

Approche par optimisation sous contraintes distribuée Divide-and-Coordinate pour le transport urbain multimodal

Laura Trujillo^{1,2}, Elsy Kaddoum¹, Gauthier Picard²

¹ Univ. Toulouse, Institut de recherche en informatique de Toulouse (IRIT), France

² ONERA/DTIS, Université de Toulouse, France

16 février 2026

Résumé

La Mobilité Serviциelle (MaaS) vise à répondre aux enjeux de congestion, de dépendance à l'automobile et de transition écologique en intégrant plusieurs services de transport au sein d'une plateforme unifiée de planification d'itinéraires personnalisés. La planification réaliste d'itinéraires multimodaux demeure toutefois un défi majeur : les approches existantes considèrent peu de modes, négligent les ressources de mobilité personnelles et reposent principalement sur des schémas centralisés. Nous introduisons le Personalized and Resource-aware Multimodal Routing Problem (PRMRP), une formalisation de la planification multimodale et multi-utilisateurs. Deux formulations multi-agents sont proposées, basées sur l'optimisation distribuée sous contraintes (DCOP) et une approche Divide-and-Coordinate (DaC), et évaluées sur des scénarios réalistes à l'échelle d'une ville.

Mots-clés

Mobilité servicielle, Transports, Optimisation Distribuée, Systèmes Multi-Agents

Abstract

Urban mobility systems face growing challenges related to congestion, car dependency, and ecological transition. Mobility as a Service (MaaS) addresses these issues by integrating multiple transportation services within a unified personalized trip planning platform. However, realistic multi-modal itinerary planning remains one of the central challenges. Existing approaches typically consider a limited set of transport modes, rarely account for personal mobility resources, and rely on centralized schemes that decouple service composition from resource allocation. We introduce the Personalized and Resource-aware Multimodal Routing Problem (PRMRP), a formal formulation for multi-modal and multi-user itinerary planning with resource allocation. We propose two multi-agent formulations for the itinerary planning component : a Distributed Constraint Optimization (DCOP) model that could be solved by dedicated algorithms and a Divide-and-Coordinate (DaC) model based on the dual decomposition of the problem into subproblems to coordinate. We evaluate both approaches against a centralized complete solver based on a MILP formula-

tion and a greedy heuristic baseline, on realistic city-scale multi-modal transportation scenarios.

Keywords

Mobility-as-a-Service, Transportation, Distributed Optimization, Multi-Agent Systems

1 Introduction

Les zones urbaines font face à des défis croissants liés à la congestion, à la dépendance à l'automobile et à la transition écologique. Dans ce contexte, la mobilité servicielle (Mobility as a Service, MaaS) s'impose comme un paradigme prometteur en intégrant plusieurs services de mobilité au sein d'une plateforme unifiée de planification et de réservation de trajets personnalisés [16]. La mise en œuvre effective du MaaS repose notamment sur la recommandation d'itinéraires multimodaux et multi-utilisateurs, qui reflète pleinement ce paradigme en intégrant des services émergents tels que le transport à la demande ou le covoiturage [25, 30]. Cependant, les approches existantes restent limitées : elles considèrent un petit sous ensemble de modes, souvent issus d'un unique opérateur, offrent une personnalisation restreinte (souvent temporelle uniquement) et relèguent les ressources de mobilité personnelles au rôle d'options de secours. Elles s'appuient majoritairement sur des formulations centralisées en programmation linéaire en nombres entiers mixtes (MILP), dont la complexité croît rapidement avec le nombre d'utilisateurs, de modes et d'arcs temporels, dépassant les capacités des solveurs. Ces approches requièrent en outre une vision globale des données privées des utilisateurs, soulevant des enjeux de confidentialité et de conformité au RGPD [9], et induisent une latence incompatible avec des scénarios MaaS réalistes. Enfin, la séparation récurrente entre allocation des ressources et composition modale peut conduire à des solutions sous-optimales. Sur la base de cette analyse, nous défendons une approche décentralisée de la planification multimodale intégrée dans le cadre du MaaS, fondée sur les contributions suivantes :

- (i) l'introduction du *Personalized and Resource-aware Multimodal Routing Problem* (PRMRP), un cadre unifié permettant une composition flexible de tout mode de transport conditionnée par la disponibilité

des ressources et la prise en compte explicite des préférences utilisateurs ;

- (ii) une formulation distribuée de la composante de routage du PRMRP sous forme de problème d'optimisation sous contraintes (DCOP) ;
- (iii) une adaptation de cette formulation dans une approche *Divide-and-Coordinate* (DaC), résolue par un algorithme dédié, DaCSA-PRMRP ;
- (iv) une évaluation expérimentale comparant DaCSA-PRMRP à des algorithmes DCOP, à un programme linéaire en nombres entiers centralisé et à une heuristique gloutonne.

La section 2 présente les approches classiques et distribuées de la planification d'itinéraires multimodaux. La section 3 introduit le modèle PRMRP. La section 4 décrit la formulation DCOP et l'approche DaC proposées. La section 5 détaille le protocole expérimental et les résultats obtenus, avant que la section 6 ne conclue et esquisse des perspectives de recherche.

2 Travaux antérieurs

Transport multimodal Les problèmes de planification des transports peuvent être caractérisés selon deux dimensions : (i) le nombre d'utilisateurs impliqués (mono- ou multi-utilisateur) et (ii) le nombre de modes de transport intégrés (mono- ou multimodal), dont le croisement définit quatre grandes familles de problèmes. Les travaux sur la planification d'itinéraires ont progressivement évolué de formulations mono-modales vers des approches multimodales plus intégrées. Les premiers modèles traitaient le cas mono-utilisateur et mono-modal à travers des problèmes classiques de plus court chemin sur des réseaux à une seule couche [7, 4]. Des extensions ultérieures ont considéré le cas mono-utilisateur multimodal en combinant des réseaux hétérogènes (route, rail, vélo, marche) [27, 2], permettant une planification porte-à-porte réaliste via des représentations dépendantes du temps ou de la fréquence [3, 6], mais sans coordination entre requêtes.

Des approches plus récentes ont introduit des perspectives multi-utilisateurs, notamment via des problèmes de gestion de flottes mono-modales tels que le Transport à la Demande [5, 23, 12] et les services de transport partagé [20], qui optimisent le routage, l'affectation des passagers et la planification sous contraintes de capacité et de fenêtres temporelles. Ces approches coordonnent les utilisateurs au sein d'un mode unique. Des formulations pleinement multimodales et multi-utilisateurs ont ensuite été proposées en combinant des modèles de routage centrés utilisateur avec des services de mobilité partagée [15, 25, 18], en étendant les modèles classiques de Transport à la Demande par l'intégration de transferts intermodaux et des contraintes de partage de ressources.

Malgré ces avancées, plusieurs limites persistent. L'intégration modale demeure souvent partielle, restreinte à un nombre réduit de modalités [15], tandis que les véhicules personnels sont généralement traités comme des solutions de secours plutôt que comme des modes pleinement com-

binables [17]. Les séquences de modes admissibles sont fréquemment prédéfinies, souvent selon un paradigme premier/dernier kilomètre, ce qui limite la flexibilité de la composition modale [17]. La personnalisation reste principalement temporelle ou budgétaire, avec une prise en compte limitée des préférences qualitatives [1, 31]. Enfin, les interactions entre utilisateurs sont le plus souvent confinées à des services isolés et la coordination de multiples itinéraires dans le temps et à travers plusieurs services successifs demeure peu explorée.

