que la voiture ou le vélo. Finalement, des APIs définissent leurs points de transition lors de la réponse. Ainsi, certaines APIs de covoiturage sélectionnent le lieu de prise en charge et de dépose en fonction des demandes du conducteur et du passager. Ces lieux sont alors utilisés comme points de transition.

## 3 Combinaison de deux APIs

Des méthodes permettant de combiner des APIs ont été créées en se basant sur les caractéristiques définies précédemment. Le Tableau 1 présente la répartition des méthodes créées lorsqu'aucune des deux APIs ne définissent leurs points de contact dans leurs réponses. On remarque ainsi que le caractère dépendant/indépendant du temps n'affecte pas le choix de la méthode lorsqu'il concerne la première API.

API 2 \ API 1		Dépendante du temps		Indépendante du temps	
		Calcul multiple	Estimateur	Calcul multiple	Estimateur
Dépendante du temps	Calcul multiple	Méthode 1	Méthode 2	Méthode 1	Méthode 2
	Estimateur	Méthode 3	X	Méthode 3	X
Indépendante du temps	Calcul multiple	Méthode 1bis	Méthode 4	Méthode 1bis	Méthode 4
	Estimateur	Méthode 5	Méthode 6	Méthode 5	Méthode 6

TAB. 1 – Tableau des différentes méthodes utilisées en fonction des caractéristiques des APIs

Dans le cas où une des deux APIs définit ses points de transitions dans le résultat, seule deux méthodes sont proposées. Le choix de la méthode dépend alors de la capacité de l’API restante à effectuer des calculs multiples.

## 4 Généralisation

Une généralisation pour calculer des combinaisons de plus de deux APIs est aussi proposée. Tel que présenté sur la Figure 1, deux APIs successives peuvent être assimilées à une seule API. Cela permet de retomber sur les méthodes de combinaisons de deux APIs. Il est cependant nécessaire de redéfinir les calculs multiples et détaillé pour chaque combinaison pour pouvoir utiliser les dites méthodes.

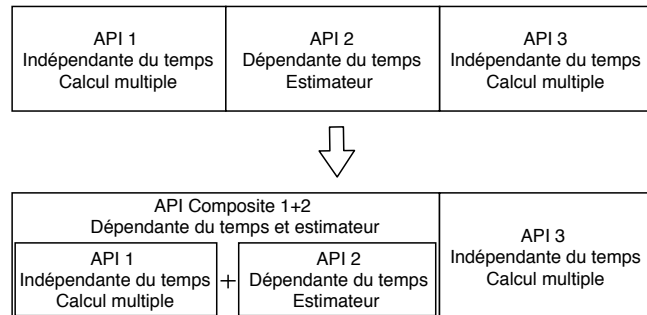


FIG. 1 – Agrégation de deux APIs en une seule

## 5 Résultats

La combinaison de deux APIs a été implémentée dans un système réel. Ce système couvre un réseau de plus de 19 000 stations de transports en commun. Trois APIs sont utilisées :

- une API fournie par Cityway. Elle propose des trajets en transport en commun, en vélo et en voiture. Elle permet d’effectuer des calculs multiples sur ces trois modes et elle est utilisée comme estimateur pour la voiture et le vélo
- l’API de TomTom pour le calcul routier. Elle fournit des résultats dépendant du temps.
- l’API de Géovélo pour les trajets en vélo. Les résultats fournis par cette API sont considérés comme indépendant du temps.

Les méthodes proposées permettent d’obtenir des itinéraires alliant transports en commun et vélo ou voiture. Elles sont cependant plus longues qu’un calculateur centralisé car elles nécessitent d’attendre les réponses des APIs.

## 6 Conclusion

Les méthodes présentées permettent d’abstraire la combinaison d’APIs pour le calcul d’itinéraire. Cette abstraction permet de ne pas imposer aux APIs des caractéristiques précises. Ces méthodes peuvent ainsi être utilisées pour créer un calculateur purement décentralisé. Elles permettent aussi à un calculateur centralisé existant d’intégrer des APIs pour proposer des trajets multimodaux variés.

## Références

- [1] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F Werneck. Route planning in transportation networks. In *Algorithm engineering*, pages 19–80. Springer, 2016.
- [2] Rolf H Möhring. Verteilte verbindingssuche im öffentlichen personenverkehr graphentheoretische modelle und algorithmen. In *Angewandte Mathematik, insbesondere Informatik*, pages 192–220. Springer, 1999.