

Le problème de livraison le jour même pour les achats en ligne

Stacy A. Voccia, une Ann Melissa Campbell, une Barrett W. Thomas une

une Département des sciences de gestion, Université de l'Iowa, Iowa City, Iowa 52242

Contact: stacy.voccia@gmail.com (SAV); ann-campbell@uiowa.edu,
barrett-thomas@uiowa.edu,

 <http://orcid.org/0000-0002-6080-6191> (BWT)

 <http://orcid.org/0000-0002-9927-687X> (AMC);

Reçu: 4 octobre 2015

Modifié: Juillet 2016

Accepté: 29 août 2016

Publié en ligne dans *Articles à l'avance*:
19 mai 2017

<https://doi.org/10.1287/trsc.2016.0732>

Copyright: © 2017 INFORMS

Abstrait. La livraison le jour même pour les achats en ligne est une tendance récente de la vente au détail en ligne. Nous introduisons un problème de ramassage et de livraison dynamique multi-véhicules avec des contraintes de temps qui intègre des fonctionnalités clés associées à la logistique de livraison le jour même. Pour prendre des décisions mieux informées, notre approche de solution intègre des informations sur les demandes futures dans les décisions de routage. Nous introduisons également un résultat analytique qui identifie quand il est bénéfique pour les véhicules d'attendre au dépôt. Nous présentons un large éventail d'expériences de calcul qui démontrent la valeur de notre approche. Les résultats montrent que davantage de demandes peuvent être remplies lorsque les fenêtres temporelles sont réparties uniformément tout au long de la journée par rapport à la période pendant laquelle de nombreuses demandes se produisent tard dans la journée. Cependant, l'anticipation des demandes futures est plus précieuse lorsque la fenêtre de temps de nombreuses demandes survient tard dans la journée.

L'histoire: Ce document a été accepté pour le *Science des transports* Numéro spécial sur les avancées récentes en transport urbain grâce à l'optimisation et à l'analyse.

Matériel supplémentaire: L'annexe en ligne est disponible à l'adresse <https://doi.org/10.1287/trsc.2016.0732>.

Mots clés: livraison le jour même • acheminement dynamique des véhicules • créneaux horaires

1. Introduction

Alors que la gratification instantanée était jadis le plus grand avantage des magasins de brique et de mortier par rapport aux détaillants en ligne, la livraison le jour même apporte désormais une gratification quasi instantanée aux acheteurs en ligne (Howard 2014). «Il se passe beaucoup de choses dans cet espace, et tout est dirigé par Amazon», explique Tom Allason, fondateur et directeur général du service de livraison le jour même basé au Royaume-Uni, Shufl (Cifford et Miller 2012, B1). Amazon propose une livraison le jour même sur plusieurs marchés urbains, et les clients Prime Nowhere les plus récemment introduits peuvent faire livrer les produits en une heure pour 7,99 \$ ou deux heures gratuitement (Amazon 2017). D'autres sociétés, comme la société de livraison d'épicerie Instacart, s'associent avec des magasins locaux de brique et de mortier. Avec un abonnement Instacart Express et une commande minimum de 35 \$, les clients peuvent demander une livraison en deux heures ou dans un délai de livraison d'une heure plus tard dans la journée gratuitement ou sélectionner une livraison d'une heure pour 5,99 \$ (Instacart 2016).

La livraison le jour même pour les achats en ligne se caractérise par une flotte de véhicules qui répondent aux demandes de livraison au cours de la journée de service. Les demandes arrivent dynamiquement; la seule information connue à l'avance est probabiliste. A chaque demande est associé un délai ou une fenêtre temporelle pendant laquelle la livraison doit avoir lieu. Les véhicules sont chargés et expédiés du dépôt, c'est-à-dire d'un entrepôt ou d'un magasin de brique et de mortier, pour répondre aux demandes. Les véhicules sont autorisés à effectuer plus d'une demande à la fois. L'objectif est de

maximiser le nombre attendu de demandes pouvant être livrées à temps. Nous appelons cela le problème de livraison le jour même (SDDP).