Méthodes centralisées La majorité des modèles multimodaux et multi-utilisateurs reposent sur des formulations centralisées, généralement sous forme de programmes linéaires en nombres entiers ou mixtes (ILP/MILP) [29, 25]. Le passage à l'échelle est abordé via des phases de génération de candidats ou des stratégies de décomposition [17], qui dissocient souvent la composition modale et l'allocation des ressources, au risque de produire des solutions globalement sous-optimales. Pour améliorer la scalabilité, des heuristiques et métaheuristiques issues du Transport à la Demande sont couramment utilisées, telles que des stratégies gloutonnes, des heuristiques d'insertion, des algorithmes génétiques, l'optimisation par essais particuliers [1] ou la recherche de voisinage [21], en échange d'un compromis entre optimalité et efficacité computationnelle. Ces approches supposent toutefois une information complète et un décideur central unique, ce qui limite leur applicabilité dans des contextes MaaS dynamiques et décentralisés et soulève des enjeux de confidentialité [14].

Méthodes décentralisées Les problèmes de recommandation d'itinéraires impliquent des entités autonomes (utilisateurs, fournisseurs de services) disposant de connaissances locales et gérant leurs propres ressources. Ils relèvent naturellement des systèmes multi-agents. Des cadres décentralisés permettent une prise de décision locale, la préservation de la confidentialité [26] et une meilleure adaptation aux environnements dynamiques, mais restent encore peu explorés pour la planification multimodale. Dans [8], les auteurs montrent qu'une approche multi-agents peut faire passer à l'échelle la planification de trajets co-modaux par la coordination de décisions locales, sans toutefois s'appuyer sur une optimisation distribuée formelle. Des cadres décentralisés modélisent explicitement la coopération entre agents, c'est le cas du formalisme des problèmes d'optimisation distribuée sous contraintes (DCOP), qui permet de coordonner des décisions locales vers un objectif global [11]. Bien que les DCOP aient été appliqués à des problèmes de logistique et de routage, leur utilisation pour la planification de la mobilité multimodale n'existe pas à notre connaissance. Malgré un coût potentiel en communication, ils offrent des garanties d'optimalité et une représentation naturelle de la coordination décentralisée, ce qui en fait une base prometteuse pour ce travail.

3 Définition du problème

Nous introduisons le *Personalized and Resource-aware Multimodal Routing Problem* (PRMRP), défini comme le

tuplet $\langle \mathcal{Q}, \mathcal{N}, \mathcal{O}, \mathcal{S}, \mathcal{R} \rangle$, où \mathcal{Q} , \mathcal{N} , \mathcal{O} , \mathcal{S} et \mathcal{R} désignent respectivement les ensembles des requêtes, des nœuds de transferts, des opérateurs, des services et des ressources.

Requêtes utilisateurs Soit \mathcal{U} l'ensemble des utilisateurs soumettant des requêtes. Chaque requête $q_i \in \mathcal{Q}$ est définie par :

$$q_i = (u_i, d_i, a_i, [t_i^{ED}, t_i^{LA}]),$$

où :

- $u_i \in \mathcal{U}$ est l'utilisateur associé, caractérisé par : (1) $pr_i = (pr_i^{\text{véhicules}}, pr_i^{\text{avantages}})$, représentant ses ressources de mobilité personnelles (p. ex. véhicules possédés) et ses avantages liés à des abonnements (p. ex. train, bus); (2) $w_i \in \mathbb{R}_{\geq 0}^k$, un vecteur de préférences normalisé pondérant les critères d'optimisation (p. ex. temps, coût, confort, émissions); et (3) β_i , l'ensemble des contraintes spécifiques à l'utilisateur (p. ex. accessibilité, distance de marche, restrictions modales);
- $d_i, a_i \in \mathcal{N}$ sont respectivement les points de départ et d'arrivée;
- $[t_i^{ED}, t_i^{LA}]$ correspond aux instants de départ au plus tôt (ED) et d'arrivée au plus tard (LA).

Services Soit $\mathcal{S} = \{s_1, \dots, s_k, \dots, s_{|\mathcal{S}|}\}$ l'ensemble des services de transport disponibles. Chaque service $s_k \in \mathcal{S}$ est géré par un opérateur unique. Pour chaque requête q_i , $S_i \subseteq \mathcal{S}$ désigne l'ensemble des services accessibles.

Opérateurs Soit $\mathcal{O} = \{o_1, \dots, o_p, \dots, o_{|\mathcal{O}|}\}$ l'ensemble des opérateurs de mobilité. Chaque opérateur $o_p \in \mathcal{O}$ gère un ou plusieurs services de transport et détermine ceux accessibles à chaque requête.

Nœuds Un nœud représente un point d'entrée ou de sortie pour un ou plusieurs services. Chaque nœud $n \in \mathcal{N}$ peut avoir des caractéristiques opérationnelles selon le type de service qui s'y trouve. On note $\kappa_{n,s}^{\text{out}}$ et $\kappa_{n,s}^{\text{in}}$ respectivement le nombre maximal de départs et d'arrivées simultanés autorisés au nœud n pour le service $s \in \mathcal{S}$. Ces paramètres modélisent des goulots d'étranglement opérationnels tels que des capacités limitées de quais, de zones de chargement ou de rotation des véhicules. Certains services pouvant être restreints à des requêtes spécifiques (p. ex. véhicules privés) ou exclus par un utilisateur, chaque requête $q_i \in \mathcal{Q}$ dispose de son propre ensemble de nœuds de correspondance accessibles N_i . L'ensemble global des nœuds de correspondance est alors défini par :

$$\mathcal{N} = \bigcup_{i=1}^{|\mathcal{Q}|} N_i.$$

On compte aussi parmi N_i , les points de départ et d'arrivée de q_i

Ressources Pour chaque service $s_k \in \mathcal{S}$, $\mathcal{R}_k = \{r_1, \dots, r_\ell, \dots, r_{|\mathcal{R}_k|}\}$ est l'ensemble des ressources associées. Chaque ressource $r_\ell \in \mathcal{R}_k$ représente une unité élémentaire de transport, de capacité $cap(r_\ell) \in \mathbb{N} \cup \{+\infty\}$. Les ressources disposant d'horaires prédéfinis sont dites *fixes*, tandis que les autres sont dites *libres*.

A partir de ces définitions, nous pouvons définir la notion d'itinéraire et l'objectif du problème, en passant par les concepts de tronçon candidat et de tronçon instancié.

Tronçon candidat Pour chaque requête $q_i \in \mathcal{Q}$, soit $L_i = \{l_1, \dots, l_m, \dots, l_{|L_i|}\}$ l'ensemble des tronçons candidats accessibles. Un tronçon candidat correspond à un trajet potentiel, opéré par un service entre deux nœuds. Formellement, chaque tronçon $l_m \in L_i$ est défini par $l_m = (d_m, a_m, s_m)$, où $d_m, a_m \in \mathcal{N}$ sont les nœuds d'entrée et de sortie du service $s_m \in \mathcal{S}$.

Tronçon instancié L'affectation d'une ressource $r_\ell \in \mathcal{R}^k$ à un tronçon candidat $l_m \in L_i$ produit un tronçon instancié via l'application :

$$\begin{aligned} \psi &: (L_i \times \mathcal{R}^k) \rightarrow \mathcal{D}_i, \\ \psi(l, r) &= (t, dur, z) \end{aligned}$$

où \mathcal{D}_i ne contient que les tronçons instanciés pour les paires (l, r) faisables c.à.d. pour lesquelles la ressource effectivement peut opérer le service associé. Les variables t , dur et z désignent respectivement l'heure de départ, la durée et un vecteur de caractéristiques opérationnelles liées aux critères d'optimisation (p. ex. coût, énergie, confort). Ces données sont déduites de l'allocation de la ressource au tronçon.

