# Constrained Time-Dependent Adaptive Eco-Routing Navigation System

Matěj Kubička

université
PARIS-SACLAY

NNT : 2017SACLS434

# Constrained Time-Dependent Adaptive Eco-Routing Navigation System

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

École doctorale n°580: Sciences et technologies
de l'information et de la communication (STIC)
Spécialité de doctorat: Automatique

Thèse présentée et soutenue à Gif-sur-Yvette, le 16/11/2017, par

**M. Matěj Kubička**

Composition du Jury :

| | |
|---|---|
| Mme Dorothée Normand-Cyrot<br>Directrice de recherche, CNRS | Présidente |
| M. René Natowicz<br>Professeur, ESIEE Paris | Rapporteur |
| M. Michel Basset<br>Professeur, Laboratoire MIPS, UHA | Rapporteur |
| M. Antonio Sciarretta<br>Expert Scientist, IFPEN | Examinateur |
| M. Arben Cela<br>Professeur, ESIEE Paris | Examinateur |
| Mme Brigitte D’Andrea-Novel<br>Professeure, Mines ParisTech | Examinatrice |
| M. Hugues Mounier<br>Professeur, L2S | Directeur de thèse |
| M. Silviu-Iulian Niculescu<br>Directeur de recherce, CNRS | Invité |
| M. Philippe Moulin<br>CEO, DriveQuant | Invité |

Thèse de doctorat

# *Résumé*

L'éco-routage est une méthode de navigation du véhicule qui sélectionne les trajets vers une destination minisant la consommation de carburant, la consommation d'énergie ou les émissions de polluants. C'est l'une des techniques qui tentent de réduire les coûts d'exploitation et l'empreinte environnementale du véhicule. Ce travail passe en revue les méthodes actuelles d'éco-routage et propose une nouvelle méthode pour pallier leurs insuffisances.

La plupart des méthodes actuelles attribuent à chaque route du réseau routier un coût constant qui représente la consommation du véhicule ou la quantité de polluants émis. Un algorithme de routage optimal est ensuite utilisé pour trouver le chemin qui minimise la somme de ces coûts. Différentes extensions sont considérées dans la littérature. L'éco-routage contraint permet d'imposer des limites sur le temps de trajet, la consommation d'énergie et les émissions de polluants. L'éco-routage dépendant du temps permet le routage sur un graphique avec des coûts qui sont fonction du temps. L'éco-routage adaptatif permet de mettre à jour la solution d'éco-routage au cas où elle deviendrait invalide en raison d'un développement inattendu sur la route.

Il existe des méthodes d'éco-routage optimales publiées qui résolvent l'éco-routage dépendant du temps ou l'éco-routage contraint ou l'éco-routage adaptatif. Chacun vient avec des frais généraux de calcul considérablement plus élevés par rapport à l'éco-routage standard et, à la connaissance de l'auteur, aucune méthode publiée ne prend en charge la combinaison des trois: éco-routage adaptatif dépendant du temps contraint.

On soutient dans ce travail que les coûts d'acheminement sont incertains en raison de leur dépendance au trafic immédiat autour du véhicule, du comportement du conducteur et d'autres perturbations. Il est en outre soutenu que puisque ces coûts sont incertains, il y a peu d'avantages à utiliser un routage optimal car l'optimalité de la solution ne tient que tant que les coûts de routage sont corrects. Au lieu de cela, une méthode d'approximation est proposée dans ce travail. La charge de calcul est plus faible car la solution n'est pas requise pour être optimale. Cela permet l'éco-routage adaptatif dépendant du temps contraint.

***Mots clés:*** *eco-routage, map-matching, systémes de navigation, transport vert*

# Abstract

Eco-routing is a vehicle navigation method that selects those paths to a destination that minimize fuel consumption, energy consumption or pollutant emissions. It is one of the techniques that attempt to lower vehicle's operational cost and environmental footprint. This work reviews the current eco-routing methods and proposes a new method designed to overcome their shortcomings.

Most current methods assign every road in the road network some constant cost that represents either vehicle's consumption there or the amount of emitted pollutants. An optimal routing algorithm is then used to find the path that minimizes the sum of these costs. Various extensions are considered in the literature. Constrained eco-routing allows imposing limits on travel time, energy consumption, and pollutant emissions. Time-dependent eco-routing allows routing on a graph with costs that are functions of time. Adaptive eco-routing allows updating the eco-routing solution in case it becomes invalid due to some unexpected development on the road.

There exist published optimal eco-routing methods that solve either the time-dependent eco-routing, or constrained eco-routing, or adaptive eco-routing. Each comes with considerably higher computational overhead with respect to the standard eco-routing and, to author's best knowledge, no published method supports the combination of all three: constrained time-dependent adaptive eco-routing.

It is argued in this work that the routing costs are uncertain because of their dependence on immediate traffic around the vehicle, on driver's behavior, and other perturbations. It is further argued that since these costs are uncertain, there is little benefit in using optimal routing because the optimality of the solution holds only as long as the routing costs are correct. Instead, an approximation method is proposed in this work. The computational overhead is lower since the solution is not required to be optimal. This enables the constrained time-dependent adaptive eco-routing.

**Keywords:** *eco-routing, map-matching, navigation systems, green transportation*

# *Acknowledgements*

I would like to express my sincere gratitude to my advisors Antonio Sciarretta, Hugues Mounier, Arben Cela and Silviu I. Niculescu for their continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I would also like to thank my fellow doctoral students for their feedback, cooperation and of course friendship.

Last but not least, I would like to thank to my parents Dana and Tomáš, who have provided me through moral and emotional support in my life. I am also grateful to my amazing partner, Eva, and all other family members and friends who have supported me along the way.

# Contents

*Mým rodičům*

*To my parents*

# Chapter 1

# Résumé substantiel en langue française

L'éco-routage est une méthode de navigation de véhicule qui choisit les chemins en minimisant la consommation de carburant, la consommation d'énergie ou l'émission de polluants pour un voyage vers une destination donnée. Le transport routier a de nombreux effets négatifs sur l'environnement. Il est le plus grand contributeur au réchauffement climatique par les émissions de $CO_2$ (Fuglestvedt et al., 2008) et est responsable de la détérioration de la qualité de l'air dans les zones à réseau routier dense par les émissions de particules et autres polluants. Par exemple, la matière particulaire est un polluant dangereux: l'étude de Raaschou-Nielsen et al., 2013 a révélé que pour chaque augmentation de 10 $\mu$g/m$^3$ de particules dans l'air, le taux de cancer du poumon dans la région a augmenté de 22%. En dehors de cette dimension environnementale, il y a aussi un côté économique du problème. Le coût monétaire associé à la consommation d'énergie (carburant) est considérable. Cela motive des techniques telles que l'éco-routage visant à réduire les émissions de polluants, ou la consommation d'énergie (carburant), ou les deux.

Les premières études sur le sujet ont confirmé la dépendance des émissions de polluants et de la consommation d'énergie (carburant) sur le trajet emprunté. Ahn and Rakha, 2007 ont montré dans une étude de cas que prendre un chemin plus lent peut économiser du carburant au prix d'un temps de trajet prolongé. Ericsson et al., 2006 ont montré que les conducteurs ne choisissent pas toujours les chemins avec la meilleure économie de carburant. Le premier système de navigation d'éco-routage a été publié par Barth et al., 2007. De nombreuses études de cas et méthodes d'éco-routage ont été publiées depuis.

Différents auteurs considèrent différentes variantes de l'éco-routage. Comme indiqué ci-dessus, certains visent à réduire les émissions de polluants tandis que d'autres visent à réduire la consommation. Dans le premier cas, le terme "éco-routage" signifie "écologisation" alors que dans le dernier cas, il pourrait être à la fois "écologique" et "économique" (certains auteurs suggèrent que les émissions de $CO_2$ sont réduites avec la consommation). Les méthodes de routage sont généralement conçues en fonction d'un type de véhicule spécifique. Certaines sont conçues pour des véhicules conventionnels avec des moteurs à combustion interne, tandis que d'autres envisagent des véhicules électriques et hybrides. Les véhicules conventionnels émettent des polluants et consomment du carburant. Les véhicules électriques consomment de l'énergie plutôt que du carburant et n'émettent aucun polluant. Les véhicules hybrides ont à la fois un moteur électrique et un moteur à combustion interne. Il existe une grande variété de topologies, et leur comportement n'est pas facile à caractériser.

Il existe également des différences dans la façon dont les méthodes d'éco-routage sont évaluées. Certains auteurs définissent les économies en comparant la quantité

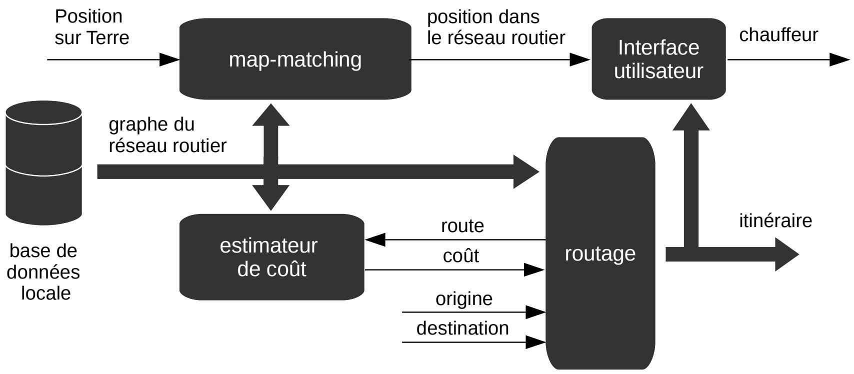


FIGURE 1.1: Système de navigation de base pour l'éco-routage (version française).

d'intérêt (le consommation de carburant, le consommation d'énergie ou les émissions de polluants) aux trajets les plus courts, tandis que d'autres se comparent aux trajets les plus rapides. Les chemins les plus courts sont ceux qui minimisent la distance jusqu'à la destination. Les chemins les plus rapides sont ceux qui minimisent le temps de trajet prévu.

On sait peu sur les avantages que peut offrir l'éco-routage. La performance de l'éco-routage dépend des propriétés du réseau routier et du véhicule. Les études de cas publiées sont souvent conçues comme des preuves du concept de l'éco-routage, et elles se concentrent donc sur des scénarios où l'éco-routage est susceptible d'exceller. Ceci, cependant, ne dit rien sur les économies typiques observées dans l'utilisation quotidienne. Certaines études de cas et la plupart des publications proposant de nouvelles méthodes d'éco-routage contiennent néanmoins une évaluation de l'épargne moyenne. Toutefois, le même modèle de consommation ou d'émission de polluants est souvent utilisé pour identifier les éco-itinéraires et d'estimer les économies sur elles. Ce type d'évaluation ne peut jamais aboutir à des économies négatives, ce qui signifie qu'il est incapable de détecter les échecs.

Il existe une technique étroitement liée à l'éco-routage appelée éco-conduite. Elle essaie d'atteindre les mêmes objectifs que l'éco-routage mais utilise différents moyens pour le faire. Au lieu de réduire les émissions de polluants ou la consommation par le choix du trajet, l'eco-conduite est prédéterminée et fixée en optimisant le comportement du conducteur pendant le trajet. Puisque les deux techniques partagent le même objectif, il devrait être possible de les combiner pour obtenir de meilleurs résultats. Elles peuvent être utilisées indépendamment, mais il est probable qu'il existe un certain degré de co-dépendance entre elles.

## 1.1 Structure du système de navigation éco-routage

La structure de base d'un système de navigation par éco-routage est schématisée à la Figure 1.1. Il y a quatre éléments principaux: la base de données locale, le bloc d'estimation des coûts, le bloc de routage et le bloc de "map-matching". Le bloc d'interface utilisateur est hors de portée de ce travail et n'est plus discuté ici.

- *La base de données locale* contient le graphe du réseau routier. C'est un graphe orienté dont le but est de décrire la connectivité entre différents endroits sur la carte. Il contient également l'organisation spatiale du réseau routier (formes de route) et d'autres métadonnées telles que les noms de routes, les longueurs et les pentes.

- *L'estimateur de coût* est utilisé pour attribuer des coûts aux bords du graphe du réseau routier. La nature de ces coûts dépend de la manière dont le problème d'éco-routage est posé. Cela peut être la consommation de carburant, la consommation d'énergie ou les émissions de polluants.

- *Le routage* trouve le chemin dans le graphe du réseau routier entre l'origine et la destination données avec la somme la plus basse des coûts sur les bords du chemin.

- *Le map-matching* convertit la position référencée par rapport à la Terre en une position dans un graphe de réseau routier. Le système de positionnement et le graphe du réseau routier contiennent tous deux des erreurs qui peuvent rendre la conversion difficile. Il n'y a pas d'approche standard à ce problème.

La structure de la Figure 1.1 est identique au système de navigation le plus couramment utilisé. La seule différence entre l'éco-routage et le routage à plus courte distance réside dans la nature des coûts. Dans le routage à plus courte distance, le coût est simplement une longueur de route. C'est une constante non négative qui peut être enregistrée dans la base de données. En éco-routage, le coût est une fonction multivariée.

La plupart des méthodes d'éco-routage publiées utilisent l'algorithme de routage de Dijkstra (Dijkstra, 1959) ou un algorithme dérivé de celui-ci (tel que l'algorithme de recherche A*, Hart et al., 1968). Ce sont des algorithmes sophistiqués capables d'identifier la solution optimale en temps asymptotiquement optimal. Ils supposent cependant que les coûts sont des scalaires constants non négatifs. Ceci est limitant pour les applications d'éco-routage. Les quantités représentées par les coûts sont généralement des variables dépendant du temps qui peuvent être négatives si le coût représente la consommation d'énergie et que le véhicule considéré peut récupérer l'énergie de freinage. Le problème des coûts variables est examiné plus en détail à la Section 2.2. Le problème des coûts négatifs peut être résolu avec l'algorithme de Bellman-Ford (Bellman, 1958), mais l'effort de calcul sur un grand réseau routier peut devenir significatif. Une autre option consiste à rendre tous les coûts positifs sans modifier la structure des chemins minimaux dans le graphe en utilisant l'algorithme de Johnson (Johnson, 1977).

Notez que le routage naïf est intraitable par calcul pour tous les réseaux routiers, sauf les plus simples. Une méthode de routage naïf ferait d'abord une liste de tous les chemins entre l'origine et la destination données, puis rechercherait le chemin qui minimise la fonction de coût donnée. Le nombre de chemins à considérer n'est pas fini à moins que seuls les chemins simples soient considérés (chemins sans boucles). Même alors, le nombre de trajets croît exponentiellement avec le nombre d'intersections dans le réseau routier. C'est pourquoi les algorithmes de routage tels que l'algorithme de Dijkstra sont importants. Ils sont capables d'identifier la solution optimale dans un temps polynomial.

De nombreux estimateurs de coûts ont été proposés spécifiquement pour les applications d'éco-routage. Certains sont basés sur des modèles de régression, et d'autres sont dérivés de la seconde loi du mouvement de Newton. Le dénominateur commun entre eux est qu'ils sont macroscopiques. En général, la plupart des modèles publiés de consommation de carburant, de consommation d'énergie et d'émission de polluants peuvent être classés comme macroscopiques ou microscopiques. Les modèles microscopiques sont basés sur des équations différentielles. Les modèles macroscopiques sont basés sur des équations algébriques en forme fermée. Cela nécessite souvent des simplifications grossières, mais le modèle résultant est rapide à calculer et nécessite peu d'informations. La raison pour laquelle les estimateurs de coût utilisés dans l'éco-routage sont macroscopiques est que les modèles microscopiques ont besoin de données de trajet enregistrées (profil de vitesse du véhicule) afin d'estimer le coût. Ces informations ne sont pas disponible pour la planification d'un voyage.

Notez que les estimateurs de coût macroscopiques sont connus pour leurs erreurs d'estimation. Par exemple, dans l'une des premières études sur l'éco-routage, Ahn and Rakha, 2008 concluent que les outils d'estimation des émissions macroscopiques (par exemple, MOBILE6) peuvent produire des conclusions erronées étant donné qu'ils ignorent le comportement transitoire des véhicules le long d'une route. Il est peut-être surprenant alors qu'aucune méthode d'éco-routage citée dans ce travail n'en tienne compte: les coûts sont toujours supposés exacts, sans erreurs.

Le map-matching a deux applications dans l'éco-routage. Premièrement, il permet à l'interface utilisateur de diriger le conducteur vers la destination en identifiant la position actuelle du véhicule sur le réseau routier. Deuxièmement, il 1est nécessaire lors du traitement des voyages historiquement enregistrés afin de déduire des connaissances sur la consommation du véhicule sur les routes du réseau routier (c'est quelque chose qui est souvent fait dans la littérature). Au moment de l'écriture, les systèmes dominants de positionnement des véhicules sont basés sur la navigation par satellite. Ils identifient la position du véhicule en termes de latitude et de longitude. Le map-matching est nécessaire pour assigner les trajets enregistrés aux bonnes routes du réseau routier. Les erreurs du map-matching se propagent dans le système. Elles peuvent réduire la précision de l'estimateur de coût et faire en sorte que l'interface utilisateur commence à donner des conseils trompeurs au conducteur.

## 1.2 Extensions d'éco-routage

Les trois extensions de l'éco-routage de base discutées dans la littérature sont:

- *L'éco-routage contraint* permet de fixer des limites sur la consommation de carburant, la consommation d'énergie, le temps de trajet et les émissions de polluants. Il a d'abord été considéré dans l'article écrit par Juřík et al., 2014. Les auteurs proposent un algorithme de routage optimal pour résoudre un problème d'optimisation sous contrainte avec des coûts additifs. Ce problème est connu pour être NP-complet, voir Wang and Crowcroft, 1996 pour la preuve.
- *L'éco-routage dépendant du temps*, est l'éco-routage oú les coûts sont fonctions du temps et non des constantes. Il a d'abord été envisagé dans Kluge, 2011. Un tel routage permet de prendre en compte le développement de la situation de trafic lors du routage. Il est probable, par exemple, qu'il existe différentes éco-itinéraires. la nuit et l'après-midi car les densités de trafic sont différentes. Peut-être plus important encore, l'éco-route peut être affectée par des arrêts sur les routes avec des feux de circulation. L'éco-routage dépendant du temps peut, en théorie, naviguer le véhicule de telle sorte que les pertes qui en résultent sont minimisées. Il existe deux types d'éco-routage dépendant du temps reconnus dans la littérature:
  - *Le routage instantané*, qui suppose que les coûts sont des constantes. Ils sont estimés à l'heure de départ. Ce système d'éco-routage a la même structure que l'éco-routage indépendant du temps, sauf que l'estimateur de coût dépend du temps.

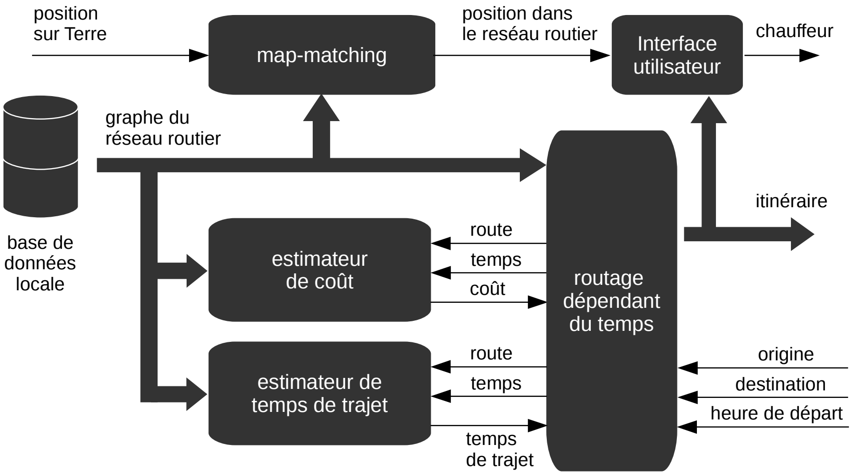


FIGURE 1.2: Système de navigation éco-routage dépendant du temps (version française).

– *Le routage dépendant du temps*, prennant en charge le routage avec des coûts qui sont fonctions du temps. Les coûts sur les routes sont alors spécifiques à l'heure d'arrivée prévue. Un modèle de temps de trajet dépendant du temps est nécessaire pour l'estimer. L'algorithme de routage doit pouvoir prendre en compte la nature variable des coûts.

Le routage dépendant du temps peut rendre le problème d'éco-routage beaucoup plus difficile à résoudre. La structure de ce système d'éco routing est représentée schématiquement à la Figure 1.2. Les estimateurs de coût et de temps de trajet sont nécessairement des modèles dépendant du temps. Le routeur identifie le chemin qui minimise la somme des coûts à une heure de départ donnée, en tenant compte du coût prévu sur ses routes à l'heure d'arrivée prévue.

- *L'éco-routage adaptatif* permet de mettre à jour la solution lorsqu'elle devient invalide. Cela pourrait devenir apparent alors que déjà sur la route que, l'éco-route identifié n'est plus valable. Il est souhaitable de détecter cette situation et de mettre à jour la solution d'éco-routage en conséquence. Il a d'abord été envisagé dans Ahn and Rakha, 2013.

  L'éco-routage adaptatif est intéressant en relation avec l'éco-routage dépendant du temps. Si le véhicule n'atteint pas la destination avec la vitesse prévue, la solution d'éco-routage n'est pas nécessairement valide. Par exemple, si le véhicule est retardé, la solution d'éco-routage n'est plus nécessairement la bonne. Cela peut être vérifié en réacheminant la position actuelle du véhicule vers la destination.

## 1.3 Les approches suivi dans ce travail

L'objectif de ce travail est de proposer un système de navigation d'éco-routage contraint, adaptatif et dépendant du temps. Les contraintes permettraient à l'utilisateur de maîtriser ses émissions et ses délais de déplacement. Le routage dépendant du temps permettrait un routage tel que la situation prédite sur les routes au moment où le véhicule est projeté pour être pris en compte. L'éco-routage adaptatif permettrait de modifier le chemin au cas où la solution originale d'éco-routage ne serait plus valide.

C'est un problème difficile à résoudre de manière optimale. L'éco-routage à contrainte optimale est NP-complet. Le routage dépendant du temps optimal recherche la solution dans un espace beaucoup plus grand par rapport au routage indépendant du temps en raison des coûts variables dans le temps. D'un autre côté, l'éco-routage adaptatif nécessite des algorithmes de routage rapides qui peuvent être ré-exécutés plusieurs fois pendant que le véhicule est en route vers sa destination. Le délai de recalcul de l'éco-route doit être minime pour minimiser la chance que le véhicule ait progressé jusqu'à présent sur l'ancien éco-route qu'il n'est plus possible de changer pour la nouvelle éco-route. Pour résumer la difficulté, il y a un conflit: le routage contraint et le routage dépendant du temps sont des tâches qui demandent beaucoup de calculs, tandis que l'éco-routage adaptatif ne fonctionne pas bien avec les routeurs intensifs en calcul.

Maintenir l'optimalité n'est pas seulement coûteux en termes de calcul, mais aussi futile. Il est soutenu dans ce travail que l'optimalité de la solution de routage est conditionnée par l'exactitude des coûts de routage dans le graphe du réseau routier, ce qui est connu pour être sujet à des erreurs considérables (comme discuté ci-dessus). Cet argument est étayé par les conclusions de ce travail car toutes les méthodes d'éco-routage étudiées n'ont pas réussi à économiser de l'énergie, dans certains cas. C'est quelque chose qui ne pourrait pas arriver avec un éco-routage optimal. Par conséquent, ce travail n'étudie pas les algorithmes de routage optimaux et se concentre plutôt sur la recherche de solutions approximatives. L'abandon de la condition d'optimalité de routage donne plus de flexibilité dans la conception des méthodes d'éco-routage. Il n'est plus nécessaire de faire les hypothèses imposées par des algorithmes de routage tels que l'algorithme de Dijkstra, par exemple. Les coûts peuvent être de véritables fonctions du temps si c'est ce que requièrent les applications d'éco-routage. Il s'agit d'identifier les bonnes approximations qui conduisent à des résultats calculables dans un temps acceptable.

La méthode d'éco-routage proposée repose sur l'hypothèse selon laquelle bien qu'il y ait de nombreux chemins vers une destination choisie, seule une petite fraction d'entre eux peuvent être des éco-routes. Il privilégie un routage simpliste et naïf, basé sur une liste de tous les chemins possibles par rapport aux algorithmes de routage optimaux connus et sophistiqués. Il suit une approche qui ne ressemble à rien de la littérature publiée. Il est basé sur des idées originales, qui peuvent ou non s'avérer utiles.

## 1.4 Sujets de recherche

Il y a trois sujets principaux étudiés dans ce travail:

1. Le modèle de consommation de véhicules pour l'éco-routage. Avoir un bon modèle d'estimation des coûts est primordial dans l'éco-routage: sa performance est conditionnée par les erreurs d'estimation des coûts. Un modèle d'estimation de consommation standard est adapté pour les applications d'éco-routage dans le Chapitre 5. Il est soigneusement reformulé sous une forme fermée et ensuite résolu, sous certaines hypothèses.

2. L'algorithme de routage. Un routage basé sur une collection présélectionnée d'éco-routes candidates est proposé au Chapitre 6. Les éco-routes candidates sont identifiées hors ligne pour une origine et une destination données, avant le départ du véhicule. Il permet de présélectionner les chemins susceptibles d'être des éco-routes dans certaines conditions. Il peut y avoir quelques dizaines ou quelques centaines de milliers de candidats. C'est une quantité négligeable, dans la plupart des cas, comparée au nombre de tous les chemins simples entre l'origine et la destination. Cela réduit l'espace de recherche et permet de prendre en charge l'éco-routage adaptatif et dépendant du temps.

3. L'évaluation fiable des performances de l'éco-routage. Comme indiqué dans les paragraphes introductifs, on sait peu de choses sur les avantages que l'éco-routage peut apporter dans un usage quotidien. Ce travail tente de faire un pas vers la résolution de ce problème. Les performances de l'éco-routage sont formellement définies au Chapitre 4. Un cadre destiné à évaluer les méthodes d'éco-routage en termes de performance est présenté au Chapitre 7. Il permet de comparer plusieurs méthodes publiées avec la méthode proposée dans ce travail.

## 1.5 Aperçu de la thèse

Ce chapitre (Chapitre 1) contient un résumé étendu de ce travail en langue française. Le problème de l'éco-routage est présenté dans le Chapitre 2. Le Chapitre 3 contient une revue des méthodes d'éco-routage et d'appariement des cartes publiées. Le Chapitre 4 contient une analyse des méthodes actuelles introduites au Chapitre 3. Le Chapitre 5 introduit un nouveau modèle de consommation de véhicules et un modèle de temps de trajet. Le Chapitre 6 propose un nouvel algorithme de routage. Le Chapitre 7 propose un cadre pour l'évaluation des méthodes d'éco-routage dans les simulations. Le Chapitre 8 traite des résultats d'identification et de validation des modèles utilisés dans ce travail. Le Chapitre 9 contient une évaluation des méthodes d'éco-routage basée sur l'algorithme de routage proposé dans le Chapitre 6 et le modèle de consommation proposé au Chapitre 5. L'évaluation est faite avec le cadre proposé au Chapitre 7.

## 1.6 Chapitre "Problem definition & analysis"

Ce chapitre définit le problème ainsi que la performance de l'éco-routage. Deux méthodes d'éco-routage publiées avec une méthode idéalisée sont étudiées pour établir une base de référence à laquelle la méthode de l'éco-routage proposée dans ce travail sera après comparée. Ensuite, un argument est avancé contre la méthodologie d'évaluation de l'éco-routage couramment utilisée dans la littérature.

Le problème de l'éco-routage est défini à la Section 4.1. Il prend en charge les contraintes et les coûts multiples dépendants du temps. La définition est assez générale pour s'appliquer à toutes les variantes de l'éco-routage dans la littérature publiée et permet en outre des variantes qui n'ont pas encore été considérées. Par exemple, il permet d'imposer des limites sur les émissions de polluants tout en minimisant la consommation du véhicule ou de ne considérer que les trajets qui ne nécessitent pas de recharge de la batterie (ou de ravitaillement) avant d'atteindre la destination.

Le modèle permet également de limiter ou interdire complètement de traverser certaines zones à certains moments. Par exemple, le routage à proximité des écoles au moment où les enfants sont là peut être interdit pour les véhicules très polluants.

La Section 4.2 traite des modèles de l'éco-routage utilisés dans la littérature publiée et définit formellement ses performances. Il résume la capacité d'une méthode de celui-ci à réduire les coûts. Les coûts peuvent correspondre à la consommation de carburant, à la consommation d'énergie ou aux émissions de polluants. Le chemin de référence est le chemin vers lequel l'éco-route est comparée. Dans ce travail, le chemin de référence est le chemin le plus court ou le chemin le plus rapide.

La performance de trois méthodes de l'éco-routage (dont deux publiées: Barth et al., 2007; Juřík et al., 2014) est étudiée à la Section 4.3. C'est une version mise à jour du travail que nous avons publié pendant la préparation de cette thèse (Kubička et al., 2016b). Un cadre d'éco-routage présenté au Chapitre 7 est utilisé pour évaluer la formation de ces méthodes. La performance des deux méthodes publiées était négatif lors de la comparaison des éco-routes vers les chemins les plus courts. Cela implique que le routage le plus court est un éco-routeur supérieur aux méthodes de Barth et al., 2007 et Juřík et al., 2014.

Alors que l'évaluation des méthodes actuelles de l'éco-routage a montré des économies négatives, il est rare de voir un tel résultat signalé dans la littérature publiée. Un examen attentif des méthodes d'évaluation utilisées dans la littérature a été effectué à la Section 4.4. Il a révélé que la méthode prédominante n'est pas capable d'observer des économies négatives par conception. Le problème est que le même modèle de consommation (ou d'émission de polluants) est utilisé à la fois pour le routage et l'évaluation. Non seulement que cette évaluation ne peux pas observer les économies négatifs, mais il est également montré dans le Théorème 1 à être sujettes à entraîner des chiffres de performance éco-routage gonflés.

## 1.7 Chapitre "Energy consumption and travel time modeling"

Les modèles de consommation et de temps de trajet sont présentés dans ce chapitre. Sa portée est limitée à leur dérivation et description. Les modèles sont ensuite identifiés et validés au Chapitre 8 et utilisés pour l'éco-routage au Chapitre 9.

La formulation standard du modèle de consommation longitudinale est examinée à la Section 5.1. Le modèle est donné dans l'équation (5.7). Il est dérivé de la deuxième loi du mouvement de Newton, en supposant que le mouvement du véhicule est limité à une direction. Il estime la puissance instantanée requise par le groupe motopropulseur et l'intègre pour obtenir la consommation globale d'énergie. La puissance instantanée est basée, entre autres, sur la vitesse instantanée du véhicule. Par conséquent, le profil de vitesse $v(t)$ doit être fourni au modèle pour estimer la consommation.

La Section 5.2 introduit une reformulation en forme fermée du modèle longitudinal (5.7). Il est soutenu dans cette section que (5.7) n'est pas bien adapté à l'éco-routage car il nécessite des informations qui ne sont généralement pas disponibles lors de la planification du voyage. Cela motive la reformulation en forme fermée du modèle longitudinal. Il est donné dans l'équation (5.14). Il repose sur la consommation énergétique d'un véhicule idéalisé (énergie sur les roues, $E_w$), sur l'énergie restituée au véhicule (énergie de freinage, $E_b$) et sur l'efficacité globale du groupe motopropulseur (efficacité globale de traction $\overline{\eta}_t$ et efficacité globale de récupération $\overline{\eta}_r$).

La solution à l'équation (5.14) pour un véhicule idéalisé est donnée dans la Section 5.3. Le véhicule idéal est un véhicule avec un groupe motopropulseur sans perte (il n'y a pas de pertes dues au chauffage et au frottement, par exemple) et avec une capacité à récupérer toute l'énergie restituée au véhicule lors du freinage. La consommation d'énergie de ce véhicule est introduite comme énergie sur roues et notée $E_w$. La solution est une équation algébrique qui dépend des statistiques de $v(t)$ (ses trois premiers moments), de la vitesse initiale, de la vitesse finale, de l'altitude initiale, de l'altitude finale et d'autres paramètres (connus). Les paramètres de vitesse initiale et finale correspondent à la variation de l'énergie cinétique du véhicule. Les paramètres d'altitude initial et final tiennent compte de la variation de l'énergie potentielle du véhicule. La force de cette solution est double. Premièrement, il parvient à réduire la dimensionnalité de l'entrée des séries temporelles de dimension arbitraire à un ensemble fermé de paramètres. Deuxièmement, il comble le fossé entre le modèle classique d'estimation de la consommation (basé sur des profils de vitesse connus) et les besoins d'un modèle de prédiction (basé sur des informations statistiques).

La limitation de la solution donnée dans la Section 5.3 est qu'elle ne s'applique qu'au véhicule idéal. Des groupes motopropulseurs plus réalistes sont examinés dans la Section 5.4. On fait valoir que bien qu'il existe une solution pratique pour l'énergie sur les roues $E_w$, trouver une solution similaire pour $E_b$ est difficile. Un modèle d'efficacité du groupe motopropulseur dont la puissance de sortie est une fonction affine de sa puissance d'entrée est proposé car il permet une solution à la consommation d'énergie $E$ qui ne dépend pas de $E_b$.

Un modèle de temps de trajet dépendant du temps est proposé à la Section 5.5. C'est un modèle simple conçu pour tirer parti de la connaissance des états de feux de circulation dans le cadre d'une simulation utilisée (présentée au Chapitre 7). Il a été incorporé ici parce que de telles informations sont susceptibles d'être disponibles dans les futures villes intelligentes. Le modèle du temps de trajet est évalué au Chapitre 8 et utilisé au Chapitre 9 lorsque l'éco-routage dépendant du temps est étudié.

## 1.8 Chapitre "Routing"

Ce chapitre concerne un algorithme de routage adapté aux applications d'éco-routage. Le concept proposé nécessite une étape de prétraitement qui identifie une collection d'éco-routes candidates. Plutôt que de router sur le graphe du réseau routier comme cela est habituellement le cas, le routage est effectué sur cette collection. La méthode proposée dans ce chapitre privilégie un routage simpliste par rapport à des algorithmes de routage optimaux sophistiqués tels que l'algorithme de Dijkstra ou de Bellman-Ford. La motivation pour cela est que ces algorithmes sont trop restrictifs, trop rigides dans leurs hypothèses. Une solution approximative basée sur une approche naïve du routage est étudiée. Il est basé sur une hypothèse selon laquelle bien qu'il y ait habituellement beaucoup de chemins possibles à prendre, il n'y a qu'un petit sous-ensemble de ceux-ci qui peuvent être des éco-routes dans des conditions réalistes.

L'algorithme de routage est décrit au Section 6.1. Il prend en entrée une collection d'éco-routes candidates entre la même origine et la même destination regroupées dans une structure de données arborescente. Le routage est une recherche exhaustive du chemin qui minimise une fonction de coût donnée et satisfait toutes les contraintes. Il n'est généralement pas possible de calculer par ordinateur tous les chemins simples entre une origine et une destination données. Par conséquent, l'arbre ne peut contenir que leur sous-ensemble. Il peut s'agir d'une poignée de chemins ou de centaines de milliers de chemins, c'est encore une petite quantité par rapport au nombre de chemins trouvés entre la plupart des origines et des destinations.

Le calcul de l'arbre peut être une opération de calcul intensif. Il peut, cependant, être préparé une fois et utilisé plus tard plusieurs fois. Le calcul de la solution de routage est alors relativement facile. Ceci motive un système de routage où le calcul de l'arbre des chemins est découplé du routage: l'arbre des chemins peut être calculé à l'avance pour les déplacements répétés, puis envoyé à l'assistant de navigation et utilisé chaque fois que nécessaire. Deux approches du calcul de l'arbre des chemins sont étudiées: la recherche exhaustive (Section 6.2), ou la recherche aléatoire (Section 6.3). Les deux génèrent des arborescences de chemin avec un sous-ensemble de chemins entre l'origine et la destination donnée. La méthode randomisée identifie directement les éco-routes prospectives, la recherche exhaustive considère tous les chemins pour lesquels les critères d'élagage n'ont pas réussi à montrer qu'il ne peut s'agir d'un éco-route.

Notez que l'arbre des chemins peut aussi être construit d'une autre manière. Ils peuvent être considérés comme des options précalculées que l'algorithme de routage peut choisir. Une façon possible de construire un tel arbre est de prendre des éco-routes basées sur divers modèles de consommation d'énergie proposés dans la littérature. L'algorithme de routage choisit alors l'éco-route qui minimise la consommation compte tenu des conditions actuelles du réseau routier.

## 1.9 Chapitre "Simulation framework for eco-routing"

L'évaluation d'une méthode de l'éco-routage nécessite des essais de terrain prohibitifs. Ce chapitre propose un cadre de simulation pour les remplacer. Les simulations ne fournissent pas de preuves tangibles par définition, mais elles sont indicatives et relativement peu coûteuses à réaliser par rapport aux essais réels sur le terrain. Il a été utilisé pour évaluer la consommation d'énergie en minimisant les méthodes de l'éco-routage. Les méthodes de réduction des émissions de polluants peuvent être évaluées de manière analogique, si nécessaire.

La réalisation d'expériences de l'éco-routage simulé nécessite un simulateur de trafic et un scénario de trafic. Le scénario devrait être vaste, de sorte que les véhicules soient sujets à des perturbations généralement observées dans le trafic réel comme l'interaction avec d'autres véhicules, la gestion des différents niveaux de congestion, l'interaction avec les feux de circulation, les priorités aux intersections, les panneaux d'arrêt , etc. Le simulateur utilisé est SUMO 0.28.0 (Krajzewicz et al., 2012), le scénario de trafic utilisé est LuST 2.1. (Trafic SUMO au Luxembourg, Codeca et al., 2015). Ils sont traités dans la Section 7.1. Les deux sont disponibles sous une licence open-source au moment de la rédaction. Le simulateur SUMO simule des véhicules individuels dans le scénario, mettant à jour leur position et leur vitesse à des intervalles d'une seconde. Le scénario LuST est basé sur le réseau routier et les trafics au Luxembourg (ville européenne de taille moyenne). Il simule plus de 200000 véhicules dans un réseau routier qui contient des routes locales, des artères et une autoroute.

La production des modèles de consommation étudiés est comparée à la consommation d'énergie estimée par deux modèles de consommation de référence. Ils sont présentés dans la Section 7.2. Deux modèles sont considérés: l'un pour un véhicule électrique, l'autre pour un véhicule conventionnel. Le véhicule électrique est représenté par le FAM F-City, le véhicule conventionnel de Renault Scénic. Les deux sont des modèles d'estimation de la consommation standard qui intègrent la puissance instantanée estimée tirée de la source d'énergie du véhicule. Ils utilisent des cartes numériques d'efficacité du groupe motopropulseur dérivées expérimentalement pour estimer la puissance consommée. Les deux modèles ont été développés et validés par un tiers.

Le graphe du réseau routier et les algorithmes de routage ont été implémentés en interne, ils sont traités dans la Section 7.3. Ces fonctions de bas niveau ont été implémentées en C. Le reste du système a été implémenté en Python. Le graphe du réseau routier a été compilé à partir du fichier de définition du réseau routier SUMO et d'un modèle d'élévation de la zone. Le fichier de définition du réseau routier fait partie de LuST 2.1; les données d'altitude ont été importées de EU-DEM (*EU Digital Elevation Model*).

Les observations historiques sur les routes individuelles sont requises par les modèles de consommation utilisés dans ce travail. Ils sont généralement utilisés pour identifier leurs coefficients. Le scénario de trafic contient plus de 200000 véhicules simulés qui constituent son trafic natif. Leurs profils de vitesse sont enregistrés et traités afin de générer les données nécessaires aux modèles de consommation. Ceci est discuté dans la Section 7.4.

Les simulations réalistes sont effectuées sur des trajets individuels: un véhicule sonde est introduit dans la simulation et sa progression est enregistrée jusqu'à ce qu'elle atteigne la destination. Le processus qui est effectué est décrit dans la Section 7.5. Les informations collectées sont utilisées pour calculer la consommation de référence (avec un modèle de consommation de référence).

## 1.10 Chapitre "Model identification & validation"

Ce chapitre traite de l'identification et de la validation des modèles de consommation de base, du modèle de temps de trajet et du modèle de consommation proposé à la Section 5.4. L'identification revient à l'identification des coefficients des modèles étudiés. Les données requises pour les identifier proviennent de l'ensemble de données de trafic natif abordé à la Section 7.4. La validation équivaut à l'estimation de la précision de prédiction des modèles étudiés. Les mesures de précision et de qualité d'ajustement utilisées sont RMSE (l'erreur quadratique moyenne), IQR (l'intervalle interquartile) et $r^2$ (coefficient de détermination). La validation croisée a été réalisée avec une double validation croisée répétée cinq fois.

Les modèles de base sont présentés, identifiés et validés dans la Section 8.1. Les modèles de consommation publiés par Barth et al., 2007 et Juřík et al., 2014 sont considérés comme des références. De plus, un modèle de consommation d'énergie dont les estimations de consommation sont des valeurs moyennes d'échantillon est considéré comme un complément aux deux modèles de référence. Il est nommé "MEC" (Mean Energy Consumption) dans ce travail. Barth et al., 2007 et Juřík et al., 2014 ont été choisis comme représentants des deux principales approches de modélisation de la consommation d'énergie utilisées dans la littérature sur l'éco-routage. Barth et al., 2007 proposent un modèle de consommation basé sur l'analyse de régression et Juřík et al., 2014 proposent un modèle dérivé du modèle de consommation longitudinale (introduit au Chapitre 5).

Le modèle de temps de trajet proposé à la Section 5.5 est identifié et évalué à la Section 8.2. Il y a quatre coefficients dans le modèle de temps de trajet (5.38) à

TABLE 1.1: Variantes du modèle de consommation (version française).

| | $T$ | $\sigma^2$ | $b$ | $v(0)$ | $v(T)$ |
|---|---|---|---|---|---|
| modéle 1 | ✓ | ✓ | ✓ | ✓ | ✓ |
| modéle 2 | $\star$ | ✓ | ✓ | ✓ | ✓ |
| modéle 3 | $\star$ | $\sim$ | $\sim$ | ✓ | ✓ |
| modéle 4 | $\star$ | – | – | – | – |
| modéle 5 | $\sim$ | $\sim$ | $\sim$ | $\sim$ | $\sim$ |
| modéle 6 | $\sim$ | – | – | – | – |

Clés: ✓valeur correcte, $\sim$ valeur moyenne, – zéro, $\star$ prédit avec (5.38)

identifier: $\mu$, $\vartheta$, $\kappa_{\mathbb{G}}$ et $\kappa_{\mathbb{S}}$. Les $\mu$ et $\vartheta$ sont des valeurs réelles, les coefficients $\kappa$ sont des entiers positifs. Un ensemble unique de coefficients a été identifié pour chaque arête du graphe du réseau routier. Les résultats montrent une réduction considérable de RMSE et d'IQR sur les routes avec des feux de circulation. Le RMSE a été réduit d'un facteur de 1,68 et l'IQR d'un facteur de 4,34 en moyenne. Les résultats pour les routes sans feux de circulation montrent que le modèle de temps de trajet proposé n'offre aucun avantage par rapport au modèle de temps de trajet moyen.

Le modèle de consommation proposé à la Section 5.4 est identifié et évalué à la Section 8.3. Six variantes ont été étudiées. Ils sont nommés "modèle 1" à "modèle 6". La différence entre eux se trouve dans les variables indépendantes fournies au modèle de consommation. Le modèle 1 utilise des valeurs observées. Le modèle 2 utilise le temps de trajet prévu (en utilisant le modèle (5.38)) au lieu du temps de trajet observé. Le modèle 3 utilise le temps de trajet prévu et les valeurs moyennes à la place des paramètres $\sigma^2$ et $b$. Le modèle 4 utilise le temps de trajet prévu tandis que toutes les autres variables indépendantes sont mises à zéro. Le modèle 5 utilise des valeurs moyennes à la place de toutes les variables indépendantes. Le modèle 6 utilise le temps de trajet moyen et suppose que toutes les autres variables sont nulles. Les modèles 1, 2, 3 et 4 dépendent du temps tandis que les modèles 5 et 6 sont indépendants du temps. Les modèles 1 à 3 utilisent des valeurs observées et ne peuvent donc être utilisés avant l'enregistrement du voyage. Cela les empêche d'être utilisés dans l'éco-routage. Les modèles 4 à 6 ne reposent pas sur des observations et peuvent être utilisés dans l'éco-routage. Les six variantes du modèle ont montré une précision supérieure sur les deux véhicules de référence par rapport à Barth et al., 2007 et Juřík et al., 2014. Surtout le modèle 6 utilise les mêmes informations que ces modèles de base et montre une précision considérablement améliorée. La précision est toutefois inférieure par rapport au modèle de consommation d'énergie moyenne (MEC) pour la plupart des variations du modèle de consommation. Seuls les modèles 1, 2 et 3 utilisés sur le véhicule électrique ont montré une performance supérieure par rapport à MEC. Dans le cas du véhicule conventionnel, la précision du modèle 1 est comparable à la précision de MEC, les modèles 2 à 6 ont montré des précisions moins bonnes.

## 1.11 Chapitre “Simulations and results”

Les résultats d'expériences de l'éco-routage simulées sont présentés dans ce chapitre. Trois méthodes d'éco-routage sont étudiées, toutes visent à minimiser la consommation d'énergie des véhicules. Deux variantes du modèle de consommation appelé "modèle 6" et "modèle 4" sont considérées. Le modèle de consommation a été proposé à la Section 5.4, les deux variantes sont présentées dans le Tableau 1.1. Le modèle 6 est indépendant du temps tandis que le modèle 4 dépend du temps. Aucune contrainte n'est imposée sur la solution de l'éco-routage. Les coûts multiples ne sont pas non plus pris en compte. Bien que la méthode de l'éco-routage proposée prenne en charge les deux, l'objectif est d'évaluer la performance de l'éco-routage de manière à ce qu'elle soit comparable aux méthodes de base examinées à la Section 4.3.

L'éco-routage avec le modèle de consommation 6 est étudié dans la Section 9.1. Il s'agit d'un éco-routage indépendant du temps avec un modèle de consommation qui utilise les mêmes informations que les modèles de référence de Barth et al., 2007 et Juřík et al., 2014. L'éco-routage avec le modèle 6 est plus performant que Barth et al., 2007 et Juřík et al., 2014, mais ne réussit pas mieux que le modèle MEC et, peut-être plus important encore, le routage simple à plus courte distance.

L'algorithme 3 est utilisé pour construire l'arbre des chemins. Son temps de calcul et le nombre de chemins identifiés sont étudiés pour diverses conditions d'arrêt dans la Section 9.2. La condition d'arrêt considérée dans cette expérience est une limite du nombre d'itérations de la boucle principale dans l'algorithme 3. Des limites allant de 100 itérations à 2000 itérations sont considérées. Les temps de calcul observés étaient inférieurs à neuf secondes dans tous les cas. La médiane du temps de calcul commence à 0,5 seconde à 100 itérations et passe à 3,8 secondes à 2000 itérations. Le nombre de chemins trouvés augmente à des pas variables en fonction des positions de l'origine et de la destination. Le pire cas observé contient 1850 chemins après 2000 itérations, ce qui signifie que dans 92,5% des itérations, un nouveau chemin a été découvert et ajouté à l'arbre. D'un autre côté, les distributions de la Figure 9.4 montrent que c'était rare. La médiane se développe à un rythme beaucoup plus lent avec 50 chemins trouvés après 100 itérations et 300 chemins trouvés après 2000 itérations. Il est néanmoins soutenu que cela n'implique pas qu'il n'y ait généralement que quelques éco-routes candidates à considérer pour une paire de destinations d'origine. La distance parcourue est un facteur de confusion, comme le montre la Figure 9.6. Il est également soutenu dans cette section qu'un moyen d'évaluer les éco-routes candidates dans les arbres de chemins est de voir s'ils contiennent les chemins les plus courts. La Figure 9.7 montre la partie des arbres qui contenait le plus court chemin par rapport au nombre d'itérations. Il montre que le chemin le plus court a été le premier chemin découvert dans 20% des cas et que dans près de 40% des cas, le chemin le plus court a été identifié dans les dix premières itérations. Environ 10% des arborescences de chemin ne contiennent pas le chemin le plus court, même après 10000 itérations.

La Section 9.3 étudie deux méthodes d'éco-routage avec le modèle de consommation 4. Il s'agit de méthodes d'éco-routage dépendant du temps et basées sur des arbres de parcours. La première variante ne prend pas en charge le réacheminement tandis que la seconde variante le fait. Ces méthodes d'éco-routage ont été évaluées sur le même ensemble de paires de destinations d'origine que les modèles de référence et la méthode d'éco-routage basée sur le modèle 6 (Section 9.1) pour assurer la comparabilité. Les résultats sont résumés dans le Tableau 1.2. En termes de performances, l'éco-routage dépendant du temps avec le modèle 4 montre une amélioration par rapport au modèle 6 avec un routage indépendant du temps. De même, l'éco-routage avec réacheminement activé est meilleur que l'éco-routage avec réacheminement désactivé. Les performances vont de 23,0% à 32,4% en comparaison avec les chemins les plus rapides. Cependant, les chiffres de performance sont considérablement plus bas lors de la comparaison avec les chemins les plus courts.

TABLE 1.2: Résultats (version française).

| par rapport au route | | plus rapide | | plus court | |
|---|---|---|---|---|---|
| | | VE[†] (%) | VC[‡] (%) | VE[†] (%) | VC[‡] (%) |
| $P$ | modéle 6 | 31.4 | 21.9 | -0.9 | -0.9 |
| | modéle 4 | 32.1 | 23.0 | 0.3 | 0.7 |
| | modéle 4 adaptatif | 32.4 | 23.5 | 0.8 | 1.4 |
| | Barth et al., 2007 | 26.7 | 20.1 | -8.0 | -3.2 |
| | Juřík et al., 2014 | 29.7 | 18.8 | -3.5 | -4.9 |
| | modéle MEC | 34.5 | 24.7 | 3.6 | 2.8 |
| $\hat{P}$ | modéle 6 | 24.7 | 15.7 | 4.1 | 6.4 |
| | modéle 4 | 24.7 | 18.0 | 5.5 | 8.3 |
| | modéle 4 adaptatif | - | - | - | - |
| | Barth et al., 2007 | 7.2 | 7.3 | 2.0 | 2.3 |
| | Juřík et al., 2014 | 50.9 | 53.4 | 2.2 | 2.7 |
| | modéle MEC | 30.3 | 18.7 | 4.9 | 3.9 |
| Probabilité d'échec | modéle 6 | 9.6 | 6.3 | 8.2 | 9.0 |
| | modéle 4 | 9.5 | 10.2 | 7.7 | 9.4 |
| | modéle 4 adaptatif | 7.5 | 7.3 | 4.6 | 5.6 |
| | Barth et al., 2007 | 8.7 | 8.6 | 5.2 | 6.1 |
| | Juřík et al., 2014 | 24.5 | 30.4 | 30.0 | 36.9 |
| | modéle MEC | 6.4 | 4.3 | 7.6 | 5.9 |
| Eco-route identique à la référence | modéle 6 | 22.8 | 31.9 | 29.6 | 22.5 |
| | modéle 4 | 16.6 | 16.4 | 30.8 | 23.6 |
| | modéle 4 adaptatif | 14.5 | 14.5 | 25.4 | 19.5 |
| | Barth et al., 2007 | 28.6 | 29.8 | 40.6 | 35.8 |
| | Juřík et al., 2014 | 9.4 | 8.7 | 37.6 | 32.2 |
| | modéle MEC | 15.5 | 22.1 | 25.0 | 29.8 |
| Économies moyennes | modéle 6 | 15.7 ± 0.19 | 11.7 ± 0.15 | -1.5 ± 0.07 | -1.2 ± 0.06 |
| | modéle 4 | 16.4 ± 0.19 | 12.6 ± 0.15 | -0.4 ± 0.07 | 0.2 ± 0.05 |
| | modéle 4 adaptatif | 16.8 ± 0.19 | 13.2 ± 0.15 | 0.1 ± 0.07 | 0.9 ± 0.05 |
| | Barth et al., 2007 | 13.9 ± 0.18 | 11.1 ± 0.15 | -5.6 ± 0.18 | -2.2 ± 0.10 |
| | Juřík et al., 2014 | 13.7 ± 0.21 | 8.2 ± 0.17 | -3.6 ± 0.07 | -4.7 ± 0.07 |
| | modéle MEC | 19.3 ± 0.19 | 14.6 ± 0.15 | 2.9 ± 0.06 | 2.4 ± 0.05 |
| Délai moyen de trajet | modéle 6 | 10.9 ± 0.16 | 8.1 ± 0.14 | -5.2 ± 0.11 | -7.4 ± 0.10 |
| | modéle 4 | 11.3 ± 0.17 | 9.9 ± 0.15 | -4.9 ± 0.10 | -6.0 ± 0.10 |
| | modéle 4 adaptatif | 9.3 ± 0.16 | 7.8 ± 0.15 | -6.6 ± 0.10 | -7.8 ± 0.10 |
| | Barth et al., 2007 | 9.7 ± 0.14 | 8.9 ± 0.13 | -5.9 ± 0.10 | -6.4 ± 0.10 |
| | Juřík et al., 2014 | 28.0 ± 0.25 | 29.4 ± 0.26 | 8.4 ± 0.16 | 9.6 ± 0.17 |
| | modéle MEC | 14.2 ± 0.18 | 11.4 ± 0.16 | -2.5 ± 0.12 | -4.9 ± 0.10 |

[†] véhicule électrique

[‡] véhicule conventionnel

Dans ce cas, les performances varient entre 0,3% et 1,4%. Le temps de parcours moyen varie de 7,8% à 11,3% par rapport aux trajets les plus rapides et de -7,8% à -4,9% par rapport aux trajets les plus courts. Cela suggère que la méthode d'éco-routage permet de réduire le temps de trajet par rapport aux trajets les plus courts. La probabilité d'échec est inférieure à 9% dans tous les cas; le réacheminement a réduit la probabilité d'échec. Les éco-routes étaient identiques aux trajets les plus rapides dans 14,5% à 30,8% des cas.