La livraison le jour même est offerte dans les zones urbaines où la densité de clientèle est suffisamment élevée pour la prendre en charge, mais c'est un service complexe et coûteux à opérer. Une façon de réduire les dépenses consiste à utiliser des stratégies de routage efficaces. Notamment, l'efficacité du routage peut être améliorée lorsque des informations probabilistes sont intégrées dans l'approche de la solution pour anticiper les demandes futures. Par exemple, au lieu de quitter le dépôt immédiatement après une demande, des informations probabilistes pourraient suggérer que le véhicule devrait attendre au dépôt en prévision de l'arrivée d'une autre demande dans un proche avenir. De même, l'anticipation des demandes futures peut aider à déterminer quelles demandes devraient être chargées dans un véhicule et traitées immédiatement et quelles demandes devraient être laissées pour être chargées et servies à l'avenir.

Dans cet article, nous apportons plusieurs contributions clés à la littérature. Nous introduisons un nouveau problème (le SDDP) qui représente une tendance actuelle dans les pratiques commerciales. Nous présentons un modèle formel de processus de décision de Markov (MDP) pour le SDDP. Nous introduisons également un résultat analytique qui identifie quand il est bénéfique pour les véhicules d'attendre au dépôt. Pour résoudre le SDDP, nous présentons une approche de solution heuristique basée sur

sur la planification d'un exemple de scénario. Il s'agit d'un sous-problème qui, à notre connaissance, constitue un nouveau problème dans la littérature et une nouvelle fonction consensuelle fondamentalement différente des fonctions consensuelles antérieures. Enfin, nous présentons les résultats d'un ensemble complet de tests de calcul qui démontrent la valeur de l'approche de solution proposée. Ces expériences computationnelles identifient les circonstances dans lesquelles la modélisation explicite des demandes futures incertaines a le plus d'impact sur la qualité de la solution. Plus précisément, les résultats montrent que davantage de demandes peuvent être remplies lorsque les fenêtres temporelles sont réparties uniformément tout au long de la journée par rapport aux nombreuses fenêtres temporelles des demandes qui se produisent tard dans la journée. Cependant, l'anticipation des demandes futures est plus précieuse lorsque la fenêtre de temps de nombreuses demandes survient tard dans la journée. En raison de la flexibilité accrue,

Ce document est organisé comme suit. Nous présentons une revue de la littérature dans la section 2 et notre modèle dans la section 3. Nous décrivons notre résultat analytique dans la section 4 et notre approche de solution de planification basée sur des scénarios dans la section 5. Nous décrivons nos expériences de calcul et nos résultats dans la section 6. Nous rapportons nos conclusions dans la section 7.

2. Revue de la littérature

Dans cette section, nous présentons une revue de la littérature relative au SDDP. D'une manière générale, le SDDP appartient à la grande classe des problèmes de routage dynamique des véhicules. Pour des analyses de la littérature générale sur le routage dynamique, voir Pillac et al. (2013) et Thomas (2011). Le problème le plus similaire au SDDP est présenté dans Azi, Gendreau et Potvin (2012). Leur papier traite d'un SDDP avec des fenêtres temporelles; leur objectif est de maximiser le total des bénéfices escomptés, lorsque les bénéfices sont spécifiques au client. En revanche, cherchant à offrir un haut niveau de service client, notre objectif est de maximiser le nombre attendu de demandes clients traitées. Comme pour notre article, Azi, Gendreau et Potvin (2012) utilisent une approche de planification par scénario, mais la longueur des segments de route est contrôlée par un paramètre fixe. L'approche solution présentée par Azi, Gendreau et Potvin (2012) ne permet pas non plus une attente stratégique. En revanche, dans notre approche, la longueur de la route est explicitement déterminée par notre fonction de consensus. De plus, notre consensus détermine quand il peut être avantageux d'attendre au dépôt. Ainsi, notre approche de solution permet des solutions moins restreintes que celles présentées dans Azi, Gendreau et Potvin (2012). En outre, nous examinons l'impact de différentes fenêtres temporelles et délais, où Azi, Gendreau et Potvin (2012) se concentrent sur un schéma de fenêtre temporelle qui motive le schéma TW.h dans nos expériences.