Itinéraire Un itinéraire I_i associé à la requête q_i est alors une séquence ordonnée de tronçons instanciés :

$$I_i = (\psi_0, \dots, \psi_j, \psi_{j+1}, \dots, \psi_{|I_i|-1})$$

Pour être faisable, il doit satisfaire la cohérence temporelle :

$$t_{j+1} \geq t_j + dur_j + \tau(a_j, s_j, s_{j+1}, u_i)$$

où τ est une fonction de temps de correspondance dépendant du nœud, des services et de l'utilisateur. D'autres contraintes de faisabilité pourraient s'ajouter à cette contrainte minimale, selon les modes choisis et la modélisation qui en est faite (contraintes de flux du point de départ à l'arrivée ou contraintes de temps minimum de marche entre deux tronçons successifs). On peut dire de façon générale que le coût d'un itinéraire est défini comme la somme des coûts des tronçons instanciés et des coûts de correspondance, y compris l'accès à l'origine et à la destination :

$$\begin{aligned} Cost_i^{\text{it}}(I_i) &= Cost_i^{\text{corr}}(d_i, \psi_0) + \sum_{j=0}^{|I_i|-1} Cost_i^{\text{tronçon}}(\psi_j) \\ &+ \sum_{j=0}^{|I_i|-1} Cost_i^{\text{corr}}(\psi_j, \psi_{j+1}) + Cost_i^{\text{corr}}(\psi_{|I_i|-1}, a_i) \end{aligned} \quad (1)$$

où $Cost_i^{\text{tronçon}}(\cdot)$ est le coût scalaire, dépendant de l'utilisateur, associé à un tronçon instancié, défini par :

$$Cost_i^{\text{tronçon}}(\psi(l, r)) = \langle w_i, \mathbf{z} \rangle.$$

Le terme $Cost_i^{\text{corr}}(\cdot)$ modélise les coûts d'accès ou de correspondance dont on connaît le vecteur de caractéristiques \mathbf{z}' , agrégé en un coût par produit scalaire avec le même vecteur de préférences w_i .

$$Cost_i^{\text{corr}}(\psi_j, \psi_{j+1}) = \langle w_i, \mathbf{z}' \rangle.$$

Objectif principal L'objectif est de sélectionner des tronçons candidats via l'allocation de ressources spécifiques, de manière à construire des itinéraires faisables et optimisant avec précision les préférences des utilisateurs.

Solution faisable Une solution $\Phi : \mathcal{Q} \rightarrow \mathcal{I}$ associe à chaque requête q_i un itinéraire $I_i = \Phi(q_i)$. Une solution est faisable si tous les itinéraires indépendants sont faisables, si toutes les contraintes utilisateur β_i sont satisfaites et si aucun conflit de ressources n'apparaît à aucun instant.

4 Modèle DaC pour le PRMRP

Dans le PRMRP, pour une requête donnée, les décisions principales sont locales : elles concernent pour chaque créneau de l'itinéraire la sélection d'un tronçon et de son instant d'entrée. Le couplage intervient principalement aux points de correspondance, où des tronçons successifs doivent être temporellement cohérents. De plus, des dépendances sont induites entre itinéraires par le partage de ressources. Nous exploitons cette structure partiellement locale, en formalisant le PRMRP comme un DCOP et en l'adaptant pour une approche de résolution *Divide-and-Coordinate*. Dans ces approches, les décisions locales sont optimisées indépendamment, tandis que la coordination est imposée uniquement pour les décisions partagées. Cela permet une recommandation d'itinéraires scalable, centrée utilisateur et respectueuse de leur confidentialité. À ce stade, le modèle se concentre sur la logique centrale de composition des services et suppose des capacités de ressources infinies. Les contraintes explicites de partage des ressources et la coordination multi-utilisateurs sont laissées à des travaux futurs. Dans ce contexte, les requêtes sont indépendantes.

4.1 Modèle DCOP pour le PRMRP

Un problème d'optimisation distribuée sous contraintes (DCOP) [11] étend le problème d'optimisation sous contraintes classique (COP) à un cadre multi-agents, dans lequel des agents, disposant uniquement de connaissances locales, doivent se coordonner par communication afin de minimiser un coût global. Formellement, un DCOP est défini comme un tuple $P = \langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu, f \rangle$, où $\mathcal{A} = \{a_1, \dots, a_m\}$ est l'ensemble des agents, $\mathcal{X} = \{x_1, \dots, x_n\}$ l'ensemble des variables de décision à domaines finis $\mathcal{D} = \{D_{x_1}, \dots, D_{x_n}\}$, $\mathcal{R} = \{r_1, \dots, r_k\}$ un ensemble de relations associant des coûts dans $\mathbb{R} \cup \{\infty\}$ aux affectations de variables, $\mu : \mathcal{X} \rightarrow \mathcal{A}$ l'application associant chaque variable à un agent, et $f : \prod_i D_{x_i} \rightarrow \mathbb{R}$ la fonction objectif globale, généralement définie comme la somme des coûts de toutes les relations. Résoudre un DCOP consiste à trouver une affectation des variables minimisant f . De nombreux algorithmes existent pour résoudre des DCOP, allant de méthodes exactes [13, 22, 24] à des approches approximées [19, 10], qui échangent des garanties d'optimalité contre une meilleure scalabilité.

La formulation DCOP proposée pour le PRMRP est illustrée en Figure 1. Chaque requête utilisateur q_i est prise en charge par un agent utilisateur (rectangle transparent) chargé de construire un itinéraire à l'aide d'un nombre fixe

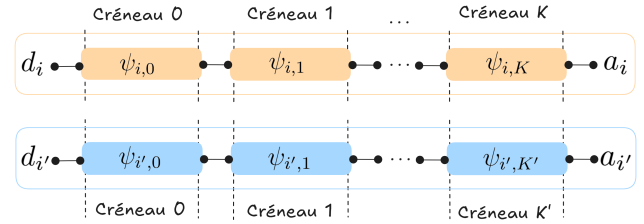


FIGURE 1 – Formulation DCOP avec décomposition des itinéraires en créneaux.

K (différent pour chaque q_i) d'agents créneau ordonnés (rectangles pleins). une variable $\psi_{i,k}$ est associée à chaque agent créneau. Elles permettent de sélectionner un tronçon instancié ou l'état inactif. La séquence de créneaux définit un itinéraire d'au plus K tronçons. Les dépendances entre décisions sont modélisées par des relations binaires partagées entre créneaux successifs (flèches circulaires). La figure illustre cette structure pour deux requêtes utilisateur, sans interaction entre elles.

L'ensemble complet des tronçons déjà instanciés est fourni au modèle de résolution puisque l'allocation de ressource n'est pas une décision. Nous supposons que les opérateurs sont capables de gérer et de prédire la disponibilité des ressources et leurs horaires. Pour une requête donnée, l'ensemble des tronçons instanciés est construit à partir des points de départ et d'arrivée, de la fenêtre temporelle et des services localement accessibles fournis par les opérateurs. La marche n'est pas considérée comme un service, car elle est réservée aux correspondances.