Enfin, la Section 9.4 donne un exemple de la manière dont la polyvalence de la méthode d'éco-routage proposée peut être exploitée pour améliorer les résultats de manière novatrice. Une méthode d'éco-routage orientée vers des chemins plus courts est proposée. L'évaluation montre des performances améliorées en ce qui concerne les chemins les plus courts, les plus rapides et une probabilité réduite de défaillance.

## 1.12 Conclusions

Les conclusions sont résumées en quatre sections: (1) principales conclusions sur la littérature actuelle; (2) les progrès de la modélisation de la consommation; (3) la méthode d'éco-routage proposée; et (4) sélection de la méthode de map-matching. La première section traite de la performance des méthodes actuelles et des limites d'une méthodologie d'évaluation d'éco-routage couramment utilisée. La deuxième section traite du modèle de consommation proposé dans ce travail. La troisième section traite de la méthode d'éco-routage proposée dans ce travail. La quatrième section fournit un guide de sélection pour les méthodes de map-matching utilisées dans le contexte des systèmes de navigation éco-routage.

### 1.12.1 Principales conclusions sur la littérature actuelle

Deux méthodes d'éco-routage publiées dans la littérature ont été étudiées en détail pour fournir des bases de comparaison avec la méthode proposée. Ils ont été proposés dans Barth et al., 2007 et Juřík et al., 2014. Ce sont des méthodes de minimisation de la consommation d'énergie indépendantes du temps. Les deux utilisent une approche différente de la modélisation de la consommation d'énergie. Le modèle proposé dans Barth et al., 2007 utilise l'analyse de régression, le modèle proposé dans Juřík et al., 2014 utilise un modèle de consommation dérivé du modèle de consommation longitudinale microscopique.

Les performances de ces méthodes dépendent de l'environnement dans lequel elles ont été testées et de l'ensemble des chemins auxquels les éco-routes sont comparées. Des économies importantes ont été observées lorsque les éco-routes sont comparées aux trajets les plus rapides. Les chiffres de performance sont toutefois beaucoup plus bas lorsque la performance est basée sur une comparaison des éco-routes vers les chemins les plus courts. Dans ce cas, Barth et al., 2007 et Juřík et al., 2014 n'ont pas réalisé d'économies. Au lieu de cela, leurs éco-routes offrent des pertes sur un voyage moyen. Ce résultat implique qu'un routage de chemin le plus court simple est supérieur aux deux méthodes dans le scénario testé. En outre, un nombre élevé de cas dans lesquels les méthodes d'éco-routage ont échoué à économiser à la fois de l'énergie et du temps de déplacement ont été observés. En particulier, la méthode de Juřík et al., 2014 a échoué dans 24,5% à 36,9% des cas. Il est difficile d'imaginer que les vrais conducteurs accepteraient un tel taux d'échecs. La conclusion est qu'il serait peut-être préférable d'utiliser un routage simple du plus court chemin plutôt que les méthodes évaluées pour l'éco-routage. Ceci motive la méthode d'éco-routage adaptatif dépendant du temps et contrainte proposée dans ce travail. Une telle méthode est, en théorie, capable de naviguer dans le véhicule à travers des séquences de feux verts sur des intersections signalisées, afin d'éviter la congestion, et de mettre à jour la solution lorsqu'elle devient invalide.

Alors que l'évaluation des méthodes actuelles d'éco-routage a montré des économies négatives, il est rare de voir un tel résultat signalé dans la littérature publiée. Un examen attentif des méthodologies d'évaluation utilisées dans la littérature a révélé que la méthode prédominante n'est pas capable d'observer des économies négatives de par leur conception. Le problème est que le même modèle de consommation (ou d'émission de polluants) est utilisé à la fois pour le routage et l'évaluation. Non seulement cette évaluation se traduira nécessairement par des économies non-négatives, mais il est également démontré dans le Théorème 1 que les résultats de l'éco-routage sont gonflés. La conclusion est que les performances et les économies rapportées dans la littérature doivent être considérées avec prudence. Ces résultats soulignent la nécessité d'un cadre d'évaluation écologique fiable. Un tel cadre est présenté dans ce travail.

Notez que ces résultats ne sont pas généralisables: les deux méthodes sont susceptibles de montrer des performances positives dans d'autres environnements. Il est probable que le scénario intra-urbain utilisé soit difficile pour l'éco-routage, car il offre de nombreux chemins similaires vers la destination avec de nombreuses sources de perturbation en cours de route.

### 1.12.2 Les avancées dans la modélisation de la consommation d'énergie

Un nouveau modèle de consommation d'énergie est proposé au Chapitre 5. Il s'agit d'un modèle macroscopique dérivé du modèle de consommation d'énergie longitudinale standard. Le modèle longitudinal est un modèle microscopique qui nécessite un profil de vitesse du véhicule enregistré pour estimer la consommation. Il est soutenu dans ce travail qu'un tel modèle ne convient pas à l'éco-routage parce que les profils de vitesse ne sont pas disponibles lors de la planification du voyage. Dans ce travail, le modèle standard est d'abord reformulé sous forme fermée en fonction de la consommation d'énergie d'un véhicule idéal et de l'énergie restituée au véhicule lors du freinage. Ensuite, une solution de forme fermée à la consommation d'énergie du véhicule idéal est dérivée et utilisée pour résoudre la consommation d'énergie d'un véhicule plus réaliste sous l'hypothèse que la puissance de sortie du groupe motopropulseur est une fonction affine de sa puissance d'entrée. Le modèle qui en résulte nécessite cinq paramètres inconnus: la vitesse initiale et la vitesse finale pour tenir compte de la variation de l'énergie cinétique du véhicule, du temps de trajet, de la variance du profil de vitesse et du coefficient d'asymétrie du profil de vitesse.

Six variantes de ce modèle de consommation ont été étudiées. Ils diffèrent par les cinq paramètres qui lui sont fournis: d'un modèle auquel tous les paramètres ont été fournis tels qu'ils ont été observés à un modèle où seul le temps moyen de déplacement a été fourni, et d'autres paramètres ont été mis à zéro. Tel est, précisément, le modèle 6. Il est directement comparable aux modèles de Barth et al., 2007 et Juřík et al., 2014. Leur seul paramètre est la vitesse moyenne qui est inversement proportionnelle à la longueur de la route au temps de parcours. Par conséquent, ils utilisent les mêmes informations que le modèle 6. Malgré cela, le modèle 6 surpasse largement les deux. Dans le cas du véhicule électrique, le modèle 6 a montré un RMSE de 28,6 Wh, tandis que Barth et al., 2007 et Juřík et al., 2014 ont montré 60,6 Wh et 47,7 Wh, respectivement. Le modèle (idéaliste) MEC a montré un RMSE de 24,83 Wh. A titre de comparaison, le RMSE du modèle 6, le modèle de Juřík et al., 2014, et le modèle de Barth et al., 2007 sont respectivement de 115%, 192% et 244% de la RMSE du modèle MEC. La précision du modèle 6 est beaucoup plus proche de la précision du modèle MEC que les autres modèles. Contrairement au modèle MEC, cependant, le modèle 6 peut être utilisé dans la pratique.

Le modèle proposé est intéressant en raison de la façon formelle dont il a été dérivé du modèle longitudinal. Les méthodes d'éco-routage qui utilisent des modèles de consommation dérivés du modèle longitudinal (comme De Nunzio et al., 2016; Juřík et al., 2014) supposent que le profil de vitesse du véhicule est une fonction constante égale à la vitesse moyenne du véhicule. Le modèle présenté ici n'a pas besoin d'une telle hypothèse drastique. Il est soigneusement développé à partir du modèle longitudinal. La contribution clé est la solution de forme fermée pour le véhicule idéal car elle ne repose sur aucune hypothèse. La solution pour des véhicules plus réalistes est basée sur elle. Ce modèle ouvre une nouvelle direction de recherche car il fournit une solution au modèle longitudinal sous forme fermée sans hypothèses irréalistes comme cela a été fait dans la littérature jusqu'à présent. Il expose les propriétés des profils de vitesse les plus importants. La seule hypothèse qu'il fait concerne l'efficacité du groupe motopropulseur. Il est difficile à éviter en raison d'une efficacité instantanée non linéaire des groupes motopropulseurs couramment utilisés.

La performance de l'éco-routage indépendant du temps avec le modèle 6 est néanmoins négative par rapport aux chemins les plus courts. C'est un résultat similaire à ce qui a été observé pour les méthodes par Barth et al., 2007 et Juřík et al., 2014. Malgré l'amélioration de la précision du modèle 6, il est toujours préférable d'utiliser le plus court chemin. L'éco-routage avec le modèle MEC est le seul éco-routeur indépendant du temps qui a permis de réaliser des économies sur les chemins les plus courts dans les expériences menées dans ce travail. Il a également surpassé l'éco-routage dépendant du temps proposé dans ce travail. Bien que le MEC soit un modèle théorique, il montre qu'il existe un potentiel d'éco-routage indépendant du temps. La question qui reste est de savoir à quel point il est réaliste d'exploiter ce potentiel.

### 1.12.3 La méthode d'éco-routage proposée

L'éco-routage a été traité dans la littérature principalement comme un problème de chemin le plus court. Cette approche offre une formulation intuitive de l'éco-routage, et de plus, des méthodes efficaces pour son calcul sont disponibles (comme les algorithmes de Dijkstra ou de Bellman-Ford). Leur principale force est qu'ils sont rapides et optimaux. Cependant, ils sont également limitants. L'éco-routage contraint, l'éco-routage dépendant du temps ou l'éco-routage adaptatif sont supportés par eux. Des algorithmes sont proposés pour chacun de ces trois types d'éco-routage, mais à la connaissance de l'auteur, il n'existe pas de méthode d'éco-routage publiée qui permette la combinaison: éco-routage dépendant du temps et adaptatif. Ce travail propose et évalue un tel système d'éco-routage.

La méthode proposée dans ce travail privilégie le routage simpliste sur des algorithmes de routage optimisés sophistiqués tels que l'algorithme de Dijkstra ou de Bellman-Ford. La motivation pour cela est que ces algorithmes sont trop restrictifs, trop rigides dans leurs hypothèses. On fait valoir ici que l'utilisation d'algorithmes de routage optimaux donne un petit avantage quand il y a une incertitude importante dans les coûts associés dans le graphe de routage. Une solution approximative basée sur une approche naïve du routage est étudiée à la place. Il est basé sur une hypothèse selon laquelle, bien qu'il y ait habituellement beaucoup de chemins possibles à prendre pour une destination, il n'y a qu'un petit sous-ensemble de ceux-ci qui peuvent être des itinéraires écologiques dans certaines conditions réalistes.

La méthode d'éco-routage proposée comprend deux étapes: le prétraitement et le routage. Le prétraitement équivaut au calcul d'une arborescence de chemin, qui est une collection d'éco-routes candidates entre une origine et une destination données. Le routage est ensuite effectué sur les chemins dans l'arborescence du chemin à un moment ultérieur. Il est irréaliste de calculer un arbre avec tous les chemins entre une origine et une destination données pour tous les cas sauf les cas triviaux. Néanmoins, tous les chemins ne peuvent pas être des éco-routes. Deux algorithmes de calcul d'arborescences ont été proposés: les algorithmes 2 et 3. L'algorithme 2 conduit une recherche exhaustive et élague uniquement les chemins pour lesquels il peut prouver qu'ils ne peuvent jamais être des éco-routes. L'algorithme 3 conduit une recherche aléatoire répétée pour les éco-routes. Le routage est effectué par l'algorithme 1. Il prend l'arbre des chemins en entrée et recherche le chemin qui minimise la fonction de coût donnée. Cette fonction peut combiner plusieurs coûts tels que le temps de trajet, la consommation d'énergie, la distance parcourue et les émissions de polluants. Il prend également en charge l'éco-routage entièrement dépendant du temps et les contraintes multiples. L'algorithme de routage est optimal en ce sens qu'il trouve le chemin qui minimise la fonction de coût donnée sur l'arbre donné. L'arbre, cependant, contient typiquement seulement un sous-ensemble de chemins entre l'origine et la destination données. Par conséquent, la solution de l'éco-routage n'est pas garantie optimale.

L'algorithme 3 a été utilisé pour calculer les arbres de chemins utilisés dans les expériences présentées car il permet de contrôler facilement la taille de l'arbre et le temps de calcul. Les temps de calcul de l'arborescence des chemins étaient inférieurs à 9 secondes dans le pire cas observé avec 2000 itérations de la boucle principale dans l'algorithme 3. Ces résultats sont acceptables car les temps de calcul de l'arbre ne sont pas critiques. Le nombre de chemins détectés était généralement faible avec une médiane de 300 chemins après 2000 itérations. Dans le pire cas observé, il y avait 1850 chemins découverts après 2000 itérations. Dans 20% des cas observés, le chemin le plus court, le premier chemin découvert et dans 40% des cas, a été découvert dans les dix premières itérations. Ceci est révélateur de la qualité des éco-routes identifiées puisque de nombreux auteurs ont indiqué que les chemins les plus courts sont des candidats viables pour être des éco-routes. En revanche, dans 10% des cas, le chemin le plus court n'a pas été découvert même après 10000 itérations. Par conséquent, le calcul de l'arbre des trajets est réalisable par calcul pour les déplacements intra-urbains similaires à ceux simulés dans le cadre présenté. La durée de ces voyages est comprise entre 1km et 12km. La question de savoir si le calcul de l'arbre des trajets est faisable lors de longs trajets reste ouverte. On a observé que le nombre de sentiers découverts augmente rapidement sur les trajets de plus de 4 km. D'un autre côté, les résultats n'ont pas montré de réduction des performances d'éco-routage sur les longs trajets.

Le routage a été effectué avec le modèle de consommation 4 et un modèle de temps de trajet (5.38) qui prend en compte les informations sur les périodes de feux de circulation connues. Le modèle 4 est une variante similaire au modèle 6 (discuté dans la section précédente), sauf que le temps de trajet utilisé est prévu avec le modèle de temps de trajet au lieu d'utiliser une moyenne simple des temps de trajet précédemment observés. Le modèle de temps de trajet et le modèle de consommation dépendent du temps. Puisque l'algorithme 1 est utilisé pour le routage, la méthode d'éco-routage dépend également du temps. Deux variantes ont été étudiées: avec réacheminement désactivé et activé. Dans le cas où il est activé, la solution d'éco-routage est adaptative. Le réacheminement est implémenté de sorte que la solution d'éco-routage soit mise à jour à chaque étape de simulation en utilisant l'algorithme 1. Elle est toujours exécutée sur un sous-arbre enraciné sur le nœud qui correspond à la position actuelle du véhicule dans l'arbre. La consommation d'énergie du véhicule était le seul coût considéré, et le routage n'était pas limité. Cette configuration a permis une comparaison directe des résultats des méthodes de base.

Les résultats de la simulation ont montré des performances élevées en comparant les éco-routes aux trajets les plus rapides et des performances modérées en comparant les éco-routes aux trajets les plus courts. La meilleure performance observée était de 32,4% par rapport aux chemins les plus rapides et de 1,4% par rapport aux chemins les plus courts. Les deux ont été observés avec un éco-routage adaptatif dépendant du temps avec le modèle 4. Le réacheminement a amélioré les performances de façon marginale, mais cohérente. Les répartitions des économies d'énergie révèlent que, pour de nombreux déplacements, la méthode a pu trouver des chemins offrant des économies par rapport au chemin le plus court, mais elle n'a pas réussi à le faire dans de nombreux autres cas. Si la méthode d'éco-routage devait choisir le chemin le plus court dans les cas où elle échouait, alors la performance par rapport aux chemins les plus courts serait de l'ordre de 2,26% à 2,64%, selon le type de véhicule. Cela montre qu'il y a un potentiel d'amélioration, bien que limité.

Les résultats montrent que la méthode de l'éco-routage proposée peut concurrencer l'éco-routage traditionnel tel quel, mais qu'elle est également plus flexible en permettant un éco-routage contraint, dépendant du temps et adaptatif. Cela permet des applications plus riches, telles que l'éco-routage avec contrainte de temps, ou l'éco-routage qui évite les écoles au moment où les enfants sont là, par exemple. Les applications qui seraient difficiles à réaliser avec l'approche standard deviennent possibles avec la méthode proposée. Une méthode d'éco-routage biaisée artificiellement pour préférer des chemins plus courts est un exemple de la polyvalence de la méthode proposée. La motivation pour cela est basée sur une observation que les chemins plus courts sont plus susceptibles d'être des éco-routes. Il est montré pour réduire la probabilité de défaillance et pour améliorer les performances en ce qui concerne les chemins les plus courts et les plus rapides. Il a réussi à économiser soit de l'énergie soit du temps de trajet par rapport aux trajets les plus courts dans 98,8% des trajets étudiés dans le cas du véhicule électrique et dans 97,6% des trajets étudiés dans le cas du véhicule conventionnel.

Le chemin le plus court s'est avéré être un éco-routeur étonnamment difficile à surperformer. Cependant, les chemins les plus courts sont lents comparés aux chemins les plus rapides. La méthode d'éco-routage proposée dans ce travail n'a pas montré une capacité à surpasser significativement le routeur simple du plus court chemin en terme d'énergie, mais a montré une capacité à économiser légèrement plus d'énergie que le routeur le plus court tout en économisant un temps considérable aux chemins les plus courts.

Le système de simulation mis au point pour mener ces expériences est conçu pour être aussi réaliste que possible. Il utilise un simulateur de trafic microscopique d'une ville entière et des modèles de consommation microscopiques basés sur des véhicules réels en tant que références. Cependant, ce sont encore des simulations qui signifient que les résultats sont indicatifs. La validation finale serait obtenue avec de nombreux tests sur le terrain, ce qui peut être prohibitif.

### 1.12.4 Sélection de la méthode de map-matching

Différentes méthodes de map-matching conviennent à différentes applications de map-matching. Il n'y a pas de méthode universelle qui conviendrait aux besoins de tous. Les compromis qui doivent être faits lors de la sélection d'une méthode de map-matching sont discutés dans cette section. L'intérêt dans le contexte de l'éco-routage est double: pour l'appariement des voyages préenregistrés et pour la navigation. Cette section est un extrait de Kubička et al., 2017.

La correspondance des trajets préenregistrés nécessite une méthode de correspondance de carte hors ligne. Un effort de calcul plus élevé peut être toléré car les trajectoires sont post-traitées après leur collecte. La méthode la plus avancée en ce qui concerne la précision d'appariement serait le filtre d'inférence de chemin (Hunter et al., 2014). Cependant, sa demande de calcul pourrait être trop prohibitive. La méthode de Newson and Krumm, 2009, offre une bonne précision d'appariement alors que sa demande de calcul est comparativement faible. Une autre option est la méthode géométrique de Wei et al., 2013a, spécialement lorsqu'elle est utilisée conjointement avec la méthode rapide d'approximation de distance Fréchet développée par Driemel et al., 2012. Si un ensemble massif de trajectoires doit être traité et que la précision d'appariement n'est pas critique, alors la méthode de Marchal et al., 2005 peut être considérée. Si l'application utilise des trajectoires faiblement échantillonnées, alors les méthodes de faible taux d'échantillonnage (Lou et al., 2009; Raymond et al., 2012; Zheng and Quddus, 2011; Chen et al., 2011b) peuvent être intéressantes. Ces méthodes sont susceptibles d'être dépassées par le filtre d'inférence de chemin, mais elles sont souvent plus faciles à implémenter.

Les applications de navigation requièrent des méthodes d'appariement de cartes en ligne à taux d'échantillonnage élevé. L'effort de calcul doit rester faible car le système doit répondre en temps réel. Lorsque la surveillance de l'intégrité est nécessaire, la méthode de Toledo-Moreo et al., 2010 doit être considérée. Cette méthode a montré une grande précision d'appariement au niveau de la voie tout en assurant une surveillance continue de l'intégrité de la sortie de correspondance de carte. Les méthodes basées sur la logique floue, telles que la méthode de Quddus et al., 2006, auraient une excellente précision d'appariement, mais nécessitent des connaissances spécialisées pour leur réglage. Les méthodes basées sur une technique d'hypothèses multiples (Pyo et al., 2001; Kubička et al., 2014) pourraient être en mesure d'offrir un compromis intéressant entre la demande informatique et la précision d'appariement. Les méthodes basées sur le modèle de Markov caché et les méthodes géométriques ne sont pas bien adaptées car elles nécessitent des ressources de calcul considérables. La technique de la fenêtre glissante peut être utilisée pour remédier à ce problème: seuls quelques derniers échantillons sont utilisés pour mapper le point de correspondance actuel. Lorsque la demande sur l'effort de calcul n'est pas stricte, la méthode de Hummel, 2006 devrait également être considérée car elle a des perspectives d'être robuste contre les erreurs de positionnement.

# Chapter 2

# Introduction

Eco-routing is a vehicle navigation method that chooses those paths that minimize fuel consumption, energy consumption or pollutant emissions for a trip to given destination. Road transportation has numerous adverse effects on the environment. It is the largest contributor to global warming through emissions of $CO_2$ (Fuglestvedt et al., 2008) and is responsible for the deterioration of air quality in areas with dense road networks through emissions of particulate matter and other pollutants. For example, the particulate matter is a dangerous pollutant: the study by Raaschou-Nielsen et al., 2013 revealed that for every increase of 10 $\mu$g/m$^3$ of particulate matter in the air the lung cancer rate in the area rose by 22%. Apart of this environmental dimension, there is also an economical side to the problem. The monetary cost associated with energy (fuel) consumption is considerable. This motivates techniques such as eco-routing that aim to lower pollutant emissions, or energy (fuel) consumption, or both.

Initial studies on the topic confirmed the dependency of pollutant emissions and energy (fuel) consumption on the taken path. Ahn and Rakha, 2007 shown in a case study that taking a slower path can save some fuel at the cost of prolonged travel time. Ericsson et al., 2006 have shown that drivers don't always choose paths with best fuel economy. The first eco-routing navigation system was published by Barth et al., 2007. Numerous case studies and eco-routing methods were published since.

Different authors consider different variants of eco-routing. As was stated above, some aim to lower pollutant emissions while others aim to lower consumption. In the former case the term "eco-routing" stands for "ecologic routing" while in the latter case it might be both "ecologic" and "economic" (some authors imply that $CO_2$ emissions are reduced together with consumption). Eco-routing methods are usually designed with a specific vehicle type in mind. Some are designed for conventional vehicles with internal combustion and compression-ignition engines, while other consider electric and hybrid vehicles. Conventional vehicles emit pollutants and consume fuel. Electric vehicles consume energy instead of fuel and do not emit any pollutants. Hybrid vehicles have both electric motor and an engine. There is a wide variety of existing topologies, and their behavior is not easily characterized.

There are also differences in the way eco-routing methods are evaluated. Some authors define savings by comparing the quantity of interest (fuel consumption, energy consumption, or pollutant emissions) to shortest paths, while others compare to fastest paths. The shortest paths are those paths that minimize the distance to destination. The fastest paths are those paths that minimize expected travel time.

Little is known on what benefits can eco-routing offer. Eco-routing performance has been shown to depend on the properties of the road network and of the vehicle. The published case studies are often designed as proofs of the eco-routing concept, and so they concentrate on scenarios where eco-routing is likely to excel.

This, however, does not say anything about the typical savings attainable in everyday use. Some case studies and most publications that propose new eco-routing methods nevertheless contain an evaluation of average savings. However, the same consumption or pollutant emission model is often used to identify the eco-route and to estimate the savings on it. This type of evaluation can never result in negative savings, meaning that it is unable to detect failures.

There is a closely-related technique to eco-routing called eco-driving. It tries to achieve the same goals as eco-routing but uses different means to do it. Instead of lowering consumption or pollutant emissions by path choice, eco-driving optimizes driver's behavior while the path is predetermined and fixed. Since both techniques share the same goal, it should be possible to combine them for better results. They can be used independently, but it is likely that there exists some degree of co-dependence between them.

## 2.1 Structure of the eco-routing navigation system

The basic structure of an eco-routing navigation system is shown diagrammatically in Figure 2.1. There are four main elements: local database, cost estimation block, routing block and map-matching block. The user interface block is out of the scope of this work and is not further discussed here.

- *Local database* holds the road network graph. It is a directed graph whose purpose is to describe connectivity between different places in the map. It also contains spatial organization of the road network (road shapes) and other metadata such as road names, lengths, and slopes.
- *Cost estimator* is used to assign costs to edges of the road network graph. The nature of these costs depends on the way the eco-routing problem is posed. It can be fuel consumption, energy consumption, or pollutant emissions.
- *Routing* finds the path in the road network graph between given origin and destination with the lowest sum of the costs on the edges in the path.
- *Map-matching* converts Earth-referenced position to a position in a road network graph. Both the positioning system and the road network graph contain errors that can make the conversion difficult. There is no standard approach to this problem.

The structure in Figure 2.1 is identical to the commonly used shortest-distance navigation system. The only difference between eco-routing and shortest-distance routing is in the nature of the costs. In shortest-distance routing, the cost is simply a road length. This is a nonnegative constant that can be saved in the database. In eco-routing, the cost is a multivariate function.

Most published eco-routing methods use Dijkstra's routing algorithm (Dijkstra, 1959) or an algorithm derived from it (such as A* search algorithm; Hart et al., 1968). These are sophisticated algorithms that can identify the optimal solution in asymptotically optimal time. They, however, assume that the costs are nonnegative constant scalars. This is limiting for eco-routing applications. The quantities represented by the costs are usually time-dependent variables that can be negative if the cost represents energy consumption and the vehicle under consideration can recuperate braking energy. The problem with variable costs is further discussed in Section 2.2. The problem with negative costs can be solved with Bellman-Ford algorithm (Bellman, 1958), but the computational effort on a large road network can

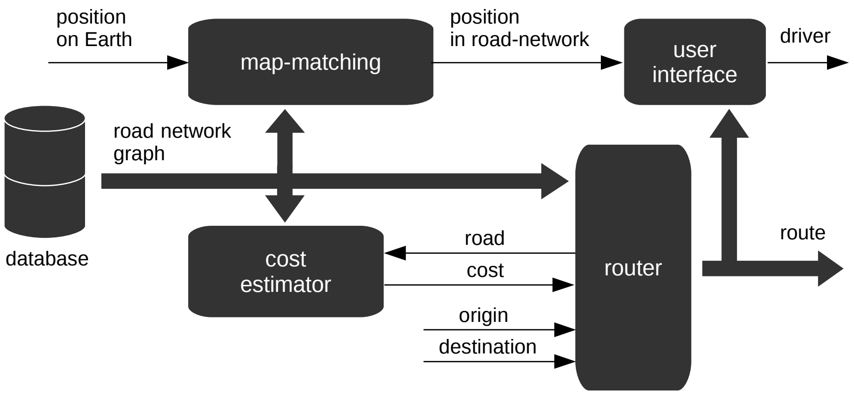


FIGURE 2.1: Basic eco-routing navigation system.

become significant. Another option is to make all costs positive without changing the structure of the shortest paths in the graph using Johnson's algorithm (Johnson, 1977).

Note that naive routing is computationally intractable for any but the simplest road networks. A naive routing method would first make a list all paths between given origin and destination and then search for the path that minimizes given cost function. The number of paths that has to be considered is not finite unless only simple paths are considered (paths without loops). Even then, the number of paths grows exponentially with the number of intersections in the road network. This is why routing algorithms such as Dijkstra's algorithm are important. They are capable of identifying the optimal solution in a polynomial time.

Numerous cost estimators were proposed specifically for eco-routing applications. Some are regression models, and others are derived from Newton's second law of motion. The common denominator between them is that they are *macroscopic*. In general, most published fuel consumption, energy consumption and pollutant emission models can be classified as either macroscopic or microscopic. Microscopic models are based on differential equations. Macroscopic models are based on closed-form algebraic equations. This often requires crude simplifications, but the resulting model is fast to compute and requires little information. The reason why the cost estimators used in eco-routing are macroscopic is that microscopic models need recorded trip data (vehicle's speed profile[1]) in order to estimate the cost. Such information is not available at the trip planning stage.

Note that macroscopic cost estimators are known for their estimation errors. For example, in one of the first studies on eco-routing, Ahn and Rakha, 2008 conclude that "macroscopic emission estimation tools (e.g., MOBILE6) can produce erroneous conclusions given that they ignore transient vehicle behavior along a route". It is perhaps surprising then that not a single eco-routing method cited in this work takes that into account: the costs are always assumed to be exact, without errors.

Map-matching has two applications in eco-routing. Firstly, it enables the user interface to navigate the driver to the destination by identifying current vehicle position in the road network. Secondly, map-matching is needed when processing historically recorded trips in order to infer knowledge about the consumption of the vehicle on the roads in the road network (this is something that is often done in literature). At the time of writing, the dominant vehicle positioning systems are based on satellite navigation. They identify vehicle position in terms of latitude and longitude. The map-matching is needed to assign the recorded trips to the right roads in

[1] The speed profile, denoted $v(t)$, is vehicle speed as a function of time.

the road network. Matching errors get propagated through the system since map-matching is a low-level task. They can reduce cost estimator's accuracy and can make the user interface to start giving misleading advice to the driver.

## 2.2 Eco-routing extensions

Three extensions of the basic eco-routing are discussed in the literature.

- *Constrained eco-routing* allows setting limits on fuel consumption, energy consumption, travel time, and pollutant emissions. It was first considered in Juřík et al., 2014. The authors propose an optimal routing algorithm to solve a constrained optimization problem with additive costs. This problem is known to be NP-Complete, see Wang and Crowcroft, 1996 for the proof.

- *Time-dependent eco-routing* is eco-routing with costs that are functions of time instead of constants. It was first considered in Kluge, 2011. Such routing allows taking into account developing traffic situation when routing. It is likely, for example, that there are different eco-routes at night and in the afternoon as the traffic densities are different. Perhaps more importantly, the eco-route can be affected by stops on the roads with traffic lights. Time-dependent eco-routing can, in theory, navigate the vehicle such that the losses due to them are minimized. There are two types of time-dependent eco-routing recognized in the literature:

  - *Snapshot routing* assumes the costs are constants. They are estimated at the departure time for every road in the road network. Such eco-routing system has the same structure as time-independent eco-routing, except that the cost estimator is time-dependent.
  - *Time-dependent routing* supports routing with costs that are functions of time. The costs on the roads are then specific to the expected time of arrival there. Time-dependent travel time model is needed to estimate it. The routing algorithm needs to be able to consider the time-varying nature of the costs.

  The time-dependent routing can make the eco-routing problem considerably more difficult to solve. The structure of such eco-routing navigation system is depicted diagrammatically in Figure 2.2. Both the cost and travel time estimators are necessarily time-dependent models. The router identifies the path that minimizes the sum of costs at a given departure time, considering predicted cost on its roads at the predicted time of arrival there.

- *Adaptive eco-routing* allows updating the eco-routing solution when it becomes invalid. It might become apparent while already on the way that the identified eco-route is no longer valid. It is desirable to detect this situation and to update the eco-routing solution accordingly. It was first considered in Ahn and Rakha, 2013.

  Adaptive eco-routing is interesting in connection with time-dependent eco-routing. If the vehicle does not proceed to destination with the predicted speed, then the eco-routing solution is not necessarily valid. For example, if the vehicle gets delayed the eco-routing solution is not necessarily the correct one anymore. This can be verified by rerouting from vehicle's current position to the destination.

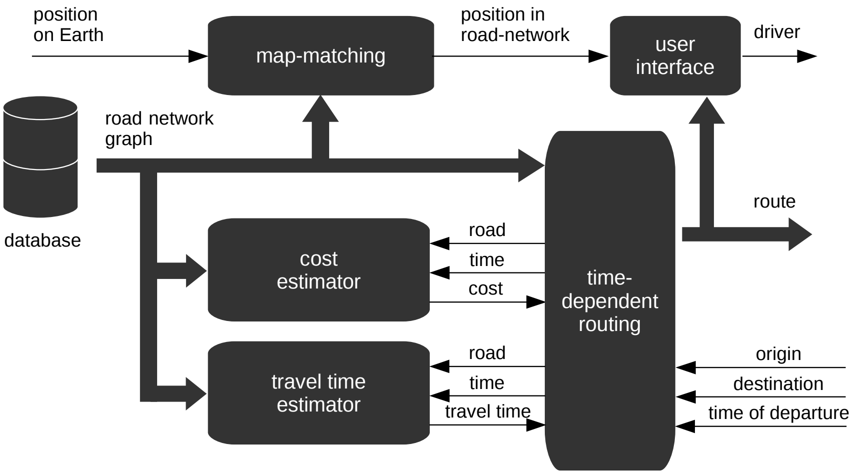


FIGURE 2.2: Time-dependent eco-routing navigation system.

## 2.3 Approach followed in this work

The goal of this work is to propose constrained time-dependent adaptive eco-routing navigation system. The constraints would allow the user to gain control over their emissions and travel time delays. The time-dependent routing would allow routing such that predicted situation on the roads at the time the vehicle is projected to be there is taken into account. Adaptive eco-routing would allow changing the path in case the original eco-routing solution is no longer valid.

It is a difficult problem to solve optimally. Optimal constrained eco-routing is NP-Complete. Optimal time-dependent routing searches for the solution in much larger space with respect to time-independent routing because of the time-variable costs. On the other hand, adaptive eco-routing needs fast routing algorithms that can be re-run multiple times while the vehicle is on the way to its destination. The time delay in which the eco-route is recomputed needs to be minimal to minimize the chance that the vehicle in the mean time progressed so far on the old eco-route that it is no longer possible to change for the new eco-route. To summarize the difficulty, there is a conflict: both constrained and time-dependent routing are computationally intensive tasks, while adaptive eco-routing does not work well with computationally intensive routers.

Sustaining optimality is not only computationally expensive, but also futile. It is argued in this work that optimality of the routing solution is conditioned on the correctness of the routing costs in the road network graph, something that is known to be subject to considerable errors (as discussed above). This argument is supported by findings in this work as all studied eco-routing methods failed to save energy, in some instances. This is something that could not happen with optimal eco-routing. Consequently, this work does not study optimal routing algorithms and concentrates instead on finding approximative solutions. Dropping the routing optimality condition gives more flexibility in designing eco-routing methods. It is not necessary anymore to make the assumptions imposed by sophisticated optimal routing algorithms such as Dijkstra's algorithm, for example. The costs can be real functions of time if that's what eco-routing applications require. It is a question of identifying the right approximations that lead to results that are computable in an acceptable time.

The proposed eco-routing method is based on a hypothesis that while there might be many paths to a chosen destination, only a small fraction of them can ever be eco-routes. It favors simplistic, naive even, routing based on a list of all possible paths over known and sophisticated optimal routing algorithms. It follows an approach that does not resemble anything from the published literature. It is based on original ideas, that may, or may not, prove useful.

## 2.4 Research topics

There are three main topics studied in this work:

1. *Vehicle consumption model for eco-routing.* Having a good cost estimation model is of prime importance in eco-routing: its performance is conditioned on the cost estimation errors. A standard consumption estimation model is adapted for eco-routing applications in Chapter 5. It is carefully reformulated in a closed form and then solved, under some assumptions.

2. *Routing algorithm.* A routing based on a preselected collection of candidate eco-routes is proposed in Chapter 6. The candidate eco-routes are identified offline for a given origin and destination, before vehicle's departure. It allows to preselect those paths that are likely to be eco-routes in some conditions. There can be few dozen or a few hundred thousand candidates. It is still a negligible amount, in most cases, compared to the number of all simple paths between the origin and the destination. This reduces the search space and allows to support constrained time-dependent adaptive eco-routing.

3. *Reliable eco-routing performance evaluation.* As was stated in the introductory paragraphs, little is known on what benefits can eco-routing offer in everyday use. This work tries to make a step towards resolution of this problem. Eco-routing performance is formally defined in Chapter 4. A framework designed to evaluate eco-routing methods in terms of their performance is presented in Chapter 7. It allows comparing several published methods with the method proposed in this work.

## 2.5 Thesis outline

The eco-routing problem is introduced in this chapter. The Chapter 1 contains an extended summary of this work in the French language. The Chapter 3 contains a review of published eco-routing and map-matching methods. The Chapter 4 contains an analysis of the current methods that were introduced in Chapter 3. The chapter 5 introduces a novel vehicle consumption model and a travel time model. The Chapter 6 proposes a novel routing algorithm. The Chapter 7 proposes a framework for the evaluation of eco-routing methods in simulations. The chapter 8 discusses identification and validation results of the models used in this work. The Chapter 9 contains an evaluation of eco-routing methods based on the routing algorithm proposed in the Chapter 6 and consumption model proposed in the Chapter 5. The evaluation is done with the framework proposed in the Chapter 7.

# Chapter 3

# The state of the art

Published eco-routing and map-matching methods are reviewed in this chapter. A selection of eco-routing case studies is discussed in Section 3.1.1 and a selection of published methods is discussed in 3.1.2. Then, map-matching methods are reviewed in Section 3.2. Map-matching is an important subproblem in eco-routing. This subject is not further studied in this thesis. An extensive review of the literature on map-matching is provided in this chapter instead. It is an adapted extract from Kubička et al., 2017.

## 3.1 Eco-routing

There are diverse publications on eco-routing. Some authors report results of their case studies, some authors propose a method and evaluate it in a simulation, and other authors describe fully mature eco-routing systems deployed in real-life applications. The publications that propose new methods tend to have a similar structure: the authors first describe their consumption or pollutant emissions model and then their routing algorithm. The model is used to assign estimates of consumption or pollutant emissions to every road in the road network. The assigned values are then used as routing costs by the routing algorithm: it searches for the path that minimizes the sum of the costs between given origin and destination. The reviews below follow a similar structure: first, the aim of the method is introduced, then the consumption model followed by the routing method, and at the end, results are discussed (if the method was evaluated).

Different authors evaluate their results in different ways. Many evaluate them in terms of savings, where the consumption (or pollutant emissions) on the eco-route is compared to the consumption or pollutant emissions on some reference path. The most common reference paths are the fastest paths and the shortest paths. The kind of reference path used in the evaluation by the authors is always indicated in the reviews below since the two are not comparable.

### 3.1.1 Case studies

Ahn and Rakha conducted three case studies on eco-routing (Ahn and Rakha, 2007; Ahn and Rakha, 2008; Ahn and Rakha, 2013). The first one is an experiment designed to “evaluate energy and environmental impacts of driver route choice decisions”. The authors recorded thirty-nine trips on two different paths between the same origin and destination during a morning commute. Eighteen of them took the arterial path and twenty-one the highway. These two paths have different speed limits and different commute patterns. The authors estimate energy consumption, travel time and HC, CO, $NO_x$, $CO_2$ emissions based on the recorded data. The study demonstrated that, for the case in hand, the motorists could have saved on average

17% of travel time on the highway. However, 23% of energy consumption could be saved on the arterial path. Additionally, 63%, 71%, 45% and 20% of HC, CO, $NO_x$, $CO_2$ respectively could be also saved on this path. The authors also studied different energy consumption and pollutant emission models. They compare two microscopic models (CMEM and VT-Micro) to a macroscopic model (MOBILE6). They conclude that the macroscopic models are not adequate to quantify the energy and environmental impact of path choice decisions as they fail to capture the impact of transient changes in vehicle's speed and acceleration levels. In the follow-up study (Ahn and Rakha, 2008) the authors extend the previous work with sensitivity analyses to various traffic assignment scenarios and different vehicle types. The authors conclude that the commonly used user-equilibrium and system-optimum traffic assignments do not necessarily minimize vehicle energy consumption or pollutant emissions. They also conclude that vehicle powertrain characteristics need to be carefully examined before implementing emission- or energy-optimized traffic assignments. In the latest study (Ahn and Rakha, 2013) the authors conduct an extensive numerical experiment with eco-routing in a microscopic simulation of the road network of Cleveland and Columbus in Ohio, US. The study aims to quantify the impact of an eco-routing system on the road transportation system. The authors consider various levels of eco-routing market penetration and study the system-wide savings, the impact of congestion and the impact of vehicle type. They observed average fuel savings ranging from 3.3% to 9.3% when compared to the fastest paths, depending on eco-routing market penetration, road network configuration and vehicle type. The authors also observed that eco-routing typically reduces the travel distance, but not necessarily travel time. System-wide benefits of eco-routing increased with increasing eco-routing market penetration. Higher savings in terms of both energy consumption and pollutant emissions were observed in more congested situations.

Ericsson et al., 2006 published the first case study on eco-routing. The authors tried to estimate the potential of eco-routing in Lund, Sweden. Their goal was to explore possible means to lower fuel consumption and $CO_2$ emissions through driver support tools. They used a collection of 15,437 commutes over the period of three years to classify the streets in Lund and to build a macroscopic consumption model based on this classification. Every road in the road network was classified as one of 22 classes according to road type, function, speed limit, signalization, used traffic-calming measures and observed traffic flows in peak and non-peak hours. Then, the authors estimated fuel consumption factors and emission factors for each class. The fuel consumption factor is normalized fuel consumption in liters per 10 km. The emission factors are in grams per kilometer. The fuel consumption and $CO_2$ emissions were estimated for each road in the road network as its length multiplied by the fuel consumption factor and emission factors respectively. The consumption model was used on a subset of 109 commutes longer than 5 minutes to estimate how many of them could be optimized and what is the expected gain for those that could be. The authors found that fuel-efficiency could be enhanced for 46% of the trips and that fuel savings would be 8.2% on average. This corresponds to average 4% savings overall. The authors have also noticed that the eco-routes were the shortest paths in 82% of the cases, which agrees with the findings of Ahn and Rakha, 2008. Note that these results depend on the accuracy of the proposed consumption model, which was not validated by the authors.

Minett et al., 2011 recorded forty trips between the cities Delft and Zoetermeer in Netherlands on three different paths and evaluated the differences in consumption and travel times on these paths. The three paths were chosen such that they have different characteristics: one uses a highway, the other provincial roads, and the last one only local roads. The experiment was conducted with three Ford Focus vehicles equipped with GPS loggers. The consumption was estimated using microscopic VT-CPFEM model (Rakha et al., 2011). On the way from Delft to Zoetermeer, the authors report 45% average fuel savings observed on local roads when compared to the highway path. Similar savings were observed on the provincial roads when compared to the highway. In both cases, the fuel savings came at the cost of 25% travel time increase. Similar results were observed for the way back (from Zoetermeer to Delft). The authors also propose a method to generate synthetic speed profiles[1]) that can be used to estimate consumption with a microscopic consumption model. The synthetic speed profile is constant on individual roads with smoothed transitions between them. The authors evaluated the accuracy of the fuel consumption on this speed profile and conclude that it underestimates fuel consumption because it neglects speed variations.

### 3.1.2 Methods

Andersen et al., 2013 proposed an eco-routing method that uses an extensive collection of pre-recorded commutes to compute the average consumption on each road in the road network. These averages are used as routing costs in place of a consumption model. The routing is done with Dijkstra's algorithm. Missing data pose a problem for this method: if there are roads with no pre-recorded commutes, then the method is unable to devise a cost for them. The authors did not validate their method, only demonstrated what savings they expect to obtain between some chosen origin and destination. Note that these savings were estimated using the same consumption model that was used to identify the eco-route.

Barth et al., 2007 proposed an "Environmentally friendly navigation". It is the first eco-routing method described in the literature. The authors use CMEM microscopic consumption and pollutant emissions model on a set of pre-recorded commutes to estimate pollutant emissions and fuel consumption on individual roads in the network. This is used to build a regression model for pollutant emissions and fuel consumption where average vehicle speed and road grade are explanatory variables. The road grade parameters are assumed known for every road in the road network. The average vehicle speed parameter can be based on historical observations or real-time data. The routing was done with the Dijkstra's algorithm. The authors conducted four case studies between the same origin and destination in Los Angeles, California region. They have chosen two paths on freeways between the same origin and destination and compared their fuel consumption and $CO_2$, CO, HC, $NO_x$ emissions with the proposed models. Four different situations were studied: (1) when there is no congestion on both paths, (2) when one path is moderately congested, while the other one is not, (3) when one path is heavily congested while the other one is not and (4) when one path is moderately congested while the other one is heavily congested. The authors show that considerable savings in energy and emissions can be obtained if the right path is taken in the right situation and conclude that congestion can cause the minimal energy path to change.

Boriboonsomsin et al., 2012 (co-authored by Barth) followed on the work by Barth et al., 2007. Unlike the preceding work, the authors propose a functionally complete eco-routing navigation system that integrates historical and real-time traffic information to identify the paths that minimize pollutant emissions or fuel consumption. The method uses real-time traffic information from probe vehicles and

[1] "speed profile" is a term often used for vehicle speed as a function of time

local traffic information system. Historical data from traffic simulation models and travel demand models are used on roads for which real-time data are not available. The authors reused the methodology for fuel consumption and pollutant emission modeling proposed by Barth et al., 2007. They compile a database of vehicle, road and traffic characteristics and use it to estimate fuel consumption and pollutant emissions on pre-recorded commutes with the CMEM model. Then, they use multivariate nonlinear regression to fit six coefficients $\beta_0 \dots \beta_5$ in

$$\ln(f_k) = \beta_0 + \beta_1\bar{v} + \beta_2\bar{v}^2 + \beta_3\bar{v}^3 + \beta_4\bar{v}^4 + \beta_5\alpha \tag{3.1}$$

where $f_k$ is the fuel consumption or pollutant emissions in grams per mile on the road $k$, $\bar{v}$ is the average speed, and $\alpha$ is road grade. This enables fuel consumption or pollutant emission estimation on any road based on its length, road grade and the average speed there. The routing is based on Dijkstra's algorithm. The fuel consumption and pollutant emission model was validated in a field test. The authors report estimated versus actual fuel consumption on a chosen path. The results show fuel consumption estimation accuracy in the range between -12.7% and -25% depending on used traffic information sources. The authors also evaluate the eco-routes identified by this system. They simulate a large collection of paths from different origins and destinations to estimate savings of the identified eco-routes when compared to fastest paths. The results indicate 13% average savings at the cost of 21% average travel time increase. Note that the same fuel consumption model was used to identify these paths and to estimate the savings.

De Nunzio et al., 2016 proposed an eco-routing method for electric vehicles. The method uses Bellman-Ford algorithm instead of the Dijkstra's algorithm for routing. While Dijkstra's algorithm is asymptotically faster, it also assumes the routing costs are nonnegative. This is a problem for electric vehicles since they can recuperate energy, which can result in negative consumption (meaning negative cost). The authors propose a consumption model derived from the longitudinal model (Guzzella and Sciarretta, 2005). It assumes the vehicle speed is constant, like many other models used for eco-routing do, but it also considers the changes in speed between two roads (or two parts of a road) with different average speeds. The authors base their consumption model on vehicle equivalent resistance force as formulated by the longitudinal model (see Equation (5.4) in Section 5.1, where the longitudinal model is discussed in detail) and replace the instantaneous speed with the average speed $\bar{v}_k$ on a road $k$

$$\bar{F}_k = c_a\bar{v}_k^2 + c_b\bar{v}_k + c_c + mg\sin(\alpha) \tag{3.2}$$

where $c_a$, $c_b$, $c_c$, $m$ are vehicle-specific constants, $g$ is gravitational acceleration and $\alpha$ is road grade. The consumption on road $k$ is estimated as

$$f_k = \begin{cases} F_k D_k \eta_t^{-1} \eta_b^{-1} & \text{if } F_k > 0 \\ F_k D_k \eta_t \eta_b & \text{if } F_k \leqslant 0 \end{cases} \tag{3.3}$$

where $D_k$ is the length of road $k$ and $\eta_t$, $\eta_b$ are transmission and electric drive efficiencies respectively. Note that this is not the exact formulation used by the authors. The model also limits $F_k$ such that motor torque is kept within given boundaries, see the original publication for details. When the vehicle passes on a boundary between two roads (or road segments) with different average speeds the transition is modeled with constant acceleration from original speed to the new speed. The authors compute the energy associated with the change in speed using the same framework like the one described above, except that the speed is no longer considered constant.

The consumption on the road $k$ is a sum of consumption for traveling there and consumption associated with the change of speed from the previous road to the current one. Another novelty of this method is what authors call “adjoint graph”. The authors conjecture that the usually used model for the road network is a directed graph where nodes represent intersections and edges represent roads between them. They propose to use the adjoint graph, where nodes represent roads and edges represent the connections on intersections. This model allows assigning different costs to the same road depending on the heading of the vehicle on the downstream intersection. The authors have evaluated the validity of their consumption model with two field trials. HERE maps provided the real-time average speed information. In the first case, the proposed model overestimated actual consumption by 7%, and in the second case, it underestimated it by 9%. The eco-routes were evaluated in simulations. The authors have chosen a thousand origin-destination pairs at random and plotted the eco-routes, the shortest paths, and the fastest paths. The results show 6% average savings when compared to the shortest path and 10% average savings when compared to the fastest path. Note that the same energy consumption model was used to both identify these paths and to estimate the savings.

Juřík et al., 2014 proposed an eco-routing method for hybrid vehicles with a time-constraint. The authors view eco-routing as a trade-off between fuel (resp. energy) consumption and travel time. This is supported by previously published case studies (such as Ahn and Rakha, 2007) that conclude that eco-routing does not necessarily lower travel time. The authors propose a consumption model derived from the longitudinal model (Guzzella and Sciarretta, 2005). The model is discussed in Section 8.1. It is designed for vehicles that can recuperate energy. The recuperation efficiency is considered constant. This can result in negative routing costs. Since Dijkstra’s algorithm-based routing does not allow this, the authors render the costs positive by making recuperation efficiency negative when braking. The consumption model also neglects powertrain inefficiency. This is not necessarily a problem because scaling errors does not affect eco-routing performance: the structure of shortest paths in a graph will not be altered when all costs are rescaled by the same positive amount. The authors propose a novel routing algorithm that identifies the shortest path under constraint. They formulate eco-routing as a multi-constrained optimization problem and show that it is NP-Complete (the formal proof is given in Wang and Crowcroft, 1996). Then they give an algorithm not dissimilar to Dijkstra’s algorithm that solves it. Several search space reduction techniques are used to lower the computational overhead. They pre-compute lower bounds on travel time from any node in the road network to the destination. This allows early identification of paths that can never satisfy the travel time constraint. Another technique is path dominance testing, which identifies paths that can never be the eco-routing solution. Both techniques were previously described in Van Mieghem and Kuipers, 2004. The authors did not validate their method, only shown the estimated savings on a few examples. Note that the same energy consumption model was used to identify these eco-routes and to estimate the savings on them.

Kluge et al., 2013 considers time-dependent eco-routing for electric vehicles. The authors used about one year worth of recorded commutes from a fleet of about 100 taxis in Ingolstadt, Germany. They derive a combined distribution of speed and acceleration for every road in the road network at five-minute intervals through the day. The used consumption model is described in other works that were not accessible at the time of writing. It reportedly takes vehicle speed, acceleration, and road grade as inputs and returns an instantaneous power draw from the battery. The authors assign a cost function to every road in the road network that corresponds to the integral of the expected power draw. The routing is done with two algorithms: DOT* and TD-APX. They were both proposed in Kluge, 2011. The DOT* algorithm is not dissimilar to Dijkstra's algorithm. It is an extension of previously published work by Chabini, 1998 and Dean, 1999 with a heuristic search similar to the A* algorithm (also derived from Dijkstra's algorithm). The TD-APX is an approximative algorithm that finds a solution whose cost differs from the optimal solution by some chosen accuracy $\epsilon$. It provides the first estimates of the solution quickly and then refines it until sufficient accuracy is achieved. The method was evaluated in five numerical experiments in a road network graph of Ingolstadt, Germany. The first three experiments were used to evaluate the computation time of the DOT* and TD-APX algorithms. They consider the time of day with peak traffic. The first experiment is restricted to fifty random trips in Ingolstadt city center (1 $\mathrm{km}^2$ area) in a time frame between 7:50 a.m. and 8:00 a.m. The second experiment is based on 243 random trips in the whole road network graph between 7:00 a.m. and 8:00 a.m. The third experiment is based 121 random trips between 7:00 a.m. and 9:00 a.m. The last two experiments were designed to evaluate savings of the proposed eco-routing method. The fourth experiment is based on 300 random trips in the city center (this time authors consider an area of 16 $\mathrm{km}^2$ as city center) at different days of week (Tuesdays to Thursdays and Sundays) and in various times of day (1:00 a.m. to 2:00 a.m. and 7:30 a.m. to 8:30 a.m.). Finally, the fifth experiment has the same setup as the fourth experiment without the limitation to the city center. In the first experiment, the authors notice that the DOT* algorithm requires too much time to finish its computation for the method to be suitable for practical applications. The TD-APX algorithm shows acceptable computation times for various $\epsilon$. In the second and third experiments, the authors evaluate the computation time of the TD-APX algorithm for longer time frames. The longest computation time was 31 seconds, observed in the third experiment for $\epsilon = 0$ kWh. The fourth and fifth experiments have shown 10-14% savings on eco-routes when compared to fastest paths, and 13-24% savings when compared to shortest paths. The savings on eco-routes with respect to fastest paths are comparable to results reported by other authors. The savings on eco-routes with respect to shortest paths does not agree with observations made by other authors. Multiple other works report that eco-routes tend to minimize trip length as well (Ahn and Rakha, 2013; Ericsson et al., 2006). Note that the same energy consumption model was used to identify these eco-routes and to estimate the savings on them.