Klapp, Erera et Toriello (2018) étudient une simplification version du SDDP discutée dans cet article. À Klapp,

Erera et Toriello (2018), les décisions de répartition sont prises à intervalles fixes, tous les clients sont en ligne et toutes les demandes ont le même délai, c'est-à-dire la fin de la journée. Les auteurs présentent a priori, le déploiement et des approches de solution de programmation linéaire approximative (ALP). Les résultats computationnels démontrent que les deux dernières approches dynamiques fonctionnent bien mieux que l'approche a priori, l'approche ALP étant la plus performante. Semblable à Klapp, Erera et Toriello (2018), notre approche de solution est dynamique en ce que la solution est spécifique à l'état du problème. Contrairement à l'approche de la solution que nous présentons ici, cependant, il n'est pas évident que l'approche présentée dans Klapp, Erera et Toriello (2018) peut être appliqué à une flotte de véhicules et de réseaux généraux.

Comme Klapp, Erera et Toriello (2018), Ulmer, Thomas et Mattfeld (2016) étudient une seule variante de véhicule du SDDP mais sur un graphique général comme c'est le cas dans cet article. Contrairement à Azi, Gendreau et Potvin (2012), Klapp, Erera et Toriello (2018), et ce travail, Ulmer, Thomas et Mattfeld (2016) permettent au véhicule de se détourner des clients en service pour retourner au dépôt pour charger des colis supplémentaires. Pour surmonter le grand espace d'état associé au SDDP, les auteurs s'appuient sur l'agrégation de l'espace d'états et une approche de table de correspondance. Alors qu'Ulmer, Thomas et Mattfeld (2016) démontrent l'efficacité de leur approche dans le cas d'un seul véhicule, il est difficile de savoir si l'approche évolue en fonction du paramètre de véhicule multiple que nous présentons ici.

Le SDDP est également lié aux travaux sur le problème de routage des véhicules avec les demandes de ramassage dynamiques et stochastiques (VRPDSR). Les travaux les plus récents et les revues de littérature complètes sur le VRPDSR sont présentés à Ulmer, Mattfeld et Köster (2018) et Ulmer et al. (2019, 2015). Dans le VRPDSR, les demandes de ramassage arrivent au cours de la journée. Les demandes sont intégrées dans les itinéraires de livraison existants. Comme pour ce travail, dans le VRPDSR, le décideur détermine quel sous-ensemble de demandes sera traité. En revanche, parce que le SDDP exige qu'un véhicule retourne au dépôt avant de servir de nouvelles demandes, le SDDP exige également de déterminer quand les retours de dépôt se produisent et combien de paquets doivent être chargés lors de la visite suivante. Dans cette optique, le travail sur plusieurs périodes d'Ulmer, Mattfeld et Soeffker (2016) peut être considéré comme un cas particulier du problème dans cet article. Dans Ulmer, Mattfeld et Soeffker (2016), le décideur peut choisir de récupérer les demandes dès aujourd'hui ou de les reporter jusqu'au lendemain. Ainsi, Ulmer, Mattfeld et Soeffker (2016) créent un itinéraire pour le deuxième jour; ceci est similaire à notre document dans lequel nous déterminons quelles demandes sont satisfaites sur quelle route. D'un autre côté, nous contrôlons ici la longueur des itinéraires pour faciliter les retours de dépôts multiples en une journée. Ulmer, Mattfeld et Soeffker (2016) cherchent à maximiser le nombre de services le jour même et

fait donc face à un problème de sélection de sous-ensemble différent de celui rencontré dans notre article. D'autres problèmes liés à plusieurs périodes sont présentés dans Angelelli et al. (2009), Wen et al. (2010), et Albareda-Sambola, Fernández et Laporte (2014).