4.1.1 Domaines (\mathcal{D})

Ensemble des tronçons instanciés disponibles pour la requête i , augmenté d'une valeur nulle \perp :

$$D_i^{\text{tronçon}} = \mathcal{D}_i \cup \{\perp\}.$$

4.1.2 Variables (\mathcal{X})

Pour chaque requête i et chaque créneau $k \in \{1, \dots, K\}$, une variable unique $\psi_{i,k} \in \mathcal{D}_i^{\text{tronçon}}$ est définie. Pour chaque variable, on note :

- $\ell_{i,k}$: le tronçon candidat associé au tronçon instancié ($\ell_{i,k} = \perp$ pour un tronçon nul) ;
- $t_{i,k}^{\text{in}}$: l'instant d'entrée du tronçon instancié (0 indique un créneau inactif). L'instant de sortie $t_{i,k}^{\text{out}}$ est déduit de $t_{i,k}^{\text{in}}$ et de la durée du tronçon.

4.1.3 Agents (\mathcal{A})

Nous définissons un agent par créneau pour chaque requête :

$$\mathcal{A} = \{A_{i,k} \mid i \in \mathcal{Q}, k = 1, \dots, K\}.$$

Chaque agent $A_{i,k}$ contrôle $\psi_{i,k}$. Ainsi, chaque agent représente un créneau disponible pour un tronçon dans l'itinéraire d'une requête.

4.1.4 Contraintes et relations de coût (C)

Les contraintes dures définissent les relations de faisabilité entre les variables de décision et restreignent l'ensemble des affectations admissibles. Les relations de coût ne sont évalués que sur les affectations faisables.

Contraintes Les contraintes dures sont exprimées sous forme d'ensembles caractérisant les décisions valides ; elles peuvent être traduites directement en relations DCOP équivalentes en attribuant un coût infini aux configurations invalides.

Activation du premier créneau : impose l'activation du premier créneau afin d'éviter des scénarios ne comportant que de la marche.

$$C_{i,0}^{\text{premier_tronçon}} = \{\psi_{i,0} \mid \ell_{i,0} = \perp\}.$$

Compacité de l'itinéraire : un itinéraire doit être compact ; dès qu'un créneau est inactif, tous les créneaux suivants doivent l'être également ; retrouver le dernier créneau actif avec cette contrainte devient trivial.

$$C_{i,k}^{\text{compact}} = \{(\psi_{i,k}, \psi_{i,k+1}) \mid \ell_{i,k} \neq \perp \vee \ell_{i,k+1} = \perp\}.$$

Temps minimal de correspondance : Soit la fonction $ok(a, b, t_1, t_2)$ retournant vrai si la correspondance est faisable entre deux noeuds i.e. :

$$t_2 \geq t_1 + \text{min_transfer_time}(a, b)$$

On définit alors la contrainte de temps minimal de correspondance comme :

$$C_{i,k}^{\text{min_corr}} = \{(\psi_{i,k}, \psi_{i,k+1}) \mid \text{condition}\}$$

avec $\text{condition} = ((\ell_{i,k} = \perp) \vee (\ell_{i,k+1} = \perp) \vee ok(a_{i,k}, d_{i,k+1}, t_{i,k}^{\text{out}}, t_{i,k+1}^{\text{in}}))$. Des contraintes supplémentaires de correspondance portent sur le choix des modes : deux tronçons consécutifs de même mode ne sont pas autorisés, sauf pour les transports publics avec des lignes distinctes.

Temps minimal de correspondance à l'origine : La correspondance entre le point de départ et le premier créneau doit être faisable :

$$C_i^{\text{min_dep_corr}} = \{(\psi_{i,0} \mid ok(d_i, d_{i,0}, t_i^{\text{ED}}, t_{i,0}^{\text{in}}))\}.$$

Temps minimal de correspondance vers la destination : La correspondance entre le créneau terminal et la destination doit être faisable. Un créneau k est dit "terminal k " si $\ell_{i,k} \neq \perp$ et $\ell_{i,k+1} = \perp$. Il est "terminal K " si $\ell_{i,k} \neq \perp$ et $k = K$. On définit alors :

$$C_{i,k}^{\text{min_arr_corr}} = \{(\psi_{i,k}, \psi_{i,k+1}) \mid \text{cond}_k\},$$

où $\text{cond}_k = \text{term}_k(\psi_{i,k}, \psi_{i,k+1}) \Rightarrow ok(a_{i,k}, a_i, t_{i,k}^{\text{out}}, t_i^{\text{LA}})$. Et pour le cas terminal K :

$$C_{i,k}^{\text{min_arr_corr_K}} = \{\psi_{i,k} \mid \text{cond}_K\}.$$

où $\text{cond}_K = \text{term}_K(\psi_{i,k}) \Rightarrow ok(a_{i,k}, a_i, t_{i,k}^{\text{out}}, t_i^{\text{LA}})$.

Dans la suite du papier, nous feront référence aux relations qui représentent ces contraintes dures avec les notations : $R^{\text{premier_tronçon}}$, R^{compact} , $R^{\text{min_corr}}$, $R^{\text{min_dep_corr}}$ et $R^{\text{min_arr_corr}}$.

Relations de coût Dans la formulation DCOP, les relations de coût implémentent les composantes du coût d'itinéraire définies à la Section 3. Les coûts ne sont appliqués que lorsque les créneaux et correspondances sont actives et ils correspondent exactement au coût utilisateur $Cost_i^{\text{it}}$.

Coût de tronçon : Pour chaque requête i et créneau k , la sélection d'un tronçon non nul $\ell_{i,k} \neq \perp$ à l'instant $t_{i,k}^{\text{in}}$ engendre le coût du tronçon instancié correspondant :

$$R^{\text{tronçon}}(\psi_{i,k}) = \mathbb{I}[\ell_{i,k} \neq \perp] Cost_i^{\text{tronçon}}(\psi_{i,k}),$$

Coût de correspondance : Une correspondance entre deux créneaux consécutifs actifs engendre le coût suivant :

$$R^{\text{corr}}(\psi_{i,k}, \psi_{i,k+1}) = \begin{cases} 0 & \text{si } \text{fin}, \\ Cost_i^{\text{corr}}(\psi_{i,k}, \psi_{i,k+1}) & \text{sinon.} \end{cases}$$

avec $\text{fin} = ((\ell_{i,k} = \perp) \vee (\ell_{i,k+1} = \perp))$

Coût d'accès à l'origine : Le coût de correspondance entre le point de départ et le premier créneau actif est défini par :

$$R^{\text{dep}}(\psi_{i,0}) = Cost_i^{\text{corr}}(d_i, \psi_{i,0})$$

Coût d'accès à la destination : Le coût d'arrivée à la destination est modélisé par des relations binaires $R_i^{\text{arr}_k}$ et $R_i^{\text{arr}_K}$, qui n'interviennent que lorsque la fin d'itinéraire est détectée.

$$R_i^{\text{arr}}(\psi_{i,k}, \psi_{i,k+1}) = \mathbb{K}_{\text{term}_k(\psi_{i,k}, \psi_{i,k+1})} Cost_i^{\text{corr}}(\psi_{i,k}, a_i)$$

et, pour le cas limite $k = K - 1$:

$$R_i^{\text{arr_K}}(\psi_{i,k}) = \mathbb{K}_{\text{term}_K(\psi_{i,k})} Cost_i^{\text{corr}}(\psi_{i,k}, a_i)$$

4.1.5 Fonction objectif (f)

La fonction objectif globale est définie comme la somme de tous les coûts pour l'ensemble des requêtes et des créneaux.