Nie and Li, 2013 described an eco-routing method that aims to find a path that minimizes the monetary value of the fuel combined with travel time while meeting a given $CO_2$ constraint. The authors propose a consumption and pollutant emission model that depends on vehicle speed (similarly like Barth et al., 2007 and Yao and Song, 2013), but also consider the cost associated with switching from one road to another. The speed changes are modeled with a constant acceleration model, like De Nunzio et al., 2016. The proposed routing model minimizes the monetary value of the trip (a linear combination of the fuel cost and of the travel time cost) under the $CO_2$ constraint. The authors do not propose to use any specific routing algorithm. They define their eco-routing model as an optimization problem instead. They conducted three numerical experiments, which were solved using CPLEX optimization software. They were designed to show that ignoring travel speed variations and vehicle type causes $CO_2$ estimation errors.

Yao and Song, 2013 described an eco-routing method conceptually not dissimilar to the method by Barth et al., 2007. The authors propose a pollutant emission and consumption model based on vehicle-specific power (VSP; Jimenez-Palacios, 1998). They consider an empirical formula with fuel consumption and pollutant emission factors in grams per kilometer as a function of average speed (hence the similarity to Barth et al., 2007). The coefficients were obtained using multiple linear regression. Unique coefficients were identified for all 288 five-minute intervals in 24 hours from midnight to midnight. The routing was implemented with a time-dependent variant of the Dijkstra's algorithm. The authors conducted a proof-of-concept case study with a field trial between a single origin and destination. They observed 5.3% fuel savings on a fuel-minimizing eco-route and 5.9% $CO_2$ emission reduction on a $CO_2$-minimizing eco-route when compared to the fastest path. They also simulated eco-routing between 200,000 randomly generated origin-destination pairs in Beijing area. They computed the eco-routes for all and estimated that their method cuts carbon dioxide emissions by 4.7% and lower consumption by 4% in heavy traffic when compared to the fastest paths. Note that the same fuel consumption (resp. pollutant emission) model was used to identify these paths and to estimate the savings on them.

## 3.2 Map-matching

Roughly speaking, map-matching aims to identify vehicle position or path on a map (Quddus et al., 2007; White et al., 2000; Bernstein and Kornhauser, 1996). Figure 3.1 shows its role in the context of navigation systems and other services. Map-matching has been under active research since the dawn of the global navigation satellite systems in 1990's. A few contributions dating before that (for example by Honey et al., 1989) exist, but the interest in the context of vehicular navigation appeared after GPS became widely available. Original interest was in connection with navigation assistants, but a large area of applications emerged after the mobile internet became available. One can cite, among others, location-based services, floating car data, "pay-as-you-go" services, automatic emergency beacons and fleet surveying.

The interest in the context of eco-routing is twofold: for matching pre-recorded trips and for navigation. In the former case, map-matching is used during preprocessing. Many eco-routing methods require historically recorded trips to identify their coefficients. Map-matching is used to convert the recorded sequence of positions in terms of latitude and longitude to a path in the road network graph. Assigning recorded data to correct roads is important. Assigning the wrong record to the wrong road would introduce errors in the consumption and pollutant emission models. In the latter case, map-matching is used during navigation, when the current position of the vehicle in terms of latitude and longitude is matched to a position in the road network. Incorrect matching would result in giving incorrect instructions to the driver.

When the map and the positioning data are sufficiently accurate, the problem is trivial. Hence, it might seem that map-matching is not a difficult problem. Nowadays, we have satellite navigation with sub-meter precision and detailed maps on scales allowing to distinguish millimeters. However, neither of the preceding is guaranteed, and errors in both can be intricate. The satellite-based positioning systems are unreliable, especially in urbanized areas where satellite visibility is limited and reflected satellite signals are common. This leads to outliers in positioning data or missing parts of trajectories when the satellite signal is weak. This in itself can make correct matching difficult in some situations. However, the map can also be outdated (missing some roads), or the map model itself can introduce ambiguity

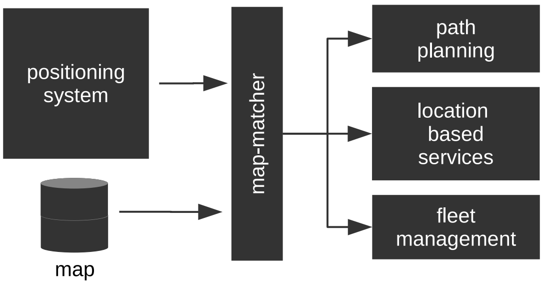


FIGURE 3.1: Map-matching converts raw position to position in the road network (originally printed in Kubička et al., 2014).

into the matching process. Commonly used maps are directed graphs where intersections are represented by their center points, and roads are represented by polygonal curves. Such a simplistic representation of otherwise complex road network can make correct matching difficult. Consequently, map-matching is not always trivial. Over a hundred of different methods have been published in the last two decades. They are based on various methods and approaches: from simple geometry and topology to advanced concepts that involve fuzzy logic, particle filtering, belief theory, conditional random fields, Kalman filtering, neural networks, genetic algorithms, ant colony optimization and others.

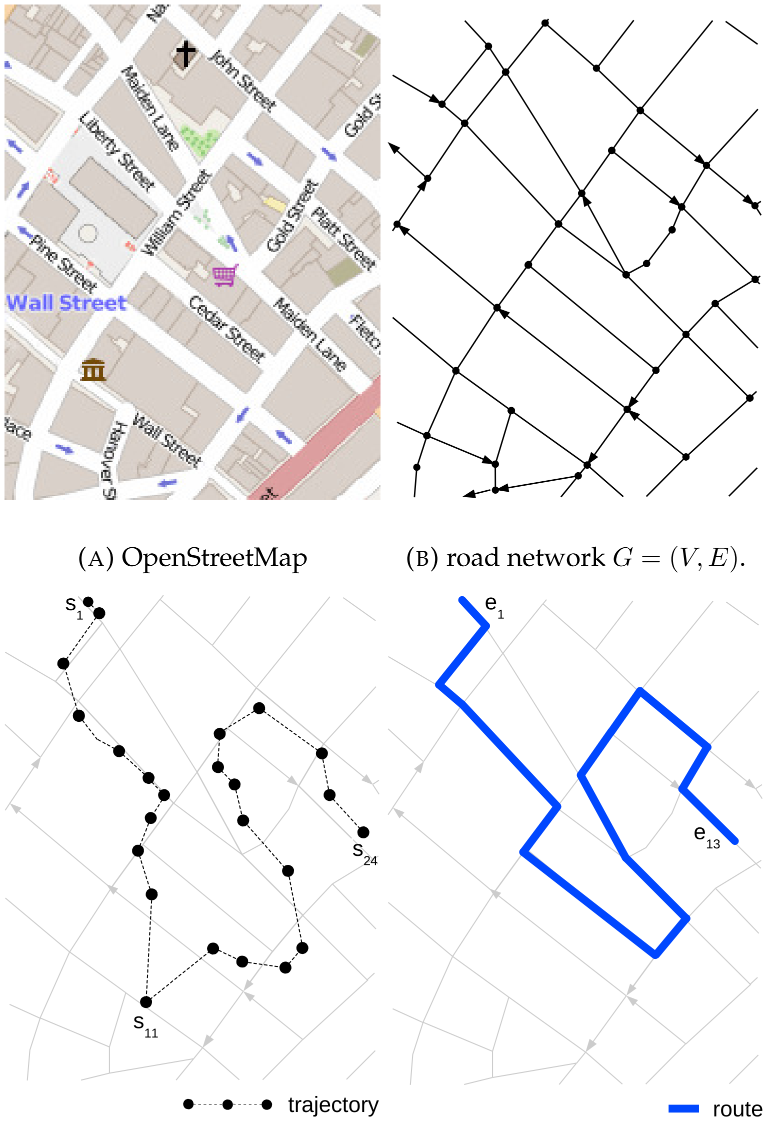
By inspecting the results concerning the map-matching problem, it is not immediately clear how the performance should be evaluated. This fact leads to different evaluation methodologies proposed by different authors. Unfortunately, these are often not mutually compatible. For example, some authors evaluate map-matching performance as a ratio of correctly matched positions with respect to the total number of samples. Other authors consider the degree of overlap between the matched path and the correct path. There is no common basis between the two methodologies, and the direct comparison seems not possible. This makes it difficult to argue on which method is best suited for any given use.

### 3.2.1 Published map-matching reviews

Quddus et al., 2007 published an overview of map-matching methods in 2007. The authors classify them as *geometric*, *probabilistic*, *topological* and *advanced*. More precisely, geometric methods make use of geometric closeness criteria for matching, topological methods leverage contiguity of edges in road network graphs and probabilistic methods make use of uncertainties in reported position to approximate error-bounding regions. Advanced methods are the methods that do not fit this classification scheme otherwise. For example, methods based on fuzzy logic or Kalman filtering. Most of the methods published in the last fifteen years could be viewed as advanced since recent map-matching algorithms often combine the ideas of geometric, topological and sometimes probabilistic methods.

The most recent review of map-matching methods was proposed by Hashemi and Karimi, 2014. The authors focused on map-matching for navigational applications. They reviewed a selection of publications and categorized them as *simple*, *weight-based* and *advanced*. The motivation for this categorization and used classification criteria is not clearly explained to the reader.

Wei et al., 2013a provide a short review of map-matching methods. The authors classify them as *incremental max-weight*, *global max-weight* and *global geometric*. This was adopted by other authors since (e.g., Li et al., 2013b). The max-weight methods



(A) OpenStreetMap (B) road network $G = (V, E)$.

(C) trajectory $s = (s_1, \dots, s_{24})$. (D) route $r = (e_1, \dots, e_{13})$.

FIGURE 3.2: Example road network, trajectory and map-matched route (originally printed in Kubička et al., 2017).

search for the route that maximizes some fitness function. The geometric methods search for a route geometrically similar to given trajectory. The incremental methods match individual samples while still collecting them. The global methods process complete trajectory after it was collected.

Quddus et al., 2007 list performance of 10 map-matching methods, Hashemi and Karimi, 2014 list performance of 19 methods. The performance figures in both reviews are based on results reported in the reviewed papers, which are not entirely comparable. Wei et al., 2013b compare 15 methods. The authors reimplemented them and tested using a unified methodology which assures comparability of the results.

### 3.2.2 Terminology used in this section

Consider a road network graph $G = (V, E)$ such as the one defined in Section 4.1 in context of eco-routing. Let ordered set $s = (s_1, s_2, s_3, \ldots, s_n)$ be a sequence of observed states $s_i$ at discrete time instants $i \in \{1 \ldots n\}$, where $s_i$ is $i$-th *sample* and $s$ is a *trajectory*. Some methods return the matched position for each sample. Let such output be denoted as *matching* $m$ and defined $m = (p_1, p_2, \ldots, p_n)$ where $p_i = (a_i, o_i)$ is $i$-th matched position on the road network. The $a_i$ is the road on which vehicle travels and the $o_i$ is the distance from the upstream intersection. Other methods do not match individual samples. Instead, they concentrate on travel route identification. Let such output be denoted *route* $r$ and defined as a sequence of contiguous edges $r = (e_1, e_2, \ldots)$ in $G$. Finally, the path inference filter discussed in Section 3.2.7 defines output as a sequence of waypoints interleaved with paths between them. Let us denote such output as $r'$ and define it $r' = (p_1, r_{12}, p_2, r_{23}, \ldots, p_n)$ where $p_i$'s are the matched positions and $r_{ij}$ is a sequence of contiguous edges in $E$ between $p_i$ and $p_j$.

For an example of a map, trajectory and route see Figure 3.2. Figure 3.2a shows OpenStreetMap render of a part of the Manhattan island in the New York, US (© OpenStreetMap contributors). Corresponding road network graph is given in Figure 3.2b. Note the arrows, they indicate traffic direction on one-way streets. Missing arrow indicates a two-way street. Example trajectory made of samples $(s_1, s_2, \ldots, s_{24})$ is shown in Figure 3.2c. A route matched to this trajectory is shown in Figure 3.2d. Note that the sample $s_{11}$ is an outlier: as can be observed in Figure 3.2d the route passes on Cedar Street rather on Wall Street, even the Wall Street is closer to sample $s_{11}$ (compare to OpenStreetMap render in Figure 3.2a). This example is studied later in the text when effects of such outliers on some map-matching methods are discussed.

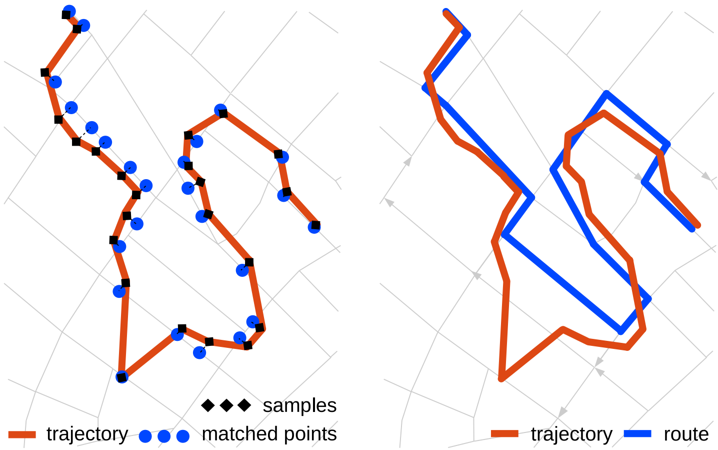
### 3.2.3 Classification

A classification based on four criteria according to which most published methods describe themselves is introduced in this section.

- *Indoor/outdoor methods* - while outdoor map-matching makes use of satellite navigation (Newson and Krumm, 2009; White et al., 2000; Alt et al., 2003), indoor map-matching has to use other positioning systems such as inertial navigation, radio beacons, etc. (Xiao et al., 2014).

- *Pedestrian/vehicular/multimodal map-matching* - pedestrian map-matching (Ren, 2012) is more challenging than the vehicular map-matching as pedestrians can be both indoors and outside. This is, strictly speaking, possible even with vehicular map-matching but authors often assume that vehicles are restricted to outdoor movement. The multimodal map-matching is used in the general case when the traveler can combine different travel modes (Chen and Bierlaire, 2015).

- *Online/offline methods* - online map-matching (White et al., 2000; Kubička et al., 2014) methods output matched positions while still collecting the trajectory. Some authors use the term “real-time” instead of “online” (Kubička et al., 2014; Hashemi and Karimi, 2014) to underline the fact that such map-matching returns results immediately with incoming samples. Other authors use the term



(A) point-to-curve

(B) curve-to-curve

FIGURE 3.3: Basic map-matching methods (originally printed in Kubička et al., 2017).

"incremental" (Wei et al., 2013a; Li et al., 2013b). Offline map-matching processes whole trajectory after it has been collected. Some authors use the term "global" instead of "offline" (Wei et al., 2013a; Li et al., 2013b).

- *Low/high sampling rate methods* - low sampling rate methods are designed for situations where positioning data are sampled on periods longer than about thirty seconds. This number is somewhat arbitrary as authors consider various thresholds. There are methods designed specifically for low-sampling-rate applications (Lou et al., 2009; Yang et al., 2005; Raymond et al., 2012; Zheng and Quddus, 2011; Chen et al., 2011b) and also methods that try to be competitive over the full range of sampling rates (Newson and Krumm, 2009; Hunter et al., 2014).

The map-matching methods that are used for preprocessing in eco-routing are outdoor, vehicular, offline and high sampling rate methods. The methods that are used for navigation are the same except online. In Kubička et al., 2017 the former is classified as *tracking application* and the latter as *navigational application*.

### 3.2.4 Early methods

Early methods are known as *point-to-point*, *point-to-curve*, or *curve-to-curve*. This naming convention was introduced by Bernstein and Kornhauser in "An Introduction to Map Matching for Personal Navigation Assistants" in Bernstein and Kornhauser, 1996. The point-to-point matching matches each sample $s_i$ to the nearest node in $V$. The point-to-curve matches $s_i$ to the nearest point on the nearest road in $E$. The curve-to-curve method matches the trajectory $s$ to the most similar route $r$ in the road network. An example of map-matching with these methods is in Figure 3.3. It is based on the road network and the trajectory from Figure 3.2. In Figure 3.3a the point-to-curve method matches trajectory samples (squares) to their respective closest points on the road network (circles). In Figure 3.3b the curve-to-curve method matches the geometric shape of a candidate path (blue curve) to the geometric shape of the trajectory (red curve). Historically, the point-to-curve methods lead the development of online map-matching while curve-to-curve methods lead the research on offline methods.

TABLE 3.1: Baseline matching accuracies

| method | performance[1] |
|---|---|
| point-to-curve | $53 - 67\%$ |
| point-to-curve, considers heading | $66 - 85\%$ |
| point-to-curve, enforces route contiguity | $66 - 85\%$ |
| curve-to-curve | $61 - 72\%$ |

[1] extracted from White et al. White et al., 2000

White et al., 2000 experimented with four basic map-matching methods. First is classical point-to-curve, it ignores past samples and makes no effort to make the resulting route contiguous. The second method is a modified version of the first one where vehicle heading is taken into consideration. The third method is based on the second but additionally considers road network topology. The fourth method is the curve-to-curve matching. The results are summarized in Table 3.1. It was tested on four routes; the worst-to-best performance range is reported for each. The percentage values represent the correctly matched samples with respect to the number of samples in the trajectory. The authors found that the enhanced point-to-curve methods perform better than curve-to-curve matching. They attributed it to high sensitivity to outliers of the latter.

### 3.2.5 Geometric methods

Given some trajectory $s$ the geometric methods search for the most resembling route in the road network using some shape similarity metric $\delta$. Consider $\mathcal{R}$ a set with geometric shapes of all paths in $G$. Then, geometric methods search for a path in $\mathcal{R}$ that maximizes similarity $\delta$ to the trajectory $s$. Hence, a generic model of geometric methods reads

$$r = \operatorname*{argmax}_{x \in \mathcal{R}} \delta(s, x) \tag{3.4}$$

A typical application is in offline settings. This makes geometric methods suitable for mapping and tracking. Online map-matching is possible but computationally demanding. Sparsely sampled trajectories are challenging for these methods since they are based on a hypothesis that the trajectory is geometrically similar to the correct route. The trajectory contains less detail when sampled sparsely. Nevertheless, there are successful applications of geometric methods on sparse trajectories. This is achieved by combining purely geometrical approach with heuristics that help to resolve ambiguities.

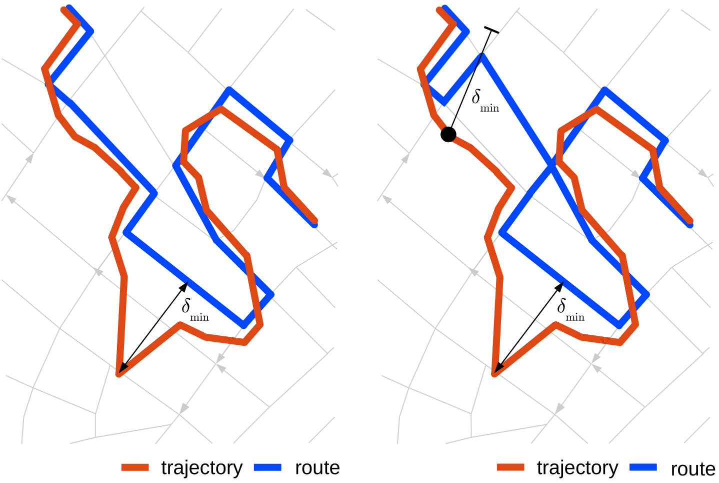
Two similarity metrics received attention in the literature: the *Hausdorff distance* and the *Fréchet distance*. Consider two polygonal curves A and B. The *one-sided Hausdorff distance* from curve A to curve B is defined as

$$\delta'_H(A, B) = \max_{a \in A} \min_{b \in B} d(a, b) \tag{3.5}$$

where $d(a, b)$ is the distance between $a$ and $b$. The metric is either the Euclidean distance or the great-circle distance. The Euclidean distance is suitable on short ranges. The great-circle distance is used when the curvature of Earth becomes a concern. The Haversine or Vincenty's formulæ can be used to compute it.

The Hausdorff distance $\delta_H$ is defined as the maximum of the two one-sided Hausdorff distances:

$$\delta_H = \max(\delta'_H(A, B), \delta'_H(B, A)) \tag{3.6}$$



(A) correct route (B) incorrect route

FIGURE 3.4: An example of sensitivity to outliers with Fréchet distance based geometric map-matching: both routes have the same Fréchet distance $\delta_{\text{min}}$ to the trajectory (originally printed in Kubička et al., 2017).

Its computation on polygonal curves is straightforward from the definition in $O(nm)$ time, with $n$ being the number of samples in the trajectory and $m$ the number of nodes in the route. The asymptotic run time $O((n+m)\log(n+m))$ can be achieved if Voronoi diagrams are used to answer the nearest neighbor queries faster (Alt and Guibas, 1999).

The Hausdorff distance ignores the course of the two curves. Consider two mutually overlapping curves in a reversed order (i.e. for curve $A = (a_1, a_2, \dots a_n)$ the $B = (a_n, a_{n-1}, \dots, a_1)$). The Hausdorff distance between them is zero by definition. Consequently, a map-matching method that uses the Hausdorff distance cannot distinguish between two routes that go on the same roads in the opposite direction. In general, any two curves that occupy the same area can have small Hausdorff distance even if the two shapes are wildly different. See Figure 3.5 for one such example. This is why the Hausdorff distance is seldom used in geometric map-matching.

The Fréchet distance is more common. It was introduced in Maurice Fréchet's thesis "Sur quelques points du calcul fonctionnel" (Fréchet, 1906). The definition reads

$$\delta_F(f,g) = \inf_{\alpha,\beta} \max_{t\in[0,1]} d(f(\alpha(t)), g(\beta(t))) \tag{3.7}$$

where $f$, $g$ are parametrizations of the two curves ($f, g : [0,1] \to \mathbb{R}^2$) and $\alpha$, $\beta$ are continuous, monotone, increasing reparametrizations ($\alpha, \beta : [0,1] \to [0,1]$). The reparametrization functions are introduced to enforce continuous and monotonically increasing parameters for $f$ and $g$. Famous illustration goes as follows: suppose a man is walking his dog, both the man and the dog walk along their own trajectory. The maximum length of the leash is then equal to the Fréchet distance between trajectory curves of the man and his dog.

Map-matching methods based on both Hausdorff and Fréchet distances are sensitive to outliers. This is due to the maximization operator in definitions (3.6), (3.7). See Figure 3.4 for an example. The Figure 3.4a shows the trajectory and correctly matched route. Figure 3.4b shows incorrectly matched route. Both routes have the same Fréchet distance $\delta_{\text{min}}$ to the trajectory due to an outlier in another part of the trajectory (the sample $s_{11}$; see Figure 3.2c). Since both these routes have minimal

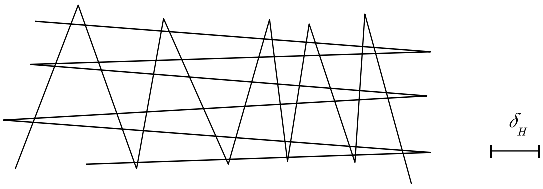


FIGURE 3.5: Two curves and their Hausdorff distance. The $\delta_H$ indicates the Hausdorff distance in scale (adaptation of example given by Alt and Godau, 1995, originally printed in Kubička et al., 2017).

Fréchet distance to the trajectory the map-matching algorithm is by definition (3.4) indifferent between them. As can be seen in Figure 3.4b this can produce obviously wrong matching.

In order to compute the Fréchet distance one has to identify the reparametrization functions $\alpha$, $\beta$ such that the maximum distance between the two curves $f$, $g$ is minimized. This has proven difficult, no algorithm for continuous curves has been found (Alt and Guibas, 1999). Alt and Godau, 1995 discovered an algorithm to compute the Fréchet distance between two polygonal curves. It runs in $O(nm\log^2(nm))$, where $n$ and $m$ are cardinalities of the two polygonal curves.

Cao and Wolfson, 2005 used the Hausdorff distance for "nonmaterialized trajectory representation". The authors formulate map-matching as a spatial mismatch correction problem and assume that the vehicle is always on the road to solve it. Their algorithm adjusts the trajectory to the road network such that the Hausdorff distance between the route and the trajectory is minimized. The strength of their contribution is that their solution is exact with respect to their problem formulation, however, practical application is likely to be limited due to properties of the Hausdorff distance discussed above. The asymptotic complexity of the algorithm is $O(n|E|^2)$, where $E$ is the set of edges in the road network graph $G$.

Alt et al., 2003 published the first map-matching method based on the Fréchet distance. The authors first solve a decision problem whether $\delta_F(f,g) \leqslant \varepsilon$, for some $\varepsilon > 0$ and then use the solution to find minimal $\varepsilon$. The method is optimal in the sense that it finds a route whose Fréchet distance to the trajectory is minimal. Nevertheless, if there are outliers in the trajectory, then there can be multiple routes with minimal Fréchet distance as discussed above. The computational demand on reasonably sized maps is high since the algorithm runs in $O(n|E|\log(n|E|)\log(|E|))$ time.

Brakatsoulas et al., 2005 followed on the work of Alt et al. and introduced a *weak Fréchet distance* in attempt to deal with the high computational demand of the original. The weak Fréchet distance[2] is a relaxed version of the Fréchet distance that does not impose the non-decreasing property on reparametrizations $\alpha(\cdot)$ and $\beta(\cdot)$. It is asymptotically faster (runs in $O(n|E|\log(n|E|))$ time) and, according to authors, produces results same as (3.7), most of the time. This, however, is not guaranteed. It is possible to find two curves whose weak Fréchet distance is different from the Fréchet distance.

Another proposition to solve the map-matching problem using the weak Fréchet distance is by Wenk et al., 2006. Their Adaptive Clipping algorithm speeds up the weak Fréchet distance computation. The asymptotic run time is slightly worse than the run time of the method proposed by Brakatsoulas et al., 2005. The authors report that average run time is reduced significantly.

[2] In some sources referred to as nonmonotonic Fréchet distance.

Further speed up of the Fréchet distance computation was reported by Chen et al., 2011a (co-authored by Driemel and Wenk). The authors extend the results of Driemel et al., 2012 for Fréchet distance approximation in near linear time. The authors report speed up with respect to Alt et al., 2003 on the order of $10^3$, in some cases.

Wei et al., 2013a revisited the problem with the sensitivity to outliers. The authors first find all routes with minimal Fréchet distance to the trajectory. If there is a unique solution, then it is returned as a correct match. A fitness function is used to find the most likely match if there are multiple solutions. Its performance depends on the choice of the fitness function. It is often based on empirical formulas that can be biased to work in the environment they were identified in. Still, this is a sophisticated approach with outlook to perform well in terms of accuracy and speed if used in conjunction with near linear time approximation of the Fréchet distance (proposed by Driemel et al., 2012 and adapted for map-matching by Chen et al., 2011a, see above).

In summary, Alt et al., 2003 pioneered the geometric map-matching with their method to compute the Fréchet distance. The authors Brakatsoulas et al., 2005, Wenk et al., 2006 and Chen et al., 2011a contributed to significant advances in lowering the computational requirements of the Fréchet distance based map-matching. However, sensitivity to outliers, which is the main drawback of this technique, remained unresolved. Wei et al., 2013a introduced a combined approach that uses a heuristic fitness function to solve this issue.

### 3.2.6 Multiple hypothesis technique based methods

The Multiple hypothesis technique (MHT) was originally developed for object tracking. A number of its variations were used in the context of map-matching as well. Multiple hypothesis methods make use of road network topology to infer where the vehicle might have gone from its prior positions. This has a number of advantages: it enforces route contiguity and it allows recursive solution to the map-matching problem. However, the set of hypotheses can grow exponentially in worst case.

As the name suggests, these methods maintain a set of *hypotheses*. The term hypothesis is used as a synonym to any candidate route under consideration. A set of seed hypotheses is generated from the first sample, usually with point-to-curve method. Then as further samples arrive the method updates its set of hypotheses and estimates the likelihood of each being the correct route. The updating process consists of *hypothesis branching* and *hypothesis pruning*. Hypothesis is branched when the vehicle arrives at an intersection: the original (parent) hypothesis is replaced with new (child) hypotheses. Each child is a clone of the parent extended into one of the directions the vehicle can take on the intersection. This guarantees that there will always be a hypothesis spanning the whole trajectory. Hypothesis pruning is used to remove outdated hypotheses. Three pruning criteria are currently used by authors: (1) limit the maximum number of hypotheses; (2) threshold likelihood scores; (3) keep only those hypotheses that lead close to latest position.

The multiple hypothesis paradigm is recursive: partial results can be obtained before the trajectory is fully processed. Each sample can be processed in time that scales linearly with the number of hypotheses, while the number of hypotheses is usually kept low due to hypothesis pruning. These properties make it particularly suitable for online map-matching and navigational applications. Another advantage is that basic failure detection is implicit: if there are no hypotheses then a failure must have occurred.

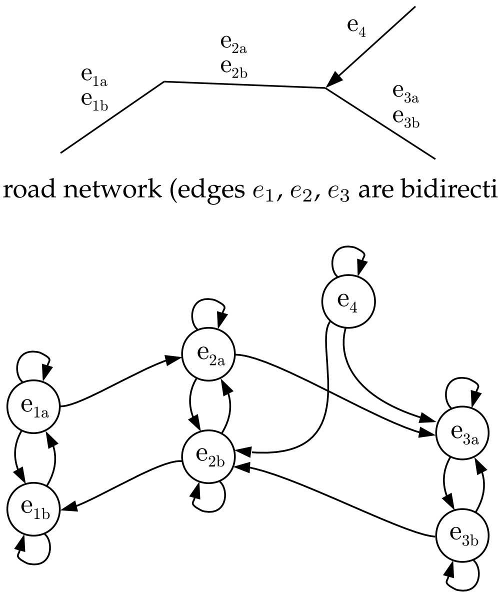
Pyo et al., 2001 designed a method for robust online map-matching. The authors derive the probability of each hypothesis using the Bayes rule, asserting all hypotheses to be true and then computing probability of the assertion being correct for each. The pruning is done by thresholding the probabilities: the hypothesis gets pruned if it gets below some predefined threshold. The authors also threshold the probabilities to decide whether the hypothesis is tentative or confirmed. This serves as simple integrity monitoring. Finally, authors always add an "off road" hypothesis between the candidates to account for the case when the road is missing in the map. The method was validated in a field test. The authors used GPS augmented with deduced reckoning based on a gyroscope and odometer. The method have failed to decide on correct matching in 4-12% cases depending on the context, but no mismatchings were observed. It also failed to identify off road condition 17% of time, but it never matched the trajectory to incorrect road.

Marchal et al., 2005 developed a simple and fast Multiple hypothesis technique based method for offline map-matching. The authors use simple scoring based on the distance between individual samples and their matched counterparts on the hypothesized routes. The hypotheses are branched when the vehicle gets to the second half of current road. The list of candidate hypotheses is trimmed to $N$ most likely hypotheses, where the $N$ is chosen experimentally. The hypothesis with the highest score is returned as the matched route. The method was validated in a field test. The authors report that 3.3% of the samples were matched to roads more distant than accuracy of the road network while computation time was $10^3$ times faster than sampling period of the GPS receivers. This method shows that interesting results can be obtained fast and with relatively little effort. Schuessler and Axhausen, 2009 followed on work of Marchal et al. They observed that the original does not perform well when there are two parallel roads close to each other. They adapted the scoring function such that differences between the vehicle speed and the speed limit are punished with lower scores in response to that. They report that this helps to discern position of the vehicle if the speed limits are different on two parallel roads.

Kubička et al., 2014 developed a method for online map-matching using Multiple hypothesis technique. The authors use *gating technique* for branching and pruning. The gate is defined as a spherical area around the latest reported position with radius larger than maximum positioning error. Hypotheses are branched whenever an intersection is found within the gate and pruned whenever hypothesis drops out from the gate. This guarantees that correct hypothesis will not get pruned. The authors compare the travel distance reported by the positioning system with the distance traveled along the hypothesized routes. The difference is considered in the scoring function together with average positioning error between the trajectory and the route. The method was validated in two short campaigns in rural and urban environments. The authors did not observe any error on rural roads. In urban areas they observed 0.5% samples temporarily mismatched on intersections.

### 3.2.7 HMM and CRF based methods

The map-matching based on Hidden Markov models received considerable attention in connection with tracking applications. The first method that uses it is by Hummel, 2006. Over ten other methods were proposed since, many are well cited. Majority of them is designed for offline map-matching. Online map-matching is possible, however the matching is usually limited to last few samples to reduce computational effort. This is known as the *sliding window technique* (Goh et al., 2012).



(A) a road network (edges $e_1$, $e_2$, $e_3$ are bidirectional)

(B) corresponding Markov model structure

FIGURE 3.6: Markov model structure according to Hummel, 2006 and Pink and Hummel, 2008 (originally printed in Kubička et al., 2017).

To shortly review the classic theory, a Markov chain is a stochastic model of a system that can randomly change state such that probability of the next state depends only on the current state. The set of states is finite. It is often visualized with a graph where nodes represent states and edges represent transitions between them. Each state defines *transition probability* distribution over the set of states. It describes the probability of transition from the current state to any other state. Hidden Markov model is a generalized Markov chain where current state is not directly observable. The relation between observations and internal states is described using *emission probability* distribution $p(l|x)$. It is a conditional distribution that observation $x$ was made due to the system being in state $l$. In the context of map-matching the $x$ is observed position while the $l$ is the road on which the vehicle might have been when the measurement took place. See Rabiner, 1989 for more information on Hidden Markov models.

Most methods model emission probabilities with a model that considers only the distance between the trajectory and the road. Positioning errors are assumed to follow isoradial Gaussian distribution. Probability $p(x|l)$ (not to be confused with $p(l|x)$) of observing some position $x$ given that the vehicle is on road $l$ is

$$p(x|l) \approx \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-d^2}{2\sigma^2}} \tag{3.8}$$

where $d$ is minimum distance between observation $x$ and any position on the road $l$. The emission probability $p(l|x)$ can be computed using Bayes rule. It simplifies to

$$p(l|x) = \frac{p(x|l)p(l)}{\sum\limits_{k=1}^{n_l} p(x|k)p(k)} \tag{3.9}$$

where $n_l$ is the number of roads in the road network and $p(l)$ is the prior probability of being on the road $l$ (prior distribution). Most methods don't have any upfront preference about the road on which the vehicle might be. If that is the case, then the prior distribution is uniform over the set of candidate roads, i.e. $p(l) = {n_l}^{-1}$.

With the Hidden Markov model set up, it can be used to perform inference on it. The goal is to recover the most likely sequence of states (that is, the route $r$), given a sequence of observations (that is, the trajectory $s$). The standard method to compute it is with the *Viterbi algorithm* (Viterbi, 1967). Its asymptotic run time is $O(nm^2)$, where $n$ is the number of observations (trajectory samples) and $m$ is the number of states (roads). All map-matching methods discussed below have their run time asymptotically dominated by the run time of the Viterbi algorithm.

Hummel, 2006 developed Bayesian classifier to match a single trajectory sample and used the Hidden Markov model to identify the route with it. The method was designed for online applications. Author identifies most likely road on which the vehicle travels using a search for road that minimizes Mahalanobis distance $\delta_M$

$$\delta_M = \left(\frac{d}{\sigma_d}\right)^2 + \left(\frac{\delta\phi}{\sigma_\phi}\right)^2 \tag{3.10}$$

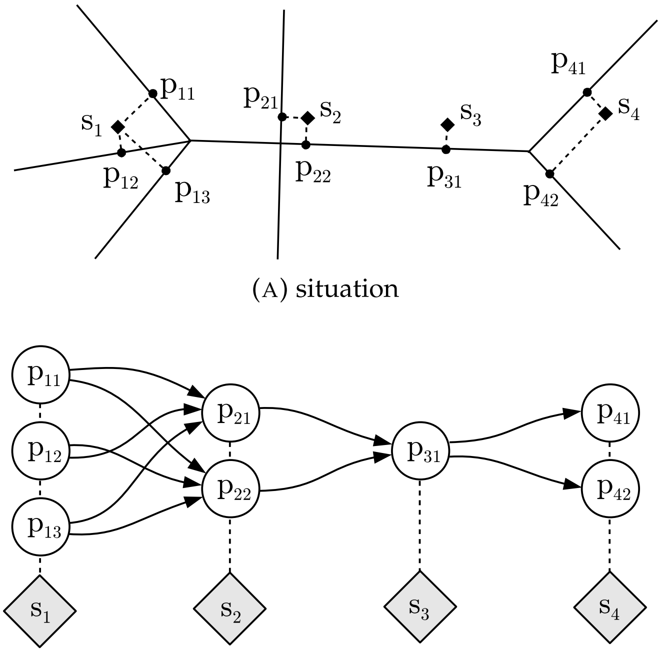
where $d$ and $\delta\phi$ are positioning and heading errors between the sample and the road. Parameters $\sigma_d$ and $\sigma_\phi$ are standard deviations in position and heading respectively. The Hidden Markov model structure follows the structure of the road network: Markov states represent roads and transitions represent the turns the vehicle can take on an intersection. Transition probabilities are distributed uniformly between the turns the vehicle can make on intersections. See Figure 3.6 for comparison between the road network graph and corresponding Hidden Markov model. Note the self-transitions on each state, it allows the vehicle to stay on the same road for a time. The dependence on vehicle heading makes this method robust, but it can have the opposite effect in some situations:

- Uncertainty in heading can be significant when moving slowly.
- The heading errors can be artificially high (up to 45 degrees) when performing a sharp turn on an intersection.

The method was validated with field test in urban environment. Duration of the test was roughly four hours. The trajectories were collected using a low cost GPS receiver at 1 Hz sampling rate in Karlsruhe, Germany. All trajectories were matched correctly, although authors report that 0.4% of samples were temporarily mismatched. These errors occurred in short duration until belief in correct route built up sufficiently. They can be considered as mismatches in online map-matching, but they bear no consequence in offline map-matching.

Pink and Hummel, 2008 extended results by Hummel, 2006. The authors introduced a number of innovations with respect to the original method. They preprocessed the trajectory using a Kalman filter in order to suppress outliers. Another innovation was in road network modeling: authors used cubic spline interpolation to generate continuous curved path. It models the shapes conventional vehicles normally follow. This was motivated by the problems with the sensitivity to heading of (3.10). This method is designed to be robust against positioning errors, however, authors do not consider trajectories with missing samples due to lacking satellite signal. This makes it suitable to mapping applications as its positioning systems are usually designed to provide position fixes without interruption.

Krumm et al., 2007 developed offline map-matching method that constrains the solution to those routes whose expected travel time is congruent with observed



(A) situation

(B) corresponding Hidden Markov model

FIGURE 3.7: Hidden Markov model according to Krumm et al., 2007 and Newson and Krumm, 2009. An example. (originally printed in Kubička et al., 2017).

travel time. The authors used a novel structure of the Hidden Markov model to achieve this. Instead of recycling road network topology as Hummel did, Krumm et al. builds his Markov model from trajectory data. See Figure 3.7 for an example. A set of states that represents possible positions of the vehicle is created for each sample. Let us denote this set $P_i$ for sample $s_i$ and let us denote $j$-th matching in $P_i$ as $p_{ij}$. The authors allow only transitions from positions in the set $P_i$ to the set $P_{i+1}$. The transition probability from some $p_{ij} \in P_i$ to $p_{(i+1)k} \in P_{i+1}$ is estimated according to the agreement between observed and expected travel times. The observed travel time is the time difference between the two samples. The expected travel time is estimated using the average speed along the roads on the shortest path between the two candidate positions $p_{ij}$ and $p_{(i+1)k}$. Emission distributions are based on the model (3.9). The authors have shown on numerous examples that their method is able to deal with difficult matching scenarios but did not quantify its performance.

Newson and Krumm, 2009 argued that estimated travel time might be affected by immediate traffic on the road. They addressed this issue and extended the method of Krumm et al., 2007. They changed the way transition probabilities are modeled: instead of working with travel times authors make use of distances. They compare the great-circle distance between two consecutive samples with the distance along the road network. The routes with minimal difference between the two are preferred. Another novelty introduced by the authors is preprocessing. They downsample the trajectory such that samples within two standard deviations from the last accepted sample are ignored. Median absolute deviation (Hampel, 1974) is used to estimate the standard deviation. It is a robust measure of variability in univariate dataset defined as

$$\mathrm{MAD} = \mathrm{med}_i\,|X_i - \mathrm{med}_j(X_j)| \tag{3.11}$$

where $X$ is the dataset whose variability is to be estimated. It has to be scaled in order to obtain consistent estimator of standard deviation. This depends on distribution type. The authors assumed the positioning error follows the Gaussian distribution for which $\sigma \approx 1.4826\text{MAD}$. They tested the method on 80-kilometer trajectory collected in Seattle, US. The performance was evaluated in novel way by comparing total length of the roads that were matched incorrectly to length of those matched correctly. The error is computed as

$$e = \frac{d_+ + d_-}{d_0} \tag{3.12}$$

where $d_0$ is the length of correctly matched roads, $d_+$ the length of mismatched roads that were added to the route and the $d_-$ length of mismatched roads that were missing in the route. The authors report no errors on sampling periods shorter than 30 seconds. Wei et al., 2013b tested this method independently and observed $98.5\%$ matching accuracy on one-second sampling period and $95\%$ matching accuracy on a 60-second sampling period.

Hunter et al., 2014 developed a "Path inference filter". The authors show that map-matching based on Hidden Markov models is subject to selection bias and propose the Path inference filter as the solution. It is based on Conditional random fields, a generalization of Hidden Markov models. The authors build directed graph with similar structure as the model used by Krumm et al. (Figure 3.7b). A set $P_i$ of candidate positions on the road network is created for each sample $s_i$ in the trajectory. Each candidate position $p_{ij} \in P_i$ has assigned a score that represents the likelihood that vehicle was in position $p_{ij}$ when $s_i$ was observed. The probabilistic model (3.9) is used for this. The authors create edge for each path the vehicle can take between each pair $p_{ij}$ and $p_{(i+1)k}$ in $P_i \times P_{i+1}$. Each such path has assigned score according to a *driver model* that represents the likelihood that vehicle took this path. The number of paths for each pair in $P_i \times P_{i+1}$ is upper bounded by speed limits on the roads in question. The authors list all complete paths in this graph (candidate routes from origin to destination) and define *potential* of each as a product of the scores of the candidate positions and of the candidate routes between them. The potential function becomes probability mass function when normalized to one. With normalized potentials authors use the Viterbi algorithm to identify the most probable route. The Path inference filter was primarily developed for tracking applications. The authors report that it stays competitive over the full range of sampling rates. Testing has shown ability to match all routes correctly on trajectories with high sampling rates and 75% of the trajectories with two minute sampling period. This method is computationally demanding in both memory and time.

### 3.2.8 Map-matching integrity

Correct map-matching is not always possible when there is serious incongruence between the trajectory and the road network or when the situation is ambiguous. The reliability of map-matching in such situations is questionable. Hence, some map-matching methods perform *integrity monitoring* to report on the reliability of its output. This is often critical, especially in applications related to safety and electronic fee collection. The term integrity monitoring was imported from aerial navigation where integrity monitoring is used to verify reliability of satellite navigation based position fixes. Integrity monitoring in context of map-matching can be formulated independent of the matching method. It takes the trajectory, the road network and sometimes the map-matching output as input and provide *integrity indicators* that indicate how reliable the map-matching output is.

Integrity monitoring performance is typically evaluated with overall correct detection rate derived from counts of missed detections and false alarms. The results are sensitive to selected alarm levels against which the integrity indicators are compared. The sensitivity of the monitoring system is determined with them. They should be set experimentally such that both missed detections and false alarm occurences are kept at minimum. There is a trade-off: too many false alarms are due to high sensitivity while too many missed detections are due to low sensitivity. Note that in the context of map-matching there can always be unobservable missed detections since both positioning and road network errors are not bounded. Seemingly congruent scenarios where trajectory is aligned with the wrong roads can theoretically be observed. It implies that there can never be any kind of guarantee that the matching is correct.

First mention of map-matching integrity was in Quddus's thesis (Quddus, 2006) and in publications published during its preparation. Author proposed a method that uses uncertainty in position, distance error and heading error to produce heuristic integrity indicator based on Fuzzy inference system. Author validated his integrity monitoring with three map-matching methods and reports correct detection rate of at least 91%, missed detection rate below 8.5% and false alarm rate below 10.1%.

Velaga, 2010 developed integrity method based on the work by Quddus. Author uses two Fuzzy inference systems: one is used when RAIM (receiver autonomous integrity monitoring) labels positioning fix reliable and other when not. Reported missed detection and false alarm rates are both below 1%. This gives overall correct detection rate over 98%.

Li et al., 2013a (co-authored by Quddus) proposed a tightly-coupled integrity monitoring for map-matching. The authors review drawbacks of classical RAIM when used in the context of map-matching and then propose an integrated navigation system with adapted RAIM functionality that is better suited for this task. The navigation system integrates information from GNSS, gyroscope and digital elevation model. The integration is achieved using extended Kalman filter. Uncertainty in measurements and related filter residuals are then used in the adapted RAIM to compute the horizontal protection level (HPL). The authors tested their solution in urban and suburban field trials. They report both false and missed detection rate less than 0.1% with 99.83% overall correct detection rate. These are excellent results, however, specialized navigation system with a gyroscope and GNSS receiver that is able to report raw pseudorange measurements is required.

Jabbour et al., 2008 proposed a multiple hypothesis technique based method and presented a simple integrity method with it. The authors use two integrity indicators: (1) the number of effective hypotheses, $N_{\mathrm{eff}}$, and (2) the "normalized innovation squared", NIS. The $N_{\mathrm{eff}}$ is the number of hypotheses with high likelihood of being correct and the NIS is similar to Mahalanobis distance between the trajectory sample and its matching in position and heading. The authors conducted a field trial with a navigation system based on GPS, odometer and fiber-optic gyroscope. Overall correct detection rate over 88.8% was reported. An adaptation of this integrity method was later used by Bonnifait et al., 2009.

Toledo-Moreo et al., 2010 proposed lane-level map-matching method with integrity monitoring. This solution uses an integrated navigation system based on GNSS, odometry and gyroscope. The lane-level matching is enabled via novel road network model named *eMap* (enhanced map). The eMap models lane shapes as a clothoids (also known as Euler Spirals). Since clothoids are used in highway engineering authors argue that similar mathematical structures can be expected to emerge in sampled trajectories. The map-matching algorithm is based on a Particle filter. Two integrity indicators are used: (1) lane occupancy probability ($\mu$LO) and (2) lane position protection level (LPPL). The $\mu$LO is a sum of normalized particle weights that occupy the matched lane and LPPL is positioning protection level analogous to horizontal protection level (HPL) used in the classical RAIM. The authors tested their method in three short field trials. Reported overall correct detection rate ranged from 98.7% to 99.2%. The missed detection rate was in range from 0% to 1.2%. The false alarm rate ranged from 0.7% to 1.7%.

# Chapter 4

# Problem definition & analysis

This chapter defines the general eco-routing problem and the eco-routing performance. It is then used to measure the performance of three eco-routing methods, two of which were proposed in the literature. This establishes a baseline for the eco-routing method proposed in later chapters. The observed performance figures are, in some cases, negative which implies that some of the tested eco-routing methods do not save energy. It is rare to see such a result reported in the literature. This motivated a closer look at the used evaluation methodologies, which revealed that the prevalent method is not capable of observing negative savings by design. The problem is that the same consumption (or pollutant emission) model is used for both routing and evaluation. Not only that such evaluation will necessarily result in non-negative savings, but it is also shown to be prone to result in inflated eco-routing performance figures.

The used definition of eco-routing is general enough to apply to all eco-routing methods reviewed in Section 3.1. It supports time-dependent eco-routing as well as constrained eco-routing. The cited works don't formally define eco-routing performance; authors typically report average observed savings instead. The definition used in this work allows comparison between multiple methods. It makes the ability of eco-routing to save costs measurable.

The study of the baseline methods is an updated version of Kubička et al., 2016b. The publication is based on an older version of the traffic scenario where the simulations were conducted.

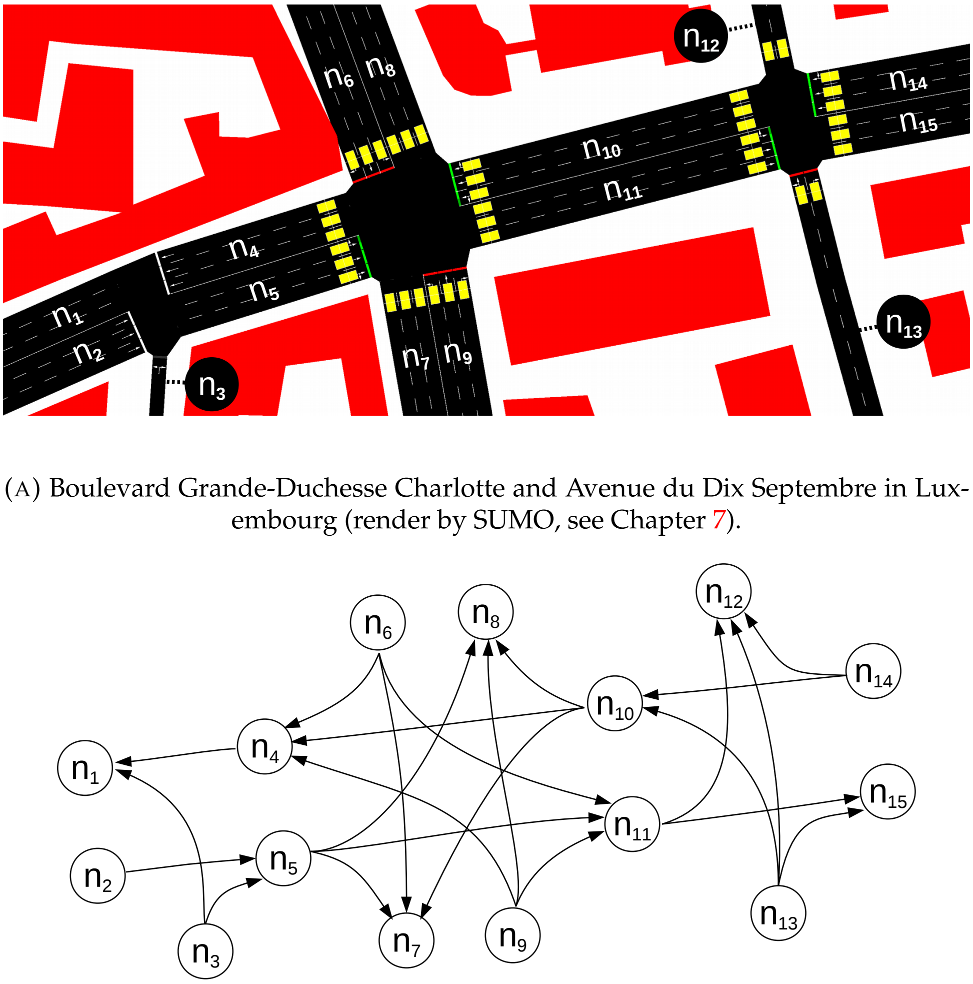
## 4.1 Problem definition

Consider a road network graph $G = (V, E)$, where nodes $n \in V$ represent roads and edges $e \in E$ represent connections between them. The Figure 4.1 illustrates this. It shows the relation between a road network (Figure 4.1a) and the corresponding road network graph $G$ (Figure 4.1b). The nodes are shown as circles, the edges as arrows. There are fifteen nodes $n_1$ to $n_{15}$ representing the roads in the road network. The edges (arrows) spanning from nodes indicate to which nodes (roads) can the vehicle turn to at the downstream intersection.

Let each edge $e = (n_1, n_2) \in V \times V$ have assigned a column vector $c_e(t) \in \mathbb{R}^k$ with routing costs. Consider a path $p = (e_1, e_2, \ldots)$ in $G$. Let the cost of this path be denoted $F(p,t)$ for departure at the time $t$ and defined as the sum of costs on individual edges along $p$.

$$F(p,t) = \sum_{e \in p} c_e(t + \lambda_e) \tag{4.1}$$

where $\lambda_e$ is the travel time from the origin to the edge $e$. Let also $D(p)$ be the length of the path $p$ and $T(p,t)$ the travel time on a path $p$ when departed at some time $t$.



(A) Boulevard Grande-Duchesse Charlotte and Avenue du Dix Septembre in Luxembourg (render by SUMO, see Chapter 7).

(B) The corresponding road network graph $G$.

FIGURE 4.1: The relationship between the road network and the road network graph.

Let $l_e$ be the length of road $e$ and $\tau_e(t)$ the travel time on $e$ if arrived there at time $t$, then

$$D(p) = \sum_{e \in p} l_e \tag{4.2}$$

$$T(p,t) = \sum_{e \in p} \tau_e(t + \lambda_e) \tag{4.3}$$

Note that $T(p,t)$ could also be defined directly with $\lambda_e$. The used formulation has the advantage of highlighting the similarity between (4.3) and (4.1).

Let $\mathbb{P}$ be a set of all simple[1] paths in $G$. Let a column vector $w \in \mathbb{R}^k$ be weights of individual costs and let $C \in \mathbb{R}^k$ be constraints on $F(p,t)$. Let $o \in E$ be the origin edge and $d \in E$ the destination edge. Then, the eco-route $p_e = (e_1, \ldots, e_n)$ of length

[1] Simple path is a path without cycles: its edges are distinct

$n$ is defined as the solution to the optimization problem

$$\begin{aligned} p_e(o,d,t) = \ & \operatorname*{argmin}_{p \in \mathbb{P}} w^T F(p,t) \\ \text{s.t.} \quad & e_1 = o \\ & e_n = d \\ & C - F(p,t) \geqslant 0 \end{aligned} \tag{4.4}$$

This model is general enough to cover most, if not all, eco-routing applications. The classical minimum distance and minimum time routing are its special cases. Its distinctive feature is that it is fully time-dependent: the costs are allowed to change in time. It also allows combining multiple costs according to a given linear weighting scheme, provided that the costs satisfy the additivity criterion imposed by (4.1). The consumption, pollutant emissions, travel time, travel distance, and possibly other costs can be minimized simultaneously. For example, it can be used to penalize paths with excessive pollutant emissions or prolonged travel times. The constraints can be used to impose limits on any of the costs. Some existing applications consider consumption minimization with a constraint on maximum travel time (like Juřík et al., 2014), but other applications are possible. For example, one can limit pollutant emissions or consider only paths that do not require battery recharging (respectively refueling) before reaching the destination. The model also allows to limit or completely disallow passing through certain areas at certain times. For example, one can disallow routing around schools in the time when the children are there.

## 4.2 Eco-routing problem as modeled in literature

Most eco-routing methods discussed in Chapter 3 are similar: they consider only a single cost to minimize, do not impose any constraints and assume that the cost is time independent. Let us consider a reduced version of the optimization problem (4.4) that reflects this. Let the eco-route $p_e$ be defined as the solution to the following variant of the problem (4.4)

$$\begin{aligned} p_e(o,d) = \ & \operatorname*{argmin}_{p \in \mathbb{P}} F(p) \\ \text{s.t.} \quad & e_1 = o \\ & e_n = d \end{aligned} \tag{4.5}$$

where the path cost $F$ is a scalar (since there is only one cost to minimize) and without the dependency on the time of departure. This model is optimal under the assumption that the costs $c_e$ are known exactly and constant over time.

Note that while this model is commonly used in the literature, there are exceptions. For example Juřík et al., 2014 consider routing under a time-constraint and Kluge et al., 2013 consider time-dependent routing. The model used by Juřík et al., 2014 is reduced to (4.5) by setting the time constraint to $+\infty$. The model by Kluge et al., 2013 does not comply with (4.5).

To account for imperfections of current consumption (or pollutant emission) models let us also introduce an eco-routing model that has cost functions augmented with random errors. Let $\hat{c}_e$ be an estimate of cost $c_e$ defined as

$$\hat{c}_e = c_e + \delta_e \tag{4.6}$$

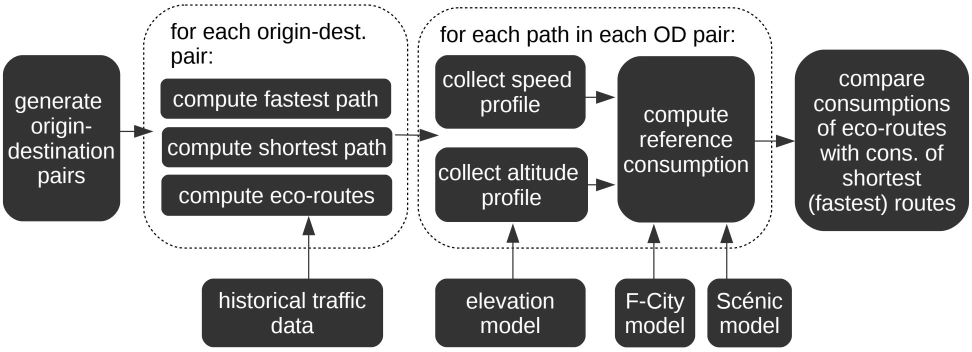

generate origin-destination pairs
for each origin-dest. pair:
compute fastest path
compute shortest path
compute eco-routes
for each path in each OD pair:
collect speed profile
collect altitude profile
compute reference consumption
compare consumptions of eco-routes with cons. of shortest (fastest) routes
historical traffic data
elevation model
F-City model
Scénic model

FIGURE 4.2: Experiment toolchain (simplified; originally printed in Kubička et al., 2016b).

where $\delta_e \in \mathbb{R}$ is an unknown estimation error. It is considered a random variable with distribution specific to every edge $e$. No assumptions need to be made about the random variable: it can have any mean, moments, variance and any distribution shape. The estimated cost of some path $p$ in $G$, denoted $\hat{F}$, is then

$$\hat{F}(p) = \sum_{e \in p} c_e + \sum_{e \in p} \delta_e \tag{4.7}$$

and the estimated eco-route $\hat{p}_e$ can be defined as the solution to

$$\begin{aligned} \hat{p}_e(o,d) = \; & \underset{p \in \mathbb{P}}{\operatorname{argmin}} \hat{F}(p) \\ \text{s.t.} \quad & e_1 = o \\ & e_n = d \end{aligned} \tag{4.8}$$

### 4.2.1 Eco-routing performance

Eco-routing *performance* is defined in this section as a quantity that summarizes the ability of an eco-routing method to save costs. Let $p_{\text{ref}}(o,d) : V \times V \to \mathbb{P}$ be reference paths to which costs on eco-routes are compared. The reference paths considered in this work are shortest and fastest paths. Let $\mathbb{V}_{od}$ be a set of pairs $(o,d)$, where $o \in V$ is origin and $d \in V$ is destination. The performance $P$ of eco-routing $p_e$ with respect to $p_{\text{ref}}$ on the set of trips $\mathbb{V}_{od}$ is defined as

$$P = 1 - \frac{\sum\limits_{(o,d) \in \mathbb{V}_{od}} F(p_e(o,d))}{\sum\limits_{(o,d) \in \mathbb{V}_{od}} F(p_{\text{ref}}(o,d))} \tag{4.9}$$

where the numerator is the sum of costs on eco-routes between all origin-destination pairs in the road network graph, and the denominator is an analogous sum of costs on the reference paths. It summarizes the ability of the eco-routing method to lower costs with respect to the reference paths.

Consider also a performance for the problem (4.8). Distinct formulation is needed as this model uses path cost function $\hat{F}$ rather than $F$. Let the *estimated performance*

$\hat{P}$ be defined as

$$P = 1 - \frac{\sum\limits_{(o,d)\in\mathbb{V}_{od}} \hat{F}(\hat{p}_e(o,d))}{\sum\limits_{(o,d)\in\mathbb{V}_{od}} \hat{F}(p_{\text{ref}}(o,d))} \tag{4.10}$$

The important thing to notice with $\hat{P}$ is that $\hat{F}$ uses internally costs $\hat{c}_e$, which are also used by $\hat{p}_e$ for eco-routing. In another words, the same costs are used for routing and evaluation of the eco-routing method. These costs have perfectly correlated errors in both cases. Most methods in published literature were evaluated in this way. The Section 4.4 discusses in detail the drawbacks of this methodology.

The performance as defined is not the mean savings. The mean savings are always reported together with performance in this work. The evaluation of eco-routing methods is, however, based on a comparison of their performances and not the mean savings. This is because the performance summarizes the actual savings on all trips from the set of origin-destination pairs $\mathbb{V}_{od}$, while the mean savings only describe their central tendency.

Note that performance is also evaluated on time-dependent methods in this work. The given definitions of performance do not consider their departure time. It does not need to be considered as another variable since all experiments were conducted at the same (known) departure time.

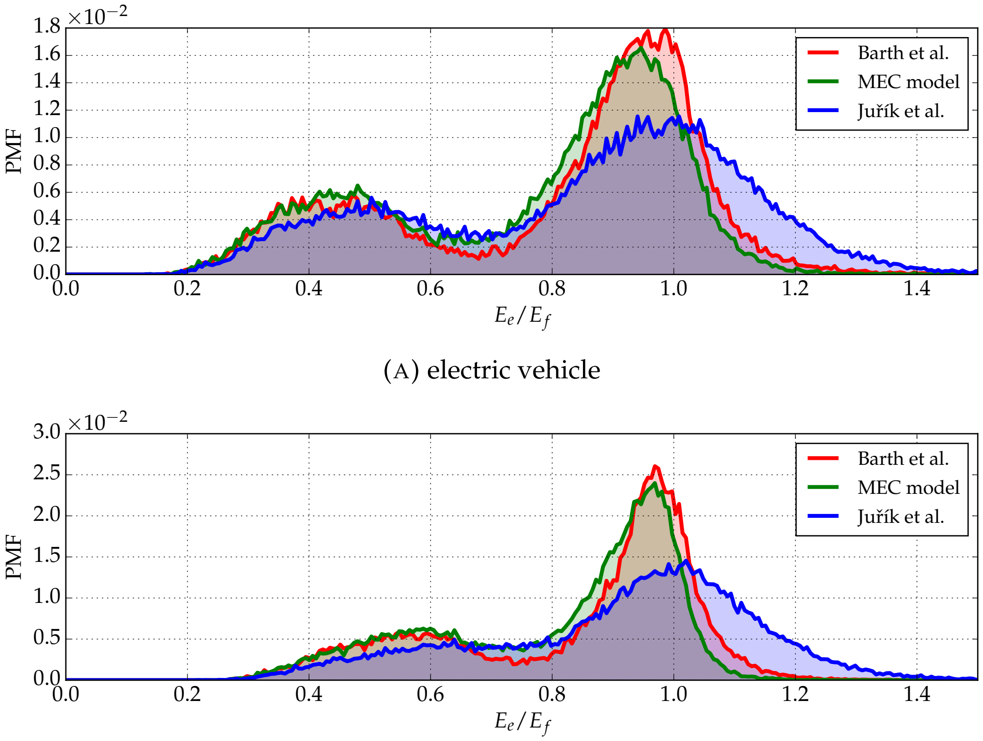
## 4.3 Performance of current methods

We have studied the performance of current eco-routing methods in Kubička et al., 2016b. This section presents an updated version of the experiment from this paper. It is based on LuST traffic scenario version 2. The original results are based on LuST 1.1. For more information on LuST see Chapter 7.

The toolchain used to conduct the experiment is depicted diagrammatically in Figure 4.2. The methodology is discussed in detail in Chapter 7. The experiments were conducted within a traffic simulation of a real European city (Luxembourg). The departure time was set to midnight.

Three time-independent unconstrained eco-routing methods were studied. Their implementation is discussed in Section 8.1. The routing was done with Dijkstra's algorithm. Two methods used consumption models proposed in the literature (Barth et al., 2007, Juřík et al., 2014). The third method used the mean energy consumption (abbrev. MEC). It assigns a cost to each edge according to the mean observed consumption there. Note that all three consumption models were tested in two variants: with an electric vehicle and with a conventional vehicle. Microscopic consumption estimation models of these vehicles were used to compute reference consumption. They are introduced in Section 7.2.

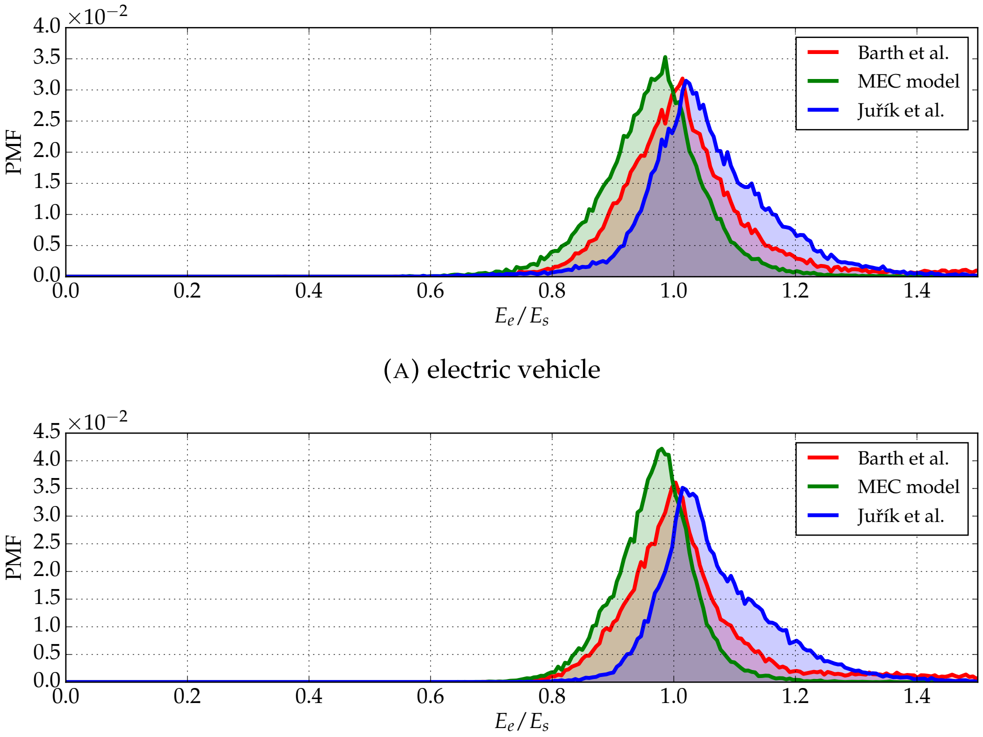
The goal is to evaluate performances $P$ and $\hat{P}$. A set of 60,478 origin-destination pairs was extracted from scenario's trip origins and destinations. This constitutes the set $\mathbb{V}_{od}$ on which the performance is evaluated. Eight paths were computed between each origin and destination: three eco-routes according to Barth et al., 2007, Juřík et al., 2014 and MEC model for both electric and conventional vehicles plus shortest and fastest paths as references. A vehicle traveling on each of these paths was tried in the simulation. Vehicle speed profiles were collected and used to estimate consumption with the reference consumption models. Note that the study considers energy consumption even the conventional vehicle consumes fuel rather



(A) electric vehicle

(B) conventional vehicle

FIGURE 4.3: Distribution of savings relative to fastest paths.



(A) electric vehicle

(B) conventional vehicle

FIGURE 4.4: Distribution of savings relative to shortest paths.

TABLE 4.1: Routing results.

| | | wrt fastest | | wrt shortest | |
|---|---|---|---|---|---|
| | | EV† (%) | ICEV‡ (%) | EV† (%) | ICEV‡ (%) |
| $P$ | Barth et al., 2007 | 26.7 | 20.1 | -8.0 | -3.2 |
| | Juřík et al., 2014 | 29.7 | 18.8 | -3.5 | -4.9 |
| | MEC model | 34.5 | 24.7 | 3.6 | 2.8 |
| $\hat{P}$ | Barth et al., 2007 | 7.2 | 7.3 | 2.0 | 2.3 |
| | Juřík et al., 2014 | 50.9 | 53.4 | 2.2 | 2.7 |
| | MEC model | 30.3 | 18.7 | 4.9 | 3.9 |
| Probability of failure | Barth et al., 2007 | 8.7 | 8.6 | 5.2 | 6.1 |
| | Juřík et al., 2014 | 24.5 | 30.4 | 30.0 | 36.9 |
| | MEC model | 6.4 | 4.3 | 7.6 | 5.9 |
| Eco-route same as reference | Barth et al., 2007 | 28.6 | 29.8 | 40.6 | 35.8 |
| | Juřík et al., 2014 | 9.4 | 8.7 | 37.6 | 32.2 |
| | MEC model | 15.5 | 22.1 | 25.0 | 29.8 |
| Mean savings | Barth et al., 2007 | 13.9 ± 0.18 | 11.1 ± 0.15 | -5.6 ± 0.18 | -2.2 ± 0.10 |
| | Juřík et al., 2014 | 13.7 ± 0.21 | 8.2 ± 0.17 | -3.6 ± 0.07 | -4.7 ± 0.07 |
| | MEC model | 19.3 ± 0.19 | 14.6 ± 0.15 | 2.9 ± 0.06 | 2.4 ± 0.05 |
| Mean travel time delay | Barth et al., 2007 | 9.7 ± 0.14 | 8.9 ± 0.13 | -5.9 ± 0.10 | -6.4 ± 0.10 |
| | Juřík et al., 2014 | 28.0 ± 0.25 | 29.4 ± 0.26 | 8.4 ± 0.16 | 9.6 ± 0.17 |
| | MEC model | 14.2 ± 0.18 | 11.4 ± 0.16 | -2.5 ± 0.12 | -4.9 ± 0.10 |

† electric vehicle
‡ internal combustion engine (conventional) vehicle

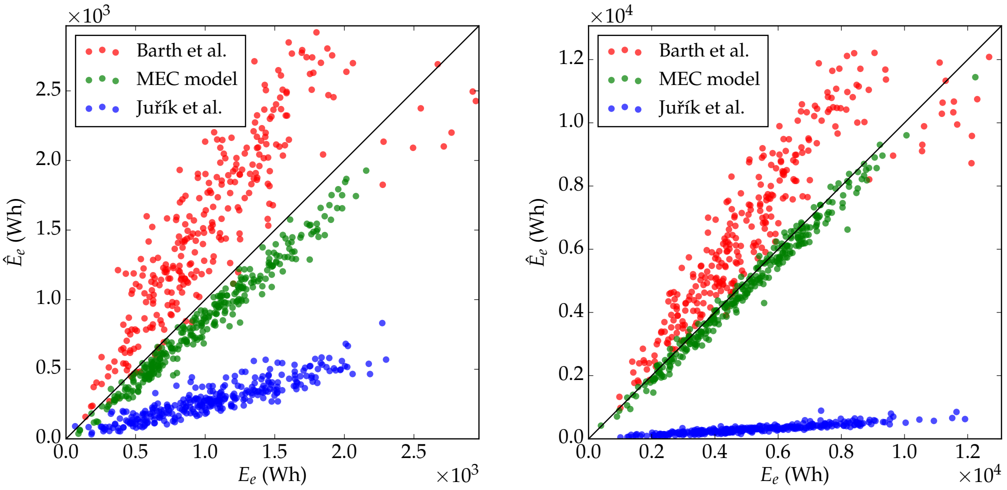
than energy. Fuel consumption can be computed from energy consumption on the basis of fuel's lower heating value.

Each of the 60,478 resulting records contains the true[2] consumption on the eco-routes and the true consumption on the fastest and shortest paths. This allows generating a set of 60,478 observed savings for each eco-routing method. The computation took an equivalent of 1154 hours of computing time distributed on a computing cluster.

The observed distributions of savings are shown in figures 4.3 and 4.4. For some fixed origin, destination and time of departure let $E_e$ be the consumption on the eco-route, $E_s$ the consumption on the shortest path and $E_f$ the consumption on the fastest path. The Figure 4.3 shows the distribution of observed savings on the eco-routes with respect to the fastest paths (the distribution of the ratios $E_e/E_f$). The Figure 4.4 shows the distribution of observed savings with respect to the shortest paths (the distribution of the ratios $E_e/E_s$). There are three curves in each plot, one for each method. The area under the curves below 1.0 on the horizontal axis are the cases when the eco-routing method saved some energy while the area under the curve above 1.0 on the horizontal axis are the cases when the eco-route required more energy than the reference path. These are the cases in which the eco-routing method failed with respect to the reference. An exceptionally high probability of the eco-route being identical to the shortest (or fastest) path was observed in the results. This appears as a discontinuity in the distributions in Figures 4.3 and 4.4. It was removed from the plots and reported in Table 4.1 where the results are summarized.

Correlations between observed and estimated consumptions are shown in Figure 4.5. Results for the electric vehicle are in Figure 4.5a, for the conventional vehicle in Figure 4.5b. Each dot represents a single trip. The results are color-coded. The red dots are for Barth et al., 2007, the green dots are for MEC model, and the blue dots

[2] The consumption estimated by the reference models is considered to be the true consumption.



(A) electric vehicle

(B) conventional vehicle

FIGURE 4.5: Correlation of estimated and reference consumptions.

are for Juřík et al., 2014. The reference consumption on the eco-route $E_e$ (computed with the reference consumption models) is on the horizontal axis while the estimated consumption $\hat{E}_e$ is on the vertical axis. Perfect estimation would be achieved if all dots were aligned on the black diagonal line. The dots above are the cases when the consumption was overestimated, and the dots below it are the cases when it was underestimated.

The results are summarized in Table 4.1. All fields are relative quantities (in percent) with eco-routes being compared either to the shortest paths or the fastest paths. Both reference consumption models (electric vehicle and conventional vehicle; presented in Section 7.2) are shown. Hence, the last four columns in the table are from left to right the results (1) for the electric vehicle with eco-routes compared to the fastest paths; (2) for the conventional vehicle with eco-routes compared to the fastest paths; (3) for the electric vehicle with eco-routes compared to the shortest paths; (4) for the conventional vehicle with eco-routes compared to the shortest paths. The 95% confidence intervals are listed for the mean values given in the table. The table has six sections:

1. $P$ and $\hat{P}$ is the performance and estimated performance, respectively. The $P$ is the relative amount of energy saved on a typical trip, while $\hat{P}$ is the estimate of $P$ when the same consumption model was used for the eco-routing and to estimate the savings on the eco-route (see its definition given in Section 4.2.1). Negative values indicate losses rather than savings.

2. *"Probability of failure"* is the sample probability that the eco-route is less economical than the fastest (resp. shortest) path and that it requires longer travel time than the fastest (resp. shortest) path.

3. *"Eco-route same as reference"* is the portion of cases for which the eco-route is identical to the fastest (resp. shortest) path.

4. *"Mean savings"* is the average reduction of energy consumption on the eco-route when compared to the fastest (resp. shortest) path.

TABLE 4.2: Eco-routing evaluation techniques in literature.

| | Method | Evaluation |
|---|---|---|
| Methods | Andersen et al., 2013 | demonstration |
| | Boriboonsomsin et al., 2012 | macroscopic |
| | De Nunzio et al., 2016 | macroscopic |
| | Juřík et al., 2014 | demonstration |
| | Kluge et al., 2013 | macroscopic |
| | Nie and Li, 2013 | macroscopic |
| | Yao and Song, 2013 | macroscopic |
| Case studies | Ahn and Rakha, 2008 | field trial |
| | Ahn and Rakha, 2013 | microscopic |
| | Ericsson et al., 2006 | macroscopic |
| | Kubička et al., 2016b | microscopic |
| | Minett et al., 2011 | field trial |

5. *"Mean travel time delay"* is the average increase in travel time on the eco-route when compared to the fastest (resp. shortest) path.

The MEC-based eco-routing shows the highest performance $P$ in all categories. While all three methods manage to save energy (or fuel) when compared to the fastest paths, the MEC-based eco-routing is the only one which manages the same when compared to the shortest paths. The other methods show performance inferior to shortest paths. The performance is in five out of six cases higher for the electric vehicle. This is likely because the electric vehicle has simpler powertrain than the conventional vehicle (fixed gearing, electric motor instead of the internal combustion engine). The probability of failure is less than 9% in all cases for MEC-based eco-routing and Barth et al., 2007. However, Juřík et al., 2014 shows the probabilities of failure between 24% and 37% depending on the case. This corresponds with distributions in Figures 4.3 and 4.4 where Juřík et al., 2014 shows heaviest right tails among the three methods in all four cases. The mean travel time delay is higher when comparing it to the fastest paths, which is to be expected. With respect to the shortest paths, the method by Barth et al., 2007 and MEC-based eco-routing show negative mean travel time delays. This implies that the eco-routes save travel time on average when compared to the shortest paths.

The Juřík et al., 2014 is the most practical method. There is no need to fit any coefficients. The Barth et al., 2007 requires a set of prerecorded trips. The MEC model requires a large amount of data from identical vehicles, which makes the feasibility of its deployment in a real-world application doubtful. To summarize, there seems to be inverse proportionality between the feasibility of the method and its performance since the less feasible methods have shown higher performance.

## 4.4 Conditions under which methods overestimate own performance

The methodology used to evaluate several published eco-routing methods is questioned in this section. The methodologies used in the literature are listed in Table 4.2. There are four approaches:

- *Demonstrations* are used to show savings in a few specific instances. Authors run their eco-routing method for a trip with chosen origin and destination and then report savings by comparing the consumption (or pollutant emission) estimates on the eco-route to consumption (or pollutant emission) estimates on the shortest path or the fastest path. The same consumption (or pollutant emission) model is used for both eco-routing and the evaluation.

- *Field trials* are sometimes used to estimate savings in real traffic. Only limited campaigns were conducted in past as large-scale field trials are cost prohibitive.
- *Macroscopic evaluation* is technically similar to the demonstration with the difference that the savings are estimated for a large number of trips, each with a unique origin and destination. This allows to approximate eco-routing performance for the method under study. The same consumption (or pollutant emission) model is used for both eco-routing and the evaluation.
- *Microscopic evaluation* is the method used in this work. It is done in a microscopic traffic simulator and with a microscopic consumption (or pollutant emission) estimation model for evaluation. The eco-routes and reference (shortest or fastest) paths are computed for a large number of trips, similarly like in a macroscopic evaluation. All identified paths are then tried in a microscopic simulation. Resulting speed profiles are used to estimate the "true" consumption on these paths with a microscopic reference consumption (or pollutant emission) model. They are used to obtain a reliable approximation of the performance.

Note that all eco-routing methods in Table 4.2 in section "methods" use either demonstration or macroscopic evaluation. The key observation is that these publications propose some consumption (or pollutant emission) model $\hat{F}$ and then use it both for eco-routing and its evaluation. Such evaluation can never result in negative savings. Negative savings would be observed for a given trip if the consumption (or pollutant emissions) would be higher on the eco-route than on the reference (shortest or fastest) path. This can never happen in demonstrations and macroscopic evaluations because the same model is used for eco-routing and the evaluation. The cost estimation errors are perfectly correlated for the two tasks. If the reference path would have lower cost, then it would have to be the eco-route as well, as follows from the definition of eco-routing.

The $\hat{P}$ models this: it is defined such that it uses cost $\hat{F}$ to estimate performance while in the same time it considers eco-routes based on eco-routing model (4.8) that minimize the same cost $\hat{F}$. It is argued below that due to this the estimated performance $\hat{P}$ might also be higher than the actual performance $P$, under some conditions.

From the cost function (4.7) it follows that the difference between $\hat{F}$ and $F$ is a sum of random errors

$$\hat{F}(p) - F(p) = \sum_{e \in p} \delta_e \tag{4.11}$$

which does not necessarily converge to zero in the long run. Let us denote average of this error on eco-routes as $\chi$ and on reference paths as $\psi$. Let us also denote $\mathbb{P}_e$ the set of studied eco-routes as $\mathbb{P}_e = \{(o,d) \in E \times E : \hat{p}_e(o,d)\}$. Similarly let $\mathbb{P}_{\text{ref}}$ be the set of all reference paths in the road network, $\mathbb{P}_{\text{ref}} = \{(o,d) \in E \times E : p_{\text{ref}}(o,d)\}$. Then,

$$\chi = \frac{1}{||\mathbb{P}_e||} \sum_{\hat{p}_e \in \mathbb{P}_e} \sum_{e \in \hat{p}_e} \delta_e \tag{4.12}$$

$$\psi = \frac{1}{||\mathbb{P}_{\text{ref}}||} \sum_{p_{\text{ref}} \in \mathbb{P}_{\text{ref}}} \sum_{e \in p_{\text{ref}}} \delta_e \tag{4.13}$$

where $||\mathbb{P}_e||$ and $||\mathbb{P}_{\text{ref}}||$ are cardinalities of the sets $\mathbb{P}_e$ and $\mathbb{P}_{\text{ref}}$ respectively.

**Theorem 1.** *$\hat{P} > P$ if $\chi < \psi$, $\chi < 0$, $\psi \neq 0$*

*Proof.* Consider the relationship between $P$ and $\hat{P}$. Let us suppose $P = \hat{P}$, then

$$\frac{\sum\limits_{o \in E} \sum\limits_{d \in E} F(\hat{p}_e(o,d))}{\sum\limits_{o \in E} \sum\limits_{d \in E} F(p_{\text{ref}}(o,d))} = \frac{\sum\limits_{o \in E} \sum\limits_{d \in E} \hat{F}(\hat{p}_e(o,d))}{\sum\limits_{o \in E} \sum\limits_{d \in E} \hat{F}(p_{\text{ref}}(o,d))} \tag{4.14}$$

which can be rewritten in form

$$\frac{F_e}{F_{\text{ref}}} = \frac{F_e + K\chi}{F_{\text{ref}} + K\psi} \tag{4.15}$$

where $K = ||\mathbb{P}_e|| = ||\mathbb{P}_{\text{ref}}||$ and $F_e$, $F_{\text{ref}}$ read

$$F_e = \sum_{o \in E} \sum_{d \in E} F(\hat{p}_e(o,d)) \tag{4.16}$$

$$F_{\text{ref}} = \sum_{o \in E} \sum_{d \in E} F(p_{\text{ref}}(o,d)) \tag{4.17}$$

Provided that $\psi \neq 0$, it follows from (4.15) that $P = \hat{P}$ if and only if

$$\frac{F_e}{F_{\text{ref}}} = \frac{\chi}{\psi} \tag{4.18}$$

Both $F_e$ and $F_{\text{ref}}$ are positive. In terms of pollutant emissions, there cannot be negative emissions. In terms of consumption, only the kinetic and the potential energy changes can contribute negatively to consumption. The kinetic energy is zero when the vehicle departs from the origin, and when it arrives at the destination, hence it cannot render consumption negative for the whole trip. The potential energy can make the consumption negative only if the origin is at a higher altitude than the destination. The path consumption is lower-bounded by the potential energy difference between the origin and the destination. The potential energy is an odd function of altitude. Since both $F_e$ and $F_{\text{ref}}$ sum over all $(o, d)$ origin-destination pairs there is for any path from $o$ to $d$ the opposite path from $d$ to $o$. It follows that if consumption for a path from $o$ to $d$ is at least, say, $c$, then for a path from $d$ to $o$ it is at least $-c$. Note that it does not matter which path the vehicle takes. Consequently, the sum over all such paths is necessarily nonnegative. The only case when $F_e$ or $F_{\text{ref}}$ is zero is when $\hat{F} \equiv 0$, which can be rejected.

Consider $F_e \leqslant F_{\text{ref}}$ which implies that eco-routing performance $P$ is positive (some amount of energy is saved on a typical trip), or zero. It follows that $F_e/F_{\text{ref}} \in (0, 1]$. Recall that $\chi < \psi$, $\psi \neq 0$. Consequently, there can be three cases according to the sign of $\chi$ and $\psi$.

$$\chi < 0, \psi > 0 \implies \frac{\chi}{\psi} < 0 \implies \frac{F_e}{F_{\text{ref}}} > \frac{\chi}{\psi} \tag{4.19}$$

$$\chi < 0, \psi < 0 \implies \frac{\chi}{\psi} > 1 \implies \frac{F_e}{F_{\text{ref}}} < \frac{\chi}{\psi} \tag{4.20}$$

$$\chi > 0, \psi > 0 \implies \frac{\chi}{\psi} \in [0, 1] \tag{4.21}$$

Immediate result is that $\hat{P} \neq P$ in cases (4.19), (4.20). Only when $\chi$ is positive (4.21) it is possible to have the equality. This case is ruled out by the conditions of the theorem.

With $\chi < 0$ the performance is necessarily overestimated. In case (4.19) it suffices to convert to the form of (4.15) and substitute back. We obtain $\hat{P} > P$. In case (4.20) we can follow the same steps, but when converting to the form of (4.15) we are forced to multiply both sides by $\psi$, which is negative. The inequality gets switched, and the same result follows: $\hat{P} > P$. This concludes our proof for the case when $F_e \leqslant F_{\text{ref}}$.

Let us now consider $F_e > F_{\text{ref}}$. Then the performance $P$ is negative. It follows that one can expect to actually lose energy on a typical trip. Observe that $F_e + \chi \leqslant F_{\text{ref}} + \psi$. This can be shown by contradiction. Assume $F_e + \chi > F_{\text{ref}} + \psi$. Then there must be at least one $(o, d)$, $o \in E$, $d \in E$ pair for which $\hat{F}(\hat{p}_e(o, d)) > \hat{F}(p_{\text{ref}}(o, d))$. It comes from definition of model (4.8) that this can never be the case, which contradicts the assumption. Consequently, $F_e + \chi \leqslant F_{\text{ref}} + \psi$. However, in the same time $F_e > F_{\text{ref}}$. In other words

$$\frac{F_e}{F_{\text{ref}}} > \frac{F_e + \chi}{F_{\text{ref}} + \psi} \tag{4.22}$$

which yields $\hat{P} > P$: estimated performance $\hat{P}$ is necessarily higher than reference performance $P$ when $P$ is negative. □

The Theorem 1 shows that the estimated performance $\hat{P}$ is artificially high if its conditions are satisfied. These conditions are not unrealistic:

- Condition $\chi < \psi$: the average cost estimation error on eco-routes, $\chi$, must be less than average cost estimation error on the reference paths, $\psi$. There is no apparent reason why commonly used reference paths (shortest and fastest paths) should have lower than typical (in an average sense) cost estimation error. On the other hand, there is such a reason for $\chi$ as it is based on eco-routes: the cost $\hat{F}$ on eco-routes is minimized by the eco-routing model (4.8). Both actual path cost and the path cost estimation error are partially minimized as can be seen from Equation (4.7). Hence, it is reasonable to assume that this condition is easily satisfied. It is nevertheless possible to find a set of reference paths $\mathbb{P}_{\text{ref}}$ such that $\chi \geqslant \psi$. Such is for example the set that minimizes $\psi$, or the set $\mathbb{P}_{\text{ref}} \equiv \mathbb{P}_e$.

- Condition $\chi < 0$: As shown in the proof, $\hat{P} = P$ is satisfiable if $\chi$ is non-negative. Hence, the consumption (or pollutant emission) model $\hat{F}$ must not overestimate significantly for the Theorem 1 to apply. Slight tendency to overestimate can be tolerated (the $\psi$ can be positive). It is reasonable to assume that this condition is easily satisfied unless models that deliberately overestimate are considered.

- Condition $\psi \neq 0$: This is required in Equation (4.18) as it is not defined for $\psi = 0$. The likelihood of the average cost estimation error on reference paths $\psi$ being exactly zero is infinitesimal.

The studies that avoid these issues are by Ahn and Rakha, 2013 and Kubička et al., 2016b. Both study eco-routing in a microscopic simulation with the evaluation based on independent microscopic consumption models. Conducting such evaluation is technically demanding. It requires microscopic traffic simulation software, sufficiently-sized traffic scenario and the simulation itself is computationally heavy. For example, Kubička et al., 2016b report that their simulation took an equivalent of 559 hours of computation time. A technically easier way to estimate performance might be to use a macroscopic model for evaluation that produces errors uncorrelated with the model used in eco-routing. The performance estimate is likely to be less reliable, but it is free of the type of bias discussed above.

## 4.5 Summary

This chapter defines the eco-routing problem and the eco-routing performance. Two published eco-routing methods together with one idealized method are studied to establish a baseline to which the eco-routing method proposed in this work is later compared. Then, an argument is made against eco-routing evaluation methodology commonly used in the literature.

The eco-routing problem is defined in Section 4.1. It supports both constraints and multiple time-dependent costs. The definition is general enough to apply to all eco-routing variants in published literature and further allows variants that were not yet considered. For example, it allows imposing limits on pollutant emissions while minimizing vehicle consumption or to consider only paths that do not require battery recharging (resp. refueling) before reaching the destination. The model also allows to limit or completely disallow passing through certain areas at certain times. For example, routing close to schools at the time when the children are there can be disallowed for heavy polluting vehicles.

The Section 4.2 discusses eco-routing models used in published literature and formally defines eco-routing performance. It summarizes the ability of an eco-routing method to lower costs. The cost can be any of fuel consumption, energy consumption, or pollutant emissions. The reference path is the path to which the eco-route is compared to. In this work, the reference path is either the shortest path or the fastest path.

The performance of three eco-routing methods (two of them published; Barth et al., 2007; Juřík et al., 2014) is studied in Section 4.3. This is an updated version of work we published during the preparation of this thesis (Kubička et al., 2016b). An eco-routing framework that is presented in Chapter 7 is used to evaluate performance of these methods. The performance of the two published methods was negative when comparing the eco-routes to shortest paths. This implies that shortest path routing is superior eco-router to the methods by Barth et al., 2007 and Juřík et al., 2014.

While the evaluation of current eco-routing methods has shown negative savings, it is rare to see such a result reported in the published literature. A close look at the evaluation methodologies used in the literature was conducted in Section 4.4. It revealed that the prevalent method is not capable of observing negative savings by design. The problem is that the same consumption (or pollutant emission) model is used for both routing and evaluation. Not only that such evaluation will necessarily result in nonnegative savings, but it is also shown in Theorem 1 to be prone to result in inflated eco-routing performance figures.

# Chapter 5

# Energy consumption and travel time modeling

This chapter introduces a novel vehicle consumption model and a travel time model. The consumption model is derived from a standard estimation model and adapted for the needs of eco-routing applications. The standard formulation of this model is reviewed in Section 5.1. Then, the model is reformulated to a form suitable for eco-routing in Section 5.2. The model is solved in Section 5.3 for a special case of a vehicle with lossless powertrain and perfect recuperation. It is used in Section 5.4 to derive a closed-form solution for a realistic vehicle. A travel time predictor is proposed in Section 5.5. It is a simple model designed specifically to take into account known traffic light periods. Pollutant emission models are not discussed in this chapter. While they are essential for some eco-routing applications, they are discussed in this work only to the point of how they can be incorporated into the proposed eco-routing system.

The Section 5.1 is based mainly on Guzzella and Sciarretta, 2005. The content of sections 5.2 to 5.5 is original. We have published the closed-form solution of the longitudinal consumption for the lossless vehicle (the Section 5.3) during the preparation of this work, see Kubička et al., 2016a.

## 5.1 The longitudinal model of consumption

The longitudinal model is commonly used vehicle consumption estimation model (see Guzzella and Sciarretta, 2005). This model ignores vehicle heading and losses associated with it. It takes instantaneous vehicle speed and road slope as inputs to estimate instantaneous propelling power. It is derived from Newton's second law of motion. Consider $F_w$ to be the propelling force on wheels, $F_{\text{res}}$ the equivalent force with which the environment acts on the vehicle, $m$ vehicle mass, $v$ its instantaneous speed and $a$ its instantaneous acceleration. Then,

$$F_w - F_{\text{res}} = ma \tag{5.1}$$

The instantaneous propelling power $P_w$ and energy consumption $E_w$ during a trip of duration $T$ read

$$P_w = (ma + F_{\text{res}})v \tag{5.2}$$

$$E_w = \int_0^T P_w \mathrm{d}t \tag{5.3}$$

The $E_w$ is the energy spent on vehicle's propulsion, it does not account for its internal losses due to friction or heating, for example. A vehicle with no such losses is considered an "ideal vehicle" in this work. It is discussed in Section 5.3.

The force $F_{\text{res}}$ has various sources. The aerodynamic drag, rolling friction, road slope and other factors contribute to it. Note that $F_{\text{res}}$ can be negative in some circumstances, in which case it helps vehicle movement. It can be modeled analytically. The standard formulation is a second order polynomial of instantaneous speed plus losses dependent on road slope

$$F_{\text{res}} = c_a v^2 + c_b v + c_c + mg\sin(\alpha) \tag{5.4}$$

where $c_a$, $c_b$, $c_c$ are the *coasting coefficients*, the $g$ is the gravitational acceleration constant and the $\alpha$ is road slope (assumed constant). The climbing losses are expressed as a change in vehicle's potential energy. This is derived from Newton's laws of motion. The coasting coefficients can be functions of many variables, depending on how detailed the model is. For example, the aerodynamic drag depends on vehicle shape and air pressure. Rolling friction depends on road pavement, tire pressurization, precipitation and other factors. They are assumed constant in this work, as it is often done in the literature (Guzzella and Sciarretta, 2005). They can be identified experimentally by letting the vehicle slow down in coasting mode on a flat road (see Section 5.1.1).

The vehicle's energy consumption depends on how efficient the powertrain is. The energy on wheels $E_w$ does not account for internal losses in the powertrain (e.g., the energy dissipated as heat). Consider $P$ to be the power draw from vehicle's energy source and $\eta_t(t)$ to be the instantaneous *traction efficiency* defined as

$$P_w = \eta_t P \tag{5.5}$$

Then, $\eta_t \in [0,1]$ as long as both $P_w$ and $P$ remain positive and $P_w < P$ (which is always the case for a realistic powertrains). The trouble is that when the vehicle is in the braking mode the $P_w$ is negative as the power transfer is reversed: the vehicle receives power. The $\eta_t$ is not well defined in this situation. It can be either undefined or larger than one depending on powertrain's ability to recuperate. Consider then also a *recuperation efficiency* $\eta_r(t)$ to be the reciprocal of $\eta_t(t)$

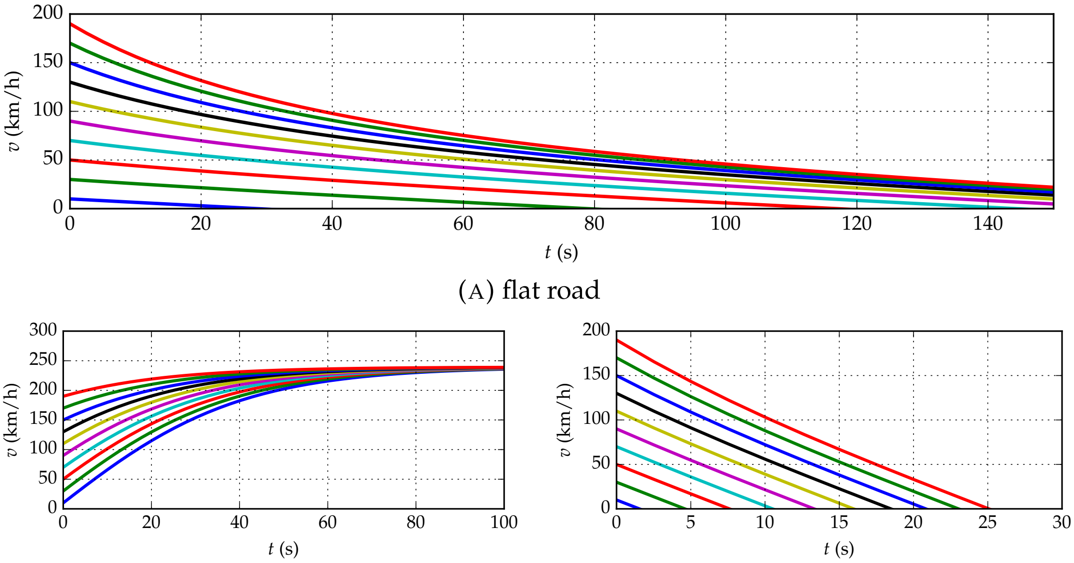
$$P = \eta_r P_w \tag{5.6}$$

The commonly used formulation of consumption splits the consumption to the times when the vehicle is in traction mode and the times when it is braking (Guzzella and Sciarretta, 2005)

$$E = \int\limits_{P_w>0} P_w \frac{1}{\eta_t} \mathrm{d}t + \int\limits_{P_w<0} P_w \eta_r \mathrm{d}t \tag{5.7}$$

This formulation has a clear physical meaning and is well defined for $(P, P_w) \in \mathbb{R}^2$ and $(\eta_t, \eta_r) \in [0,1]^2$. Notice that if the vehicle is unable to recuperate then this can be modeled with $\eta_r \equiv 0$. Also note that this form becomes equivalent to $E_w$ for $\eta_t \equiv 1$ and $\eta_r \equiv 1$.

There are two ways how (5.7) can be used for eco-routing. Either it is used directly on a synthesized shape of the speed profile $v(t)$, or reformulated in a closed form. The latter approach is preferred in this work as it exposes what properties of the speed profile are important and how they can be summarized. Indeed, the solution identified in Section 5.3 requires only initial and final speeds (to account for



(A) flat road

(B) 10-degree fall

(C) 10-degree climb

FIGURE 5.1: Coasting speed on different road slopes.

the change in vehicle's kinetic energy), initial and final altitudes (to account for the change in potential energy) and the first three central moments of the speed profile. Other parameters can be assumed to be known constants.

### 5.1.1 Traction versus braking mode

The threshold between the traction and braking modes depends on instantaneous speed and road slope. This section discusses in detail where this threshold is and what affects it. The vehicle is said to be in one of three modes: *traction*, *braking*, or *coasting*. Which mode the vehicle resides in depends on $P_w$. For $P_w > 0$ the vehicle is in the traction mode: it is giving away power. For $P_w < 0$ the vehicle is in braking mode: it is receiving power. The received power can be recuperated or turned to heat in friction brakes. For $P_w = 0$, the vehicle is in coasting mode. Typically, the vehicle slows down and reaches the full stop in a finite amount of time. However, it is also possible that the vehicle will accelerate due to gravitational pull if the road is sloped.

The coasting acceleration can be obtained by solving (5.1) for acceleration with $F_w = 0$ (which implies $P_w = 0$)

$$a_c = -\frac{F_{\text{res}}}{m} \tag{5.8}$$

where $a_c$ is the coasting acceleration. The equation is a nonlinear Ricatti differential equation. In its full form

$$m\frac{\mathrm{d}v_c}{\mathrm{d}t} = -c_a v_c^2 - c_b v_c - c_c + mg\sin(\alpha) \tag{5.9}$$

where $v_c$ is *coasting speed*. Assuming that the road slope $\alpha$ is constant the equation can be solved by direct integration for coasting speed $v_c > 0$

$$v_c(t) = \frac{1}{2a}\left[\tan\left(\arctan\left(\frac{b+2av_c(0)}{\Delta}\right)+\frac{\Delta}{2}t\right)\Delta - b\right] \tag{5.10}$$

where $a = -\frac{c_a}{m}$, $b = -\frac{c_b}{m}$, $c = -\frac{c_c + mg\sin(\alpha)}{m}$ and $\Delta = \sqrt{-b^2 + 4ac}$.

The solution (5.10) is plotted in Figure 5.1. It shows a vehicle in coasting mode for initial speeds from the set {10, 30, 50, . . . , 190} km/h on roads with various slopes. The vehicle-specific constants were taken from Table 7.2: $c_a = 0.4212$ N/(m/s)$^2$, $c_b = 0.774$ N/(m/s), $c_c = 113.5$ N, $m = 1190$ kg. Note that a vehicle going downhill (Figure 5.1b) will accelerate until an equilibrium state where $F_{\text{res}} = 0$ is reached.

## 5.2 The longitudinal model adapted for eco-routing

This section proposes a reformulation of (5.7) suitable for prediction. The (5.7) is a functional of vehicle speed $v(t)$ and road slope $\alpha(t)$. It is useful for consumption estimation, but not for prediction since the kind of information that is required is not available at the trip planning stage. The instantaneous efficiencies $\eta_t$ and $\eta_r$ are replaced with overall efficiencies $\overline{\eta}_t$ and $\overline{\eta}_r$ and taken out of the integrals. The integrals of instantaneous power are rewritten as energy on wheels $E_w$ and as braking energy $E_b$.

Let $E_b$ be *braking energy* defined as the total energy returned to the vehicle.

$$E_b = \int\limits_{P_w<0} P_w \mathrm{d}t \tag{5.11}$$

Let $\overline{\eta}_t$ be *overall traction efficiency* defined as the mean of the reciprocal of $\eta_t$ weighted with the power on wheels $P_w$

$$\overline{\eta}_t = \frac{\int\limits_{P_w>0} P_w \mathrm{d}t}{\int\limits_{P_w>0} P_w \frac{1}{\eta_t} \mathrm{d}t} \tag{5.12}$$

and let $\overline{\eta}_r$ be *overall recuperating efficiency* defined as the mean of $\eta_r$ weighted with the power on wheels $P_w$.

$$\overline{\eta}_r = \frac{\int\limits_{P_w<0} P_w \eta_r \mathrm{d}t}{\int\limits_{P_w<0} P_w \mathrm{d}t} \tag{5.13}$$

Notice that with these definitions the $\overline{\eta}_t$ and $\overline{\eta}_r$ can be substituted into (5.7) such that the instantaneous efficiencies $\eta_t$ and $\eta_r$ can be taken out of the integrands.

**Lemma 1.** *With given definitions of $E_b$, $\overline{\eta}_t$ and $\overline{\eta}_r$ the (5.7) can be rewritten as*

$$E = \frac{1}{\overline{\eta}_t} E_w + \left( \overline{\eta}_r - \frac{1}{\overline{\eta}_t} \right) E_b \tag{5.14}$$

*Proof.* With definitions of $\overline{\eta}_t$ in Equation (5.5) and $\overline{\eta}_r$ in Equation (5.6) the Equation (5.7) can be rewritten with overall instead of instantaneous efficiencies. This allows for their extraction out of the integrals.

$$E = \frac{1}{\overline{\eta}_t} \int\limits_{P_w>0} P_w \mathrm{d}t + \overline{\eta}_r \int\limits_{P_w<0} P_w \mathrm{d}t \tag{5.15}$$

With the definition of $E_w$ in Equation (5.3) and $E_b$ in (5.11) we can rewrite the left integral as $E_w - E_b$ and the right integral as $E_b$. The Equation (5.14) ensues. □

## 5.3 Ideal vehicle

By an ideal vehicle is considered a vehicle compliant to (5.7) with lossless powertrain ($\eta_t \equiv 1$) and perfect recuperation ($\eta_r \equiv 1$). Such vehicle has consumption equal to the energy on wheels $E_w$. All of the energy exchange between the vehicle and the environment is due to the power $F_w$. There are no losses due to heating, friction or on-board instrumentation.

**Theorem 2.** *The integral Equation (5.3) for $E_w$ can be solved as*

$$\begin{aligned} E_w = \frac{1}{2}mv^2(T) - \frac{1}{2}mv^2(0) + mg\Delta h + c_c D + \\ c_b\frac{D^2}{T} + c_a\frac{D^3}{T^2} + \sigma^2(c_b T + 3c_a D) + b\sigma^3 c_a T \end{aligned} \tag{5.16}$$

*where $v(T)$ is final speed, $v(0)$ is initial speed, $\Delta h$ is the difference between the initial and final altitude, $\sigma^2$ is speed profile variance, and $b$ is speed profile skew.*

*Proof.* Let us first fully expand (5.16) with (5.1), (5.2) and (5.4).

$$E_w = \int_0^T (mav + c_c v + c_b v^2 + c_a v^3 + mgv\sin(\alpha))\,\mathrm{d}t \tag{5.17}$$

Now we solve individual terms in the integrand separately. Observe that $\mathrm{d}v = a\mathrm{d}t$. This can be used to solve the first term as

$$\int_0^T mav\,\mathrm{d}t = \frac{1}{2}mv^2(T) - \frac{1}{2}mv^2(0) \tag{5.18}$$

which is vehicle's final kinetic energy minus vehicle's initial kinetic energy. Further, the term that accounts for climbing losses can be solved using $\sin(\alpha)\mathrm{d}s = \mathrm{d}h$ where $\mathrm{d}h$ is altitude differential as

$$\int_0^T mgv\sin(\alpha)\,\mathrm{d}t = mg\int_0^D \mathrm{d}h = mg\Delta h \tag{5.19}$$

which is the final potential energy minus initial potential energy. In order to solve the first-order speed dependent term we can use $\mathrm{d}s = v\mathrm{d}t$, where $\mathrm{d}s$ is a position differential

$$\int_0^T c_c v\,\mathrm{d}t = c_c D \tag{5.20}$$

The solution to remaining terms requires the introduction of further notation. Let us denote $\mu_i$ the $i$-th central moment of the speed $v(t)$ and $\mu'_i$ its $i$-th raw moment. First raw moment $\mu'_1$ of the speed profile is the mean speed, which is equal to traveled distance $D$ divided by travel time $T$. Second central moment $\mu_2$ is speed profile variance (denoted $\sigma^2$). Third central moment $\mu_3$ divided by $\sigma^3$ is Pearson's moment coefficient of skewness (denoted $b$)

$$\mu'_1 = \frac{D}{T} \tag{5.21}$$

$$\mu_2 = \sigma^2 \tag{5.22}$$

$$\mu_3 = b\sigma^3 \tag{5.23}$$

Higher raw moments can be expressed in terms of central moments using inverse binomial transformation (Papoulis and Pillai, 1985). Second and third raw moments are

$$\mu'_2 = \mu_2 + \mu'^2_1 \tag{5.24}$$

$$\mu'_3 = \mu_3 + 3\mu_2\mu'_1 + \mu'^3_1 \tag{5.25}$$

Note that $\mu'_2$ is the mean squared speed and $\mu'_3$ is the mean cubed speed. The terms with $v^2$ and $v^3$ in (5.17) can be solved using (5.24) and (5.25)

$$\int_0^T c_b v^2 \,\mathrm{d}t = c_b T(\mu_2 + \mu'^2_1) \tag{5.26}$$

$$\int_0^T c_a v^3 \,\mathrm{d}t = c_a T(\mu_3 + 3\mu'_1\mu_2 + \mu'^3_1) \tag{5.27}$$

By combining results in (5.18), (5.20), (5.19), (5.26) and (5.27) with (5.17) we obtain closed form formulation of the consumption model. We can further substitute the moment terms with travel distance $D$, travel time $T$, speed profile variance $\sigma^2$ and skewness coefficient $b$ using (5.21), (5.22) and (5.23). The Equation (5.16) ensues. □

The solution in Theorem 2 depends on five unknowns: initial speed $v(0)$, final speed $v(T)$, travel time $T$, speed profile variance $\sigma^2$ and its skew $b$. Parameters $\Delta h$, $D$ and the vehicle-specific coefficients $m$, $c_a$, $c_b$ and $c_c$ are supposed to be known constants. The initial and final speeds $v(0)$, $v(T)$ are zero if the vehicle starts and ends its trip in a stationary state.