4.2 Formulation de DaCSA-PRMRP

Nous adoptons un schéma *Divide-and-Coordinate* [28] dans lequel chaque agent $A_{i,k}$ résout un problème d'optimisation local construit à partir du graphe de relations du DCOP global, ce qui conduit à notre méthode dédiée DaCSA-PRMRP.

Divide-and-Coordinate (DaC) Au-delà des méthodes de recherche locale et tabulaires, les approches *divide-and-coordinate* traitent la complexité des DCOP en *divisant* le DCOP global et en *coordonnant* itérativement les décisions locales [28]. Dans DaC, un DCOP binaire est décomposé en $|\mathcal{X}|$ sous-problèmes, où chaque sous-problème ϕ_s est défini sur x_s (contrôlé par l'agent s) ainsi que sur des copies de ses variables voisines dans le graphe de facteurs ; il inclut la relation unaire de x_s et la moitié de chaque relation binaire partagée avec ses voisins. Chaque sous-problème ϕ_s est formalisé comme un Programme Linéaire en Nombres Entiers (PLNE) binaire impliquant des variables $x_{i,k}^s$ valant 1 lorsque la variable x_i prend la valeur k , et des variables $x_{ij;kl}^s$ valant 1 lorsque x_i et x_j prennent respectivement les

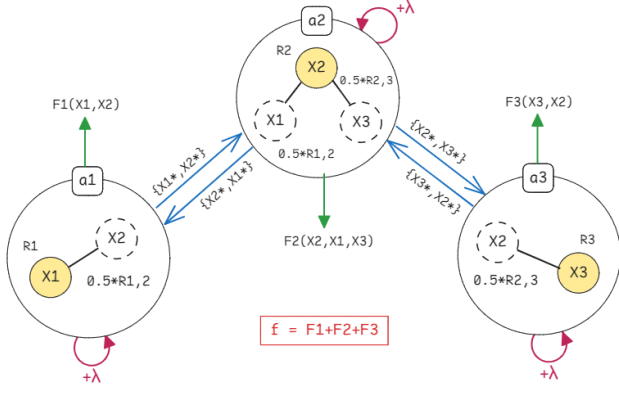


FIGURE 2 – Schéma de l'approche Divide-and-Coordinate.

valeurs k et l . Les contraintes suivantes doivent être vérifiées pour chaque ϕ_s : (i) des *contraintes d'unicité*, garantissant que chaque copie locale d'une variable reçoit exactement une valeur de son domaine ; (ii) des *contraintes de cohérence intra-sous-problème*, assurant la cohérence entre les représentations unaires et binaires d'une même variable au sein d'un même sous-problème ; (iii) des *contraintes de cohérence inter-sous-problèmes*, imposant l'accord entre copies locales de variables partagées et paires de variables partagées entre sous-problèmes voisins. Comme la troisième contrainte est globale, elle est relâchée par l'introduction de multiplicateurs de Lagrange dans l'objectif, sous la forme $\lambda_{i,j}^j \cdot (x_{i,j}^i - x_{i,j}^j)$ $\forall x_i \in \mathcal{X}, \forall x_j \in N(x_i), \forall k \in \mathcal{D}_i$ et $\lambda_{i,j;kl} \cdot (x_{i,j;kl}^i - x_{i,j;kl}^j)$ $\forall r_{ij} \in \mathcal{R}, \forall k \in \mathcal{D}_i, \forall l \in \mathcal{D}_j$. Ainsi, chaque fonction objectif locale est réécrite pour intégrer ces termes lagrangiens, produisant un ensemble modifié de sous-problèmes dont les objectifs se somment en l'objectif initial du DCOP, constituant une *division valide* au sens où l'utilité globale est exactement retrouvée en agrégeant les utilités locales. Lors de l'étape de coordination, les agents échangent leurs solutions locales avec leurs voisins et mettent à jour les multiplicateurs de Lagrange en cas de désaccord ; ces multiplicateurs agissent comme des signaux de coordination pénalisant localement les affectations inconsistantes et guidant les agents vers un consensus.

4.2.1 Division du DCOP global

L'objectif du DCOP global est la somme des relations unaires (attachées à un seul créneau k) et des relations binaires couplant deux créneaux consécutifs $(k, k+1)$ d'une même requête i . Afin d'obtenir des sous-problèmes locaux indépendants tout en préservant l'objectif global, chaque agent $A_{i,k}$ définit un problème local $\phi_{i,k}$ formulé comme un Programme Linéaire en Nombres entiers (PLNE) binaire. Les variables utilisées ont déjà été décrites dans la section 4.2 ; on adapte les notations à notre problème :

- $x_{k;\psi}^{(i,k)}$,
- $x_{k-1;\psi'}^{(i,k)}$ ($k > 1$), $x_{k+1;\psi''}^{(i,k)}$ ($k < K$)
- $x_{k-1,k;\psi',\psi}^{(i,k)}$ ($k > 1$), $x_{k,k+1;\psi,\psi''}^{(i,k)}$ ($k < K$)

avec $\psi \in \mathcal{D}_{i,k}$, $\psi' \in \mathcal{D}_{i,k-1}$, $\psi'' \in \mathcal{D}_{i,k+1}$

Les relations du DCOP correspondant à des contraintes dures ne sont pas intégrées à la fonction objectif du sous-problème, mais traitées comme des contraintes du PLNE. La valeur de l'objectif repose ainsi exclusivement sur les relations de coût unaires et binaires, distinguées en deux ensembles :

(i) Toutes les relations unaires pour le créneau (i, k) , définie par :

$$\mathcal{R}_{i,k}^{\text{un}} = \{\mathcal{R}^{\text{tronçon}}, \mathcal{R}^{\text{dep}}, \mathcal{R}^{\text{arr-K}}\}$$

(ii) Toutes les relations binaires couplant les créneaux (i, k) et (i, j) . On note cet ensemble :

$$\mathcal{R}_{i,k}^{\text{bin}} = \{\mathcal{R}^{\text{corr}}, \mathcal{R}^{\text{arr}}\}$$

L'objectif local de l'agent $A_{i,k}$ est :

$$F_{i,k} = \sum_{R \in \mathcal{R}^{\text{un}}} R(\psi_{i,k}) + \frac{1}{2} \sum_{j \in \{k-1, k+1\}} \sum_{R \in \mathcal{R}^{\text{bin}}} R(\hat{\psi}_{i,j}, \psi_{i,k})$$

La relaxation de consensus dans DaCSA-PRMRP est obtenue en introduisant six familles de multiplicateurs de Lagrange imposant l'accord entre une variable globale et ses copies dans les sous-problèmes voisins : deux ensembles unaires par voisin ($\lambda_{i,k}^{\{k-1\} \rightarrow k}$ et $\lambda_{i,k}^{\{k+1\} \rightarrow k}$) et deux ensembles pour la direction inverse ($\lambda_{i,k}^{k \rightarrow \{k-1\}}$ et $\lambda_{i,k}^{k \rightarrow \{k+1\}}$), ainsi que deux ensembles par paires ($\lambda_{i,k}^{\{k-1\}, k}$ et $\lambda_{i,k}^{k, \{k+1\}}$) gérant l'interaction entre copies adjacentes. Ces multiplicateurs définissent un terme lagrangien local $\mathcal{L}_{i,k}(\mathbf{x}, \lambda)$ qui pénalise tout désaccord entre les affectations choisies par des sous-problèmes voisins, à l'aide de fonctions indicatrices $\mathbb{1}[\cdot]$ pour activer les termes appropriés aux frontières de la chaîne.