Note the difference between $E_w$ in (5.3) and in (5.16). The classical formulation in (5.3) depends on speed and road slope, both are continuous functions of time. The formulation in (5.16) requires only basic statistics about the speed profile and a few other parameters while being equal to (5.3). The solution does not rely on any assumptions. Specifically, it does not make any assumption about the speed profile. The speed profile $v(t)$ is not necessarily stochastic (partially unknown; with noise term) even if some of the parameters are speed profile statistics. The solution is exact with respect to (5.3). The speed profile statistics used in the solution come from their formal definitions.

## 5.4 Willans vehicle

The Equation (5.14) puts in relation the ideal vehicle and realistic vehicles. A solution for braking energy $E_b$ is needed to solve (5.14). However, finding a solution for $E_b$ that would offer similar advantages as the solution to $E_w$ is difficult. A straightforward approach is to rewrite the braking energy as $E_b = \overline{P}_b T_b$, where $\overline{P}_b$ is mean braking power and $T_b$ the total time the vehicle spent in the braking mode. This approach is feasible, but not necessarily practical. Both the time spent in braking mode and mean braking power depends on the instantaneous travel speed, the driver, behavior of other drivers on the road, and traffic density.

A convenient solution might be to consider a specific powertrain model with which the energy consumption $E$ (Equation 5.14) does not depend on $E_b$. Such is the powertrain whose output power is an affine function of input power. Formally,

$$P_w = Pe - P_0 \tag{5.28}$$

where $e$ and $P_0$ are powertrain-specific parameters. A similar approach to powertrain modeling was proposed by Willans (Guzzella and Sciarretta, 2005) in a different context. It is a successful model that is sometimes used instead of nonlinear engine efficiency maps. There are differences with respect to Willans's proposal, however. Willans considers $e$ and $P_0$ to be functions of instantaneous motor shaft rotational velocity, but here they are assumed constant. It might be possible to consider it since the instantaneous motor shaft rotational velocity depends on instantaneous vehicle speed, but a multiple-gear transmission would make it difficult, especially if the vehicle has a manual transmission. Another difference is that the powertrain output power is considered to be the power on wheels $P_w$, while Willans allows for the power received during braking to be redirected to friction brakes (they are considered outside of the powertrain).

**Proposition 1.** *The Willans's powertrain model (5.28) simplifies the consumption model (5.14) to*

$$E = \frac{1}{e}E_w + \frac{P_0}{e}T \tag{5.29}$$

*Proof.* Note that (5.29) can be obtained by direct integration of (5.28). However, we also need to show that this solution is compliant with the consumption model (5.14). Let us first solve (5.28) for $P$ as a function of $P_w$ and divide both sides by $P_w$. We obtain

$$\frac{P}{P_w} = \frac{P_w + P_0}{P_w e} \tag{5.30}$$

where the left hand side is, by definition, the instantaneous recuperation efficiency $\eta_r$. The instantaneous traction efficiency $\eta_t$ is its reciprocal. If we plug these formulations of $\eta_t$ and $\eta_r$ to equations (5.12) and (5.13) for overall efficiencies $\overline{\eta}_t$, $\overline{\eta}_r$ we obtain

$$\overline{\eta}_t = \frac{(E_w - E_b)e}{(E_w - E_b) + P_0(T - T_b)} \tag{5.31}$$

$$\overline{\eta}_r = \frac{E_b + P_0 T_b}{E_b e} \tag{5.32}$$

where $T_b$ is the travel time spent in braking mode (when $P_w < 0$). Now we can plug the right-hand sides to (5.14) in the place of $\overline{\eta}_t$ and $\overline{\eta}_r$ and simplify. We obtain (5.29). □

Fully expanded solution (5.14) with powertrain model (5.28) reads

$$\begin{aligned} E = \frac{1}{e}\bigg(&\frac{1}{2}mv^2(T) - \frac{1}{2}mv^2(0) + mg\Delta h + c_c D + \\ &c_b\frac{D^2}{T} + c_a\frac{D^3}{T^2} + \sigma^2(c_b T + 3c_a D) + T(b\sigma^3 c_T + P_0)\bigg) \end{aligned} \tag{5.33}$$

Unlike the efficiencies $\eta_t$, $\eta_r$ in Equation (5.7), the efficiency $e$ in (5.28) is not well defined for $P = 0$. This is a problem especially for vehicles that cannot recuperate. When braking with such a vehicle the $P_w < 0$ while $P = 0$ which implies $e = -\infty$. The $e$ can be also greater than one, contrary to the intuition. The $e$ in the braking mode is the reciprocal of efficiency, this can be observed by comparing (5.7) and

(5.29). It follows that $e > 1$ during braking. Despite these issues the consumption (5.29) is always well defined. This is because $e$ is assumed to be a known constant.

### 5.4.1 More advanced powertrain model

There are more complex powertrain models than (5.28) that can be considered. For example, consider a powertrain model that accounts for speed dependent losses such as

$$P_w = Pe - P_0 - \beta_1 v - \beta_2 v^2 - \beta_3 v^3 - \beta_4 v^4 \tag{5.34}$$

where $v$ is instantaneous vehicle speed and $\beta_1 \dots \beta_4$ are regression coefficients. The solution to this model can be obtained analogically to the previous case. The overall efficiencies $\overline{\eta}_t$ and $\overline{\eta}_r$ read

$$\overline{\eta}_t = \frac{(E_w - E_b)e}{(E_w - E_b) + P_0(T - T_b) + \int\limits_{P_w \geq 0} (\beta_1 v + \beta_2 v^2 + \beta_3 v^3 + \beta_4 v^4)\mathrm{d}t} \tag{5.35}$$

$$\overline{\eta}_r = \frac{E_b + P_0 T_b + \int\limits_{P_w < 0} (\beta_1 v + \beta_2 v^2 + \beta_3 v^3 + \beta_4 v^4)\mathrm{d}t}{E_b e} \tag{5.36}$$

which can be plugged in (5.14) and simplified. The solution for energy consumption $E$ the reads.

$$\begin{aligned} E = \frac{E_w}{e} + \frac{D}{e}\left(\beta_1 + 3\beta_3 + \sigma^2 + 4\beta_4 b\sigma^3\right) + \\ \frac{T}{e}\left(P_0 + \beta_2\sigma^2 + \beta_3 b\sigma^3 + \beta_4\kappa\sigma^4\right) + \\ \frac{D^2}{eT}\left(\beta_2 + 6\beta_4\sigma^2\right) + \frac{D^3}{eT^2}\beta_3 + \frac{D^4}{eT^3}\beta_4 \end{aligned} \tag{5.37}$$

where $\kappa$ is a new parameter: speed profile kurtosis. It appears because (5.34) includes the fourth power of vehicle speed.

The powertrain model (5.34) might perspectively be used instead of (5.28). It is nevertheless just an example meant to show that more complex powertrain models can be considered. It is not studied further in this work.

## 5.5 Travel time

A travel time model is proposed in this section. As was discussed in the introduction a travel time model is required for time-dependent eco-routing applications. The model proposed here is a simple travel time predictor that makes use of known traffic light periods in the simulations. It predicts travel time on a given road in the road network at a given time of arrival there.

The model proposed below is based on assumptions that are not backed by evidence: it is a conjecture. The main assumption is that one of the three following cases must occur:

1. the vehicle will pass without stopping
2. the vehicle will be forced to a full-stop on a red traffic light
3. the vehicle will get caught in a traffic jam

The second case is introduced to the model because the travel time is difficult to predict by simple averaging if there is a traffic light at a downstream intersection. In that case, the waiting time depends on the phase of the traffic light at the time when the vehicle arrives at it. If the road is not congested, then the vehicle will leave the road within the next green period. If the road is congested, then the vehicle will not be able to leave the next period. The third case is introduced to account for this. The travel time does not depend so strongly on the traffic light state if there is a queue so long that the vehicle won't be able to leave the road during the first green period.

The model is described in Proposition 2. It is based on a $k$-NN ($k$ nearest neighbors) predictor. The $k$-NN is used for regression, not classification. It uses a set with a collection of previously observed travel times stored as pairs $(t_i, \tau_i)$, where $t_i$ is $i$-th time of day when the observation was made, and $\tau_i$ is $i$-th observed travel time. The query for travel time at some time of day $t_q$ that is not in the set is predicted as the average of $k$ closest historical observations to the time of day $t_q$. This is the principle of the $k$-NN regression.

**Proposition 2.** *Let $\mathbb{S}_e$ be a set of pairs $(t_i, \tau_i)$ of observed travel times $\tau_i$ at times $t_i$ on a road $e$. Let $\mathbb{G}_e$ be a subset of $\mathbb{S}_e$ with those observations that originated from vehicles that did not stop on the road $e$. Let $\mu \in \mathbb{R}$, $\kappa_{\mathbb{S}} \in \mathbb{N}^+$, $\kappa_{\mathbb{G}} \in \mathbb{N}^+$, and $\vartheta \in \mathbb{R}$ be known coefficients. Let $\overline{\tau}_{\mathbb{S}}(t)$ be a k-NN predictor trained on the set $\mathbb{S}$, with $k = \kappa_{\mathbb{S}}$. Let $\overline{\tau}_{\mathbb{G}}(t)$ be a k-NN predictor trained on the set $\mathbb{G}$ with $k = \kappa_{\mathbb{G}}$. Let $t_{tg}(t)$ be the remaining time to green on a downstream intersection, if there is a red traffic light at time t. If there is no traffic light or the traffic light is green at time t, let $t_{tg}(t) = 0$. The travel time $\tau_e(t)$, where t is time of arrival to road e, is then*

$$\tau_e(t) = \begin{cases} \overline{\tau}_{\mathbb{S}}(t) & \text{if } \overline{\tau}_{\mathbb{S}}(t) \geqslant \vartheta \\ \overline{\tau}_{\mathbb{G}}(t) & \text{if } t_{tg}(t + \overline{\tau}_{\mathbb{G}}(t)) = 0 \text{ and } \overline{\tau}_{\mathbb{S}}(t) < \vartheta \\ \overline{\tau}_{\mathbb{G}}(t) + \mu + t_{tg}(t + \overline{\tau}_{\mathbb{G}}(t)) & \text{otherwise} \end{cases} \tag{5.38}$$

Every road in the road network has its own travel time model with a specific set of coefficients $\vartheta$, $\mu$, $\kappa_{\mathbb{G}}$, $\kappa_{\mathbb{S}}$ and sets $\mathbb{S}_e$, $\mathbb{G}_e$. The roads are implicitly assumed to have a traffic light. If that is not the case, then the part that considers traffic lights can be disabled by considering $\vartheta = 0$. The parameter $\vartheta$ is a threshold on travel time above which the road is considered congested. The parameter $\mu$ is intended as a correction for bias when the vehicle is forced to stop on a red traffic light. The parameters $\kappa_{\mathbb{S}}$, $\kappa_{\mathbb{G}}$ are search radii of the two $k$-NN regressors. The three lines correspond to the three cases enumerated above:

1. *Case 1: the vehicle will pass without stopping.* (line 2 in the Equation (5.38)) If the road is not congested ($\overline{\tau}_{\mathbb{S}}(t) < \vartheta$) and the predicted state of the traffic light when arriving at it is green, then the predicted travel time is the average of previously observed travel times when the vehicles passed without stopping at a similar time of day.

2. *Case 2: the vehicle will be forced to a full stop on a red traffic light.* (line 3 in the Equation (5.38)) If the road is not congested ($\overline{\tau}_{\mathbb{S}}(t) < \vartheta$) and the predicted state of the traffic light when arriving at it is red, then the predicted travel time is the average of previously observed travel times when the vehicles passed without stopping at a similar time of day plus predicted waiting time plus a small constant $\mu$ that accounts for delays during slow-down and subsequent acceleration.

3. *Case 3: the vehicle will get caught in congestion.* (line 1 in the Equation (5.38)) The road is considered congested when $\overline{\tau}_{\mathbb{S}}(t) \geqslant \vartheta$. The predicted travel time is then the average of previously observed travel times at a similar time of day.

This model is designed for roads that lead to signalized intersections. If the intersection in question is not signalized, then setting $\vartheta = 0$ will disable the part of the model that reacts to traffic light states. The parameters $\mu$ and $\kappa_{\mathbb{G}}$ will then be irrelevant. The only parameter that would remain relevant is $\kappa_{\mathbb{S}}$.

Validation of this model is discussed in Section 8.2. A specific set of parameters is identified for each road in the road network. This results in a large number of unique models. Each model is cross-validated to obtain a robust estimate of the prediction error.

## 5.6 Summary

The consumption and travel time models are introduced in this chapter. Its scope is limited to their derivation and description. The models are later identified and validated in Chapter 8, and used for eco-routing in Chapter 9.

The standard formulation of the longitudinal consumption model is reviewed in Section 5.1. The model is given in Equation (5.7). It is derived from Newton's second law of motion, under an assumption that the vehicle movement is limited to one direction. It estimates the instantaneous power required by the powertrain and integrates it to obtain the overall energy consumption. The instantaneous power is based, among other things, on vehicle's instantaneous speed. Hence the speed profile $v(t)$ must be provided to the model to estimate consumption.

The Section 5.2 introduces a closed-form reformulation of the standard longitudinal model (5.7). It is argued in this section that (5.7) is not well suited for eco-routing as it requires information that is typically not available when planning the trip. This motivates the closed-form reformulation of the longitudinal model. It is given in equation (5.14). It relies on energy consumption of an idealized vehicle (the energy on wheels, $E_w$), on energy that was returned to the vehicle (the braking energy, $E_b$) and on overall powertrain efficiencies (overall traction efficiency $\overline{\eta}_t$ and overall recuperation efficiency $\overline{\eta}_r$).

The solution to (5.14) for an idealized vehicle is given in Section 5.3. The ideal vehicle is a vehicle with a lossless powertrain (there are no losses due to heating and friction, for example) and an ability to recuperate all energy returned to the vehicle when braking. Energy consumption of such vehicle is introduced as the energy on wheels and denoted $E_w$. The solution is an algebraic equation that depends on statistics of $v(t)$ (its first three moments), initial speed, final speed, initial altitude, final altitude and other (known) parameters. The initial and final speed parameters account for the change in vehicle's kinetic energy. The initial and final altitude parameters account for the change in vehicle's potential energy. The strength of this solution is twofold. Firstly, it manages to reduce the dimensionality of the input from arbitrary-dimension time series to a closed set of parameters. Secondly, it bridges the gap between classical consumption estimation model (based on known speed profiles) and the needs of a prediction model (based on statistical information).

The limitation of the solution given in Section 5.3 is that it applies only to the ideal vehicle. More realistic powertrains are considered in Section 5.4. It is argued that while there is a convenient solution for the energy on wheels $E_w$, finding a similar solution for $E_b$ is difficult. A powertrain efficiency model whose output power is an affine function of its input power is proposed as it enables a solution to energy consumption $E$ that does not depend on $E_b$.

A time-dependent travel time model is proposed in Section 5.5. It is a simple model designed to leverage the knowledge of traffic light states in the used simulation framework (presented in Chapter 7). It was incorporated here because such information is likely to be available in future smart cities. The travel time model is evaluated in Chapter 8 and used in Chapter 9 when time-dependent eco-routing is studied.

# Chapter 6

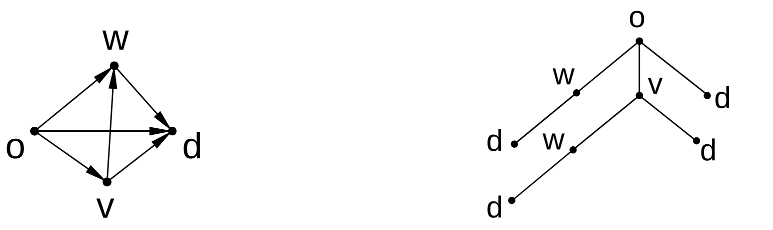
# Routing

As evidenced by the literature review in Chapter 3 the eco-routing has been treated mostly as a minimum path problem. This approach offers an intuitive formulation of eco-routing, and moreover, efficient methods for its computation are available (e.g., Dijkstra's algorithm and its derivates). The main strength of these algorithms is that they are fast and optimal. However, an evaluation of three eco-routing methods in Chapter 4 have shown poor results. Two have shown inferior performance with respect to the shortest path router. The third studied method has shown mild savings but is likely to be too cost-prohibitive for a successful real-world deployment. Main reasons for this performance regression are discussed in the introduction. In summary:

- Current methods often assume that the energy consumption, pollutant emissions and travel time do not change in time. While this assumption is known to be false, it seems essential to sustain computational tractability.

- Current methods ignore uncertainty in predicted energy consumption, pollutant emissions and travel time. The situation in the road network is continually evolving and predicting its state in future is arguably difficult. For this reason, an eco-routing method should be able to adapt to the evolving situation by updating its solution whenever necessary.

- Most published methods do not offer any prospect for routing in road network graphs with negative costs (an exception is De Nunzio et al., 2016). Negative costs may arise when routing for a vehicle that can recuperate energy. The energy consumption can be negative when the vehicle goes downhill or on short roads when the vehicle slows down. Dijkstra's algorithm, a commonly used algorithm in eco-routing context, assumes the costs are nonnegative. The Bellman-Ford (Bellman, 1958) algorithm can be used instead to route on graphs with negative costs. Both Dijkstra's method and Bellman-Ford method can be considered fast since their runtime is polynomially bounded. However, while Dijkstra's method visits only those nodes that can potentially be on the shortest path, Bellman-Ford operates on every node in the graph. Hence the computational effort associated with Dijkstra's method is dominated by trip properties while the road network size dominates the computational effort associated with the Bellman-Ford algorithm. This property makes routing with the Bellman-Ford algorithm on large road network graphs impractical. Another problem with routing in a graph with negative costs is that negative cycles can appear due to uncertainties in the routing costs: there might be cycles in the road network graph whose sum of costs is negative. Routing in such graphs is not possible, as the vehicle can gain any amount of energy simply by running along these cycles sufficiently many times.



(A) the road network graph $G$

(B) corresponding $\mathcal{T}_{od}$ tree

FIGURE 6.1: A $\mathcal{T}_{od}$ tree example.

The method proposed in this chapter favors simplistic routing based on a list of all possible paths over sophisticated optimal routing algorithms such as Dijkstra's or Bellman-Ford algorithm. The motivation for this is that these algorithms are too restrictive, too rigid in their assumptions. It is argued in the introduction that finding optimal paths gives a little advantage when there is sizable uncertainty in the associated costs that are to be minimized. An approximative solution based on a naive approach to routing is studied instead. It is based on a hypothesis that while there can be many paths to consider, there is usually only a limited set of candidate eco-routes. Then a sufficiently small tree can be identified, and powerful eco-routing ensues.

## 6.1 Routing in path trees

Let us consider a naive approach to routing. As a reminder, $\mathbb{P}$ was introduced in Chapter 4 as a set of all simple paths in the road network graph $G = (V, E)$. Consider a subset of $\mathbb{P}$ such that it contains only paths that start at some chosen origin $o$ and end at some chosen destination $d$. The key notion required to understand the content of this section is that paths in this subset can be stored efficiently in a tree[1] since all paths in this subset share the same origin. An algorithm that solves the eco-routing problem (4.4) posed in Chapter 4 using such a tree is proposed below. The discussion about practical considerations about its computational tractability is postponed to later sections.

Let a tree $\mathcal{T}_{od} = (V', E')$ be a connected graph without cycles that contains a collection of roads that start at $o \in V$ and end at $d \in V$ (hence the suffix "od"). Each node $n' \in V'$ refers to a node $n \in V$. Let us define this relationship with a surjective function $\text{REF}(\cdot)$ that constitutes a mapping from the set $V'$ to the set $V$. Let also all edges $e' = (n'_1, n'_2) \in E'$ satisfy condition $(\text{REF}(n'_1), \text{REF}(n'_2)) \in E$. This ensures that the tree $\mathcal{T}_{od}$ contains only paths contiguous in $G$. Finally, let the root of the tree refer to the origin $o$ and all leafs refer to the destination $d$, formally let $\text{REF}(n_r) = o$ for the tree root $n_r$ and $\text{REF}(n_l) = d$ for all $n_l$ in the set of all leafs of the tree.

To illustrate the relationship between the road network graph $G$ and the tree $\mathcal{T}_{od}$ see the example given in Figure 6.1. The Figure 6.1a shows the graph $G$ with $V = \{o, d, v, w\}$ and $E = \{(o, w), (o, d), (o, v), (w, d), (v, w), (v, d)\}$. This graph contains four paths $\{(o, w, d), (o, v, w, d), (o, d), (o, v, d)\}$ from $o$ to $d$. The Figure 6.1b shows the corresponding tree with the four paths.

Let us assume that the tree $\mathcal{T}_{od}$ is *full*: it contains all simple paths between the origin $o$ and destination $d$. The solution to the eco-routing problem (4.4) posed in Chapter 4 can then be identified using Algorithm 1. It is an exhaustive search in

[1] tree: a connected graph without cycles

**Algorithm 1** $p_e(\mathcal{T}_{od}, t_d)$

```
if $||V'|| = 0$ then
    return $\emptyset$
end if
min $\leftarrow \infty$
argmin $\leftarrow \emptyset$
$o \leftarrow$ root node of $\mathcal{T}_{od}$
$F[o] \leftarrow 0$
$t[o] \leftarrow t_d$
$q \leftarrow$ new QUEUE()
$q$.ENQUEUE($o$)
while $q$ not empty do
    $u \leftarrow q$.DEQUEUE()
    for all child nodes $v$ of $u$ do
        $e \leftarrow (\text{REF}(u), \text{REF}(v))$
        $t[v] \leftarrow t[u] + \tau_e(t[u])$
        $F[v] \leftarrow F[u] + c_e(t[u])$
        if $C - F[v] \geqslant 0$ then
            if $v$ is not a leaf node of $\mathcal{T}_{od}$ then
                q.ENQUEUE($v$)
            else if $\omega^T F[v] <$ min then
                min $\leftarrow \omega^T F[v]$
                argmin $\leftarrow v$
            end if
        end if
    end for
end while
return argmin
```

the tree for the path from the root to a leaf that minimizes the weighted cost $\omega^T F$ while satisfying the constraints. The variables "min" and "argmin" keep the partial solution during the computation: "argmin" contains the minimal path found insofar and "min" contains the cost associated with this path. The $F$ is an array of costs from origin to every node in $G$ (defined in Section 4.1). The array $t$ contains times of arrival at the nodes in the tree. The $q$ represents a queue. This version of the algorithm uses a standard queue with operations ENQUEUE and DEQUEUE. A stack can be used instead of a queue. It would have the advantage that it can return partial results while still computing. The algorithm returns a leaf node of the tree. This output is sufficient to identify the path $p_e$. The path can be extracted from the tree by listing the sequence of nodes from tree root to the returned leaf node. This sequence is unique and can be retrieved in reversed order by walking backward from the returned node up to the origin.

**Theorem 3.** *(proof of correctness) Algorithm 1 solves the eco-routing problem (4.4) for a set of paths $\mathbb{P}$ represented by a tree $\mathcal{T}_{od} = (V', E')$ at a departure time $t_d$.*

*Proof.* By induction on the tree $\mathcal{T}_{od}$ we prove that:

(a) For every reachable $v \in V'$ the $t[v]$ is predicted time of arrival at $v$ and $F[v]$ is predicted cost for the path from the tree root to node $v$.

(b) The algorithm finds a path with minimal cost that satisfies given constraints. If such path does not exist, then the algorithm returns an empty set.

For $||V'|| = 0$ (the $|| \cdot ||$ denotes set cardinality) we have an empty tree and the algorithm returns immediately. For $||V'|| = 1$ we have a base case when there is no edge. The algorithm returns immediately as there is no path in the tree, with the $F[o]$ set to zero and $t[o]$ set to $t_d$, for $o \in V'$. Hence the base case satisfies both (a) and (b).

The inductive step, embodied by line 19, adds the node $v$ to the queue, which guarantees that it will be processed later. Since there are no cycles in $\mathcal{T}_{od}$, the induction step is invoked strictly on the children of the currently processed node. It follows that each node will be processed at most once. Both the time of arrival at $v$ (the $t[v]$) and the vector of costs of a path from the tree root to $v$ (the $F[v]$) are computed before the $v$ is considered to be added to the queue. Since there are no cycles in the tree, we can predict the time of arrival at $v$ by adding the time of arrival at $u$ with the travel time from $u$ to $v$. This step is done on line 15. Additionally, because the costs in the vector $F$ are additive (Equation (4.1) imposes this), we can predict the costs of traveling from the root to $v$ (the $F[v]$) by adding the costs $F[u]$ of the parent node $u$ with the costs for travel from $u$ to $v$. This step is done on line 16. By induction hypothesis, the $F[v]$ and $t[v]$ are correct if $F[u]$ and $t[u]$ are correct. This was shown in the base case, hence (a) holds.

It remains to show that (b) holds in the induction step. Line 17 requires current node $v$ to satisfy time and energy constraints to be enqueued for later processing. If the constraints imposed by $C$ are not satisfied, then the node will be ignored. The constraint that imposes that the identified path starts at the origin $o$ and end in the destination $d$ is satisfied implicitly by the definition of the $\mathcal{T}_{od}$ tree. It follows that no path that does not respect the constraints can ever be returned as a solution. If the current node $v$ satisfies the constraints and if it is a leaf node (tested on line 18), then we check the weighted cost $\omega^T F[v]$. If it is smaller than the cost of any previously discovered path, then we assign this node to the "argmin" variable. This technique guarantees that the minimum cost path that satisfies the constraints is identified after searching the whole tree. An empty set is returned if no path that would meet the constraints was found. Hence, (b) holds as well. This concludes the proof.

□

Assuming the Algorithm 1 is computationally tractable (which is a concern yet to be addressed), it solves the problem (4.4) without suffering from any of the tree drawbacks discussed above. The first one is that current methods often assume that the energy consumption, pollutant emissions and travel time do not change in time. Note how time is treated in the Algorithm 1. Predicted arrival time at each node is stored in the array $t$ and used to predict the costs $F$. The time discretization is not needed at all. This method supports arbitrarily fast changes of road network states. It can, in particular, find paths that minimize the consumption cost imposed by stops on red traffic lights. The second drawback is that current methods ignore uncertainty in predicted energy consumption, pollutant emissions and travel time. This problem can be addressed by running the Algorithm 1 whenever a new information that might change the solution becomes available. The advantage of the Algorithm 1 is that it can be run on a subtree rooted at the node that corresponds to the current position of the vehicle. This reduces the computational effort required to update the solution as the vehicle progresses to its destination. The third drawback is that most published methods do not offer any prospect for routing in road network graphs with negative costs. The negative costs are not a problem when routing in path trees using the Algorithm 1, as it assigns a cost to every path in the tree separately.

While the proposed approach resolves the drawbacks of the current methods, other problems arise. Specifically, the computational effort associated with the building of the tree is prohibitive: a full tree can be computed in a reasonable time only for unrealistically small road network graphs. Methods to build such a tree are studied in following sections.

## 6.2 Computing path trees using an exhaustive search

Computing a full $\mathcal{T}_{od}$ tree is tractable only on small road network graphs $G$. In the case when $G$ is a complete graph there is

$$\sum_{k=0}^{||V||-2} \binom{||V||-2}{k} k! \tag{6.1}$$

paths between any pair of two nodes ($||\cdot||$ denotes set cardinality). The situation is not as bad in practice as road network graphs tend to be sparse, but the superpolynomial rate of growth stays an issue. The disproportion of the size between the $\mathcal{T}_{od}$ tree and the road network graph is apparent even in the example in Figure 6.1: while the $G$ has only four nodes the $\mathcal{T}_{od}$ tree has eight nodes.

A method that performs an exhaustive search to compute the $\mathcal{T}_{od}$ tree is given in Algorithm 2. The first three lines initialize an empty queue (line 1), create a tree with a single node (line 2) and add this node to the queue (line 3). The root of the tree refers to the origin $o$. The nodes in the queue are processed one by one in the loop on lines 4 to 15. The node is taken out of the queue and assigned to $u$ on line 5. A path in the road network graph $G$ that corresponds to the path from the tree root to $u$ is then assigned to $p$ on line 6 using a helper function named EXTRACT_PATH. Possible expansion of the tree is then considered. The connectivity of the roads in the road network is examined by listing nodes connected to $u$. For every connecting road $v$ the algorithm first checks if the path $p$ remains simple if it is extended with $v$. This check is implemented on line 8 by verifying that $v \notin p$. A new node is added to the tree if the condition is met. This is implemented by line 9. It adds a child node $v'$ to node $u$ and sets $\text{REF}(v') = v$. The argument $v$ is provided to ADD_CHILD to set the reference. The $v'$ is put in the queue if the node $v$ is not the destination $d$ (checked on line 10). After the main loop on lines 4 to 15 is finished the tree might contain paths that do not end in the destination $d$. They are removed with the routine CLEANUP on line 16.

The computational effort required to obtain a solution with the Algorithm 2 is extensive. The $O((||V||-2)!)$ asymptotically bounds its runtime and space complexity. To illustrate this consider an example: let there be an origin $o$, destination $d$ and some intermediate node $x$. If there is $n$ paths from $o$ to $x$ then all paths from $x$ to $d$ are repeated $n$ times as $n$ subtrees at different parts of the tree. It is this behavior that leads to the factorial in the computational complexity given above. Nevertheless, the given asymptotical bound assumes the graph is complete, which is typically not the case with road networks graphs. The sparseness reduces the complexity. Also, the interesting paths are only those that can be eco-routes: many paths can never be eco-routes and can be pruned. A number of pruning techniques to reduce the search space can be adopted in the context of eco-routing. Two techniques are discussed below: pruning based on travel time constraints and pruning based on path dominance.

**Algorithm 2** $\mathcal{T}_{od}(G, o, d)$

```
$q \leftarrow$ QUEUE()
$\mathcal{T}_{od} \leftarrow$ TREE(root = $o$)
$q$.ENQUEUE(root node of $\mathcal{T}_{od}$)
while $q$ not empty do
    $u \leftarrow$ q.DEQUEUE()
    $p \leftarrow$ EXTRACT_PATH($u$)
    for all child $v$ of REF($u$) do
        if $v \notin p$ then
            $v' \leftarrow u$.ADD_CHILD($v$)
            if $v \neq d$ then
                $q$.ENQUEUE($v'$)
            end if
        end if
    end for
end while
CLEANUP($\mathcal{T}_{od}$)
return $\mathcal{T}_{od}$
```

### 6.2.1 Time-constraint based pruning

A time constraint can be leveraged to reduce the search space of the Algorithm 2. This idea was first considered in eco-routing context by Juřík et al., 2014. The authors posed the eco-routing problem as a trade-off between two resources: energy consumption and travel time. They proposed an algorithm that solves the time-constrained optimal eco-routing problem. The time constraint based pruning is one of the techniques used by the authors. It can be adopted for path tree computation.

The road network graph can be reduced to a subgraph with only those nodes that are reachable under given time constraint. Let us denote $t_c \in \mathbb{R}$ the limit on maximum travel time (the time constraint), origin $o \in V$, destination $d \in V$ and let $\tau_{\text{min}}(m, n)$ where $m, n \in V$ be minimum travel time from $m$ to $n$. The pruning can be done by taking an induced subgraph of $G$: a subgraph with a subset of the nodes of $G$ together with any edges whose endpoints are both in this subset. The subset is defined as

$$\{n \in V : \tau_{\text{min}}(o, n) + \tau_{\text{min}}(n, d) \leqslant t_c\} \tag{6.2}$$

While this reduces the graph size, it does not necessarily remove all paths which do not satisfy the time constraint. Consider a path $p$ from origin $o$ to $n \in V$ and let $\tau'_{\text{min}}(p)$ be the low-bound on travel time on the path $p$ This path can constitute part of the solution if and only if

$$\tau'_{\text{min}}(p) + \tau_{\text{min}}(n, d) \leqslant t_c, \tag{6.3}$$

which leads to another, more effective way to prune travel times: when only paths that satisfy the condition (6.3) are considered. The difference between $\tau'_{\text{min}}(p)$ and $\tau_{\text{min}}(o, n)$ is that the former is a minimum travel time on a specific path while the latter is a minimum travel time on the path that minimizes travel time itself. It follows that $\tau_{\text{min}}(o, n) \leqslant \tau'_{\text{min}}(p)$, which explains the claim above that pruning specific paths is more effective than reducing the graph.

The first method (reducing the graph) can be a preprocessing step: the Algorithm 2 can be run on the induced subgraph directly. In the latter case, the method must be implemented inside the Algorithm 2: the condition on line 8 must additionally verify that (6.3) is met.

The time constraint based pruning can be effective when a stringent time constraint is imposed but will have little effect on the size of the search space if the constraint is lenient. Nevertheless, it can be used to remove overly long paths from the search space even when time-constraints are not considered. This comes at the expense of losing optimality when using the resulting $\mathcal{T}_{od}$ tree for routing.

### 6.2.2 Path dominance based pruning

Consider two distinct paths $p_1$, $p_2$ in $G$ from origin to some node $n \in V$. In some cases, one path can always have a lower cost than the other. Formally,

$$\forall t \in \mathbb{R} : \omega^T F(p_1, t) \leqslant \omega^T F(p_2, t) \tag{6.4}$$

In this case the path $p_1$ dominates the path $p_2$ in the sense that the routing algorithm would always choose the path $p_1$ over $p_2$ (path $p_2$ is said to be dominated by $p_1$). Moreover, all paths from the origin to the destination that pass through the node $n$ and take the subpath $p_1$ necessarily dominate all such paths that take the subpath $p_2$ to $n$. This is because the path cost $F$ is additive (implied by (4.1) in Section 4.1). This gives rise to a pruning technique that aims to reduce the search space by removing dominated paths from the solution.

The derivation of this criterion depends on how exactly is the eco-routing problem posed. Consider, as an example, that the routing cost is the energy consumption alone. The general definition of energy $E = \int P \mathrm{d}t$ can be rewritten as $E = \overline{P}T$, where $\overline{P}$ is the average power and $T$ is the travel time. Expanding the (6.4) with it yields a simple criterion:

$$\frac{\overline{P}_1}{\overline{P}_2} \leqslant \frac{T_2}{T_1} \tag{6.5}$$

where $\overline{P}_2 > 0$ and $\overline{T}_1 > 0$. The $\overline{P}_1$ and $\overline{P}_2$ are the average powers drawn by the powertrain on paths $p_1$ and $p_2$. Similarly, the $T_1$ and $T_2$ are travel times on the two paths. None of these quantities are known, but their bounds can be used to derive the criterion. Let $\tau'_{\mathrm{min}}(p)$ be travel time low-bound on a path $p$ (like in the previous section) and let $P_{\mathrm{max}}$ and $P_{\mathrm{min}}$ be bounds on powertrain's power draw. Then

$$\frac{P_{\mathrm{max}}}{P_{\mathrm{min}}} \leqslant \frac{\tau_{\mathrm{min}}(p_2)}{\tau_{\mathrm{min}}(p_1)} \implies \forall t \in \mathbb{R} : \omega^T F(p_1, t) \leqslant \omega^T F(p_2, t) \tag{6.6}$$

which is a condition easily verifiable with information about the powertrain and about speed limits in the road network (speed limits are used to construct the travel time low-bound).

Note that the applicability of this criterion is conditioned to the case when $P_{\mathrm{min}} > 0$. The condition is met by conventional vehicles with internal combustion engines and no ability to recuperate, but not by vehicles that can recuperate. In that case, it is theoretically possible to obtain $\overline{P} \leqslant 0$ which implies $P_{\mathrm{min}} < 0$. While the trips that leave the vehicle with more energy than what it had when started are rare (and typically short, downhill), the possibility has to be considered if the goal is to build a $\mathcal{T}_{od}$ tree with a guarantee that no candidate eco-routes are missed. The solution for the case $\overline{P} \leqslant 0$ is the same, except for the relation sign in the premise for the implication in (6.6) that must be inverted.

The strength of this pruning criterion depends on how the $P_{\text{min}}$ and $P_{\text{max}}$ parameters are chosen. It is not likely to reduce the search space sufficiently to make Algorithm 2 computationally tractable in practice. Vehicles can draw a wide range of powers from their power source. Further, the criterion makes use of average power draw meaning that the chosen boundaries are likely to be far from typically observed mean powers drawn by the powertrain. Nevertheless, if the goal is to approximate the tree, then this approach has advantages. Mainly, the ration $P_{\text{max}}/P_{\text{min}}$ can be replaced with a parameter that allows striking the trade-off between pruning strength and the computational effort required to compute the approximated solution. It can be derived, for example, from a distribution of mean powertrain power draw such that the range between $P_{\text{min}}$ and $P_{\text{max}}$ covers 98% of observed average powertrain power draws (by choosing 99th and 1st percentiles as $P_{\text{max}}$ and $P_{\text{min}}$, respectively).

Incorporation of such pruning in the Algorithm 2 is more involved than with pruning by time-constraint. The algorithm must be adapted upon discovery of a new subpath (when a node is added to the tree; on line 9). Let the new subpath end in some node $v \in V$ (note: this notation corresponds to the notation used in the algorithm). Then all other subpaths found insofar from the origin $o$ to the node $v$ must be checked for dominance with the newly found subpath. If the newly found subpath is dominated by some other path, then the node $v'$ (as denoted in Algorithm 2) should be removed from the tree. If some other, previously found subpath is found to be dominated by the newly found subpath, then all paths spanning from the dominated subpath (its whole subtree) must be moved under the newly added subpath, and the dominated subpath must be removed from the tree. This requires support for operations that can remove nodes, and that can move subtrees. This makes an efficient implementation of the tree datastructure more involved.

## 6.3 Computing path trees using a randomized search

Computing a path tree with an exhaustive search remains a computationally intensive task even when the pruning techniques are used to reduce the search space. The resulting trees tend to be large and filled with paths that are unlikely candidates to be eco-routes. A different approach is discussed in this section: an approach that aims to discover a small[2] set of the most likely eco-routes. This is in contrast with the previous approach that adds to the tree every path it finds for which it cannot prove that it can never be an eco-route. The resulting path trees are small, which makes the routing on them cheap in terms of computational effort. It also makes the routing suboptimal as only a subset of the search space is considered. As was argued in the introduction, routing optimality has low added value in eco-routing context due to sizable uncertainties in path costs.

One way to guess the most likely eco-routes between some origin and destination is to identify the $k$ minimal paths (using $k$-Dijkstra's algorithm, for example), for some positive integer $k$. This approach identifies a set of $k$ paths between some origin and destination with lowest costs. They can be stored in the $\mathcal{T}_{od}$ path tree and used for routing. However, this approach is not likely to identify such a set of paths that would result in a successful eco-routing. The reason is that the $k$ minimal paths tend to be similar. Consider two specific paths: the minimal path and the second minimal path. The second minimal path is likely to be the first minimal path with a small perturbation since such two paths are likely to have similarly low

[2] Small in comparison to the total number of simple paths there can typically be. The tree can still hold hundreds of thousands of paths.

**Algorithm 3** $\mathcal{T}_{od}(G, \mathcal{X}, o, d)$

```
$\mathcal{T}_{od} \leftarrow$ TREE()
while not STOP_CONDITION($\mathcal{T}_{od}$) do
    for all $e$ in $E$ do
        $c_e[e] \leftarrow \mathcal{X}_e$
    end for
    $r \leftarrow$ DIJKSTRA$(G, c_e, o, d)$ b
    if $r \notin \mathcal{T}_{od}$ then
        $\mathcal{T}_{od}$.ADD$(r)$
    end if
end while
```

costs. This often leads to a set of highly overlapping paths. If the tree is made from such overlapping paths and the original cost estimates on these paths will turn out to be very wrong, then the router will have a difficulty to find a viable alternative. Consequently, it is beneficial to have a tree with paths that provide the router with different options and not variations of the same path. This motivates a stochastic approach that admits that the routing costs are uncertain.

Consider the routing costs on the roads in the road network to be random variables with arbitrary but known distributions. A random eco-route can be identified by drawing a random cost for every road in the road network and then finding a minimal path between given origin and destination. A collection of eco-routes can then be identified by repeating this process sufficiently many times. The Algorithm 3 implements this. It first initializes an empty tree $\mathcal{T}_{od}$ and then searches for new eco-routes until some stop condition is satisfied. Each iteration identifies one eco-route. It consists of three steps: (1) draw a random cost $\mathcal{X}_e$ for every edge $e$ in the road network graph $G$ and store it to the field $c_e$ (defined in the Section 4.1); (2) find minimal path $r$ between $o$ and $d$ in graph $G$ with costs $c_e$ using Dijkstra's algorithm; (3) if $r$ not in the $\mathcal{T}_{od}$ path tree, then insert it with ADD function.

This approach to path tree computation has interesting properties. Each identified path is by definition the optimal eco-route in some traffic state. It follows that the pruning techniques discussed above are no longer needed: they are designed to prove that a path can never be an eco-route. Note also that it does not have the implicit tendency to identify paths that are likely overlapping, as it is the case with the $k$ minimal paths approach discussed above. The algorithm runtime depends on the choice of the stop condition. The stop condition used in this work is always a simple limit on the number of paths in the tree. In this case, the asymptotic bound on the runtime is polynomial: it is a multiple of the asymptotic runtime of Dijkstra's algorithm. Other stop conditions are possible. For example, one can construct a stop condition such that the algorithm stops when the probability of finding new paths drops below some threshold.

Consider, for example, that the routing cost is the vehicle energy consumption. The Algorithm 3 then requires the knowledge of consumption probability distribution function on every road in the road network. Collecting large enough dataset to approximate these distributions sufficiently well might be cost-prohibitive in practice. Nevertheless, these distributions can be approximated with considerable flexibility. If there is no information available for a given road, then it can be approximated using the Normal distribution with large enough variance to cover all plausible cases. While this approach does not require any information about the road in question, it is clear that better eco-routes can be identified with tighter distributions.

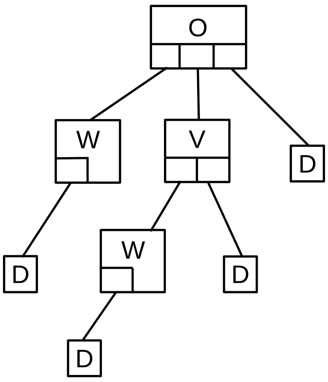


FIGURE 6.2: The path tree made of TREE_NODE instances that corresponds to the tree in example in Figure 6.1.

The consumption model proposed in Section 5.4 can be used to approximate it if there are enough recorded trips on the road in question. The five parameters of this model are then treated as random variables with their own distributions. This approach has the advantage that approximate consumption distributions can be constructed from information that does not depend on the specific vehicle: only the distributions of initial speeds, final speeds, mean travel times, speed profile variances and speed profile skews are required. Prospectively these distributions can also be constructed using more advanced models. For example, there is a limit on minimum travel time that can be derived from road length and speed limit. This can be reflected in the travel time distribution.

## 6.4 High performance path tree

Previous sections considered the path tree to be an abstract data type. This section discusses its implementation. The path tree is at the core of the routing method described in this chapter. Its efficient implementation is therefore essential.

The intersections in a typical road network rarely connect to more than a few roads. In the context of the road network graph $G$, the number of connecting roads is the out-degree of the node that represents the current road. The path tree is a collection of paths from the same origin. Since these paths are required to be contiguous in $G$, the out-degree limits the number of children any tree node can have. This property makes the tree sparse which means that the adjacency list is a more suitable representation that the adjacency matrix. The proposed implementation uses an approach similar to the adjacency list. It implements the tree as a collection of nodes, where every node contains references to its parent and children. There is exactly one parent (except the root node which has no parent) and any number of children. Every node also carries one attribute: the reference to the road network graph $G$. It implements the REF function defined in Section 6.1. Let the tree nodes be represented with a datastructure called TREE_NODE. Let the references to the parent and the children be pointers to other TREE_NODE instances and the reference to the road network graph be a pointer to a node in $G$.

Consider the example given in Section 6.1 in Figure 6.1. The corresponding path tree made of the TREE_NODE instances is shown in Figure 6.2. The boxes are the TREE_NODE instances. The reference to the node in the road network graph is shown in the upper part. The lower parts show the slots to connect the children. The edges represent the links between TREE_NODE instances in memory.

### 6.4.1 Design challenges

The basic operation on the tree is to add a node. The Algorithm 2 uses operation ADD_CHILD to do this. The Algorithm 3 uses operation ADD that adds a whole path to the tree. It can be broken into a sequence of calls to ADD_CHILD. The ADD_CHILD operation needs to (1) allocate some memory for the node, (2) set its reference to the road network graph $G$ and (3) connect to the parent. The parent must be linked with the child in both ways (parent to child and child to parent) since the tree is an undirected graph. A node can have only one parent and an unlimited number of children. Being able to handle tree nodes with a varying number of children is one of the issues that must be considered when designing the path tree datastructure. This topic is treated in Section 6.4.2.

Implementing support for path dominance pruning (Section 6.2.2) requires support for three operations: (1) remove a node, (2) move a node; and (3) query all tree nodes that refer to a given node in the road network graph $G$. Only leaf nodes can be removed. If the node is not a leaf, then there is a subtree rooted at this node which must be removed first. Node moving is easy as it suffices to disconnect the node from its parent and reconnect it with the new parent. Any subtree under the moved node will be moved implicitly with it. As for the third operation, the list of nodes that refer to the same chosen node in the road network graph determines a set of paths on which the path dominance is evaluated. The naive way to produce it would require scanning the whole tree. It is desirable to have a more efficient way to do so. It is proposed in Section 6.4.3

Implementation considerations are not limited to the four methods discussed above. The properties and limitations of modern computers also have to be taken into account. Specifically, following criteria should be respected in the implementation:

- *Obey locality principle.* The processors are typically much faster than memories. Computer cache mitigates the impact of slow memory access by storing recently accessed parts of memory locally in the processor. This is motivated by the locality principle: that recently accessed parts of memory are likely to be reaccessed. If the related information is kept in the same parts of the memory, then the computational overhead is lower due to faster memory access.

- *Avoid memory reallocation.* Changing the size of allocated memory using REALLOC system call can result in the data being moved to a different part of the address space. It would make any pointers pointing inside that memory area invalid.

- *Keep* TREE_NODE *memory footprint small.* The tree can potentially be a huge datastructure. The size of the TREE_NODE datastructure will have a strong effect on the tree size as it is a collection of TREE_NODE instances.

There are conflicting goals. The TREE_NODE datastructure should be as small as possible, but it also should be able to connect to any number of child nodes. The memory reallocation should be avoided, but a datastructure of a variable size is to be managed. The locality principle should be obeyed, but arranging tree nodes that refer to the same node in $G$ together is also preferred as it would allow implementing fast lookups of these node groups. The design proposed below aims to strike a trade-off between all these considerations.

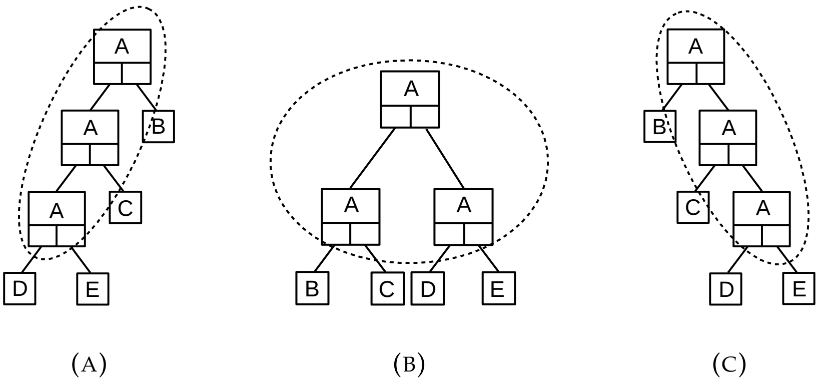


FIGURE 6.3: Three ways to build a complex tree node (in the dashed circle) from TREE_NODE datastructures.

### 6.4.2 Tree nodes without limitations on the number of connected children

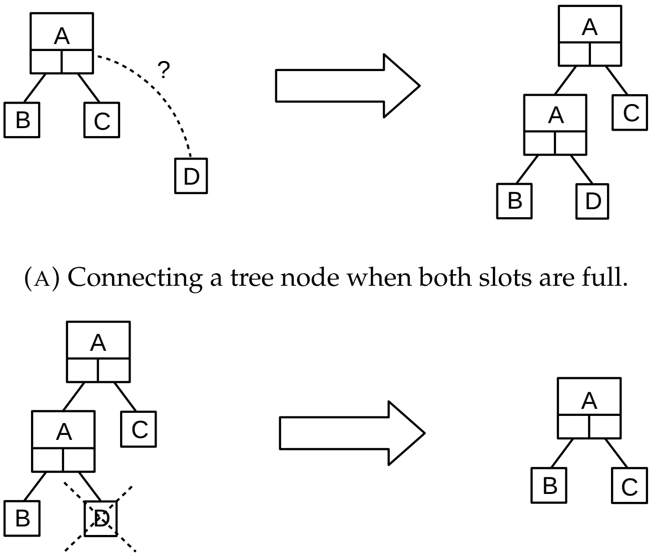
Allowing tree nodes to have an unlimited number of connected child nodes is the first design challenge mentioned in the previous section. A naive way to deal with it would be to equip each TREE_NODE datastructure with a fixed-length array that would carry the references to the children, where the size of the array is the maximum out-degree of nodes in the road network graph. This guarantees the array is always large enough. It is not effective, however, as these arrays can be expected to be largely unused. The ineffectivity can be fixed by using dynamical arrays instead of fixed-sized arrays. But this would bring other issues. Mainly, memory allocation and subsequent reallocations would introduce time delays. The approach proposed here is designed to allow connecting any number of children to a tree node, but which does not require explicit memory allocation.

Consider a TREE_NODE datastructure that can carry references to two children. Then two or more TREE_NODE datastructures can be combined to create a tree node with any number of children. Let us call such nodes *complex nodes*. Consider the example in Figure 6.3. It shows three ways how a complex node with four children can be built from three instances of TREE_NODE datastructures, each with two children.

There must be a way to distinguish complex nodes from the other nodes in order to use this approach. Each node carries a reference to the node in the road network graph that it represents. If this reference is identical between two mutually connected TREE_NODE instances, then both instances can be considered to belong to the same complex node. The limitation of this approach is that it will not work on road network graphs with edges that start and end on the same node. This corresponds to road networks with roads that start and end in the same intersection.

Efficient adding and removing of children nodes can be realized with the complex nodes. As shown in Figure 6.3 there are multiple ways to build a tree node from multiple TREE_NODE instances. Hence there are multiple options to construct the complex nodes. Consider the case in Figure 6.3a, where the complex node is linked via first child pointers. Connecting and disconnecting child nodes can then be implemented as follows:

- *Connect a child.* If there is a free slot to connect the child through, then connect there and return. If not, then the tree node needs to be expanded to create a new free slot for the child. The expansion is indicated in Figure 6.4a.



(A) Connecting a tree node when both slots are full.

(B) Removing the connection.

FIGURE 6.4: Connecting children to complex nodes.

- *Disconnecting a child.* If the node is complex, then disconnect the child and remove the free slot. The slot can be freed by removing the TREE_NODE instance through which the child was connected. This is shown in Figure 6.4b.

The proposed approach keeps the child slots of the complex tree nodes always fully occupied. Hence, there is no need to scan the complex node when connecting a child since there can never be any free slots. This enables implementation of the connecting (resp. disconnecting) process that runs in a constant time. Also, note that the proposed approach implicitly turns simple nodes to complex nodes when connecting children and complex nodes to simple nodes when disconnecting children.

To summarize, a tree node is either simple or complex. The simple nodes and can connect to at most two children. They are represented by the TREE_NODE datastructure. The complex nodes are made of multiple TREE_NODE instances. They can connect to $n$ children through exactly $(n-1)$ TREE_NODE instances. The TREE_NODE instances that belong to the same complex node are uniquely identified by their reference to the road network graph: they all refer to the same node in $G$. Child nodes can be connected (respectively disconnected) efficiently in a constant time in this way.

Note that it is possible to consider more than two slots in the TREE_NODE datastructure, but then the mechanics of efficient child connection (respectively disconnection) would be more complex. It is yet to be determined whether there is any potential in TREE_NODE datastructure with more than two child slots.

### 6.4.3 Grouping tree nodes by their reference to the road network

The second design challenge mentioned in the previous section is the need to be able to produce a list of all tree nodes that refer to the same node in the road network graph $G$. This list is used every time a new partial path is found for path dominance testing since all other paths in the tree that lead to the same node are compared to the newly found path. An efficient implementation is therefore essential. The naive implementation would have to scan the whole tree, which is time-consuming. Since

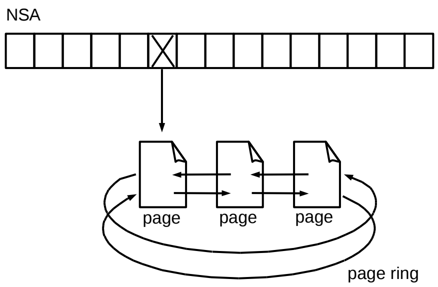


FIGURE 6.5: The node supplementary array (NSA) with a page ring that holds the tree nodes grouped by their reference.

the tree is linked together via its parent/child references (pointers), the position of the nodes in the memory is not limited. Hence, tree nodes with the same reference can be grouped in the memory. Producing lists of all tree nodes with the same reference is then a matter of referencing to the right group of tree nodes, which can be implemented in a constant time.

Consider the arrangement in Figure 6.5. It shows an array called *NSA* with elements that point to doubly-linked lists of objects called a *page*. The NSA array has a fixed size with the number of elements that correspond to the cardinality of the set of nodes $V$ in the road network graph $G$. The pages are datastructures that hold the TREE_NODE instances. They have a fixed size and can, therefore, hold only a limited number of TREE_NODE instances. This limitation can be overcome by arranging the pages in circular doubly-linked lists. Let us call this list a *page ring*. If there is a bijective mapping from $V$ to the NSA, then it can be used to retrieve the page ring that contains all TREE_NODE instances in the tree with the desired reference to $G$. One such bijective mapping is with the elements in the NSA organized in the same order as the nodes in $V$. This particular mapping allows looking up the page ring with nodes that refer to $n$-th node in $V$ simply by taking the $n$-th element in the NSA.

Consider the page structure outlined in Figure 6.6. It has three parts: header, padding, and payload. The header contains information about the page, the payload is an array of TREE_NODE instances, and the padding is used to offset the size of the datastructure to have exactly 4096 bytes. The indicated offsets in the memory (left side; Figure 6.6) assume the computer uses 64-bit memory addressing.

Let the pages be aligned with system page frames: then they take exactly 4096 bytes (typically) and start at an address that is divisible by 4096 without any remainder (such addresses end with 0x000)[3]. Allocating the area that corresponds to system page frames is faster than general allocation. Moreover, the knowledge of the lower twelve bits of the page base address (it is 0x000) allows converting any pointer to a TREE_NODE instance into a pointer to the page where it resides by setting the lower twelve bits of the address to zero.

There are four fields in the page header: reference to the road network graph $G$, the pointers to the previous and next pages, and the number of occupied TREE_NODE instances within the page. The tree nodes within one page necessarily refer to the same node in the road network graph $G$. It would be redundant if every TREE_NODE instance in the page would carry the same reference. Hence, it can be taken out of the TREE_NODE datastructure and placed in the page header (as shown in Figure 6.6). This trick lowers the memory footprint of each TREE_NODE instance, and allows to store 169 TREE_NODE instances in the page. The TREE_NODE then

[3] Such allocation can be made with POSIX system call POSIX_MEMALIGN.

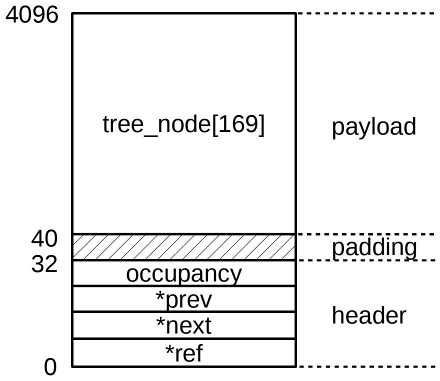


FIGURE 6.6: The memory organization of a page.

consists of three pointers: (1) reference to the parent, (2) reference to the first child, (3) reference to the second child.

### 6.4.4 Memory management

The design of path tree's memory management affects to two main operations: allocating a TREE_NODE instance and freeing a TREE_NODE instance. The proposed implementation avoids system calls that relate to memory management, if possible. It pre-allocates some memory during initialization. Assuming there is some preallocated memory available, then the proposed design achieves optimal performance: both operations are done in a constant time.

The path tree datastructure is stored in pages (as discussed above), except for the fixed-sized NSA array. TREE_NODE allocation (respectively TREE_NODE freeing) can trigger page allocation (respectively page freeing). Node allocation triggers a new page allocation when there is no more space for the new tree node in the current page. Page freeing is triggered when it becomes empty after node removal.

The implementation keeps some free pages at hand instead of allocating and freeing individual pages. Specifically, some pages are preallocated during initialization and placed in a linked list. The fields in page headers that are used to keep the page in the page ring can be used for that. A page is taken from this list of free pages whenever needed. If there is no page left in the list, then multiple pages are preallocated again. Freed pages are not returned to the operating system but pushed back to the list of free pages. They are put on top of the list because the next time a page is needed the recently freed page is likely to be still present in the processor cache.

Node allocation is about finding a free place in the right page ring. It is advantageous to identify the page where the node should be inserted quickly. The NSA array can be used to identify the right page ring in constant time, but finding an empty slot might require scanning the ring. This operation can be avoided if all pages are kept full except for the page to which the field in the NSA points. Let us call the single non-full page in the ring the *top page*. Node allocation is then easy: it is a matter of incrementing the occupancy counter in the top page. If the top page becomes full after the node is added there, then a new page should be allocated. This page then becomes the new top page.

Keeping the ring full except for the top page is not trivial as any node in the ring might be removed. This is why node freeing is more complicated: it has to remove the node such that the pages in the page ring remain fully occupied, except for the top page. The process goes as follows. First, it is necessary to check if the top page in the ring is empty (its occupancy counter value is zero). If that is the case, the page should be removed from the ring and freed. Then, it is necessary to swap the to-be-deleted tree node with the tree node on top of the top page. The swapping can be done in a constant time by updating the linkage of the TREE_NODE instances with their parents and children. The node freeing is easy after the swapping: it is a matter of decrementing the occupation counter of the top page. Note that this implies that the swapping routine moves the position of an unknown node in memory. This operation does not impact the coherency of the tree as the swapping routine updates all relevant links carefully, but any pointer to the moved TREE_NODE instance from outside of the tree will become invalid. This is an important issue that motivates the support for path queues as discussed in the next section.

### 6.4.5 Queue support

The Algorithm 1 uses a queue to store tree nodes for later processing. It is a datastructure separate from the path tree: a list of references to path tree nodes. These references are addresses in memory where the TREE_NODE instances reside. As discussed in the Section 6.4.4, the tree nodes can change their position in memory anytime a node is deleted. Consequently, the references from the queue to the tree can become invalid without notification. It motivates an implementation of a queue that is embedded in the path tree. The path tree and the queue can co-operate to keep all references valid, and it can also reuse path tree's memory management.

Consider a queue stored in pages that have the same structure as the pages used in the path tree, except that its payload is not a collection of TREE_NODE instances but a collection of pointers to TREE_NODE instances. Multiple pages can be linked together in a linked list. The queue can then be implemented by pushing (respectively popping) references to tree nodes to (respectively from) the top of the first page in the linked list. The path tree pages have four fields in the header: the pointer to the next and previous pages, the reference to the road network graph $G$ and the occupancy counter. The pointer to the next page and the occupancy counter can be used to implement the queue. The value of the field with reference to $G$ can be set to zero to indicate that this page is used for the queue.

An efficient approach to making sure that the references inside of the queue are always up-to-date with the tree is to trick the swapper routine (Section 6.4.4) to think that there is a fake child node which is, in fact, the back-referencing pointer in the queue. Consider some tree node that is referenced from the queue, and that is about to be moved. The swapper will check whether the node has any children and if so, update their pointers to the parent node to its new position. If the first field in the TREE_NODE datastructure is the pointer to the parent node, then this operation is the same as updating a single remote pointer (since the rest of the TREE_NODE instance will remain untouched). The nodes in the queue can't have any children because the Algorithm 1 did not yet process them. Hence, there is always a space to add the fake child node. In summary: with this method the swapper routine will always update entries in the queue, considering them to be another TREE_NODE instance, while they are are in fact only pointers. The described method keeps the queue references valid at no extra cost in terms of larger memory footprint or prolonged computation time.

## 6.5 Summary

This chapter is concerned with a routing algorithm suitable for eco-routing applications. The proposed concept requires a preprocessing step that identifies a collection of candidate eco-routes. Rather than routing on the road network graph as it is usually done, the routing is conducted on this collection. The method proposed in this chapter favors simplistic routing over sophisticated optimal routing algorithms such as Dijkstra's or Bellman-Ford algorithm. The motivation for this is that these algorithms are too restrictive, too rigid in their assumptions. An approximative solution based on a naive approach to routing is studied. It is based on a hypothesis that while there are usually many possible paths to take, there is only a small subset of them that can be eco-routes in some realistic conditions.

The routing algorithm is described in Section 6.1. It takes as an input a collection of candidate eco-routes between the same origin and destination bundled in a tree datastructure. The routing is an exhaustive search for the path that minimizes a given cost function and satisfies all constraints. It is usually not computationally feasible to consider all simple paths between a given origin and destination. Hence, the tree can contain only their subset. It can be a handful of paths or hundreds of thousands of paths, it is still a small amount in comparison to the number of paths found between most origins and destinations.

The tree computation can be a computationally intensive operation. It can, however, be prepared once and used later multiple times. Computing the routing solution is then comparatively cheap. This motivates a routing system where the path tree computation is decoupled from the routing: the path tree can be computed in advance for repeated trips, then sent to the navigation assistant and used whenever required. Two approaches to path tree computation are studied: exhaustive search (Section 6.2), or randomized search (Section 6.3). Both generate path trees with a subset of paths between given origin and destination. The randomized method directly identifies the prospective eco-routes, the exhaustive search considers every path for which the pruning criteria failed to show that it cannot be an eco-route.

Note that path trees can be constructed in other ways as well. They can be considered to be pre-computed options the routing algorithm can choose from. One possible way to construct such a tree is to take eco-routes based on various proposed energy consumption models in the literature. The routing algorithm would then choose the eco-route that minimizes the consumption given the current conditions in the road network.

# Chapter 7

# Simulation framework for eco-routing

The evaluation of an eco-routing method requires field trials. Evaluating it in this way is nevertheless cost prohibitive, as evidenced by the limitations of field trials conducted in the past (see Chapter 3). This motivates evaluation in simulation as it can approximate results of large-scale field trials. Such simulation framework is presented in this chapter. A specialized simulation software is used to run a traffic simulation of a real European city. Two microscopic vehicle consumption estimation models that were validated on real vehicles are used as references. Note that while the simulations were designed to be as realistic as possible, they only indicate whether the studied method is likely to succeed in deployment. The ultimate eco-routing evaluation is still with field trials.

There is only one relevant eco-routing simulator available: the INTEGRATION framework. Ahn and Rakha, 2013 conducted a case study with this software. It was not used in this work because a suitable traffic scenario to use with it was not available at the time of writing. Another traffic simulator for which such scenario is available was adopted. This simulator was not designed to be used in context of eco-routing. All the experiments conducted in the context of this work were implemented as separate programs that connect remotely to the simulator. The experiments were implemented in Python 2.7 except for some computationally intensive subroutines that were implemented in C.

## 7.1 Resources and software

SUMO (Simulation of Urban MObility) simulator by Krajzewicz et al., 2012 in version 0.28.0 was used to conduct the traffic simulations. LuST 2.1 (Luxembourg SUMO Traffic) scenario by Codeca et al., 2015 was used to simulate realistic traffic in

TABLE 7.1: Luxembourg traffic scenario summary[†].

| | |
|---|---|
| simulation area | 155.95 km$^2$ |
| simulation time span | 24 hours |
| number of vehicles | 284,184 |
| total length of roads | 929.5 km |
| number of intersections[‡] | 4,477 |
| peak traffic | ≈ 4,800 active vehicles |

[†] Data from Codeca et al., 2015

[‡] Includes 203 signalized intersections

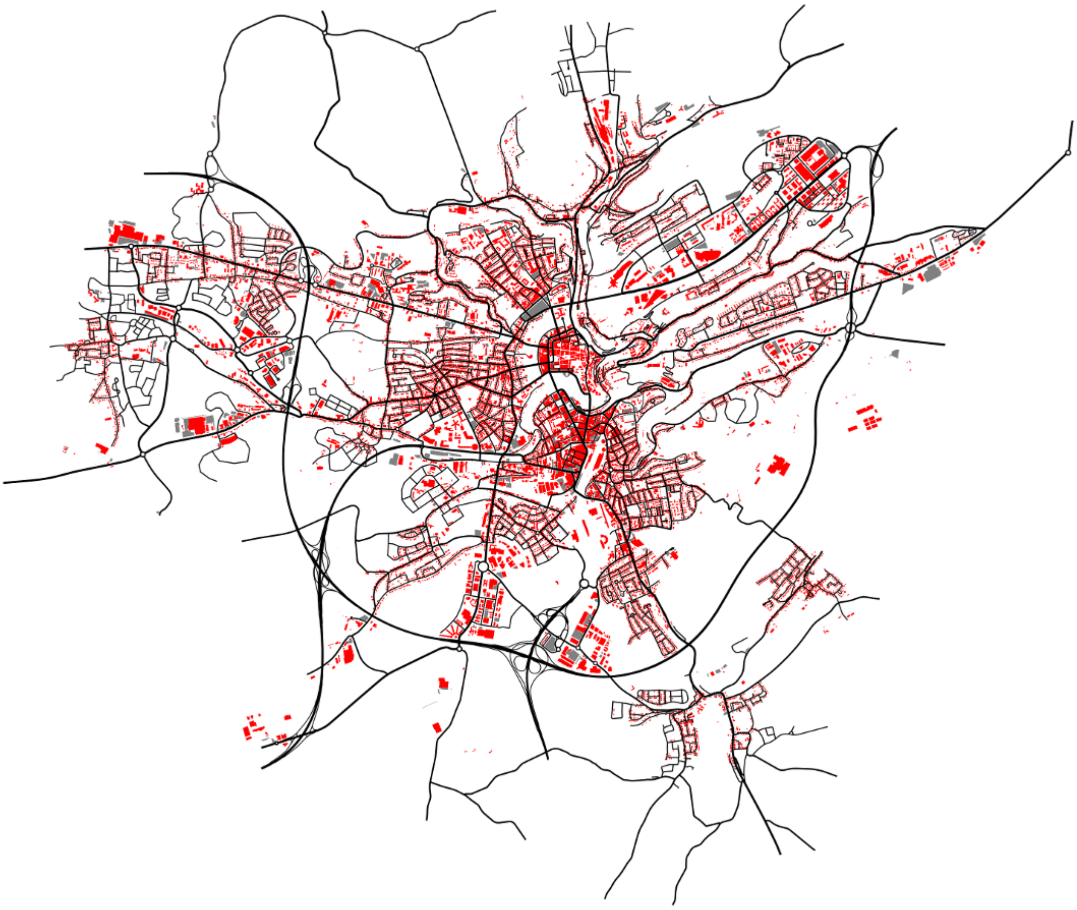


FIGURE 7.1: Luxembourg traffic scenario (Codeca et al., 2015).

a mid-sized European city. Road elevation data were extracted from *EU Digital Elevation Model* by European Environment Agency. All these resources were available under open licenses at the time of writing.

The SUMO software is a microscopic urban mobility simulator: it simulates the movement of individual vehicles in a road network. The vehicles follow their paths to the destination, interact with other vehicles and with the environment. Each vehicle has non-zero physical dimension and occupies correspondingly large space somewhere in the road network. Vehicles are not allowed to occupy intersecting space; this is detected by the simulator and reported as collisions. The roads in the road network have one or more lanes. The vehicles respect the right-of-way priority on intersections, or traffic light state if there is one. The traffic lights have static programming. The simulation allows adding pedestrians into the road network, but there is no support for them in LuST 2.1.

The SUMO road networks have a complex multi-layered design. The roads are called "edges", the intersections are "junctions". The edges are collections of "lanes". Each lane has own shape and can connect to different roads in the downstream intersection. Each junction features an underlying graph that contains both "internal edges" and "internal junctions". The internal junctions are used to model priority relationships between vehicles coming from different roads. The internal edges are collections of "internal lanes". They are used to model trajectories of passing vehicles.

The LuST traffic scenario has layout and mobility patterns of the Luxembourg

TABLE 7.2: The reference consumption models.

| | EV[†] | ICEV[‡] |
|---|---|---|
| referenced vehicle | FAM F-City[*] | Renault Scénic |
| maximum power | 150 kWh | 150 kWh |
| transmission type | fixed | 6-gear automatic |
| transmission efficiency | 95% | 95.06% |
| auxiliary losses | 200 W | 500 W |
| vehicle mass | 1190 kg | 1588 kg |
| tire radius | 28.48 cm | 31.73 cm |
| coefficient $c_c$ | 113.5 N | 110.45 N |
| coefficient $c_b$ | 0.7740 N/(m/s) | 1.5175 N/(m/s) |
| coefficient $c_a$ | 0.4212 N/(m/s)$^2$ | 0.5119 N/(m/s)$^2$ |

[†] Electric vehicle
[‡] Internal combustion engine vehicle (conventional vehicle)
[*] Adapted for higher maximum power, see Section 7.2

city road network. Figure 7.1 shows a render of the road network. Its main properties are summarized in Table 7.1. It covers an area of 155.95 km$^2$ with 929.5 km of roads and 4,477 intersections (Codeca et al., 2015). The simulation spans 24 hours. Different road types are available: there are highways, arterials and local roads. The traffic is generated synthetically and matched to observed traffic patterns in Luxembourg (not publicly accessible). The native traffic in the scenario consists of 284,184 vehicles. The peak traffic consists of about 4,800 vehicles simultaneously active in the simulation.

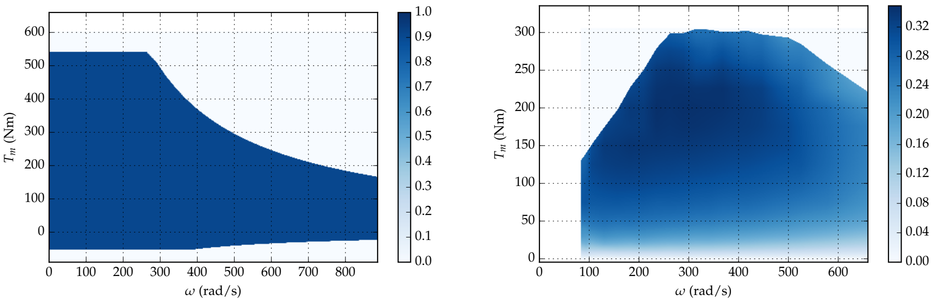
The EU Digital Elevation Model is an elevation map of the European continent. It is a numerical model with a grid of sampled elevations 25 meters apart. A section that entails the Luxembourg city was extracted from it. The elevations between the samples were interpolated with the bicubic interpolation. The model is used to approximate road slopes in the road network. They are assumed constant, based on altitude difference between the upstream and downstream intersections. This is because more precise information is not available: while the terrain shape is known the roads do not necessarily follow it (as is the case, for example, with bridges).

## 7.2 Reference microscopic consumption estimation models

Two reference microscopic consumption estimation models are discussed in this section. They are used as a golden standard when validating energy consumption models and when evaluating eco-routing performance. They are standard longitudinal models with a nonlinear engine (resp. motor) efficiency maps. Both models were validated on real vehicles: one is electric, the other conventional. Their properties are summarized in Table 7.2

The electric vehicle energy consumption model is based on FAM F-City. It is a small 50 kW electric vehicle. The model is described and validated in Dib et al., 2012. It was provided by the authors of that study. The 50 kW power is not sufficient to support the dynamics of the vehicles in the simulations. For this reason, the motor parameters were rescaled by the factor of 3 to simulate a 150 kW motor. Its efficiency map is in Figure 7.2a. The efficiency is assumed constant at 90.9%. The recuperating power is limited to 20 kW.

The conventional vehicle energy consumption model is based on a hatchback version of Renault Scénic. The authors Dib et al., 2012 provided this model too. It



(A) the electric vehicle

(B) the conventional vehicle

FIGURE 7.2: Powertrain efficiency maps for the reference vehicles. The efficiency is the ratio of powertrain's output to input power.

has a 150 kW internal combustion engine. Its efficiency map is in Figure 7.2b. The 6-gear transmission is automatic. The gear switching law does not depend on torque, only on instantaneous speed.

## 7.3 Low-level algorithms and datastructures

One of the basic requirements in eco-routing is to have a road network graph to route on. An implementation of a data structure that can hold the graph together with a set of basic operations on it is essential. The road networks tend to be extensive and so the datastructure need to be able to handle large graphs efficiently. The nodes in the road network graph relate to roads (as described in Section 4.1). Information about roads such as road length and the altitude difference is often used by eco-routing methods when computing routing costs. Hence, the datastructure must be able to store road-related information (metadata) for each node. Also, the computational overhead of queries for connected nodes in the graph has a strong impact on the computational overhead of routing algorithms. Having an efficient implementation for these queries is imperative. Also, an ability to store the graph for later use is practical as compiling a road network graph from map resources can be time-consuming. A datastructure with properties mentioned above is discussed below.

Another low-level functionality required to have is an optimal algorithm for finding minimal paths. This is addressed with an implementation of Dijkstra's algorithm presented below. It is used when computing path trees (in Algorithm 3) and by the baseline eco-routing methods (Section 4.3).

These low-level routines and datastructures were implemented in C, while all the experiments conducted in the context of this thesis were implemented in Python. The low-level functionality is accessed from Python using its "ctypes" interface.

### 7.3.1 The road network graph

There are two main approaches to represent directed graphs: using either adjacency matrix or adjacency list. The adjacency matrix is a square matrix used to represent connections between nodes in a finite graph. Consider $a_{ij}$ to be an element on $i$-th row and $j$-th column of the adjacency matrix. There is a connection from $i$ to $j$ if $a_{ij} \neq 0$. The adjacency list is a set of ordered pairs where each pair represents one edge: the first element is the origin node; the second element is the destination node.

This work uses the adjacency list approach. This representation is more compact for sparse graphs, and it also enables faster retrieval of neighboring nodes.

Each node in the set of nodes represents exactly one road between two intersections. This is in accordance with the definition given in Section 4.1. Each node carries information about the road it represents: its name, length, shape, speed limit and altitude difference between the two intersections it connects. This information can be used for various purposes from visualization to eco-routing. In particular, the information is used in the context of eco-routing to compute routing costs.

The basic operation on a graph is to query its connectivity: to retrieve successors and predecessors of a given node. Both are needed when routing with the Dijkstra's algorithm: retrieving a list of successor nodes when searching for a minimal path and retrieving a list of predecessor nodes when reconstructing it after the routing finished. The Dijkstra's algorithm makes many queries for successor nodes, so it is imperative to implement it in such a way that induces low computational overhead per query. The implementation used here makes use of a list of edges sorted by edge origin. This arrangement makes all edges originating from the same node adjacent in the list. Then, every node has assigned offset to the first edge in the list of edges that originates from the node. This allows producing a list of successor nodes in a time proportional to the number of connected nodes.

The whole road network graph was compiled once and saved to a file for later use. It is built from SUMO road network description file that comes with LuST scenario and from the EU-DEM elevation model. It is a complex structure whose compilation takes some time. Being able to compile it once and store it to a file for later use makes it easier to handle and allows faster loading. The used implementation compiles it such that the file can be directly mapped to system memory (this is facilitated by POSIX system call `mmap`). This allows fast loading of the road network graph as the file is just mapped to the address space of the process. Actual data are loaded lazily when needed (accessing data that are mapped to memory but not present in memory will cause a page fault which will prompt the OS to load it).

### 7.3.2 Dijkstra's algorithm implementation and its variants

Two variants of the Dijkstra's algorithm were used. A standard implementation was used to identify eco-routes in the classical time-independent eco-routing. The second variant is its adaptation designed to speed up the Algorithm 3.

The standard implementation uses a set of routing costs that are provided to the algorithm together with the road network graph. It uses a priority queue implemented as a standard heap datastructure. A lower asymptotic run time could be achieved with other types of priority queues, but this implementation does not make use of it. This is motivated by results reported by Chen et al., 2007 who observed that the used implementation yields lower runtime on road network graphs.

The second variant is used to render the Algorithm 3 faster. As written, the Algorithm 3 first draws a random cost for every edge in the road network graph and then runs the Dijkstra's algorithm on this graph. The adapted version of the Dijkstra's algorithm draws random costs lazily as needed, thus avoiding the need to preload it for the whole graph.

### 7.3.3 Drawing a random cost from arbitrary distribution

The Algorithm 3 draws a random cost for every road in the road network graph individually on line 4. Two things are required to be able to do that: (1) to know the

TABLE 7.3: Collected information for every passing vehicle on every road in the road network.

| property group | property | unit |
|---|---|---|
| | road name | - |
| | next road name | - |
| | vehicle name | - |
| | traveled distance | m |
| | altitude difference | m |
| speed profile | speed profile variance | $(m/s)^2$ |
| | speed profile skew | - |
| | initial speed | m/s |
| | final speed | m/s |
| timing | arrival time | s |
| | travel time | s |
| | waiting time | s |
| EV | consumption | J |
| | energy on. wheels. | J |
| | braking energy | J |
| | traction efficiency | % |
| | recuperation efficiency | % |
| | peak power | W |
| ICEV | consumption | J |
| | energy on. wheels | J |
| | braking energy | J |
| | traction efficiency | % |
| | peak power | W |

underlying distribution and (2) to have a method to draw at random from it. Inverse transform sampling method was used to draw random costs from arbitrary distributions. It takes a random number $u$ drawn from the uniform distribution between 0 and 1 and then returns $u$-th quantile of the distribution to draw from (Devroye, 1986). The quantile computation requires computing the cumulative distribution function and then inverting it. The uniformly distributed random number between 0 and 1 was obtained by reading random numbers from operating system's pseudo-random number generator[1] and by normalizing them.

The inverted cumulative distribution functions were precomputed for every node in the road network graph to lower computational overhead induced by inverse transform sampling. The consumption on individual roads in the native traffic dataset (Section 7.4) was used to approximate the distributions. They were inverted, resampled using linear interpolation, and stored as a sequence of 101 samples between 0 and 1.

## 7.4 Collecting and processing native traffic data

Historical observations are needed to build the consumption and travel time models proposed in Chapter 5. The LuST scenario includes 284,184 vehicles in a 24-hour time frame. The data from their travels can be used in place of actual historical observations.

The trips are collected by running SUMO with LuST scenario such that simulation state is dumped to a file at every simulation step. The simulator allows this

[1] `/dev/urandom`

through its "full-output" functionality. The state of every vehicle, every traffic light, and every road is reported. This is processed with a script that tracks position and speed of every vehicle. The script collects this information for the duration of life of the vehicles in the simulation. When a vehicle finishes its trip the script processes the collected record. It cuts the recorded sequences of position and speed to slices that correspond to individual roads where the vehicle have been. It then dumps a record with road name, vehicle name and other properties listed in Table 7.3 for each slice (meaning for each road where the vehicle have been traveling). Various variables are reported. It contains all information needed to build both travel time and consumption models and to run consistency checks. The records are saved to a file for later use. The file is sorted by road name, which implicitly groups information about the same roads to the same part of the file. An indexation table in a separate file is used to enable fast searching in the data. The script also dumps a file with recorded traffic light states for the full duration of the simulation. This file is compressed with run-length code to minimize the size of the resulting file.

The scenario uses one-second time discretization. This is too sparse: much information would be lost if the speed profile is sliced directly. A whole second worth of data would be lost every time the vehicle passes from one SUMO road network edge to another, which happens on the average every seven seconds. There are many short edges in SUMO road network that can be missed when sampling position once per second. Careful slicing is necessary to ensure the integrity of extracted information. The vehicle speed and road slope records were resampled with a sampling period of 0.01 second. Linear interpolation was used. Cubic interpolation was also considered, but it tends to overshoot to negative speeds when a vehicle goes to full-stop. The vehicle path is reconstructed from the observed sequence of its positions by searching for the shortest path between every two consecutive observation (they are one second apart of each other). This fills the gaps between observed positions. The integrity of the reconstructed path is checked by comparing the length of the reconstructed path with the traveled distance. The slicing is done according to crossing times between edges in the reconstructed path. There is a number of tests implemented to ensure the integrity of the result. Most notably, the consumption on every slice is added and compared to the consumption of the whole trip. The record is scraped if any inconsistency is found.

The SUMO road network distinguishes individual road lanes. This is modeled as a complex multi-layered road network graph. Such road network graph allows modeling the dependency of the energy consumption on the lane the vehicle takes: different lanes can have different occupancy, lead to different roads and have a different semaphore controlling them. A simpler approach that preserves the ability of the model to take these differences into account is used instead. As discussed in Chapter 4 roads in the road network are represented by nodes of the road network graph. The edges represent a directed connection between two roads. Each edge has some cost assigned. This is a cost for the road from which this edge spans constrained to vehicles that continue to the road to which the edge leads. This allows assigning varying costs to the same road depending on where the vehicle goes next. The complex lane-level road network graph is not needed.

The raw simulation output is massive: it has over 700 GB. The simulation normally takes about 30 minutes to complete. However, when the full-output functionality is enabled the simulation takes over 30 hours. This is because of the delay in slow I/O operations. This delay can be avoided by redirecting the "full-output" data to a named pipe: a special file in the file system typically used for interprocess communication. The data written to a pipe are stored in a memory buffer and not

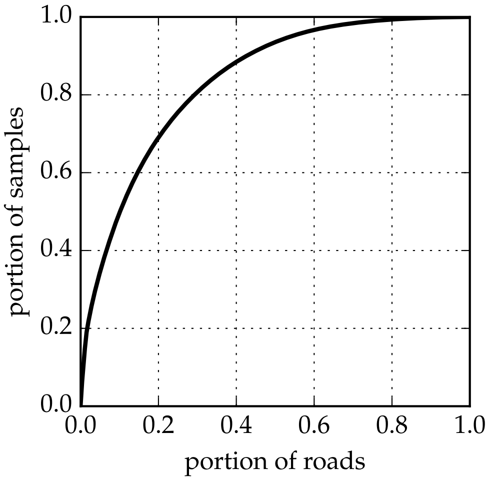


FIGURE 7.3: Inequality of traffic data distribution between edges: 70% of all records were collected on top 20% most used roads.

in a slow nonvolatile memory. Another process can open this file and read the data written there. This allows passing the data directly to the script that process them in parallel as they come. Some computation speedup can also be achieved by editing SUMO's source code. There is information in the full-output that is not used. The code responsible for writing this information can be commented out[2].

The processing script takes dozens of hours to finish. This is because of the fine-grained resampling of the collected data, path reconstruction and consumption estimation with the reference models. Especially the reference consumption models are slow to compute as they numerically integrate the estimated instantaneous power draw. The program was parallelized to allows it to scale to multiple processors. This reduces the computation time as it runs in several instances in parallel on different machines. Simple master-slave architecture is used: there is one process that sends jobs to slave processes when they are free. Each job handles a single recorded trip. When they finish, they send back the result and wait for the next job.

The resulting dataset contains 9,454,529 records. The distribution of the data on the roads in the road network is uneven. As can be seen in Figure 7.3 about 70% of all records were collected on 20% of most used roads.

## 7.5 Evaluating paths in a simulation

This section discusses the used method to compute the reference consumption. The reference consumption can be compared to the predicted consumption to validate the proposed consumption models. It is also used to evaluate performance as of the eco-routing methods studied in this work.

The eco-routing experiments are conducted within the LuST traffic scenario in the SUMO simulator (Section 7.1). It allows recording both speed and position of vehicles as they progress on their paths. This information is used to compute energy consumption with the reference consumption models (Section 7.2). The experiments were implemented in Python. The simulator is remotely controlled via SUMO's TraCI interface.

[2] See the file `src/microsim/output/MSFullExport.cpp` in SUMO's source tree

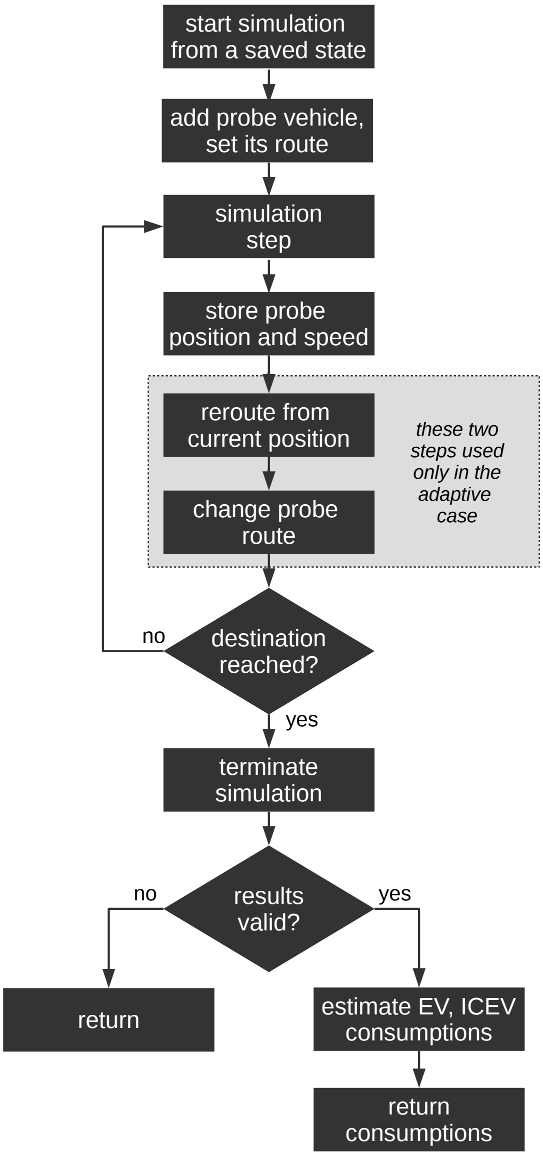

start simulation from a saved state
add probe vehicle, set its route
simulation step
store probe position and speed
reroute from current position
change probe route
these two steps used only in the adaptive case
no
destination reached?
yes
terminate simulation
no
results valid?
yes
return
estimate EV, ICEV consumptions
return consumptions

FIGURE 7.4: Simulation workflow.

A considerable number of independent simulations must be conducted to be able to evaluate eco-routing performance accurately. The program that conducts them was parallelized and run on a computer cluster to reduce the required computation time. One master process is used to assign jobs to multiple slave processes. The slave processes conduct the simulations and return the reference consumption to the master process. Each slave process runs an independent instance of the SUMO simulator.

Two algorithms are used to evaluate the paths: one for fixed paths and another for paths that can change. The former is used when the path is determined in advance and cannot change after the vehicle departed. This was used to evaluate the performance of the baseline eco-routing methods (Section 4.3) and when evaluating eco-routing based on model 6 (see Section 9.1). The latter algorithm is used to simulate the adaptive routing studied in Section 9.3.

### 7.5.1 Fixed paths

The steps involved in the simulation of fixed paths are depicted diagrammatically in Figure 7.4. The process starts by initializing a new SUMO instance with the LuST scenario. The simulation is initialized to a previously saved state at the desired departure time. Then a probe vehicle is added to the simulation, and its path is set. The vehicle appears at the origin. This concludes the initialization. The program then steps through the simulation, saving probe's instantaneous speed and position at each step. The simulation is terminated when it reaches the destination. The collected data are then validated: the program verifies that the recorded path is correct, complete and consistent. If the validation fails, the program indicates an exception and returns. If the validation succeeds, the energy consumption of both reference vehicles is computed from the collected data and returned.

### 7.5.2 Adaptive paths

The process is similar with adaptive paths. It is also depicted in Figure 7.4. The only difference is in the part when the vehicle is in the simulation. The program re-runs the routing algorithm (Algorithm 1) from the current position of the vehicle at every simulation step. If the eco-routing solution changes the probe's path is updated in the simulation.

## 7.6 Summary

The evaluation of an eco-routing method requires cost-prohibitive field trials. This chapter proposes a simulation framework to replace them. The simulations don't provide hard evidence by definition, but they are indicative and comparatively cheap to conduct compared to actual field trials. It was used to evaluate energy consumption minimizing eco-routing methods. Pollutant emission minimizing methods can be evaluated analogically, if needed.

Conducting simulated eco-routing experiments requires a traffic simulator and a traffic scenario. The scenario should be comprehensive, so the vehicles in it are subject to disturbances typically observed in real traffic such as interaction with other vehicles, dealing with various levels of congestion, interacting with traffic lights, right-of-way priorities on intersections, stop signs, and so on. The used simulator is SUMO 0.28.0 (Krajzewicz et al., 2012), the used traffic scenario is LuST 2.1. (Luxembourg SUMO traffic, Codeca et al., 2015). They are discussed in Section 7.1. Both are available under an open-source license at the time of writing. The SUMO simulator simulates individual vehicles in the scenario, updating their position and speed at one-second time intervals. The LuST scenario is based on the road network and traffic patterns of Luxembourg (a mid-sized European city). It simulates over 200,000 vehicles in a road network that contains local roads, arterial roads, and a highway.

The output of the studied consumption models is compared to energy consumption estimated by two reference consumption models. They are introduced in Section 7.2. Two such models are considered: one is for an electric vehicle, the other for a conventional vehicle. The electric vehicle is represented by FAM F-City, the conventional vehicle by Renault Scénic. Both are standard consumption estimation models that integrate estimated instantaneous power draw from vehicle's source of energy. They use experimentally derived numerical powertrain efficiency maps to approximate the power draw. Both models were developed and validated by a third party.

The road network graph and the routing algorithms were implemented in-house, they are discussed in Section 7.3. These low-level functions were implemented in C. The rest of the framework was implemented in Python. The road network graph was compiled from SUMO road network definition file and an elevation model of the area. The road network definition file is part of LuST 2.1; the altitude data were imported from EU-DEM (*EU Digital Elevation Model*).

Historical observations on individual roads are required by consumption models used in this work. They are typically used to identify their coefficients. The traffic scenario contains over 200,000 simulated vehicles that constitute its native traffic. Their speed profiles are recorded and processed in order to generate the data needed by the consumption models. This is discussed in Section 7.4.

Actual simulations are conducted on individual paths: a probe vehicle is introduced to the simulation and its progress is recorded until it reaches the destination. The process that does this is described in Section 7.5. The collected information is used to compute the reference consumption (with a reference consumption model).

# Chapter 8

# Model identification & validation

The models proposed in Chapter 5 are identified and validated in this chapter. The identification amounts to the identification of coefficients of the studied models. The validation amounts to the estimation of their accuracy.

Baseline consumption models for eco-routing are studied in Section 8.1. The travel time model that was proposed in Section 5.5 is identified and validated in Section 8.2. The consumption model that was proposed in Section 5.4 is identified and validated in Section 8.3. We have published a study similar to what is in the Section 8.3, except that the study considers ideal vehicle instead; see Kubička et al., 2016a.

The identification of the models discussed in this chapter requires a collection of records with both dependent and independent variables of these models. The native traffic dataset (Section 7.4) contains such records. In order to be able to do both identification and validation reliably, the models are cross-validated with 2-fold cross-validation repeated five times (sometimes referred to as "5x2cv"). This estimator was proposed in Dietterich, 1998. The 2-fold cross-validation is a technique that partitions the set of all observations randomly to two subsets. Then, in a first run, the first half is used for identification and the other half to estimate model's accuracy with the identified coefficients. In a second run, the roles invert as the first subset is used for evaluation and the second subset for identification. Hence, 2-fold cross-validation returns two estimates of accuracy. This process is repeated five times (there are five random partitions), giving out ten estimates of accuracy. Their mean value is reported as the model's estimation (prediction) accuracy. The 95% confidence intervals are also reported where appropriate. They equal to two standard deviations of the ten results returned by the cross-validation method. The 5x2cv method has been chosen for its low estimation variance and because it directly measures the variation of the results due to the choice of the training set.

Three validation metrics are considered: RMSE, IQR, and $r^2$. The RMSE (root-mean-square error) is a scale-dependent measure of accuracy. Let $\theta$ be a set of observed values and $\hat{\theta}$ be a set of estimated (resp. predicted) values. The RMSE is defined in this text as

$$\mathrm{RMSE} = \sqrt{\mathrm{E}[(\hat{\theta} - \theta)^2]}. \tag{8.1}$$

where $\mathrm{E}[\cdot]$ is the expectation operator. It combines information about the average error with information about the variability of errors. It also punishes more heavily larger errors which makes it sensitive to outliers.

The IQR (inter-quartile range) is used as the outlier-insensitive complement to

TABLE 8.1: $\beta$ coefficients for (8.5).

| | EV† | ICEV‡ |
|---|---|---|
| $\beta_0$ | -0.7123 | 0.9580 |
| $\beta_1$ | -2.2703$\cdot 10^{-2}$ | -6.3418$\cdot 10^{-2}$ |
| $\beta_2$ | -3.2109$\cdot 10^{-3}$ | -1.1529$\cdot 10^{-3}$ |
| $\beta_3$ | 1.0617$\cdot 10^{-4}$ | 7.0035$\cdot 10^{-5}$ |
| $\beta_4$ | -5.8308$\cdot 10^{-7}$ | -4.1114 $\cdot 10^{-7}$ |
| $\beta_5$ | 2.3320 | 1.6148 |

† electric vehicle
‡ conventional vehicle

RMSE. It is a robust metric of variability defined in this text as a measure of variability of residuals[1]: as the difference between 75-th and 25-th percentiles of the residuals $(\hat{\theta} - \theta)$, formally

$$\mathrm{IQR} = Q_3 - Q_1 \tag{8.2}$$

where $Q_3$ is the third quartile (75-th percentile) and $Q_1$ is the first quartile (25-th percentile) of the residuals. Unlike RMSE, it does not combine information about the average error and its variability: it measures only the variability.

The $r^2$ is a coefficient of determination. It measures how well are the observations replicated by the model based on the proportion of total variation in outcomes explained by the model. Its definition considered in this text reads

$$r^2 = \frac{\mathrm{E}[(\hat{\theta} - \theta)^2]}{\mathrm{E}[(\theta - \mathrm{E}[\theta])^2]} \tag{8.3}$$

It indicates how much of the variability between the two variables has been accounted for with the proposed model. It is used as a scale independent complement to the other metrics.

## 8.1 Baseline energy consumption models

The consumption models proposed in Chapter 5 are compared to three baseline models. Two of these models were proposed in the literature. They were chosen such that they are representative of the main approaches used today: one model is based on longitudinal consumption model, the other on regression analysis. The third baseline model assigns each edge in the road network the mean energy consumption observed there. It is called "MEC" in this work as short version of "Mean Energy Consumption". It is an interesting theoretical model to study.

The first baseline model was proposed in Barth et al., 2007 and revisited in Boriboonsomsin et al., 2012. The authors use Comprehensive Modal Emission Model (CMEM) to estimate both fuel consumption and pollutant emissions. The CMEM is a microscopic emission and fuel intake model (validated in Barth et al., 2001). The authors approximate the fuel consumption (denoted $f$) in grams per mile as

$$\ln(f) = \beta_0 + \beta_1\bar{v} + \beta_2\bar{v}^2 + \beta_3\bar{v}^3 + \beta_4\bar{v}^4 + \beta_5\alpha \tag{8.4}$$

[1] A note on the difference between errors and residuals: the error is the difference between an observed value and the unobservable true value. The residual is an observable estimate of the unobservable statistical error.

where $\ln(\cdot)$ denotes natural logarithm, $\beta_0 \ldots \beta_5$ are coefficients, $\bar{v}$ is the average speed and $\alpha$ is the road grade. The reasoning behind the choice of the terms in (8.4) is not clearly explained to the reader. An adapted version of this model is considered in this study: it considers energy consumption $E^b/l$ in Watt-hours per meter instead of fuel consumption in grams per mile as the independent variable. The solution for $E^b$ reads

$$E^b = l e^{\beta_0 + \beta_1 \bar{v} + \beta_2 \bar{v}^2 + \beta_3 \bar{v}^3 + \beta_4 \bar{v}^4 + \beta_5 \alpha} \tag{8.5}$$

The authors obtained the $\beta$-coefficients using multivariate nonlinear regression on a large set of recorded trips. Table 8.1 lists the regression coefficients used in this study. The $\beta$-coefficients were identified for the two reference vehicles (Section 7.2). The identification process has, however, proven difficult as it did not converge when performing nonlinear regression on (8.5). The identification was done instead with the least-squares method on a linearized model.

The second model was proposed in Juřík et al., 2014. It is based the longitudinal model (Section 5.1). The authors consider losses incurred by altitude changes, friction, and aerodynamic drag. Powertrain efficiency and heating in friction brakes is neglected. Their model reads

$$E^j = \begin{cases} E_r^j + (\gamma - 1)E_p^j & \text{if } E_p^j \leq 0 \\ E_r^j & \text{if } E_p^j > 0 \end{cases} \tag{8.6}$$

where $E_r^j$ models rolling friction and aerodynamic losses, $E_p^j$ is the change in vehicle's potential energy and $\gamma$ is a constant representing recuperation efficiency of the vehicle. The $E_r^j$ and $E_p^j$ are given as

$$E_r^j = \frac{\rho}{2} A_f C_d \bar{v}^2 l + m g \mu_r l \cos(\alpha) \tag{8.7}$$

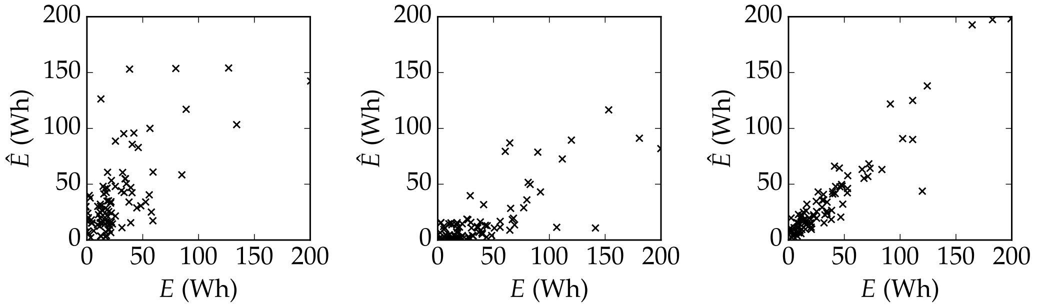
$$E_p^j = m g l \sin(\alpha) \tag{8.8}$$

where $\bar{v}$ is the average speed, and $l$ is the road length. Constants $\rho$, $A_f$, $C_d$, $m$, $\mu_r$ are vehicle-specific and $g$ is the gravitational acceleration constant. It was necessary to adapt this model as the description of the reference vehicles (Section 7.2) does not match the one used by the authors. Equation (8.7) is replaced with

$$E_r^j = (c_a \bar{v}^2 + c_b \bar{v} + c_c) l \tag{8.9}$$

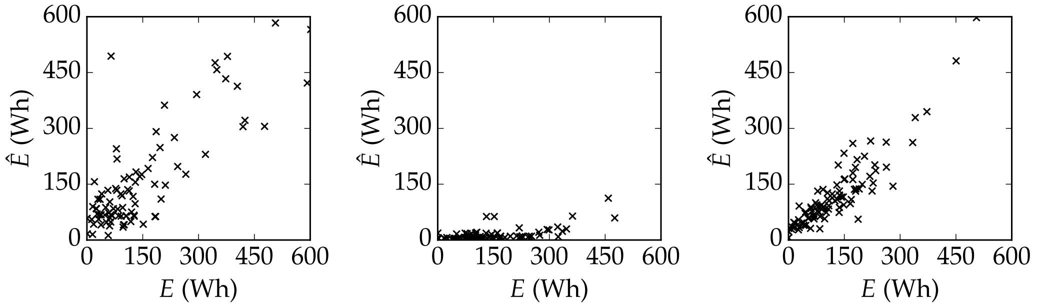
where $c_a$, $c_b$ and $c_c$ are coast-down coefficients described in Section 5.1.1 and given in Table 7.2.

The third model is the mean energy consumption (MEC) model. As the name suggests, it assigns each edge in the road network graph a cost that is the mean observed consumption there. The native traffic dataset contains recorded consumption for every passing vehicle on every edge in the road network graph. It suffices to gather all recorded consumptions for each edge and to take their averages as the costs. There are two sets of costs: one for the electric vehicle and another one for the conventional vehicle. They were computed on the same recorded trips, only with different vehicle models. The interest in this model is more theoretical than practical. Its implementation uses recorded trips of all vehicles in the simulation (over 200,000 vehicles). The mean consumptions are then as close to their respective population means as possible, given the limited road network capacity. However difficult



(A) Barth et al., 2007 (B) Juřík et al., 2014 (C) MEC

FIGURE 8.1: Relation between consumptions according to the three baseline models (vert. axis) and the consumption model of the electric vehicle (horiz. axis).



(A) Barth et al., 2007 (B) Juřík et al., 2014 (C) MEC

FIGURE 8.2: Relation between consumptions according to the three baseline models (vert. axis) and the consumption model of the conventional vehicle (horiz. axis).

it might be to use such data-intensive model in a real-world, it is still interesting because it is an unbiased estimator of consumption. The eco-routing with this model shows the potential of time-independent eco-routing. It was proposed in the literature by Andersen et al., 2013. The authors possess a large collection of recorded trips in Denmark. Their data come from GPS receivers and on-board diagnostic interfaces. They simply assign each road in Denmark the average consumption observed there. Even the MEC is formally identical to Andersen et al., 2013, they are considered distinct in this work. The MEC would not perform as well in practice as it does in simulations where it can use records from all vehicles in the road network and where all vehicles are assumed to have identical properties.

The validation of the three baseline models was done on the same dataset as the proposed models to ensure their comparability. The accuracy was evaluated on individual edges in the road network graph, not on complete trips. The consumptions estimated with the baseline models were compared to the "true" (reference) consumptions.

Table 8.2 lists the results. Each line describes one baseline model. Both reference vehicles are considered. The results for the electric vehicle are listed in columns 2, 3 and 4. The results for the conventional vehicle are listed in the last three columns (columns 5, 6 and 7). The RMSE, IQR and $r^2$ is listed for each. The RMSE and IQR are in Watt-hours, $r^2$ is unit-less. Figures 8.1 and 8.2 show scatter plots that reveal relationships between estimated consumptions using the three baseline models and their references. Figure 8.1 considers the electric vehicle; Figure 8.2 considers

TABLE 8.2: Baseline models: results.

| | EV[†] | | | ICEV[‡] | | |
|---|---|---|---|---|---|---|
| | RMSE[♣] | IQR[♣] | $r^2$ | RMSE[♣] | IQR[♣] | $r^2$ |
| Barth et al., 2007 | 60.60 | 30.79 | 0.695 | 162.75 | 87.72 | 0.838 |
| Juřík et al., 2014 | 47.71 | 27.53 | 0.811 | 358.23 | 130.60 | 0.217 |
| MEC | 24.83 | 12.54 | 0.949 | 73.63 | 41.62 | 0.967 |

[†] Electric vehicle
[‡] Conventional vehicle
[♣] Units: Wh

the conventional vehicle. In the ideal case, all the samples (depicted with crosses) would be on the diagonal from bottom left corner to the top right corner. This situation would indicate perfect estimation. Each figure shows 100 randomly selected samples. The figures have a square shape, and their axes share the same scale to make comparison visually easy.

The first model (proposed by Barth et al., 2007) has shown RMSE of 60.6 Wh and $r^2$ of 0.7 for the electric vehicle and RMSE of 162.8 Wh and $r^2$ of 0.84. The second model (proposed by Juřík et al., 2014) can be expected to show high RMSE as it neglects powertrain inefficiency and RMSE is scale-dependent. Such behavior is not necessarily a problem for eco-routing as long as the estimates correlate positively with the observed consumptions. It is the case for the electric vehicle (the $r^2$ is 0.81), but not for the conventional vehicle (the $r^2$ is 0.21). Specifically for the conventional vehicle, the results are worst between the baseline models. That is not entirely surprising as the model was designed primarily for electric vehicles. The MEC shows the highest accuracy. The RMSE is 24.8 Wh and $r^2$ is 0.95 for the electric vehicle. For the conventional vehicle, the RMSE is 73.6 Wh, and $r^2$ is 0.97.

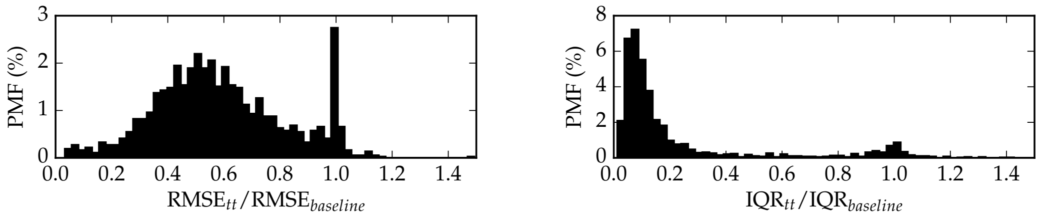
## 8.2 Travel time

There are four coefficients in the travel time model (5.38) to be identified: $\mu$, $\vartheta$, $\kappa_{\mathbb{G}}$ and $\kappa_{\mathbb{S}}$. The $\mu$ and $\vartheta$ are real-valued, the $\kappa$-coefficients are positive integers.

There are 9,770 edges in the road network graph, the tuple $(\mu, \vartheta, \kappa_{\mathbb{G}}, \kappa_{\mathbb{S}})$ needs to be identified for every edge. The sets $\mathbb{S}_e$ and $\mathbb{G}_e$ for each edge were extracted from the native traffic dataset, except for 267 edges for which there was not enough data. The travel time on these roads was approximated as the road length divided by the speed limit. The function $t_{tg}$ was implemented with pre-recorded traffic light states. 8,268 edges out of the 9,770 represent roads without traffic lights. The parameter $\vartheta$ is set to zero in these cases. It reduces the model to a $k$-NN regressor as the parameters $\kappa_{\mathbb{G}}$ and $\mu$ lose their influence.

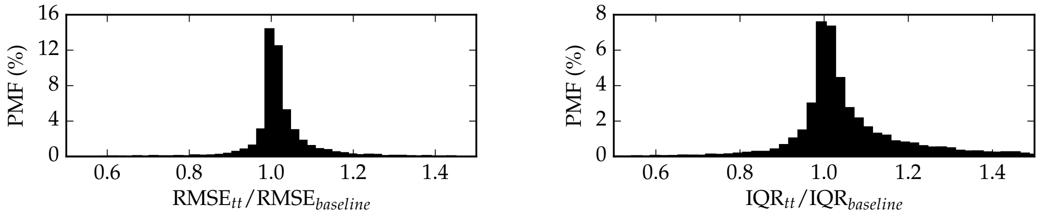
The travel time models were identified through minimization of RMSE. It is a challenging task since two of the four coefficients are integers. The model is also not linear, convex or smooth. Gradient-based numerical solvers cannot be used since numerically computed gradients are not reliable. This unreliability is due to uncertainty in the RMSE estimates that are minimized: they are estimated with the 5x2cv cross-validation method whose outcome depends on a randomized shuffling of the training dataset.