$$\begin{aligned} \mathcal{L}_{i,k}(\mathbf{x}, \lambda) = & \sum_{\mathbb{D}_{i,k}^k} \left(\mathbb{I}[k > 1] \lambda_{i,k}^{\{k-1\} \rightarrow k} + \mathbb{I}[k < K] \lambda_{i,k}^{\{k+1\} \rightarrow k} \right) x_{i,k}^{i,k} \\ & - \mathbb{I}[k > 1] \sum_{\mathbb{D}_{i,k}^{\{k-1\}}} \lambda_{i,k}^{k \rightarrow \{k-1\}} x_{i,k}^{i,k} \\ & - \mathbb{I}[k < K] \sum_{\mathbb{D}_{i,k}^{\{k+1\}}} \lambda_{i,k}^{k \rightarrow \{k+1\}} x_{i,k}^{i,k} \\ & + \mathbb{I}[k > 1] \sum_{\mathbb{D}_{i,k}^{\{k-1\}} \times \mathbb{D}_{i,k}^k} \lambda_{i,k}^{\{k-1\}, k} x_{i,k}^{i,k} \\ & - \mathbb{I}[k < K] \sum_{\mathbb{D}_{i,k}^k \times \mathbb{D}_{i,k}^{\{k+1\}}} \lambda_{i,k}^{k, \{k+1\}} x_{i,k}^{i,k} \end{aligned}$$

En combinant ce terme de consensus avec le coût local $F_{i,k}$, on obtient l'objectif local augmenté que chaque agent $A_{i,k}$ minimise à chaque itération : $\min F_{i,k} + \mathcal{L}_{i,k}$. Les détails des multiplicateurs sont fournis dans le matériel supplémentaire.

4.2.2 Mise à jour des lagrangiens

À chaque itération t , l'agent $A_{i,k}$ met à jour ses multiplicateurs à partir des solutions locales optimales des sous-problèmes adjacents.

Let $x^{*,(i,k)}$ denote the optimal solution of the subproblem (i,k) at iteration t . Soit $x^{*,t}$ la solution optimale du sous-problème considéré à l'itération t . Soit λ le multiplicateur de Lagrange à l'itération t et λ^{+} le même multiplicateur de Lagrange à l'itération suivante $t + 1$. Les mises à jour s'écrivent :

$$\begin{aligned}\lambda_{i,k}^{\{k-1\} \rightarrow k, t+1} &= \lambda_{i,k}^{\{k-1\} \rightarrow k, t} - \gamma_t (x_k^{*,t} - x_{k-1}^{*,t}), \\ \lambda_{i,k}^{\{k+1\} \rightarrow k, t+1} &= \lambda_{i,k}^{\{k+1\} \rightarrow k, t} - \gamma_t (x_k^{*,t} - x_{k+1}^{*,t}), \\ \lambda_{i,k}^{k \rightarrow \{k-1\}, t+1} &= \lambda_{i,k}^{k \rightarrow \{k-1\}, t} - \gamma_t (x_{k-1}^{*,t} - x_{k-1}^{*,t}), \\ \lambda_{i,k}^{k \rightarrow \{k+1\}, t+1} &= \lambda_{i,k}^{k \rightarrow \{k+1\}, t} - \gamma_t (x_{k,k+1}^{*,t} - x_{k+1}^{*,t})\end{aligned}$$

Les multiplicateurs ne sont mis à jour qu'en cas de désaccord sur une variable partagée entre sous-problèmes voisins, l'amplitude de l'ajustement étant contrôlée par le pas $\gamma > 0$.

5 Résultats expérimentaux

5.1 Protocole expérimental

Environnement de calcul Les expériences ont été menées sur une station Linux équipée d'un processeur Intel Xeon E5-2660 v3 (20 cœurs à 2,60 GHz) et de 64 Go de RAM. Les algorithmes ont été implémentés en Python 3.9 à l'aide d'un fork du framework *pyDCOP*¹ pour la résolution DPOP et du framework *AMAK*² pour implémenter le modèle DaC multi-agents. Les MILP sont résolus avec IBM ILOG CPLEX Optimizer v22.11.

Instances du problème Afin d'évaluer la scalabilité et la qualité des solutions, nous avons généré des instances PRMRP synthétiques à partir de données de mobilité réelles fournies par un opérateur de transport public. Chaque instance du problème correspond à une requête de transport unique sur un réseau de transport public composé de lignes de bus et de métro à horaires fixes. Les réseaux utilisés incluent 5, 10 ou 15 lignes de transport, comprenant chacune 3 à 20 stations. Cela correspond à des réseaux allant de 15 à 300 nœuds de correspondance et à des tailles d'instance allant de 28 tronçons instanciés par créneau à 3092 tronçons instanciés par créneau. La Figure 3 présente le plus grand réseau utilisé pour ces expérimentations. Les données horaires réelles sont réduites jusqu'à 98% et toutes les expériences sont répliquées avec six graines aléatoires. Avant résolution, une étape de filtrage restreint les réseaux aux nœuds atteignables compte tenu des contraintes de départ, d'arrivée et la fenêtre temporelle de chaque requête. Par conséquent, la taille effective du problème dépend à la fois du réseau et de la requête. En moyenne, ce filtrage réduit le nombre de tronçons par créneau de 81,87%, avec des réductions allant de 70,25% à 100%.

Baselines Nous comparons DaCSA-PRMRP à une heuristique *gloutonne* centralisée inspirée de A^* . Pour chaque requête, un itinéraire est construit incrémentalement en sélectionnant le meilleur tronçon suivant, minimisant son coût et

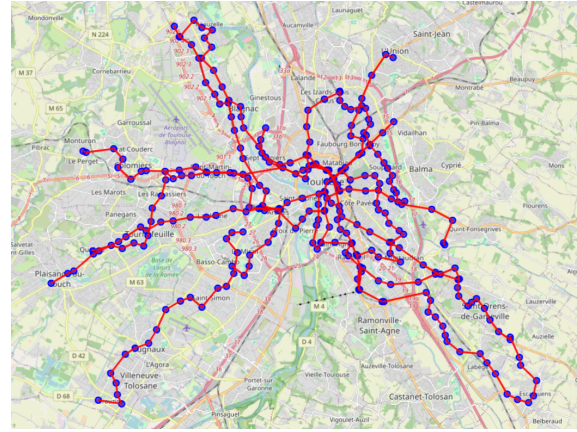


FIGURE 3 – Plus grand réseau de transport : 15 lignes, 20 arrêts par ligne.

une estimation du coût restant jusqu'à la destination. Cette stratégie rapide mais myope fournit une baseline (borne inférieure) sur la qualité des solutions. Nous comparons également les méthodes distribuées à une *formulation MILP centralisée*, implémentée avec CPLEX, qui fournit des solutions optimales et constitue une référence exacte pour la qualité.

Paramètres des modèles et des algorithmes Pour nos expériences, le nombre maximal de créneaux d'un itinéraire est fixé à $K = 5$ et le pas de discrétisation temporelle (pour obtenir les horaires de départ des ressources libres) est $\Delta = 5$ minutes. Le temps de résolution est limité à 5 minutes. Le timeout des sous-problèmes MILP dans DaCSA-PRMRP est fixé à 30 secondes. DaCSA-PRMRP s'arrête lorsque l'optimalité est atteinte avec un gap d'optimalité inférieur à $\epsilon = 0.0001$ ou lorsque la solution n'a pas été améliorée depuis 4 itérations.