The identified models are approximate as finding the optimal solution was computationally too demanding to be finished in acceptable time. Based on experimentation I have preselected a set with 42 candidate combinations of coefficients $\kappa_{\mathbb{S}}$, $\kappa_{\mathbb{G}}$ and minimized $\mu$ and $\vartheta$ for each combination. The basinhopping method (Wales,



(A) RMSE ratio (B) IQR ratio

FIGURE 8.3: Histogram of the RMSE and IQR of the proposed travel time models compared to the mean travel time model. Results for signalized intersections.



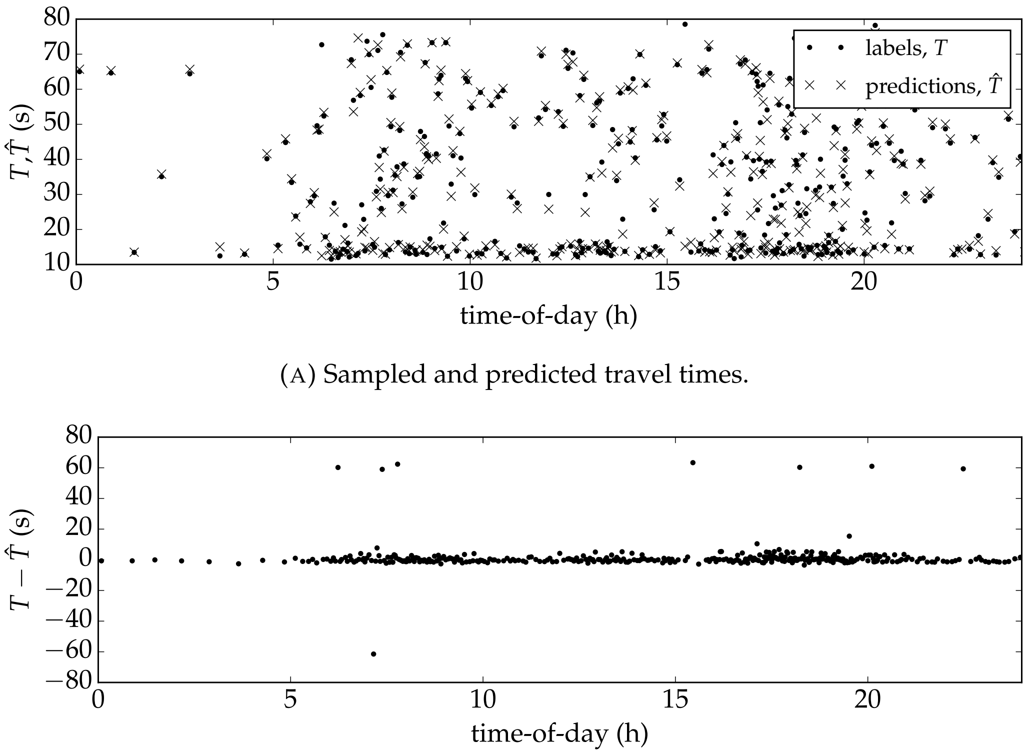
(A) RMSE ratio (B) IQR ratio,

FIGURE 8.4: Histogram of the RMSE and IQR of the proposed travel time models compared to the mean travel time model. Results for right-of-way intersections.

2003) together with the Powell solver (Powell, 1964) was used to do the minimization. The Powell'method is an algorithm for finding a local minimum of a function. It does not use the gradient of the minimized function to search for the local minimum. It was chosen for this property over more standard methods. The basinhopping method is used to overcome local minima. It runs the Powell solver several times, always with some random perturbation on the initial guesses of the two parameters $\mu$ and $\vartheta$ and returns the lowest local minimum found.

The results of cross-validation for all 9,770 models are summarized in figures 8.3 and 8.4. The Figure 8.3 shows results for roads with traffic lights, Figure 8.4 shows results for roads without traffic lights. The subfigures show histograms of ratios of RMSE (resp. IQR) compared to the RMSE (resp. IQR) of the mean travel time. The results show a considerable reduction of both RMSE and IQR on the roads with traffic lights. The RMSE was reduced by a factor 1.68 and IQR by a factor 4.34 on average. The results on the roads without traffic lights show that the proposed travel time model does not offer any advantage over the mean travel time model.

A specific example of how the model predicts the travel time is in Figure 8.5. The dots in subfigure 8.5a are observations, crosses are predictions. Subfigure 8.5b shows prediction residuals. Most of the residuals are below ten seconds but note the outliers that are on the level of ±60 seconds. These errors happen when the vehicle reaches the traffic light close to the time when it was changing its color. This behavior justifies the use of the IQR metric: it is not susceptible to these outliers.



(A) Sampled and predicted travel times.

(B) Prediction errors.

FIGURE 8.5: An example of sampled vs. predicted travel times on a road -30256#0 when continuing to -30256#1 (SUMO road names).

## 8.3 The consumption model

The assumption of the proposed consumption model is that the powertrain's output power is an affine function of its input power. It was proposed in Section 5.4 as a way to solve the longitudinal consumption model (5.14) without the dependency on braking energy (which is argued hard to predict).

Six variants of this model are considered. They are listed in Table 8.3. The difference between them is in the parameters provided to (5.29). It has five independent variables: travel time $T$ , speed variance $\sigma^2$, speed skew $b$, initial speed $v(0)$ and final speed $v(T)$. A model for travel time prediction was proposed in Section 5.5. There is no model proposed for the other parameters. Two generic ways to model them are considered: the parameter is either set to zero or set to the average of past observations.

The last five columns in the Table 8.3 define what kind of values are supplied in place of the parameters. The "✓" indicates that the observed value was provided to the model. The "∼" indicates that the parameter was replaced with a mean value. The "–" indicates that the parameter was replaced with a zero; the "⋆" indicates that predicted travel time based on model (5.38) was used. The six models were chosen such that each model is interesting in its own right. The first model "model 1" uses observed values for its parameters. This approach is useful only for estimation, not for prediction, but it allows to quantify the error introduced by the powertrain assumption itself. The other models confound modeling errors together with errors in the parameters. The models 2 to 6 consider errors introduced by various relaxations on the five parameters. The second model "model 2" considers predicted travel

TABLE 8.3: Consumption model variants.

| | $T$ | $\sigma^2$ | $b$ | $v(0)$ | $v(T)$ |
|---|---|---|---|---|---|
| model 1 | ✓ | ✓ | ✓ | ✓ | ✓ |
| model 2 | $\star$ | ✓ | ✓ | ✓ | ✓ |
| model 3 | $\star$ | $\sim$ | $\sim$ | ✓ | ✓ |
| model 4 | $\star$ | – | – | – | – |
| model 5 | $\sim$ | $\sim$ | $\sim$ | $\sim$ | $\sim$ |
| model 6 | $\sim$ | – | – | – | – |

Keys: ✓correct value, $\sim$ full-day average,
– zero, $\star$ predicted with (5.38)

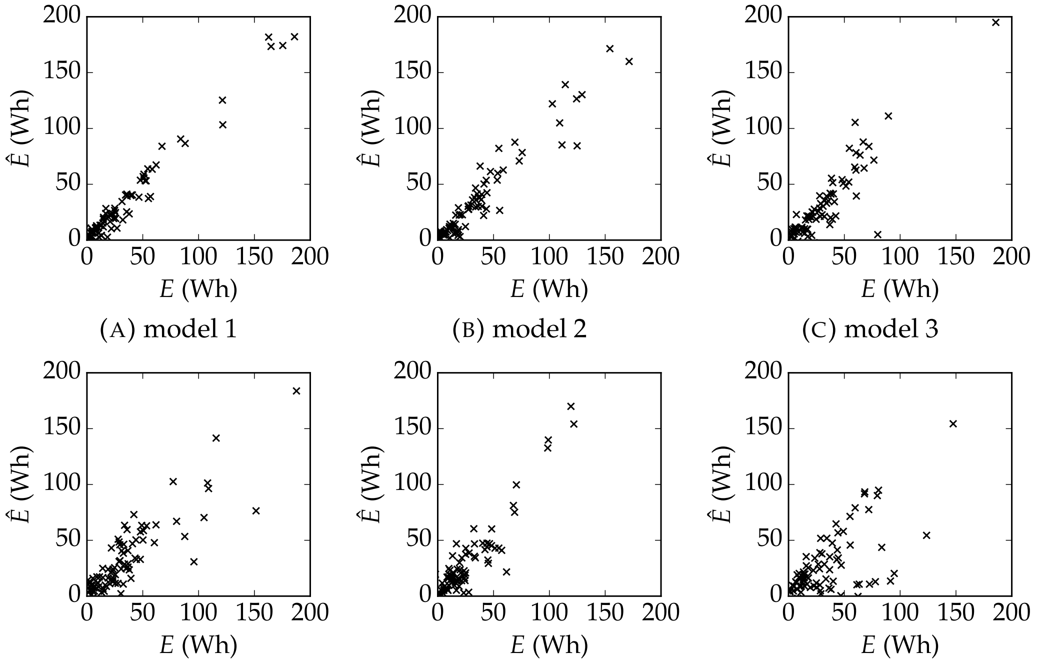
time instead of the observed travel time. This allows quantifying how the model performs with the proposed travel time model: how the errors in travel time affect its accuracy. The third model "model 3" further considers the speed variance and skew to be constants equal to the average of historically observed values. The fourth model "model 4" considers all parameters to be zeros except travel time. The fifth model "model 5" considers all parameters to be historically observed averages. The sixth model "model 6" considers the travel time to be the historically observed average and other parameters to be zeros. The models 1-3 cannot be used in eco-routing since they use information that is not available when planning the trip. The models 4-6 can be used in eco-routing. The model 4 is time-dependent; models 5 and 6 are time-independent.

The native traffic dataset was used to identify and cross-validate these models, similarly like it was done with the baseline models. Note that the results are likely sensitive to the specific topology of the road network in Luxembourg. For this reason, the accuracy of the six models should not be regarded as absolute but relative to the results reported for the baseline models (Table 8.2).

The identification of the six models requires identifying coefficients $P_0$ and $e$ for each. The $P_0$ represents the power draw that does not depend on vehicle's instantaneous speed. The $e$ is powertrain efficiency. Guzzella and Sciarretta, 2005 discuss two approaches to their identification for the original Willans's model. Willans considers $P_0$ and $e$ to be functions of shaft angular velocity. Under the assumption that they are constant the standard approach is to fit a line between observed and estimated consumptions. A different approach is proposed here. The Equation (5.29) has two independent variables: the energy on wheels $E_w$ and the travel time $T$. The coefficients $P_0$ and $e$ can be identified by fitting a plane $c_1 E_w + c_2 T = E$. Then $e = 1/c_1$ and $P_0 = c_2/c_1$ . Since this is a linear equation, the identification of $c_1$ and $c_2$ can be accomplished with the linear least-squares method.

The model (5.29) for the conventional (internal combustion engine) vehicle can output negative energy consumption. This is something the powertrain is not capable of as it cannot recuperate energy. To improve model accuracy, the used consumption model is $\max(\star, 0.0)$, where $\star$ is the energy consumption according to (5.29). This approach can always be adopted when the powertrain under consideration is unable to recuperate. The effect is that any negative consumption is trimmed to zero.

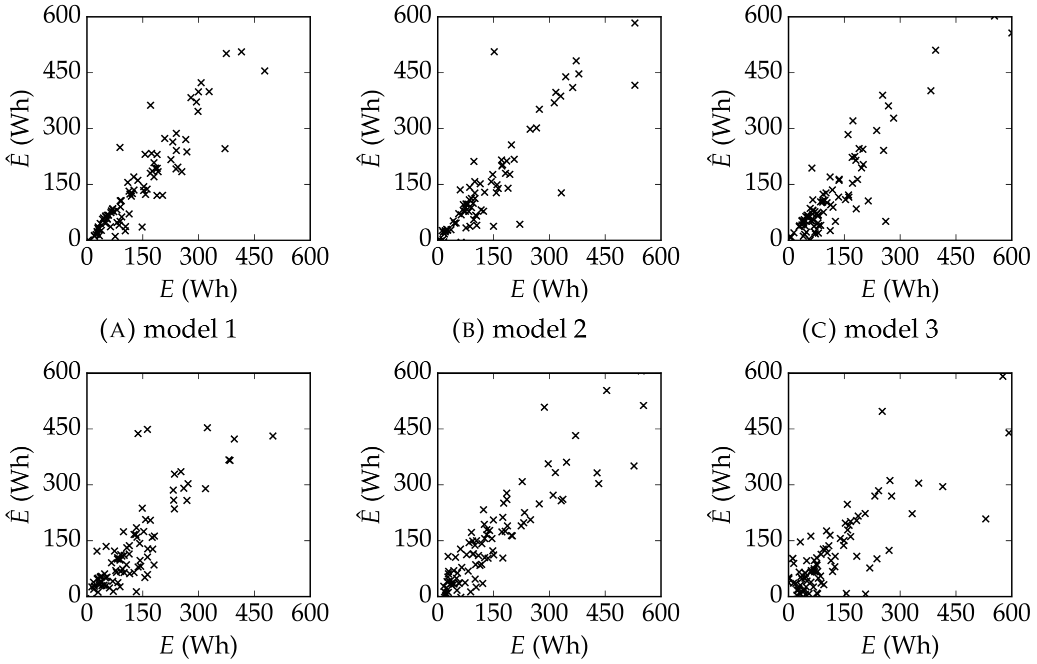
Table 8.4 lists identified coefficients and cross-validation results. All six variations of the proposed consumption model (see Table 8.3) are considered for both reference vehicles. First six lines are the results for the electric vehicle, last six lines for the conventional vehicle. The parameters $P_0$ and $e$ were identified for each model separately. Their values are listed in columns 3 and 4. The cross-validation results are in the last three columns: RMSE, IQR and $r^2$. They list the mean observed values



(A) model 1 (B) model 2 (C) model 3

(D) model 4 (E) model 5 (F) model 6

FIGURE 8.6: Scatter plots for the six versions of the proposed consumption model compared to the reference consumption of the electric vehicle.



(A) model 1 (B) model 2 (C) model 3

(D) model 4 (E) model 5 (F) model 6

FIGURE 8.7: Scatter plots for the six versions of the proposed consumption model compared to the reference consumption of the conventional vehicle.

TABLE 8.4: The proposed consumption model: results.

| | | $e$ (-) | $P_0$(W) | RMSE (Wh) | IQR (Wh) | $r^2$ (-) |
|---|---|---|---|---|---|---|
| EV[†] | model 1 | 0.889 | 2104.63 | 17.03 ± 0.11 | 7.49 ± 0.03 | 0.976 ± 0.0001 |
| | model 2 | 0.884 | 2263.09 | 23.67 ± 0.65 | 9.48 ± 0.02 | 0.954 ± 0.002 |
| | model 3 | 0.886 | 2308.95 | 23.99 ± 0.58 | 9.57 ± 0.03 | 0.952 ± 0.002 |
| | model 4 | 0.875 | 2111.75 | 27.91 ± 0.64 | 16.84 ± 0.02 | 0.936 ± 0.003 |
| | model 5 | 0.880 | 2344.17 | 26.61 ± 0.05 | 16.14 ± 0.01 | 0.941 ± 0.0001 |
| | model 6 | 0.829 | 1895.74 | 28.62 ± 0.11 | 16.82 ± 0.01 | 0.932 ± 0.0002 |
| ICEV[‡] | model 1 | 0.337 | 6534.49 | 74.25 ± 0.44 | 47.82 ± 0.06 | 0.967 ± 0.0001 |
| | model 2 | 0.338 | 6922.07 | 95.2 ± 1.8 | 52.86 ± 0.18 | 0.945 ± 0.002 |
| | model 3 | 0.339 | 6984.72 | 95.76 ± 2.15 | 52.86 ± 0.19 | 0.944 ± 0.002 |
| | model 4 | 0.330 | 6486.55 | 98.99 ± 0.51 | 65.6 ± 0.06 | 0.941 ± 0.0001 |
| | model 5 | 0.336 | 7187.53 | 90.44 ± 0.13 | 63.77 ± 0.14 | 0.950 ± 0.0003 |
| | model 6 | 0.311 | 6144.69 | 90.24 ± 0.28 | 60.71 ± 0.08 | 0.951 ± 0.0002 |

[†] Electric vehicle
[‡] Conventional vehicle

together with a confidence interval. The values of $P_0$ are in Watts, RMSE and IQR are in Watt-hours, $e$ and $r^2$ are unit-less. The reported RMSE, IQR and $r^2$ values can be directly compared to the values reported for the baseline models (See Table 8.2).

Figures 8.6 and 8.7 show scatter plots that reveal relationships between estimated consumptions using models 1-6 and the reference consumption. Figure 8.6 considers the electric vehicle; Figure 8.7 considers the conventional vehicle. All samples (depicted with crosses) would be on the diagonal from bottom left corner to the top right corner in the ideal case. This would indicate perfect estimation (resp. prediction). Each figure shows 100 randomly selected samples. The figures have a square shape, and their axes share the same scale to make comparison visually easy. They can be compared to the scatter plots for the baseline models in figures 8.1 and 8.2.

The powertrain efficiency $e$ for the electric vehicle is about 88% (except for model 6), and its speed independent losses $P_0$ are about 2.2 kW (also except for model 6). On the other hand, the efficiency of the conventional vehicle is lower (about 33%) and its static losses are higher with $P_0$ close to 6.5 kW, in most cases. These results correspond to the properties of the two vehicles: the electric vehicle is light, simple and efficient; the conventional vehicle is heavy, complex and inefficient. The cross-validation results have shown superior accuracy with respect to the baseline models proposed in Barth et al., 2007 and Juřík et al., 2014. These models require only the knowledge of the mean speed. The model 6 uses the same information and outperforms both by a considerable margin. The results are not so strong when compared to the MEC model, however. Only the models 1, 2 and 3 for the electric vehicle outperform it.

The model 1 is provided the correct (observed) parameters. Hence while this model can be used for consumption estimation, it cannot be used for prediction in eco-routing. It is nevertheless an interesting model as it allows to quantify the error introduced by the powertrain assumption alone (see Section 5.4). The results for the electric vehicle have shown superior accuracy with respect to all three baseline models. The results for the conventional vehicle have shown superior accuracy with respect to Barth et al., 2007 and Juřík et al., 2014 and comparable accuracy to the MEC model. As discussed in Section 5.4, the assumption is better suited for the electric vehicle than for the conventional vehicle. The results support that: the coefficient of determination is higher for the electric vehicle. Note also that the estimated efficiency $e$ of the electric vehicle is 88.9% while the actual powertrain efficiency is 86.3% (90.9% motor efficiency multiplied with 95% transmission efficiency). There is a discrepancy, but it is to be expected. As discussed in Section 5.4 the efficiency when braking becomes the reciprocal of the traction efficiency and hence larger than one. The effect is that the estimate of $e$ is artificially higher than vehicle's average traction efficiency.

The model 2 considers travel time predicted with (5.38); the other parameters are observations like in model 1. It injects the error from travel time predictor to the consumption model which allows quantifying the related penalty in model accuracy. The model 3 additionally considers the parameters $\sigma^2$ and $b$ to be the average of previously observed values. It allows quantifying the additional penalty in model accuracy due to average $\sigma^2$ and $b$. Both models are like model 1 useful for estimation, but not for prediction. For the electric vehicle, both models show superior performance with respect to all three baselines. For the conventional vehicle, their performance is inferior with respect to the MEC model. The results show a considerable gap in accuracy between models 1 and 2 (from 17.03 ± 0.11 Wh to 23.67 ± 0.65 Wh for the electric vehicle), but only a small difference in accuracy between models 2 and 3 (they are within the margin of their confidence intervals). These results indicate that the error introduced by the travel time prediction has a considerable effect on consumption prediction accuracy. On the other hand, the replacement of the $\sigma^2$ and $b$ parameters with their averages did not result in significant reduction of accuracy.

The model 4 differs with respect to models 1, 2 and 3 in that it can be used for eco-routing. The travel time parameter is predicted with the proposed travel time model, and the other parameters are zeros. The results show a reduction of accuracy between the models 3 and 4 (from 23.99 ± 0.58 Wh to 27.91 ± 0.64 Wh for the electric vehicle). This is likely because the difference in vehicle's kinetic energy is no longer accounted for.

The models 5 and 6 are time-independent. Unlike the other models, all of their parameters are either average values or constants. The model 5 takes average value in place of all parameters. The model 6 requires average travel time; its other parameters are assumed to be zero. These models are interesting because they can be used as drop-in replacements for consumption models in existing eco-routing applications. The model 6 uses the same information as Barth et al., 2007 and Juřík et al., 2014: the average travel time[2]. The model has shown considerably better accuracy with respect to both: the observed RMSE of the model 6 (for the electric vehicle) is 28.62 ± 0.11 Wh, while the two baseline models reached at 60.60 Wh and 47.71 Wh respectively. The accuracy of model 6 is nevertheless inferior to the MEC model as it has shown RMSE of 24.83 Wh. In the case of the electric vehicle, the model 5 is more accurate with observed RMSE lower by 2 Wh with respect to the model 6. There is no significant difference in accuracies of the two models in the case of the conventional vehicle: the differences are within the margins imposed by the confidence intervals.

## 8.4 Summary

This chapter is about identification and validation of baseline consumption models, the travel time model, and the consumption model proposed in Section 5.4. Identification amounts to the identification of the coefficients of the studied models. The

[2] Note: both Barth et al., 2007 and Juřík et al., 2014 use average speed, which is inversely proportional to average travel time via road length (a known variable).

data required to identify them come from the native traffic dataset discussed in Section 7.4. Validation amounts to the estimation of prediction accuracy of the studied models. The used metrics of accuracy and goodness-of-fit are RMSE (root-mean-square-error), IQR (inter-quartile range) and $r^2$ (coefficient of determination). The cross-validation was done with 2-fold cross-validation repeated five times.

The baseline models are introduced, identified and validated in Section 8.1. The consumption models published by Barth et al., 2007 and Juřík et al., 2014 are considered as baselines. Additionally, an energy consumption model whose consumption estimates are sample mean values is considered as a complement to the two baselines models. It is named “MEC” (Mean Energy Consumption) in this work. The Barth et al., 2007 and Juřík et al., 2014 were chosen as representatives of the main two approaches to energy consumption modeling used in eco-routing literature. Barth et al., 2007 proposes a consumption model based on regression analysis and Juřík et al., 2014 proposes a model derived from the longitudinal consumption model (introduced in Chapter 5).

The travel time model that was proposed in Section 5.5 is identified and evaluated in Section 8.2. There are four coefficients in the travel time model (5.38) to be identified: $\mu$, $\vartheta$, $\kappa_{\mathbb{G}}$ and $\kappa_{\mathbb{S}}$. The $\mu$ and $\vartheta$ are real-valued, the $\kappa$-coefficients are positive integers. A unique set of coefficients was identified for every edge in the road network graph. The results show a considerable reduction of both RMSE and IQR on the roads with traffic lights. The RMSE was reduced by a factor 1.68 and IQR by a factor 4.34 on average. The results for the roads without traffic lights show that the proposed travel time model does not offer any advantage over the mean travel time model.

The consumption model that was proposed in Section 5.4 is identified and evaluated in Section 8.3. Six variants were studied. They are named "model 1" to "model 6". The difference between them is in the independent variables provided to the consumption model. The model 1 uses observed values. The model 2 uses predicted travel time (using model (5.38)) instead of the observed travel time. The model 3 uses predicted travel time and average values in place of parameters $\sigma^2$ and $b$. The model 4 uses the predicted travel time while all other independent variables are set to zero. The model 5 uses average values in place of all independent variables. The model 6 uses average travel time and assumes all other variables are zero. The models 1, 2, 3 and 4 are time-dependent, models 5 and 6 are time-independent. The models 1 to 3 use observed values and cannot, therefore, be used before the trip was recorded. This prevents them from being used in eco-routing. Models 4 to 6 does not rely on observations and can be used in eco-routing. All six model variations have shown superior accuracy on both reference vehicles with respect to Barth et al., 2007 and Juřík et al., 2014. Especially the model 6 uses the same information as these baseline models and shows considerably improved accuracy. The accuracy is, however, inferior with respect to the mean energy consumption (MEC) model for most variations of the consumption model. Only models 1, 2 and 3 used on the electric vehicle have shown superior performance with respect to MEC. In case of the conventional vehicle, the accuracy of model 1 is comparable to the accuracy of MEC, the models 2 to 6 have shown worse accuracies.

# Chapter 9

# Simulations and results

Results of simulated eco-routing experiments are presented in this chapter. Three eco-routing methods are studied, all aim to minimize a single cost (energy consumption) without constraints. While the eco-routing methods proposed in this work support multiple costs and constraints, the used setup allows direct comparison to the baseline methods presented in Section 4.3. Eco-routing with the consumption model 6 (see Table 8.3) is studied in Section 9.1. This is a time-independent eco-routing with a consumption model that uses the same information as baseline models by Barth et al., 2007 and Juřík et al., 2014. Computational effort associated with the computation of path trees is studied in Section 9.2. It is a precursor to Section 9.3 which studies time-dependent adaptive eco-routing. Two variants are considered: with and without rerouting. Finally, the Section 9.4 gives an example how the versatility of the proposed eco-routing method can be leveraged to improve performance and to limit failures.

The summary of the results is in Table 9.1 at the end of this chapter. It is an extension of Table 4.1. All results are relative quantities (in percent) with eco-routes being compared either to the shortest paths or the fastest paths. Current literature allows both options. Both reference consumption models (electric vehicle and conventional vehicle; presented in Section 7.2) are considered. The last four columns in the table are from left to right the results (1) for the electric vehicle with eco-routes compared to the fastest paths; (2) for the conventional vehicle with eco-routes compared to the fastest paths; (3) for the electric vehicle with eco-routes compared to the shortest paths; (4) for the conventional vehicle with eco-routes compared to the shortest paths. The 95% confidence intervals are listed for the mean values given in the table. The table has six sections. Their meaning was first explained in Section 4.3. To reiterate:

1. $P$ and $\hat{P}$ is the performance and estimated performance, respectively. The $P$ is the relative amount of energy saved on a typical trip, $\hat{P}$ is the estimate of $P$ when the same consumption model is used for both eco-routing and its evaluation (see its definition given in Section 4.2.1). Negative values indicate losses rather than savings.

2. *"Probability of failure"* is the sample probability that the eco-route is less economical than the fastest (resp. shortest) path and that it requires longer travel time than the fastest (resp. shortest) path.

3. *"Eco-route same as reference"* is the portion of cases for which the eco-route is identical to the fastest (resp. shortest) path.

4. *"Mean savings"* is the average reduction of energy consumption on the eco-route when compared to the fastest (resp. shortest) path.

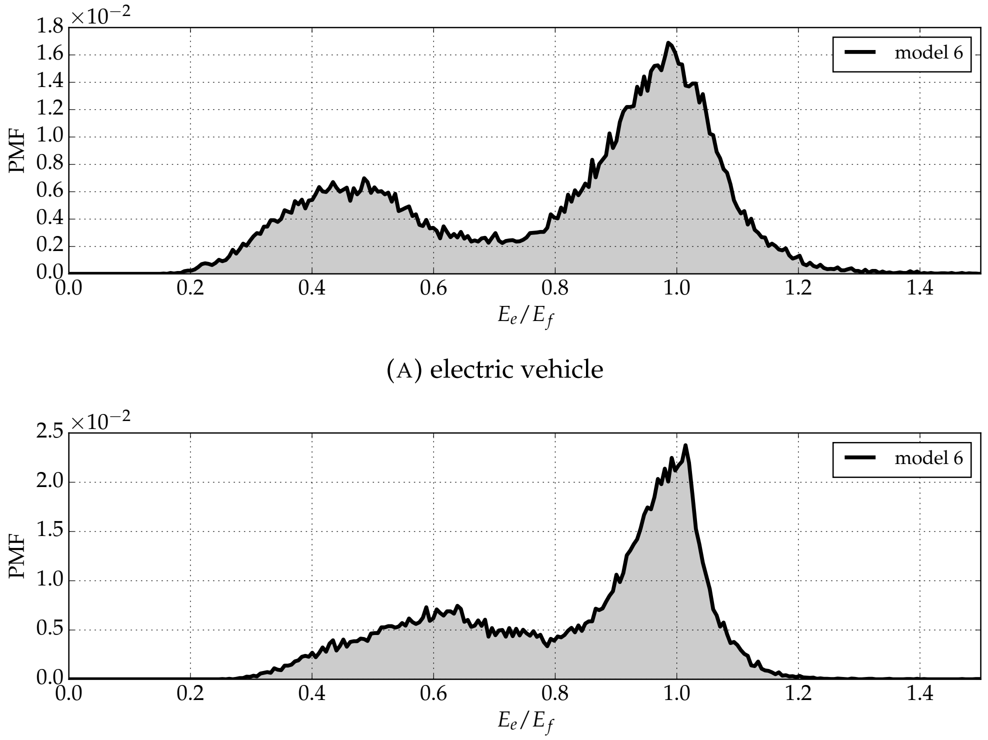
5. *"Mean travel time delay"* is the average increase in travel time on the eco-route when compared to the fastest (resp. shortest) path.

## 9.1 Eco-routing with model 6

Eco-routing with the model 6 (see Table 8.3) is studied in this section. This model is time-independent (like the baseline models). It requires only information about mean travel time on the roads in the road network. The model 6 is studied in Chapter 8, where it is cross-validated and compared to the baselines. The results have shown lower RMSE than models by Barth et al., 2007 and Juřík et al., 2014, and higher than the MEC model.

The goal is to evaluate this eco-routing method in terms of its performance $P$. The methodology is the same as with the baseline models. The framework proposed in Chapter 7 is used to conduct this experiment. The model 6 was used to assign a cost (energy consumption) to every node in the road network graph. Its only unknown parameter is the average travel time. This information was extracted from the native traffic dataset (discussed in Section 7.4). The eco-routes were computed for the same set $\mathbb{V}_{od}$ with 60,478 origin-destination pairs that were used for the evaluation of the baseline methods. The Dijkstra's algorithm was used for the routing. Negative costs were replaced with zeros. Every eco-route from the set of 60,478 was tried in the simulation that started at the same time (at midnight) from the same initial state. The simulation provided necessary information to compute the reference consumptions with the two reference consumption models (see Section 7.2). This allows comparing consumption on eco-routes to consumption on shortest and fastest paths.

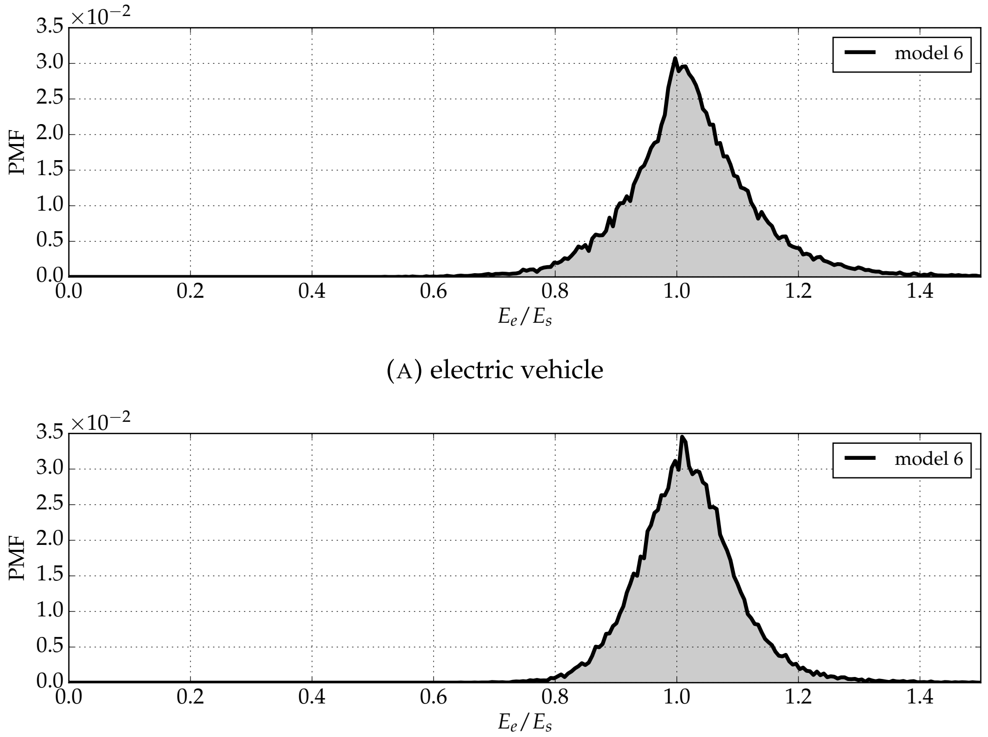
The results are in figures 9.1, 9.2, 9.3, and in Table 9.1. The distributions of savings are shown in figures 9.1 and 9.2. It can be compared to the distributions of savings for the baseline methods in figures 4.3 and 4.4. For some fixed origin, destination and time of departure let $E_e$ be the energy consumption on the eco-route, $E_s$ the energy consumption on the shortest path and energy $E_f$ the consumption on the fastest path. The Figure 9.1 shows the distribution of savings on eco-routes with respect to fastest paths (the distribution of the ratios $E_e/E_f$). The Figure 9.2 shows the distribution of savings with respect to the shortest paths (the distribution of the ratios $E_e/E_s$). The area under the curve below 1.0 on the horizontal axis represents the cases for which the eco-routing method saved some energy while the area under the curve above 1.0 on the horizontal axis represents the cases for which the eco-route required more energy than the reference path. These are the cases in which the eco-routing method failed to save energy. An exceptionally high probability of the eco-route being identical to the shortest (resp. fastest) path was observed. This appears as a discontinuity in the distributions in figures 9.1 and 9.2. It was removed from the plots and reported instead in Table 9.1 where the results are summarized. The Figure 9.3 shows the correlation between the observed (reference) energy consumptions and the consumptions estimated with the model 6. Results for the electric vehicle are in Figure 9.3a, for the conventional vehicle in Figure 9.3b. It can be compared to the results for the baseline models in Figure 4.5. Each sample represents a single trip. The reference consumption on the eco-route (denoted $E_e$) is on the horizontal axis while the estimated consumption (denoted $\hat{E}_e$) is on the vertical axis. The perfect estimation would be achieved with all samples aligned on the black diagonal line. The samples above it are the cases when the consumption was overestimated, and the samples below it are the cases when it was underestimated.



(A) electric vehicle

(B) conventional vehicle

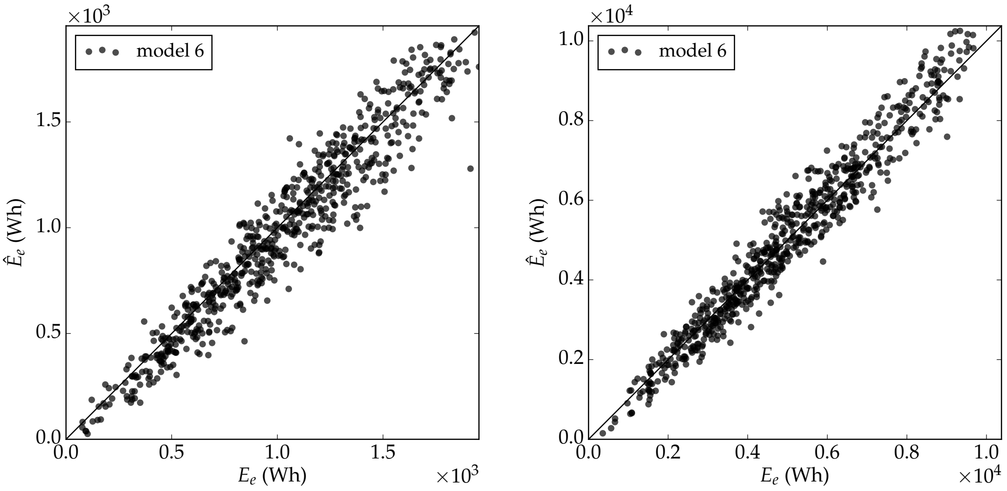
FIGURE 9.1: Eco-routing with model 6: distribution of savings relative to fastest paths.



(A) electric vehicle

(B) conventional vehicle

FIGURE 9.2: Eco-routing with model 6: distribution of savings relative to shortest paths.



(A) electric vehicle

(B) conventional vehicle

FIGURE 9.3: Eco-routing with model 6: correlation of estimated and reference energy consumptions.

The results in Table 9.1 show that eco-routing with the model 6 performs better than Barth et al., 2007 and Juřík et al., 2014, but fails to perform better than eco-routing with the MEC model and, perhaps more importantly, simple shortest-distance routing. This motivates the time-dependent adaptive routing studied in the following sections.

## 9.2 The path tree

The time-dependent adaptive eco-routing method proposed in Chapter 6 uses a pre-computed path tree with candidate eco-routes to choose from. The path tree computation time and its other properties are studied in this section. The goal is to assess the feasibility of this approach. Note that the path tree computation can be done offline, it is not a time-critical part of the system. The acceptable computation times can, therefore, reach to hundreds of seconds.

The tree is computed with the Algorithm 3; it is described in Section 6.3. The implementation of the path tree datastructure is discussed in Section 6.4. The algorithm draws a random cost for every node in the road network graph from known energy consumption distributions. The methodology for obtaining these distributions is described in Section 7.3.3. The Algorithm 3 uses internally the Dijkstra's algorithm to find random eco-routes, its implementation is discussed in Section 7.3.2. The Algorithm 3 does not specify the stop condition to be used. A simple limit on the number of iterations of the main loop in the Algorithm 3 is used in this experiment as a stop condition.

The tree computation with the exhaustive search (Algorithm 2) is not studied here as it is not used in the eco-routing experiments considered in this chapter. The randomized search is preferred over the exhaustive search. The computation time with the exhaustive search varied significantly between studied cases. This makes the choice of coefficients for pruning criteria difficult. The same setting can yield trees with a handful paths or trees with millions of paths. The randomized search allows better control over the tree size and computation time.

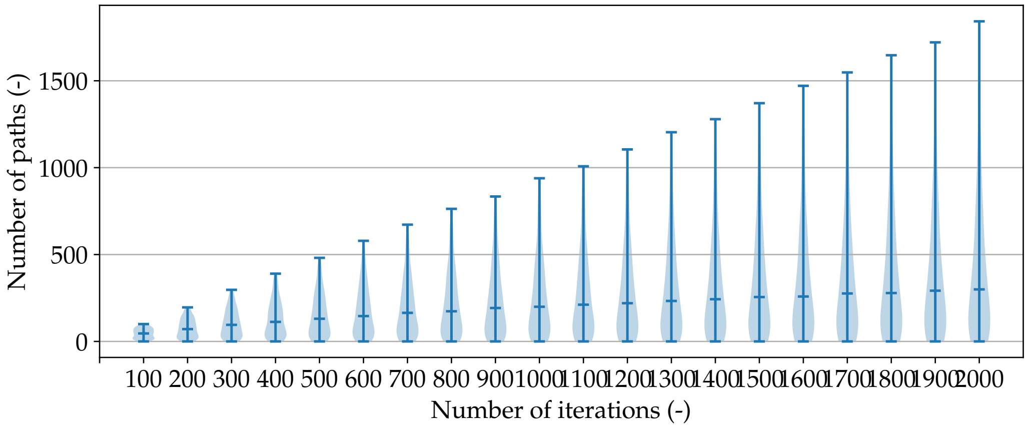


FIGURE 9.4: Number of paths in the path tree against the number of iterations.

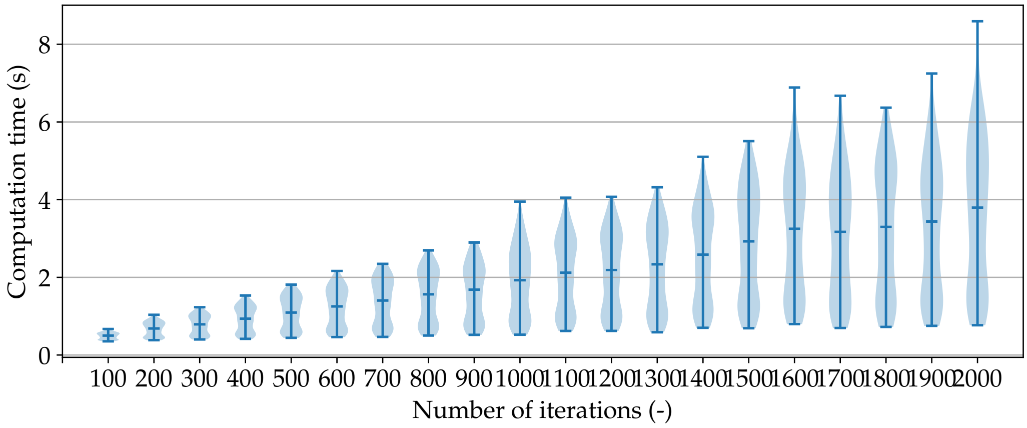


FIGURE 9.5: Path tree computation time against the number of iterations.

The experiment is designed as follows. First, a 300 random origin-destination pairs were selected. Then a tree was built for every origin-destination pair and for every limit on the number of iterations from the set $\{100, 200, \dots 2000\}$, which constitutes different stop conditions. The computation time, the number of found paths and the number of iterations before the shortest path is discovered is recorded for every created tree. The experiment was conducted in a single thread on a 64-bit personal computer running a Linux-based operating system. It had 16 GB of RAM and a four-core processor clocked at frequencies up to 2.2Ghz. The computer did not run out of memory during computation, meaning there are no delays in computation time due to memory swapping. The program that ran the experiment could have used the full capacity of a single core since the other three other cores did not saturate from running the other processes.

The results are in figures 9.4 and 9.5. They are violin plots which show for different stop conditions (different number of iterations) on the horizontal axis the number of identified paths (Figure 9.4) and the computation time (Figure 9.5). Similarly like box plots, the violin plots show for each category the median and the extrema. Additionally, they show the approximated probability density distributions overlaid over

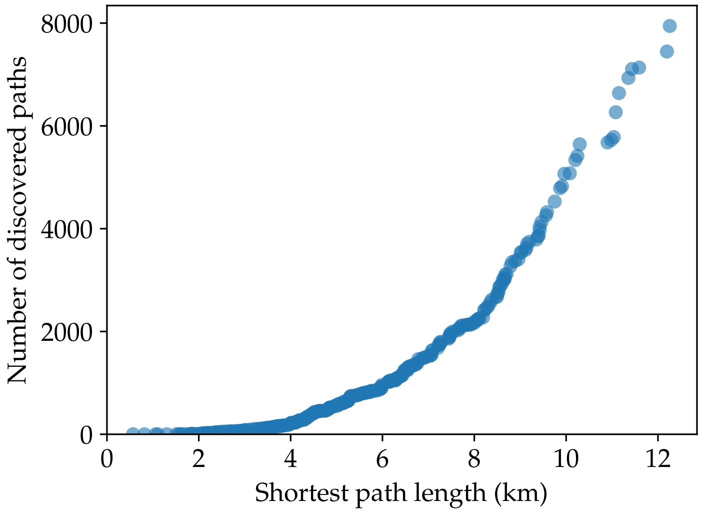


FIGURE 9.6: Trip length against the number of discovered paths.

the box plots. The distributions were approximated using kernel density estimation.

The number of found paths is the number of candidate eco-routes stored in the resulting path tree. It grows at varying paces depending on the road network and positions of the origin and destination, as can be seen in Figure 9.6[1]. In the worst observed case, there were 1850 paths after 2000 iterations, meaning that in 92.5% of iterations a new path was discovered and added to the tree. On the other hand, the distributions in the violin plot show that this was rare. The median grows at much slower pace with 50 paths after 100 iterations and 300 paths after 2000 iterations. The distributions show that for most origin-destination pairs used in this experiment there is only a handful of discovered paths (the distributions are heavier towards the bottom). This does not imply a generalization in the sense that there is typically only a few candidate eco-routes to be considered for any origin-destination pair. The Figure 9.6 reveals that trip distance is a confounding factor: for paths shorter than 4 km there is usually only a handful of discovered paths, but that number grows steeply on longer distances.

The computation times in Figure 9.5 did not go above 9 seconds even in the worst case for the maximal considered number of iterations. For 100 iterations the computation took between 0.3 and 0.6 seconds, for 2000 iterations the computation took between 0.8 and 8.9 seconds. The median of the computation time starts at 0.5 seconds at 100 iterations and grows to 3.8 seconds at 2000 iterations. Note that the minimum computation times grow only slowly with the growing number of iterations in comparison to the maximum computation times. This is likely because in the cases when the origin and destination are close to each other the Dijkstra's routing algorithm needs less time to compute. Additionally, the number of found paths is often low in these cases, meaning that the randomly discovered paths are likely to be already in the tree.

The computation times depend on used hardware and availability of its resources (memory and computational power). Nevertheless, the results in Figure 9.5 show that path tree computation with the Algorithm 3 is time-feasible for intra-urban travels with length up to 12km. It remains to be seen whether the paths stored in

[1] The data in Figure 9.6 are based on a separate run of the experiment with the limit on the number of iterations extended to 10000.

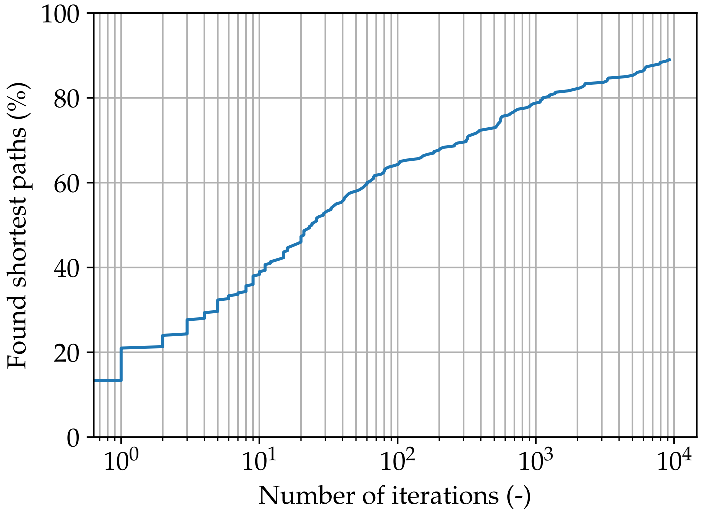


FIGURE 9.7: Portion of found shortest paths against the number of iterations.

the these trees are relevant for eco-routing. One way to assess the candidate eco-routes in the path trees is to see whether they contain the shortest paths. Numerous studies have observed that eco-routes tend to minimize travel distance (Ahn and Rakha, 2007; Ericsson et al., 2006; Kubička et al., 2016b) which implies that shortest paths are likely candidates to be eco-routes. The portion of the 300 trees (for 300 origin-destination pairs) that contain the shortest path is shown in Figure 9.7 against the number of iterations[2]. It shows that the shortest path was the first discovered path in 20% of the cases and that in almost 40% of the cases the shortest path was identified within the first ten iterations. About 10% of the trees did not contain the shortest path even after 10000 iterations.

## 9.3 Time-dependent eco-routing with rerouting

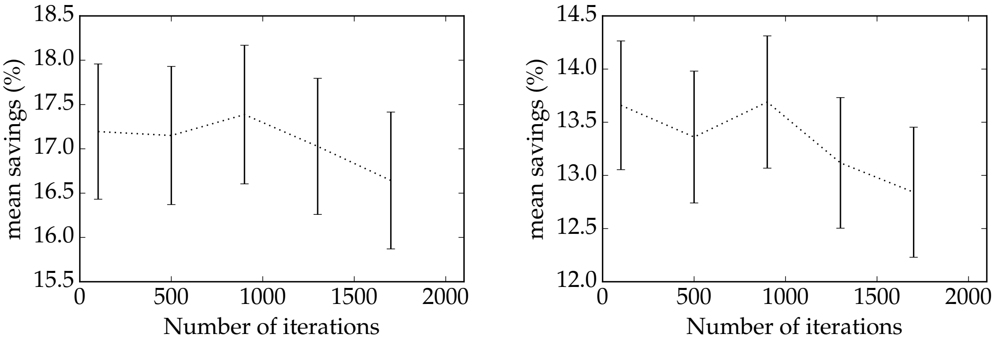
Eco-routing with the energy consumption model 4 (see Table 8.3) is studied in this section. This model is time-dependent and therefore enables time-dependent eco-routing. The only unknown parameter of the model 4 is the predicted travel time at the expected time of arrival to the road for which the energy consumption is to be predicted. The travel time model (5.38) is used for that. The routing is done with the Algorithm 1 using a path tree computed with the Algorithm 3.

### 9.3.1 The choice of a stop condition

The Algorithm 3 does not specify the stop condition to use. As in the previous section, the condition considered here is a simple limit on the number of iterations of the main loop in the Algorithm 3. The choice of this limit is studied in this section.

Consider following limits: 100, 500, 900, 1300, 1700 iterations. Let us evaluate the eco-routing method with model 4 for each limit. Considering a full set $\mathbb{V}_{od}$ (defined in Section 4.2.1) with 60,478 origin-destination pairs for all five limits would

[2] The data in Figure 9.7 are based on a separate run of the experiment with the limit on the number of iterations extended to 10000.



(A) electric vehicle (B) conventional vehicle

FIGURE 9.8: Mean savings against the number of iterations for tree computation. The error bars show the 95% confidence interval.

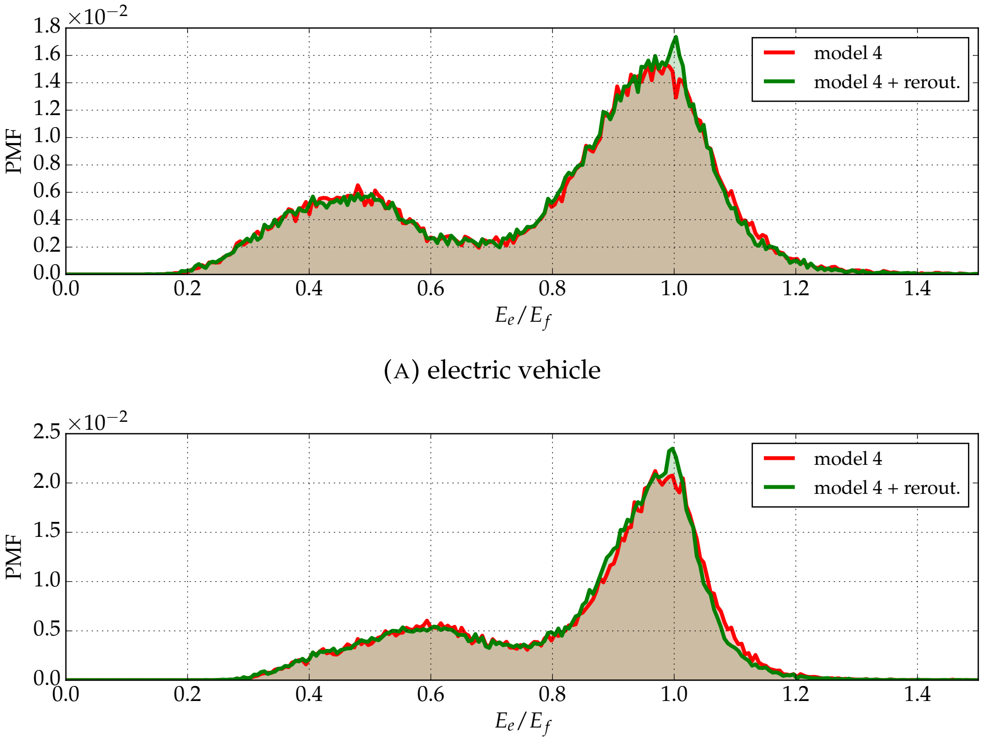
be computationally too costly. Consider a subset of $\mathbb{V}_{od}$ with 4,000 random origin-destination pairs instead. It is the same experiment like the one described below, except that a subset of the 60,478 origin-destination pairs is considered.

The results in Figure 9.8a are for the electric vehicle and in Figure 9.8b for the conventional vehicle. The evaluation was done, exceptionally, in terms of mean savings instead of performance, as it allows to construct 95% confidence intervals around the means. This is because the results were computed on a reduced set of origin-destination pairs. The mean savings are based on a comparison to fastest paths. The 900 iterations have shown best results in both cases, although given that the confidence intervals overlap it is not unlikely that this is not the optimal choice between the five studied cases. In general, the savings do not show strong sensitivity to the choice of the stop condition, although it tends to decrease with higher limits on the number of iterations. This is interesting because it suggests that larger path trees with more options for the router lead to worse results.

### 9.3.2 Time-dependent eco-routing evaluation

The goal of this experiment is to evaluate time-dependent eco-routing performance $P$. Two variants are considered: with rerouting enabled and disabled. The eco-route is computed once (before departure) if rerouting is disabled. If it is enabled, the original eco-route is continually updated on-the-go based on the progress of the vehicle.

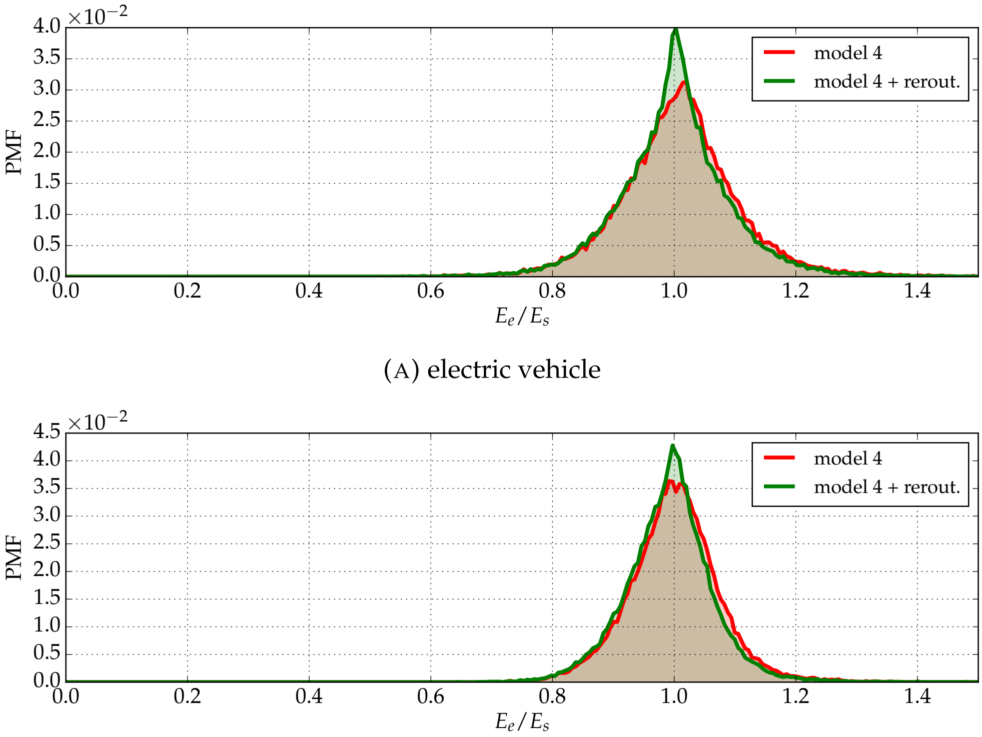
The experiment was conducted on the same set $\mathbb{V}_{od}$ with 60,478 origin-destination pairs used for the evaluation of the baseline methods and of the eco-routing with the model 6. A path tree with the stop condition set to 900 iterations was computed for each origin-destination pair. Then, the Algorithm 1 was used to identify the eco-routes. Each eco-route was then tried in the simulation. The methodology for evaluation of eco-routes in the simulation is discussed in Section 7.5. The used implementation of rerouting is also described in that section. Every simulation started at the same time (at midnight) from the same initial state. It provided necessary information to compute the reference consumptions with the two reference consumption models (see Section 7.2). The experiment was designed to provide results directly comparable to the baseline methods and eco-routing with consumption model 6. There is one exception, however. The estimated performance $\hat{P}$ is not reported as it cannot be evaluated due to changing eco-routing solution with rerouting enabled.



(A) electric vehicle

(B) conventional vehicle

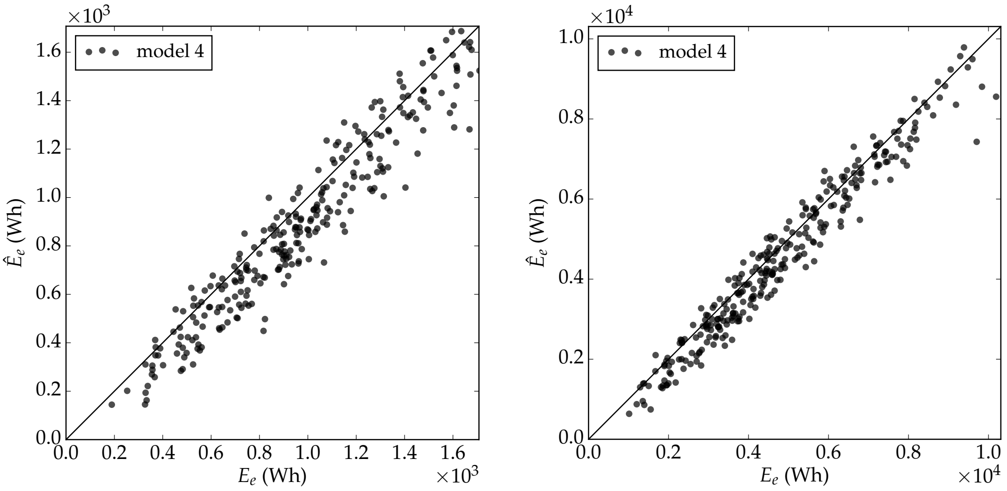
FIGURE 9.9: Distribution of savings relative to fastest paths.



(A) electric vehicle

(B) conventional vehicle

FIGURE 9.10: Distribution of savings relative to shortest paths.



(A) electric vehicle

(B) conventional vehicle

FIGURE 9.11: Correlation of estimated and reference consumption.

The results are in figures 9.9, 9.10, 9.11, and in Table 9.1 under rows "model 4" and "model 4 + rerout.". The distributions of savings are shown in figures 9.9 and 9.10. It can be compared to the distributions of savings for the baseline methods in figures 4.3 and 4.4. For some fixed origin, destination and time of departure let $E_e$ be the energy consumption on the eco-route, $E_s$ the energy consumption on the shortest path and energy $E_f$ the consumption on the fastest path. The Figure 9.9 shows the distribution of savings of the eco-routes with respect to the fastest paths (the distribution of the ratios $E_e/E_f$). The Figure 9.10 shows the distribution of savings with respect to the shortest paths (the distribution of the ratios $E_e/E_s$). The area under the curve below 1.0 on the horizontal axis represents the cases for which the eco-routing method saved some energy while the area under the curve above 1.0 on the horizontal axis represents the cases for which the eco-route required more energy than the reference path. These are the cases in which the eco-routing method failed to save energy. An exceptionally high probability of the eco-route being identical to the reference path (shortest or fastest path) was observed. This appears as a discontinuity in the distributions in figures 9.9 and 9.10. It was removed from the plots and reported instead in Table 9.1 where the results are summarized. The Figure 9.11 shows the correlation between the observed (reference) energy consumptions and the consumptions estimated with the model 6. Results for the electric vehicle are in Figure 9.11a, for the conventional vehicle in Figure 9.11b. It can be compared to the results for the baseline models in Figure 4.5. Each sample represents a single trip. The reference consumption on the eco-route (denoted $E_e$) is on the horizontal axis while the estimated consumption (denoted $\hat{E}_e$) is on the vertical axis. The perfect estimation would be achieved with all samples aligned on the black diagonal line. The samples above it are the cases when the consumption was overestimated, and the samples below it are the cases when it was underestimated.

In terms of performance, the time-dependent eco-routing with model 4 shows improvement over time-independent eco-routing with model 6. With performance based on comparison to the fastest paths, the performance $P$ is 32.1% and 32.4% for the electric vehicle with rerouting disabled and enabled, respectively. In case of the conventional vehicle, the $P$ is 23.0% and 23.5%, respectively. The performance figures are considerably lower when comparing to shortest paths, however. The performance is 0.3% and 0.8% for the electric vehicle with rerouting disabled and enabled, respectively. For the conventional vehicle, the $P$ is 0.7% and 1.4% with rerouting disabled and enabled, respectively. The mean travel time delay ranges from 7.8% to 11.3% when comparing to the fastest paths and from -7.8% to -4.9% when comparing to the shortest paths. This implies that the eco-routing method delivers savings in travel time when compared to the shortest paths. The probability of failure is lower than 9% in all cases, and rerouting has consistently lowered the probability of failure. Eco-routes were identical to fastest paths in 14.5% to 16.6% of cases and identical to shortest paths in 19.5% to 30.8% of cases. The confidence intervals associated with mean savings and mean travel time delays do not overlap across all cases.

TABLE 9.1: Routing results.

| | | wrt fastest | | wrt shortest | |
|---|---|---|---|---|---|
| | | EV† (%) | ICEV‡ (%) | EV† (%) | ICEV‡ (%) |
| $P$ | model 6 | 31.4 | 21.9 | -0.9 | -0.9 |
| | model 4 | 32.1 | 23.0 | 0.3 | 0.7 |
| | model 4 + rerout. | 32.4 | 23.5 | 0.8 | 1.4 |
| | Barth et al., 2007 | 26.7 | 20.1 | -8.0 | -3.2 |
| | Juřík et al., 2014 | 29.7 | 18.8 | -3.5 | -4.9 |
| | MEC model | 34.5 | 24.7 | 3.6 | 2.8 |
| $\hat{P}$ | model 6 | 24.7 | 15.7 | 4.1 | 6.4 |
| | model 4 | 24.7 | 18.0 | 5.5 | 8.3 |
| | model 4 + rerout. | - | - | - | - |
| | Barth et al., 2007 | 7.2 | 7.3 | 2.0 | 2.3 |
| | Juřík et al., 2014 | 50.9 | 53.4 | 2.2 | 2.7 |
| | MEC model | 30.3 | 18.7 | 4.9 | 3.9 |
| Probability of failure | model 6 | 9.6 | 6.3 | 8.2 | 9.0 |
| | model 4 | 9.5 | 10.2 | 7.7 | 9.4 |
| | model 4 + rerout. | 7.5 | 7.3 | 4.6 | 5.6 |
| | Barth et al., 2007 | 8.7 | 8.6 | 5.2 | 6.1 |
| | Juřík et al., 2014 | 24.5 | 30.4 | 30.0 | 36.9 |
| | MEC model | 6.4 | 4.3 | 7.6 | 5.9 |
| Eco-route same as reference | model 6 | 22.8 | 31.9 | 29.6 | 22.5 |
| | model 4 | 16.6 | 16.4 | 30.8 | 23.6 |
| | model 4 + rerout. | 14.5 | 14.5 | 25.4 | 19.5 |
| | Barth et al., 2007 | 28.6 | 29.8 | 40.6 | 35.8 |
| | Juřík et al., 2014 | 9.4 | 8.7 | 37.6 | 32.2 |
| | MEC model | 15.5 | 22.1 | 25.0 | 29.8 |
| Mean savings | model 6 | 15.7 ± 0.19 | 11.7 ± 0.15 | -1.5 ± 0.07 | -1.2 ± 0.06 |
| | model 4 | 16.4 ± 0.19 | 12.6 ± 0.15 | -0.4 ± 0.07 | 0.2 ± 0.05 |
| | model 4 + rerout. | 16.8 ± 0.19 | 13.2 ± 0.15 | 0.1 ± 0.07 | 0.9 ± 0.05 |
| | Barth et al., 2007 | 13.9 ± 0.18 | 11.1 ± 0.15 | -5.6 ± 0.18 | -2.2 ± 0.10 |
| | Juřík et al., 2014 | 13.7 ± 0.21 | 8.2 ± 0.17 | -3.6 ± 0.07 | -4.7 ± 0.07 |
| | MEC model | 19.3 ± 0.19 | 14.6 ± 0.15 | 2.9 ± 0.06 | 2.4 ± 0.05 |
| Mean travel time delay | model 6 | 10.9 ± 0.16 | 8.1 ± 0.14 | -5.2 ± 0.11 | -7.4 ± 0.10 |
| | model 4 | 11.3 ± 0.17 | 9.9 ± 0.15 | -4.9 ± 0.10 | -6.0 ± 0.10 |
| | model 4 + rerout. | 9.3 ± 0.16 | 7.8 ± 0.15 | -6.6 ± 0.10 | -7.8 ± 0.10 |
| | Barth et al., 2007 | 9.7 ± 0.14 | 8.9 ± 0.13 | -5.9 ± 0.10 | -6.4 ± 0.10 |
| | Juřík et al., 2014 | 28.0 ± 0.25 | 29.4 ± 0.26 | 8.4 ± 0.16 | 9.6 ± 0.17 |
| | MEC model | 14.2 ± 0.18 | 11.4 ± 0.16 | -2.5 ± 0.12 | -4.9 ± 0.10 |

† electric vehicle
‡ internal combustion engine (conventional) vehicle

TABLE 9.2: Results for the the eco-routing method with bias towards shorter paths.

| | EV[†] (%) | ICEV[‡] (%) |
|---|---|---|
| Performance $P$ (wrt fastest) | 32.8 | 24.1 |
| Performance $P$ (wrt shortest) | 1.0 | 1.9 |
| Probability of failure (wrt shortest) | 1.2 | 2.4 |
| Eco-route same as reference (wrt shortest) | 69.4 | 40.0 |

[†] electric vehicle
[‡] internal combustion engine (conventional) vehicle

## 9.4 An example with multiple costs

The results show that the proposed eco-routing method often succeeds in identifying the eco-routes that perform better than shortest paths. The problem is that they almost as often fail to do so. If the eco-routing method were to choose the shortest path in the cases when it failed to save energy, then the performance with the time-dependent eco-routing would improve to 2.26% in the case of the electric vehicle and to 2.64% in the case of the conventional vehicle. Consider then, as an example, an eco-routing method biased towards shorter paths. It was reported in the literature that shorter paths are more likely to be eco-routes. Given the uncertainty in the routing costs, the bias might help to reduce the probability of failure and, possibly, even improve performance.

To be specific, consider an eco-routing method with two costs: the energy consumption and the travel distance. Let the energy consumption have unitary weight, and the travel distance have the weight of $15000/l_{\text{min}}$, where 15000 is arbitrarily chosen constant and $l_{\text{min}}$ is the length of the shortest path between the origin and destination. The routing cost of a path under consideration is then the energy consumption with model 4 plus ratio $l/l_{\text{min}}$ multiplied by 15000. The net effect is that for every extra 1% in path's length with respect to the shortest path a constant amount (equal to 150 Wh) is added to the consumption. Note that this is a contrived example. Its sole purpose is to show how the versatility of the proposed routing method can be used to improve results in novel ways.

An extract from the results is in Table 9.2. It is based on an adapted version of the time-dependent eco-routing method with the model 4 discussed above. It shows that this method identifies the shortest paths as eco-routes more often than the original, which is to be expected since the method is artificially biased to prefer shorter paths. Nevertheless, the probability of failure was reduced from 4.6% to 1.23% in case of the electric vehicle and from 5.6% to 2.36% in case of the conventional vehicle with respect to the shortest paths. Additionally, the performance improved from 0.8% to 1.0% in case of the electric vehicle and from 1.4% to 1.9% in the case of the conventional vehicle with respect to the shortest paths. The performance with respect to fastest paths also improved, for the electric vehicle from 32.4% to 32.8% and for the electric vehicle from 23.5% to 24.1%.

## 9.5 Summary

Results of simulated eco-routing experiments are presented in this chapter. Three eco-routing methods are studied, all aim to minimize vehicle energy consumption. Two variants of the consumption model called "model 6" and "model 4" are considered. The consumption model was proposed in Section 5.4, the two variants are introduced in Table 8.3. The model 6 is time-independent while the model 4 is time-dependent. No constraints are imposed on the eco-routing solution. Multiple costs are also not considered. While the proposed eco-routing method supports both, the goal is to evaluate eco-routing performance such that it is comparable to the baseline methods discussed in Section 4.3.

Eco-routing with consumption model 6 is studied in Section 9.1. This is basic time-independent eco-routing with a consumption model that uses same information as baseline models by Barth et al., 2007 and Juřík et al., 2014. The eco-routing with the model 6 performs better than Barth et al., 2007 and Juřík et al., 2014, but fails to perform better than the MEC model and, perhaps more importantly, simple shortest-distance routing.

The Algorithm 3 is used to build path trees. Its computation time and the number of identified paths is studied for various stop conditions in Section 9.2. The stop condition considered in this experiment is a limit on the number of iterations of the main loop in the Algorithm 3. Limits ranging from 100 iterations to 2000 iterations are considered. The observed computation times were shorter than nine seconds across all cases. The median of the computation time starts at 0.5 seconds at 100 iterations and grows to 3.8 seconds at 2000 iterations. The number of found paths grows at varying paces depending on the positions of the origin and destination. The worst observed case contains 1850 paths after 2000 iterations, meaning that in 92.5% of iterations a new path was discovered and added to the tree. On the other hand, the distributions in Figure 9.4 show that this was rare. The median grows at much slower pace with 50 paths found after 100 iterations and 300 paths found after 2000 iterations. It is nevertheless argued that this does not imply that there is typically only a few candidate eco-routes to be considered for any origin-destination pair. Trip distance is a confounding factor as can be seen in Figure 9.6. It is also argued in this section that a way to assess the candidate eco-routes in path trees is to see whether they contain the shortest paths. Figure 9.7 shows the portion of trees that contained the shortest path with respect to the number of iterations. It shows that the shortest path was the first discovered path in 20% of the cases and that in almost 40% of the cases the shortest path was identified within the first ten iterations. About 10% of path trees did not contain the shortest path even after 10000 iterations.

The Section 9.3 studies two eco-routing methods with the consumption model 4. These are time-dependent eco-routing methods based on path trees. The first variant does not support rerouting while the second variant does. These eco-routing methods were evaluated on the same set of origin-destination pairs as the baseline models and the eco-routing method based on model 6 (Section 9.1) to ensure comparability. The results are summarized in Table 9.1. In terms of performance, the time-dependent eco-routing with model 4 shows improvement over time-independent eco-routing with the model 6. Similarly, eco-routing with rerouting enabled performed consistently better than eco-routing with rerouting disabled. The performance ranges from 23.0% to 32.4% when comparing to the fastest paths. The performance figures are considerably lower when comparing to shortest paths, however. In this case, the performance ranges between 0.3% and 1.4%. The mean travel time delay ranges from 7.8% to 11.3% when compared to the fastest paths and from -7.8% to -4.9% when compared to the shortest paths. This suggests that the eco-routing method delivers savings in travel time when compared to the shortest paths. The probability of failure is lower than 9% in all cases; rerouting has lowered the probability of failure. Eco-routes were identical to fastest paths in 14.5% to 30.8% of cases.

Finally, the Section 9.4 gives an example how the versatility of the proposed eco-routing method can be leveraged to improve results in novel ways. An eco-routing method biased towards shorter paths is proposed. The evaluation shows improved performance with respect to both shortest and fastest paths and reduced probability of failure.

# Chapter 10

# Conclusions

The conclusions are summarized in four sections: (1) key findings about the current literature; (2) advances in consumption modeling; (3) the proposed eco-routing method; and (4) map-matching method selection. The first section discusses the performance of current methods and limits of a commonly used eco-routing evaluation methodology. The second section discusses the consumption model proposed in this work. The third section discusses the eco-routing method proposed in this work. The fourth section provides a selection guide for map-matching methods used in the context of eco-routing navigation systems.

## 10.1 Key findings about current literature

Two eco-routing methods published in literature were studied in detail to provide baselines for comparison with the proposed method. They were proposed in Barth et al., 2007 and Juřík et al., 2014. They are time-independent energy consumption minimizing methods. Both use a different approach to energy consumption modeling. The model proposed in Barth et al., 2007 uses regression analysis, the model proposed in Juřík et al., 2014 uses a consumption model derived from the microscopic longitudinal consumption model.

The performance of these methods depend on the environment where they were tested and on the set of paths to which the eco-routes are compared to. High savings were observed when eco-routes are compared to the fastest paths. The performance figures are, however, much lower when the performance is based on a comparison of eco-routes to shortest paths. In this case both Barth et al., 2007 and Juřík et al., 2014 fail to deliver any savings. Instead, their eco-routes offer losses on an average trip. This result implies that a simple shortest path routing is superior to both methods in the tested scenario. Also, a high number of cases in which the eco-routing methods failed to save both energy and travel time were observed. Especially the method by Juřík et al., 2014 have failed in 24.5% to 36.9% of cases. It is hard to imagine that real drivers would accept such a rate of failures. The conclusion is that it might be better to use a simple shortest path routing rather than the evaluated methods for eco-routing. This motivates the constrained time-dependent adaptive eco-routing method proposed in this work. Such method is, in theory, capable of navigating the vehicle through sequences of green lights on signalized intersections, to avoid congestion, and to update the solution when it becomes invalid.

While the evaluation of current eco-routing methods has shown negative savings, it is rare to see such a result reported in the published literature. A close look at the evaluation methodologies used in the literature revealed that the prevalent method is not capable of observing negative savings by design. The problem is that the same consumption (or pollutant emission) model is used for both routing and evaluation. Not only that such evaluation will necessarily result in nonnegative savings, but it is also shown in Theorem 1 to be prone to result in inflated eco-routing performance figures. The conclusion is that the performances and savings reported in the literature should be regarded with caution. These findings underline the need for a reliable eco-routing evaluation framework. Such framework is presented in this work.

Note that these results are not generalizable: both methods are likely to show positive performance in other environments. It is likely that the used intra-urban scenario is challenging for eco-routing as it offers many similar paths to the destination with plenty perturbation sources along the way.

## 10.2 Advances in energy consumption modeling

A novel energy consumption model is proposed in Chapter 5. It is a macroscopic model derived from standard longitudinal energy consumption model. The longitudinal model is a microscopic model that requires recorded vehicle speed profile to estimate consumption. It is argued in this work that such model is not suitable for eco-routing because speed profiles are not available when planning the trip. In this work, the standard model is first reformulated in closed form as a function of energy consumption of an ideal vehicle and of energy that was returned to the vehicle when braking. Then a closed form solution to the energy consumption of the ideal vehicle is derived and used to solve the energy consumption of a more realistic vehicle under an assumption that powertrain's output power is an affine function of its input power. The resulting model requires five unknown parameters: initial and final speed to account for the change in vehicle's kinetic energy, travel time, speed profile variance and speed profile skew coefficient.

Six variants of this consumption model were studied. They differ in the five parameters provided to it: from a model to which all parameters were provided as they were observed to a model where only average travel time was provided, and other parameters were set to zero. Such is, specifically, model 6. It is directly comparable to the models by Barth et al., 2007 and Juřík et al., 2014. Their only parameter is mean speed which is inversely proportional via road length to travel time. Hence, they use the same information as model 6. Despite that, the model 6 outperforms both by a considerable margin. In the case of the electric vehicle, the model 6 have shown RMSE of 28.6 Wh, while Barth et al., 2007 and Juřík et al., 2014 have shown 60.6 Wh and 47.7 Wh, respectively. The (idealistic) MEC model has shown RMSE 24.83 Wh. To compare, the RMSE of model 6, model by Juřík et al., 2014, and model by Barth et al., 2007 is 115%, 192%, and 244% respectively of the RMSE of the MEC model. The accuracy of model 6 is much closer to the accuracy of the MEC model than the other models. Unlike MEC model, however, the model 6 can be used in practice.

The proposed model is interesting due to the formal way in which it was derived from the longitudinal model. The cited eco-routing methods that use consumption models derived from the longitudinal model (such as De Nunzio et al., 2016; Juřík et al., 2014) assume that the vehicle speed profile is a constant function equal to vehicle's mean speed. The model presented here does not need such drastic assumption. It is carefully developed from the longitudinal model. The key contribution is the closed-form solution for the ideal vehicle as it does not rely on any assumptions. The solution for more realistic vehicles is based on it. This model opens a new direction of research as it provides a solution to the longitudinal model in closed form without unrealistic assumptions as it was done in the literature so far. It exposes the properties of speed profiles that matter most. The only assumption it makes is about powertrain efficiency. It is difficult to avoid due to a nonlinear instantaneous efficiency of commonly used powertrains.

Performance of time-independent eco-routing with the model 6 is nevertheless negative with respect to shortest paths. This is a similar result to what was observed for the methods by Barth et al., 2007 and Juřík et al., 2014. Despite the improved accuracy of the model 6, it is still better to use shortest path routing. The eco-routing with the MEC model is the only time-independent eco-router that was able to deliver savings with respect to shortest paths in the experiments conducted in this work. It also outperformed time-dependent eco-routing proposed in this work. While the MEC is a theoretical model, it shows that there is potential in time-independent eco-routing. The question that remains is how realistic it is to harness this potential.

## 10.3 The proposed eco-routing method

The eco-routing has been treated in literature mostly as a minimum path problem. This approach offers an intuitive formulation of eco-routing, and moreover, efficient methods for its computation are available (such as Dijkstra's or Bellman-Ford algorithms). Their main strength is that they are fast and optimal. However, they are also limiting. Constrained eco-routing, time-dependent eco-routing nor adaptive eco-routing is supported by them. There are proposed algorithms for each of these three types of eco-routing, but to author's best knowledge, there is no published eco-routing method that would allow the combination: constrained time-dependent, adaptive eco-routing. This work proposes and evaluates such an eco-routing system.

The method proposed in this work favors simplistic routing over sophisticated optimal routing algorithms such as Dijkstra's or Bellman-Ford algorithm. The motivation for this is that these algorithms are too restrictive, too rigid in their assumptions. It is argued here that using optimal routing algorithms gives a little advantage when there is a sizable uncertainty in the associated costs in the routing graph. An approximative solution based on a naive approach to routing is studied instead. It is based on a hypothesis that while there are usually many possible paths to take to a destination, there is only a small subset of them that can be eco-routes in some realistic conditions.

The proposed eco-routing method consists of two steps: preprocessing and routing. The preprocessing amounts to the computation of a path tree, which is a collection of candidate eco-routes between given origin and destination. The routing is then conducted on the paths in the path tree at a later time. It is unrealistic to compute a tree with all paths between a given origin and destination for all but the trivial cases. Nevertheless, not all paths can be eco-routes. Two algorithms to compute path trees were proposed: the algorithms 2 and 3. The Algorithm 2 conducts an exhaustive search and prunes only those paths for which it can prove that they can never be eco-routes. The Algorithm 3 conducts a repeated randomized searches for eco-routes. The routing is done by the Algorithm 1. It takes the path tree on input and searches for the path that minimizes given cost function. This function can combine multiple costs such as travel time, energy consumption, travel distance and pollutant emissions. It also supports fully time-dependent eco-routing and multiple constraints. The routing algorithm is optimal in the sense that it finds the path that minimizes given cost function on the given tree. The tree, however, typically contains only a subset of paths between given origin and destination. Consequently, the eco-routing solution is not guaranteed optimal.

The Algorithm 3 was used to compute path trees used in presented experiments as it allows to control the tree size and computation time easily. The path tree computation times were shorter than 9 seconds in the worst observed case with 2000 iterations of the main loop in Algorithm 3. These are acceptable results since the tree computation times are not critical. The number of discovered paths was typically low with a median of 300 paths after 2000 iterations. In the worst observed case, there were 1850 discovered paths after 2000 iterations. In 20% of the observed cases the shortest path the first discovered path and in 40% of cases, it was discovered in first ten iterations. This is indicative of the quality of identified eco-routes since numerous authors reported that shortest paths are viable candidates to be eco-routes. On the other hand, in 10% of the cases, the shortest path was not discovered even after 10000 iterations. Consequently, the path tree computation is computationally feasible for the intra-urban trips similar to those simulated in the presented framework. The length of these trips is between 1km and 12km. Whether the path tree computation is feasible on longer trips remains a question. It was observed that the number of discovered paths grows quickly on trips that are longer than 4km. On the other hand, the results did not show reduced eco-routing performance on the longer trips.

The routing was done with consumption model 4 and a travel time model (5.38) that takes into account information about known traffic light periods. The model 4 is a variant similar to model 6 (discussed in the previous section), except that the used travel time is predicted with the travel time model instead of using a simple average of previously observed travel times. Both travel time model and consumption model are time-dependent. Since the Algorithm 1 is used for routing, the eco-routing method is also time-dependent. Two variants were studied: with rerouting disabled and enabled. In the case when it is enabled the eco-routing solution is adaptive. The rerouting is implemented such that the eco-routing solution is updated at every simulation step using the Algorithm 1. It is always run on a subtree rooted at the node that corresponds to the current position of the vehicle in the tree. The vehicle's energy consumption was the only considered cost, and the routing was not constrained. This setup allowed direct comparison of the results of the baseline methods.

The simulation results have shown high performance when comparing eco-routes to the fastest paths and moderate performance when comparing eco-routes to shortest paths. The best observed performance was 32.4% with respect to fastest paths and 1.4% with respect to shortest paths. Both were observed with time-dependent adaptive eco-routing with model 4. The rerouting improved performance marginally, but consistently. The distributions of energy savings reveal that for many trips the method was able to find paths that offer savings with respect to the shortest path, but it failed to do so in many other cases. If the eco-routing method were to choose the shortest path in the cases when it failed, then the performance with respect to shortest paths would be in the range from 2.26% to 2.64%, depending on vehicle type. This shows that there is potential for improvement, albeit limited.

The results show that the proposed eco-routing method can compete with the traditional eco-routing as is, but it is also more flexible by allowing constrained, time-dependent and adaptive eco-routing. This allows for richer applications such as time-constrained eco-routing, or eco-routing that avoids schools at the times when the kids are there, for example. Applications that would be difficult to realize with the standard approach become possible with the proposed method. An eco-routing method that is artificially biased to prefer shorter paths is an example of the versatility of the proposed method. The motivation for this is based on an observation that shorter paths are likelier to be eco-routes. It is shown to lower the probability of failure and to improve performance with respect to both shortest and fastest paths. It managed to save either energy or travel time with respect to shortest paths in 98.8% of studied trips in case of the electric vehicle and in 97.6% of studied trips in case of the conventional vehicle.

The shortest path routing has shown to be a surprisingly difficult eco-router to outperform. However, shortest paths are slow compared to fastest paths. The eco-routing method proposed in this work did not show an ability to outperform simple shortest path router significantly in terms of energy, but it has shown an ability to save slightly more energy than shortest path router while saving a considerable amount of travel time with respect to shortest paths.

The simulation framework developed to conduct these experiments is designed to be as realistic as possible. It uses microscopic traffic simulator of a whole city and microscopic consumption models based on real vehicles as references. However, these are still simulations which mean that the results are indicative. The ultimate validation would be achieved with extensive field tests, which can be cost prohibitive.

## 10.4 Map-matching method selection

Different map-matching methods are suitable for different map-matching applications. There is no universal method that would suit the needs of all. The trade-offs that must be made when selecting a map-matching method are discussed in this section. The interest in the context of eco-routing is twofold: for matching pre-recorded trips and for navigation. This section is an extract from Kubička et al., 2017.

The matching of pre-recorded trips requires an offline map-matching method. Higher computational effort can be tolerated as the trajectories are post-processed after they were collected. The most advanced method regarding matching accuracy is reportedly the path inference filter (Hunter et al., 2014). However, its computational demand might be too prohibitive. The method by Newson and Krumm, 2009 offers good matching accuracy while its computational demand is comparatively low. Another option is the geometric method by Wei et al., 2013a, especially when used in conjunction with fast Fréchet distance approximation method developed by Driemel et al., 2012. If a massive set of trajectories must be processed and matching accuracy is not critical, then the method by Marchal et al., 2005 can be considered. If the application makes use of sparsely sampled trajectories, then low-sampling rate methods (Lou et al., 2009; Raymond et al., 2012; Zheng and Quddus, 2011; Chen et al., 2011b) can be of interest. These methods are likely to be outperformed by the path inference filter, but they are often easier to implement.

The navigational applications require online, high sampling rate map-matching methods. The computational effort must be kept low as the system is required to respond in real time. When integrity monitoring is needed, then the method by Toledo-Moreo et al., 2010 should be considered. This method has shown high matching accuracy on lane-level while providing continuous integrity monitoring of the map-matching output. Methods based on fuzzy logic, such as the method by Quddus et al., 2006, have reportedly excellent matching accuracy but require expert knowledge for their tuning. The methods based on a multiple hypothesis technique

(Pyo et al., 2001; Kubička et al., 2014) might be able to offer an interesting trade-off between computational demand and matching accuracy. The Hidden Markov model based methods and geometric methods are not well suited as they require considerable computational resources. The sliding window technique can be used to remedy this issue: only a last few samples are used to map-match the current matching point. When the demand on computational effort is not stringent, then the method by Hummel, 2006 should also be considered as it has outlook to be robust against positioning errors.

# List of Abbreviations

| | |
|---|---|
| **CMEM** | Comprehensive Modal Emission Model |
| **CRF** | Conditional Random Fields |
| **EU-DEM** | European Union - Digital Elevation Model |
| **EV** | Electric Vehicle |
| **GNSS** | Global Navigation Satellite System |
| **GPS** | Global Positioning System |
| **HPL** | Horizontal Protection Level |
| **ICEV** | Internal Combustion Engine Vehicle |
| **IQR** | Inter-Quartile Range |
| **k-NN** | k-Nearest Neighbors |
| **LPPL** | Lane Position Protection Level |
| **LuST** | Luxembourg SUMO Traffic |
| **MAD** | Median Absolute Deviation |
| **MEC** | Mean Energy Consumption |
| **MHT** | Multiple Hypothesis Technique |
| **NIS** | Normalized Innovation Squared |
| **NSA** | Node Supplementary Array |
| **PMF** | Probability Mass Function |
| **RAIM** | Receiver Autonomous Integrity Monitoring |
| **RMSE** | Root-Mean-Square error |
| **SUMO** | Simulator of Urban MObility |
| **VSP** | Vehicle-Specific Power |

# Bibliography

Ahn, Kyoungho and Hesham Rakha (2007). "Field evaluation of energy and environmental impacts of driver route choice decisions". In: *Proceedings of 2007 IEEE Intelligent Transportation Systems Conference*. IEEE, pp. 730–735.

— (2008). "The effects of route choice decisions on vehicle energy consumption and emissions". In: *Transportation Research Part D: Transport and Environment* 13.3, pp. 151–167.

— (2013). "Network-wide impacts of eco-routing strategies: a large-scale case study". In: *Transportation Research Part D: Transport and Environment* 25, pp. 119–130.

Alt, Helmut and Michael Godau (1995). "Computing the Fréchet distance between two polygonal curves". In: *International Journal of Computational Geometry & Applications* 5.1, pp. 75–91.

Alt, Helmut and Leonidas Guibas (1999). "Discrete geometric shapes: Matching, interpolation, and approximation". In: *Handbook of computational geometry*, pp. 121–153. ISBN: 0444825371.

Alt, Helmut, Alon Efrat, Günter Rote, and Carola Wenk (2003). "Matching planar maps". In: *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, pp. 589–598.

Andersen, Olav, Christian Jensen, Kristian Torp, and Bin Yang (2013). "EcoTour: reducing the environmental footprint of vehicles using eco-routes". In: *Proceedings of 2013 IEEE 14th International Conference on Mobile Data Management*. IEEE, pp. 338–340.

Barth, Matthew, Carrie Malcolm, Theodore Younglove, and Nicole Hill (2001). "Recent validation efforts for a comprehensive modal emissions model". In: *Transportation Research Record: Journal of the Transportation Research Board* 1750, pp. 13–23.

Barth, Matthew, Kanok Boriboonsomsin, and Alexander Vu (2007). "Environmentally-friendly navigation". In: *Proceedings of 2007 IEEE Intelligent Transportation Systems Conference*. IEEE, pp. 684–689.

Bellman, Richard (1958). "On a Routing Problem". In: *Quarterly of Applied Mathematics* 16, pp. 87–90.

Bernstein, David and Alain Kornhauser (1996). *An introduction to map matching for personal navigation assistants*. Tech. rep. New Jersey TIDE Center.

Bonnifait, Philippe, Jean Laneurit, Clément Fouque, and Gérald Dherbomez (2009). "Multi-hypothesis map-matching using particle filtering". In: *Proceedings of 16th World Congress for ITS Systems and Services*, pp. 1–8.

Boriboonsomsin, Kanok, Matthew Barth, Weihua Zhu, and Alexander Vu (2012). "Eco-routing navigation system based on multisource historical and real-time traffic information". In: *IEEE Transactions on Intelligent Transportation Systems* 13.4, pp. 1694–1704.

Brakatsoulas, Sotiris, Dieter Pfoser, Randall Salas, and Carola Wenk (2005). "On map-matching vehicle tracking data". In: *Proceedings of the 31st international conference on very large databases*, pp. 853–864.

Cao, Hu and Ouri Wolfson (2005). "Nonmaterialized motion information in transport networks". In: *International Conference on Database Theory*. Springer, pp. 173–188.

Chabini, Ismail (1998). "Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time". In: *Transportation Research Record: Journal of the Transportation Research Board* 1645, pp. 170–175.

Chen, Daniel, Anne Driemel, Leonidas Guibas, Andy Nguyen, and Carola Wenk (2011a). "Approximate Map Matching with respect to the Fréchet Distance." In: *ALENEX*. SIAM, pp. 75–83.

Chen, Feng, Mingyu Shen, and Yongning Tang (2011b). "Local path searching based map matching algorithm for floating car data". In: *Procedia Environmental Sciences* 10, pp. 576–582.

Chen, Jingmin and Michel Bierlaire (2015). "Probabilistic multimodal map matching with rich smartphone data". In: *Journal of Intelligent Transportation Systems* 19.2, pp. 134–148.

Chen, Mo, Rezaul Chowdhury, Vijaya Ramachandran, David Roche, and Lingling Tong (2007). *Priority queues and Dijkstra's algorithm*. Tech. rep. Computer Science Department, University of Texas at Austin.

Codeca, Lara, Raphaël Frank, and Thomas Engel (2015). "Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research". In: *Proceedings of the 7th IEEE Vehicular Networking Conference*, pp. 1–8.

De Nunzio, Giovanni, Laurent Thibault, and Antonio Sciarretta (2016). "A Model-Based Eco-Routing Strategy for Electric Vehicles in Large Urban Networks". In: *Proceedings of 2016 IEEE Intelligent Transportation Systems Conference*.

Dean, Brian Christopher (1999). "Continuous-time dynamics shortest path algorithms". PhD thesis. Massachusetts Institute of Technology.

Devroye, Luc (1986). *Non-Uniform Random Variate Generation*. New York: Springer-Verlag. ISBN: 978-1-4613-8645-2.

Dib, Wissam, Alexandre Chasse, Domenico Di Domenico, Philippe Moulin, and Antonio Sciarretta (2012). "Evaluation of the energy efficiency of a fleet of electric vehicle for eco-driving application". In: *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles* 67.4, pp. 589–599.

Dietterich, Thomas (1998). "Approximate statistical tests for comparing supervised classification learning algorithms". In: *Neural computation* 10.7, pp. 1895–1923.

Dijkstra, Edsger (1959). "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1, pp. 269–271.

Driemel, Anne, Sariel Har-Peled, and Carola Wenk (2012). "Approximating the Fréchet distance for realistic curves in near linear time". In: *Discrete & Computational Geometry* 48.1, pp. 94–127.

Ericsson, Eva, Hanna Larsson, and Karin Brundell-Freij (2006). "Optimizing route choice for lowest fuel consumption–Potential effects of a new driver support tool". In: *Transportation Research Part C: Emerging Technologies* 14.6, pp. 369–383.

European Environment Agency. *EU Digital Elevation Model*. http://www.eea.europa.eu/data-and-maps/data/eu-dem. Accessed: 2016-01-22.

Fréchet, M Maurice (1906). "Sur quelques points du calcul fonctionnel". In: *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 22.1, pp. 1–72.

Fuglestvedt, Jan, Terje Berntsen, Gunnar Myhre, Kristin Rypdal, and Ragnhild Skeie (2008). "Climate forcing from the transport sectors". In: *Proceedings of the National Academy of Sciences* 105.2, pp. 454–458.

Goh, Chong, Justin Dauwels, Nikola Mitrovic, Muhammad Asif, Ali Oran, and Patrick Jaillet (2012). "Online map-matching based on hidden markov model for real-time traffic sensing applications”. In: *Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 776–781.

Guzzella, Lino and Antonio Sciarretta (2005). *Vehicle propulsion systems: introduction to modeling and optimization*. Springer. ISBN: 3540251952.

Hampel, Frank (1974). “The influence curve and its role in robust estimation”. In: *Journal of the American Statistical Association* 69.346, pp. 383–393.

Hart, Peter, Nils Nilsson, and Bertram Raphael (1968). “A formal basis for the heuristic determination of minimum cost paths”. In: *IEEE Transactions on Systems Science and Cybernetics* 4.2, pp. 100–107.

Hashemi, Mahdi and Hassan A Karimi (2014). “A critical review of real-time map-matching algorithms: Current issues and future directions”. In: *Computers, Environment and Urban Systems* 48, pp. 153–165.

Honey, Stanley, Walter Zavoli, Kenneth Milnes, Alan Phillips, Marvin White Jr, and George Loughmiller Jr (1989). *Vehicle navigational system and method*. US Patent 4,796,191.

Hummel, Britta (2006). “Map matching for vehicle guidance”. In: *Dynamic and Mobile GIS: Investigating Space and Time*.

Hunter, Tom, Pieter Abbeel, and Alexandre Bayen (2014). “The path inference filter: model-based low-latency map matching of probe vehicle data”. In: *IEEE Transactions on Intelligent Transportation Systems* 15.2, pp. 507–529.

Jabbour, Maged, Philippe Bonnifait, and Véronique Cherfaoui (2008). “Map-matching integrity using multihypothesis road-tracking”. In: *Journal of Intelligent Transportation Systems* 12.4, pp. 189–201.

Jimenez-Palacios, Jose (1998). “Understanding and quantifying motor vehicle emissions with vehicle specific power and TILDAS remote sensing”. PhD thesis. Massachusetts Institute of Technology.

Johnson, Donald (1977). “Efficient Algorithms for Shortest Paths in Sparse Networks”. In: *Journal of the ACM* 24.1, pp. 1–13.

Juřík, Tomáš et al. (2014). “Energy Optimal Real-Time Navigation System”. In: *IEEE Intelligent Transportation Systems Magazine* 6.3, pp. 66–79.

Kluge, Sebastian (2011). “On the computation of fuel-optimal paths in time-dependent networks”. PhD thesis. Ph. D. thesis, Technische Universität München.

Kluge, Sebastian et al. (2013). “On the computation of the energy-optimal route dependent on the traffic load in Ingolstadt”. In: *Transportation Research Part C: Emerging Technologies* 36, pp. 97–115.

Krajzewicz, Daniel, Jakob Erdmann, Michael Behrisch, and Bieker Laura (2012). “Recent Development and Applications of SUMO - Simulation of Urban MObility”. In: *International Journal On Advances in Systems and Measurements* 5.3 & 4, pp. 128–138.

Krumm, John, Eric Horvitz, and Julie Letchner (2007). *Map matching with travel time constraints*. Tech. rep. SAE Technical Paper.

Kubička, Matěj, Antonio Sciarretta, Arben Cela, Hugues Mounier, Laurent Thibault, and Silviu-Iulian Niculescu (2016a). “About prediction of vehicle energy consumption for eco-routing”. In: *Proceedings of the 2016 IEEE Intelligent Transportation Systems Conference*. IEEE, pp. 1096–1101.

— (2016b). “Performance of current eco-routing methods”. In: *IEEE Intelligent Vehicles symposium (IV), 2016*. IEEE, pp. 472–477.

Kubička, Matěj, Arben Cela, Hugues Mounier, and Silviu-Iulian Niculescu (2017). “Comparative Study and Application-oriented Classification of Vehicular

Map-Matching Methods". In: *IEEE Intelligent Transportation Systems Magazine* . In press.

Kubička, Matěj, Arben Cela, Hugues Mounier, and Silviu Niculescu (2014). "On designing robust real-time map-matching algorithms". In: *Proceedings of the 2014 IEEE Intelligent Transportation Systems Conference*, pp. 464–470.

Li, Liang, Mohammed Quddus, and Lin Zhao (2013a). "High accuracy tightly-coupled integrity monitoring algorithm for map-matching". In: *Transportation Research Part C: Emerging Technologies* 36, pp. 13–26.

Li, Yang, Qixing Huang, Michael Kerber, Lin Zhang, and Leonidas Guibas (2013b). "Large-scale joint map matching of GPS traces". In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, pp. 214–223.

Lou, Yin, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang (2009). "Map-matching for low-sampling-rate GPS trajectories". In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, pp. 352–361.

Marchal, Fabrice, J Hackney, and Kay Axhausen (2005). "Efficient map matching of large global positioning system data sets: Tests on speed-monitoring experiment in Zürich". In: *Transportation Research Record: Journal of the Transportation Research Board* 1935, pp. 93–100.

Minett, Claire, A. Salomons, Winnie Daamen, Bart Van Arem, and Sjon Kuijpers (2011). "Eco-routing: comparing the fuel consumption of different routes between an origin and destination using field test speed profiles and synthetic speed profiles". In: *Proceedings of the 2011 IEEE Forum on Integrated and Sustainable Transportation Systems*. IEEE, pp. 32–39.

Newson, Paul and John Krumm (2009). "Hidden Markov map matching through noise and sparseness". In: *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, pp. 336–343.

Nie, Yu Marco and Qianfei Li (2013). "An eco-routing model considering microscopic vehicle operating conditions". In: *Transportation Research Part B: Methodological* 55, pp. 154–170.

Papoulis, Athanasios and S. Unnikrishna Pillai (1985). *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education. ISBN: 0071226613.

Pink, Oliver and Britta Hummel (2008). "A statistical approach to map matching using road network geometry, topology and vehicular motion constraints". In: *Proceedings of the 2008 IEEE Intelligent Transportation Systems Conference*. IEEE, pp. 862–867.

Powell, Michael (1964). "An efficient method for finding the minimum of a function of several variables without calculating derivatives". In: *The computer journal* 7.2, pp. 155–162.

Pyo, Jong-Sun, Dong-Ho Shin, and Tae-Kyung Sung (2001). "Development of a map matching method using the multiple hypothesis technique". In: *Proceedings of the 2001 IEEE Intelligent Transportation Systems Conference*. IEEE, pp. 23–27.

Quddus, Mohammed (2006). "High integrity map matching algorithms for advanced transport telematics applications". PhD thesis. Imperial College London, United Kingdom.

Quddus, Mohammed, Robert Noland, and Washington Ochieng (2006). "A high accuracy fuzzy logic based map matching algorithm for road transport". In: *Journal of Intelligent Transportation Systems* 10.3, pp. 103–115.

Quddus, Mohammed, Washington Ochieng, and Robert Noland (2007). "Current map-matching algorithms for transport applications: State-of-the art and future research directions". In: *Transportation Research Part C: Emerging Technologies* 15.5, pp. 312–328.

Raaschou-Nielsen, Ole et al. (2013). "Air pollution and lung cancer incidence in 17 European cohorts: prospective analyses from the European Study of Cohorts for Air Pollution Effects (ESCAPE)". In: *The lancet oncology* 14.9, pp. 813–822.

Rabiner, Lawrence (1989). "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286.

Rakha, Hesham, Kyoungho Ahn, Kevin Moran, Bart Saerens, and Eric Van den Bulck (2011). "Simple comprehensive fuel consumption and CO2 emissions model based on instantaneous vehicle power". In: *Transportation Research Board 90th Annual Meeting*. 11-1009.

Raymond, Rudy, Tetsuro Morimura, Takayuki Osogami, and Noriaki Hirosue (2012). "Map matching with hidden Markov model on sampled road network". In: *Proceedings of the 2012 International Conference on Pattern Recognition*. IEEE, pp. 2242–2245.

Ren, Ming (2012). "Advanced map matching technologies and techniques for pedestrian/wheelchair navigation". PhD thesis. University of Pittsburgh.

Schuessler, Nadine and Kay Axhausen (2009). "Map-matching of GPS traces on high-resolution navigation networks using the Multiple Hypothesis Technique (MHT)". In: *Arbeitsberichte Verkehrsund Raumplanung* 568, pp. 1–22.

Toledo-Moreo, Rafael, David Bétaille, and François Peyret (2010). "Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps". In: *IEEE Transactions on Intelligent Transportation Systems* 11.1, pp. 100–112.

Van Mieghem, Piet and Fernando A Kuipers (2004). "Concepts of exact QoS routing algorithms". In: *IEEE/ACM Transactions on networking* 12.5, pp. 851–864.

Velaga, Nagendra (2010). "Development of a weight-based topological map-matching algorithm and an integrity method for location-based ITS services". PhD thesis. Loughborough University.

Viterbi, Andrew (1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE transactions on Information Theory* 13.2, pp. 260–269.

Wales, David (2003). *Energy landscapes: Applications to clusters, biomolecules and glasses*. Cambridge University Press. ISBN: 0521814154.

Wang, Zheng and Jon Crowcroft (1996). "Quality-of-service routing for supporting multimedia applications". In: *IEEE Journal on Selected Areas in Communications* 14.7, pp. 1228–1234.

Wei, Hong, Yin Wang, George Forman, and Yanmin Zhu (2013a). *Map matching by Fréchet distance and global weight optimization*. Tech. rep. Shanghai Jiao Tong University, Department of Computer Science and Engineering.

— (2013b). "Map matching: Comparison of approaches using sparse and noisy data". In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, pp. 444–447.

Wenk, Carola, Randall Salas, and Dieter Pfoser (2006). "Addressing the need for map-matching speed: Localizing global curve-matching algorithms". In: *Proceedings of the 18th International Conference on Scientific and Statistical Database Management*. IEEE, pp. 379–388.

White, Christopher, David Bernstein, and Alain Kornhauser (2000). "Some map matching algorithms for personal navigation assistants". In: *Transportation Research Part C: Emerging Technologies* 8.1, pp. 91–108.

Xiao, Zhuoling, Hongkai Wen, Andrew Markham, and Niki Trigoni (2014). “Lightweight map matching for indoor localisation using conditional random fields”. In: *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE, pp. 131–142.

Yang, Jae-seok, Seung-pil Kang, and Kyung-soo Chon (2005). “The map matching algorithm of GPS data with relatively long polling time intervals”. In: *Journal of the Eastern Asia Society for Transportation Studies* 6, pp. 2561–2573.

Yao, Enjian and Yuanyuan Song (2013). “Study on eco-route planning algorithm and environmental impact assessment”. In: *Journal of Intelligent Transportation Systems* 17.1, pp. 42–53.

Zheng, Yuheng and Mohammed Quddus (2011). “Weight-based shortest-path aided map-matching algorithm for low-frequency positioning data”. In: *Transportation Research Board 90th Annual Meeting*. 11-2680.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies
de l'information
et de la communication (STIC)

**Titre :** Systèmes eco-routing adaptatifs de navigation dépendant du temps avec des contraintes

**Mots clés :** éco-routage, map-matching, systèmes de navigation, transport vert

**Résumé :** L'éco-routage est une méthode de navigation du véhicule qui sélectionne les trajets vers une destination minisant la consommation de carburant, la consommation d'énergie ou les émissions de polluants. C'est l'une des techniques qui tentent de réduire les coûts d'exploitation et l'empreinte environnementale du véhicule. Ce travail passe en revue les méthodes actuelles d'éco-routage et propose une nouvelle méthode pour pallier leurs insuffisances.

La plupart des méthodes actuelles attribuent à chaque route du réseau routier un coût constant qui représente la consommation du véhicule ou la quantité de polluants émis. Un algorithme de routage optimal est ensuite utilisé pour trouver le chemin qui minimise la somme de ces coûts. Différentes extensions sont considérées dans la littérature. L'éco-routage contraint permet d'imposer des limites sur le temps de trajet, la consommation d'énergie et les émissions de polluants. L'éco-routage dépendant du temps permet le routage sur un graphique avec des coûts qui sont fonction du temps. L'éco-routage adaptatif permet de mettre à jour la solution d'éco-routage au cas où elle deviendrait invalide en raison d'un développement inattendu sur la route.

Il existe des méthodes d'éco-routage optimales publiées qui résolvent l'éco-routage dépendant du temps ou l'éco-routage contraint ou l'éco-routage adaptatif. Chacun vient avec des frais généraux de calcul considérablement plus élevés par rapport à l'éco-routage standard et, à la connaissance de l'auteur, aucune méthode publiée ne prend en charge la combinaison des trois: éco-routage adaptatif dépendant du temps contraint.

On soutient dans ce travail que les coûts d'acheminement sont incertains en raison de leur dépendance au trafic immédiat autour du véhicule, du comportement du conducteur et d'autres perturbations. Il est en outre soutenu que puisque ces coûts sont incertains, il y a peu d'avantages à utiliser un routage optimal car l'optimalité de la solution ne tient que tant que les coûts de routage sont corrects. Au lieu de cela, une méthode d'approximation est proposée dans ce travail. La charge de calcul est plus faible car la solution n'est pas requise pour être optimale. Cela permet l'éco-routage adaptatif dépendant du temps contraint.

**Title :** Constrained Time-Dependent Adaptive Eco-Routing Navigation System

**Keywords :** eco-routing, map-matching, navigation systems, green transportation

**Abstract :** Eco-routing is a vehicle navigation method that selects those paths to a destination that minimize fuel consumption, energy consumption or pollutant emissions. It is one of the techniques that attempt to lower vehicle's operational cost and environmental footprint. This work reviews the current eco-routing methods and proposes a new method designed to overcome their shortcomings.

Most current methods assign every road in the road network some constant cost that represents either vehicle's consumption there or the amount of emitted pollutants. An optimal routing algorithm is then used to find the path that minimizes the sum of these costs. Various extensions are considered in the literature. Constrained eco-routing allows imposing limits on travel time, energy consumption, and pollutant emissions. Time-dependent eco-routing allows routing on a graph with costs that are functions of time. Adaptive eco-routing allows updating the eco-routing solution in case it becomes invalid due to some unexpected development on the road.

There exist published optimal eco-routing methods that solve either the time-dependent eco-routing, or constrained eco-routing, or adaptive eco-routing. Each comes with considerably higher computational overhead with respect to the standard eco-routing and, to author's best knowledge, no published method supports the combination of all three: constrained time-dependent adaptive eco-routing.

It is argued in this work that the routing costs are uncertain because of their dependence on immediate traffic around the vehicle, on driver's behavior, and other perturbations. It is further argued that since these costs are uncertain, there is little benefit in using optimal routing because the optimality of the solution holds only as long as the routing costs are correct. Instead, an approximation method is proposed in this work. The computational overhead is lower since the solution is not required to be optimal. This enables the constrained time-dependent adaptive eco-routing.

**Université Paris-Saclay**
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France