La règle de mise à jour du pas dans DaCSA-PRMRP suit une règle de sous-gradient adaptatif proposée dans [28]. Pour des raisons de stabilité, le pas a été borné dans l'intervalle $[0.7, 20]$, ces bornes ayant été choisies empiriquement.

5.2 Analyse des résultats

Performances Notre analyse évalue la scalabilité en fonction de la taille des instances (nombre de tronçons avant filtrage) via le temps de résolution et la charge de calcul. Des tendances lissées (moyennes mobiles) sont utilisées afin de mettre en évidence les motifs de croissance tout en réduisant la variabilité inter-instances. La Figure 4 présente l'évolution du temps de résolution. En tant qu'algorithme anytime, MGM-2 atteint systématiquement la limite de temps. L'heuristique gloutonne surpasse sans surprise les autres méthodes, tandis que CPLEX reste applicable sur des instances plus grandes. Les temps de résolution de DPOP et de DaCSA-PRMRP croissent fortement et présentent des performances moyennes comparables. Ils commencent à atteindre la limite de temps respectivement autour de 1254 tronçons instanciés et 730 pour DaCSA-PRMRP. Contrairement à DPOP, DaCSA-PRMRP garantit une solution de

1. <https://github.com/rpgoldman/pyDcop.git>

2. <https://github.com/alexandrepl/AMAK-Python.git>

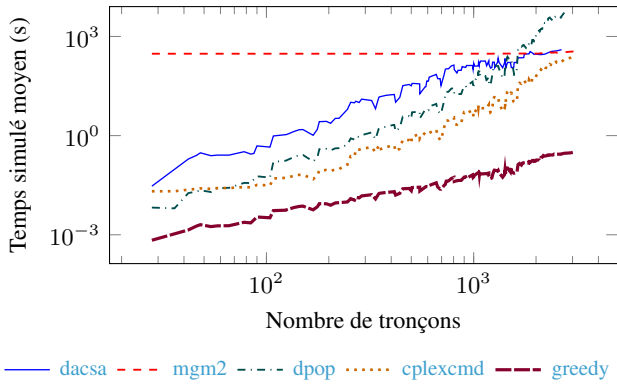


FIGURE 4 – Temps simulé moyen en secondes (axe y en log) en fonction du nombre de tronçons (axe x en log).

qualité bornée après la limite de temps.

Concernant la charge de communication, la 5-(a) montre que MGM-2 induit un surcoût de communication très élevé, tandis que DPOP échange un nombre de messages presque constant. DaCSA-PRMRP présente une complexité en messages linéaire avec le nombre d’itérations, ce qui se traduit par une variabilité visible d’une instance à l’autre, tout en conservant un comportement globalement stable. La diminution du nombre de messages observée pour certaines méthodes, en particulier MGM-2, est principalement due à des mécanismes d’arrêt anticipé qui limitent le nombre d’itérations lorsque la taille des instances augmente et que les itérations deviennent plus coûteuses. Cet effet peut également être renforcé par des propriétés structurelles des grands réseaux de transport public, qui comportent davantage de lignes traversantes et nécessitent donc moins de coordination lors des correspondances.

La 5-(b) rapporte l’évolution de la taille totale des messages. Comme attendu pour une méthode tabulaire dont le coût de communication croît avec la dimension des tables, la taille des messages de DPOP augmente rapidement avec la taille des instances. MGM-2 présente d’abord une diminution liée à l’arrêt anticipé, puis une augmentation lorsque la taille unitaire des messages devient dominante. DaCSA-PRMRP présente un comportement globalement stable, avec une légère baisse sur les plus grandes instances, probablement due au timeout ou à des effets structurels.

Qualité des solutions Au-delà du temps de calcul et de la scalabilité, nous analysons la qualité des solutions. Les violations de contraintes sont rares ($\leq 7.94\%$ des instances) : CPLEX et DPOP violent des contraintes dans 0,62% des cas, MGM-2 dans 3,70%, et DaCSA-PRMRP dans 7,94%. Les violations concernent toujours une seule contrainte parmi les 14 possibles (issues des contraintes dures DCOP) et portent sur les temps minimaux de correspondance dans 78% des instances affectées, les autres cas correspondant à des violations de la règle interdisant les solutions « marche uniquement ».

Nous avons observé l’écart relatif moyen à CPLEX en fonction de la taille des instances. Tous les algorithmes sont proches de l’optimum sur les petits graphes, et l’écart rela-

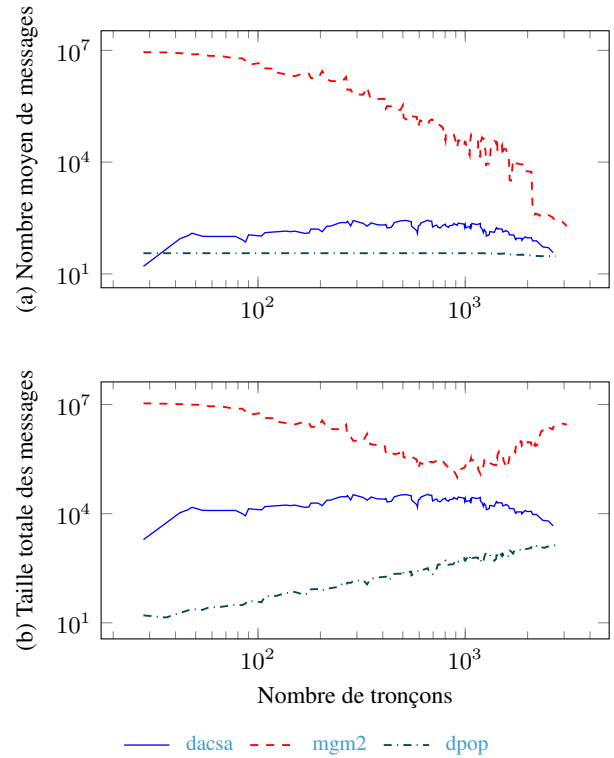


FIGURE 5 – Charge de communication en fonction du nombre de tronçons (log-log) : (a) nombre moyen de messages ; (b) taille totale moyenne des messages.

tif moyen atteint son maximum autour de 2% pour les plus grandes instances.

6 Conclusion

Dans ce travail, nous avons introduit le PRMRP, un modèle formel capturant la complexité complète de la planification d’itinéraires multimodaux et multi-utilisateurs dans un contexte MaaS. À partir de cette formulation, nous avons dérivé deux représentations distribuées complémentaires (DCOP et DaCSA-PRMRP). Des expériences menées sur des instances synthétiques dérivées de données de mobilité réelles montrent que les deux approches atteignent une bonne qualité de solution. De plus, DaCSA-PRMRP présente une meilleure scalabilité : il nécessite moins de messages que MGM-2 et, contrairement à DPOP, il garantit une solution de qualité bornée, ce qui lui permet de traiter des réseaux plus grands et des fenêtres temporelles plus contraintes. Ces résultats confirment qu’une formulation distribuée, respectueuse de la confidentialité, est non seulement faisable mais aussi avantageuse pour des systèmes MaaS à grande échelle où un contrôle centralisé est impraticable.

L’expérimentation réalisée s’est concentrée sur des instances à deux modes de transport (bus/métro) avec des ressources fixes. Néanmoins, notre formulation du problème prend en compte des ressources libres ainsi que les préférences des utilisateurs concernant leurs ressources de mobilité personnelles. L’inclusion de ressources libres dans

les expériences entraînerait un plus grand nombre de combinaisons multimodales et davantage de personnalisation. Nous prévoyons d'évaluer cela à l'aide de trois scénarios de mobilité combinant usage de voiture privée et transport public, afin de mettre en évidence la pertinence du report modal : (1) sans voiture, (2) voiture privée avec stationnement à destination, et (3) voiture privée avec stationnements situés à certains points d'entrée des lignes de transport public. En complément, une optimisation selon plusieurs critères (durée, émissions de CO_2 , prix par exemple) sous différents profils de préférences (orienté temps, orienté CO_2 , orienté coût) pourrait mettre en évidence des interactions significatives entre multimodalité et préférences utilisateurs, conduisant à des arbitrages modaux distincts selon les scénarios.

L'étude actuelle repose sur plusieurs hypothèses simplificatrices, telles que l'absence de contraintes explicites de capacité des ressources et l'indépendance des requêtes. Les travaux futurs porteront donc sur : (i) l'intégration du partage des ressources (p. ex. capacité des véhicules, limites des stations d'arrimage) directement dans les modèles DCOP/DaC, possiblement via des contraintes de couplage supplémentaires ou des couches de coordination hiérarchiques ; (ii) des stratégies hybrides combinant les heuristiques rapides de la baseline gloutonne avec l'exactitude de DaC, visant des algorithmes anytime capables de fournir des solutions progressivement meilleures sous des budgets temps stricts.

Globalement, le cadre d'optimisation distribuée proposé ouvre une voie prometteuse vers un MaaS scalable et centré utilisateur, capable de respecter conjointement les préférences personnelles, les contraintes opérationnelles et la confidentialité des agents participants.

Références

- [1] Parastoo Afrasyabi, Mohammad Saadi Mesgari, El sayed M. El-kenawy, Mehrdad Kaveh, Abdelhameed Ibrahim, and Nima Khodadadi. A crossover-based multi-objective discrete particle swarm optimization model for solving multi-modal routing problems. *Decision Analytics Journal*, 9 :100356, 2023.
- [2] Chris Barrett, Riko Jacob, and Madhav Marathe. Formal-language-constrained path problems. *SIAM J. Comput.*, 30 :809–837, 01 2000.
- [3] Hannah Bast. Car or public transport—two worlds. 5760 :355–367, 09 2009.
- [4] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. Route planning in transportation networks, 2015.
- [5] Jean-François Cordeau. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54 :573–586, 06 2006.
- [6] Daniel Delling, Thomas Pajor, and Dorothea Wagner. *Engineering Time-Expanded Graphs for Faster Timetable Information*, pages 182–206. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [7] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1) :269–271, December 1959.
- [8] Mariagrazia Dotoli, Hayfa Zgaya, Carmine Russo, and Slim Hammadi. A multi-agent advanced traveler information system for optimal trip planning in a comodal framework. *IEEE Transactions on Intelligent Transportation Systems*, 18(9) :2397–2412, 2017.
- [9] European Parliament and Council of the EU. Regulation (EU) 2016/679 of the European Parliament and of the Council. <https://data.europa.eu/eli/reg/2016/679/oj>, 2016.
- [10] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '08*, page 639–646, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
- [11] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. Distributed constraint optimization problems and applications : A survey. *Journal of Artificial Intelligence Research*, 61 :623–698, 2018.
- [12] K. Gkiotsalitis and A. Nikolopoulou. The multi-vehicle dial-a-ride problem with interchange and perceived passenger travel times. *Transportation Research Part C : Emerging Technologies*, 156 :104353, 2023.
- [13] Katsutoshi Hirayama and Makoto Yokoo. Distributed partial constraint satisfaction problem. In Gert Smolka, editor, *Principles and Practice of Constraint Programming-CP97*, pages 222–236, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.
- [14] Alexandra Hoess, Jonathan Lautenschlager, Johannes Sedlmeir, Gilbert Fridgen, Vincent Schlatt, and Nils Urbach. Toward seamless mobility-as-a-service. *Business & Information Systems Engineering*, 67(2) :213–229, 2025.
- [15] Carl H. Häll, Henrik Andersson, Jan T. Lundgren, and Peter Värbrand. The Integrated Dial-a-Ride Problem. *Public Transport*, 1(1) :39–54, May 2009.
- [16] Peraphan Jittrapirom, Valeria Caiati, Anna Maria Feneri, Shima Ebrahimigharehbaghi, Maria Alonso Gonzalez, and Jishnu Narayan. Mobility as a service : A critical review of definitions, assessments of schemes, and key challenges. *Urban Planning*, 2, 06 2017.
- [17] Pramesh Kumar and Alireza Khani. An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access. *Transportation Research Part C : Emerging Technologies*, 122 :102891, 01 2021.
- [18] Tai-Yu Ma, Saeid Rasulkhani, Joseph Y.J. Chow, and Sylvain Klein. A dynamic ridesharing dispatch

- and idle vehicle repositioning strategy with integrated transit transfers. *Transportation Research Part E : Logistics and Transportation Review*, 128 :417–442, 2019.
- [19] Rajiv Maheswaran, Jonathan Pearce, and Milind Tambe. Distributed algorithms for dcop : A graphical-game-based approach. pages 432–439, 01 2004.
- [20] Neda Masoud and R. Jayakrishnan. A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem. *Transportation Research Part B : Methodological*, 99 :1–29, 2017.
- [21] Renaud Masson, Fabien Lehuédé, and Olivier Péton. The dial-a-ride problem with transfers. *Computers & Operations Research*, 41 :12–23, 2014.
- [22] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. Adopt : asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1) :149–180, 2005. Distributed Constraint Satisfaction.
- [23] Sophie N. Parragh. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C : Emerging Technologies*, 19(5) :912–930, 2011. Freight Transportation and Logistics (selected papers from ODYSSEUS 2009 - the 4th International Workshop on Freight Transportation and Logistics).
- [24] Adrian Petcu and Boi Faltings. A scalable method for multiagent constraint optimization. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI’05*, page 266–271, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.
- [25] Marcus Posada, Henrik Andersson, and Carl Häll. The integrated dial-a-ride problem with timetabled fixed route service. *Public Transport*, 9, 07 2017.
- [26] Tamir Tassa, Tal Grinshpoun, and Avishay Yanai. A privacy preserving collusion secure dcop algorithm. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI’19*, page 4774–4780. AAAI Press, 2019.
- [27] Daniel Tischner. Multi-modal route planning in road and transit networks, 2018.
- [28] Meritxell Vinyals, Marc Pujol, Juan A. Rodríguez-Aguilar, and Jesús Cerquides. Divide and coordinate : solving dcops by agreement. In *Adaptive Agents and Multi-Agent Systems*, 2010.
- [29] Shuang Yang, Jianjun Wu, Huijun Sun, and Yunchao Qu. Trip planning for a mobility-as-a-service system : Integrating metros and shared autonomous vehicles. *Transportation Research Part E : Logistics and Transportation Review*, 176 :103217, 2023.
- [30] Xiao Yu, Huimin Miao, Armagan Bayram, Meigui Yu, and Xi Chen. Optimal routing of multimodal mobility systems with ride-sharing. *International Transactions in Operational Research*, 28(3) :1164–1189, 2021.
- [31] Yimeng Zhang, Oded Cats, and Shadi Sharif Azadeh. Dynamic preference-based multi-modal trip planning of public transport and shared mobility, 2025.