Sont également liés les problèmes liés à la livraison de produits d'épicerie, tels que ceux discutés par Campbell et Savelsbergh (2005, 2006) et Ehmke et Campbell (2014). Dans ces cas, l'épicerie est chargée sur des véhicules dans un dépôt (épicerie ou entrepôt) et est généralement livrée dans un délai strict comme dans le SDDP. Une hypothèse typique, dans la pratique et dans ces documents, cependant, est que les commandes des clients sont passées au moins un jour avant que les livraisons d'épicerie soient programmées. Ainsi, les décisions d'acceptation et d'acheminement peuvent être prises avant le jour où les itinéraires sont exécutés.

Le SDDP est également lié au problème dynamique de ramassage et de livraison (DPDP), mais diffère de la plupart des DPDP en termes de lieux de ramassage. Une majorité des DPDP dans la littérature ont des emplacements de ramassage et de livraison uniques pour chaque chargement (Arslan et al. 2016, Sheridan et al. 2013, Ghiani et al. 2009, Kiechle et al. 2009, Pureza et Laporte 2008, Sáez, Cortés et Núñez 2008). Cela est important car un véhicule peut déposer une charge à un endroit et en prendre une nouvelle à un endroit proche. Dans le SDDP, cependant, les véhicules doivent toujours retourner au dépôt pour prendre une nouvelle charge. Certains problèmes d'accès à distance (DARP) dans la littérature ont un emplacement unique qui est un arrêt obligatoire pour toutes les demandes. Un exemple est lorsque les patients font la navette entre leur domicile et un hôpital. Dans une telle situation, cependant, l'objectif est basé sur le temps d'attente des patients (Cordeau et Laporte 2007); il serait inapproprié de ramasser certains patients à l'hôpital, mais de laisser d'autres attendre le prochain véhicule. Dans le SDDP, les décisions de chargement stratégique peuvent signifier que certaines requêtes sont chargées sur un autre véhicule ultérieurement. De plus, c'est une tâche non triviale que de décider quelles requêtes doivent être chargées sur le véhicule en partance.

Notre méthode de solution s'inspire de Bent et Van Hentenryck (2004, 2007) qui utilisent l'échantillonnage pour adresser un VRPDSR avec des fenêtres temporelles difficiles. Dans ces articles, les auteurs présentent un exemple d'approche de planification de scénario qui génère des plans de routage pour les scénarios qui incluent les demandes connues et les demandes futures potentielles. Ces demandes futures sont échantillonnées à partir des distributions de probabilité connues de la demande future potentielle. Une fonction de consensus sélectionne le «meilleur» plan de routage généré à partir des scénarios; ce plan est utilisé pour déterminer la prochaine destination de chaque véhicule. Les résultats dans Bent et Van Hentenryck (2004, 2007) démontrent que l'approche de planification de scénario-échantillon avec sa fonction de consensus donne des résultats significativement meilleurs par rapport aux méthodes qui n'intègrent pas d'informations sur l'avenir

demandes. D'autres exemples de planification de scénarios types sont présentés à Hvattum, Løkketangen et Laporte (2007); Ghiani, Manni et Thomas (2012); et Srour, Agatz et Oppen (2018). Dans tous ces documents, la décision prise est de savoir où un véhicule doit aller ensuite. Notre approche de planification de scénarios d'échantillonnage utilise une fonction de consensus pour choisir un ensemble de colis à charger sur les véhicules au dépôt et l'ordre dans lequel servir les colis chargés. De plus, notre fonction de consensus détermine quand un véhicule doit attendre au dépôt en prévision de futures demandes. À notre connaissance, cette dernière caractéristique n'est prise en compte que dans Bent et Van Hentenryck (2007).

3. Formulation du modèle

Dans cette section, nous présentons notre modèle MDP formel du problème. Ce modèle fournit une description précise du problème. La formulation démontre clairement que le problème vient des malédictions de dimensionnalité dans les espaces d'état et d'actions. Dans cet esprit, nous présentons notre méthode de solution heuristique dans la section 5. Le SDDP se caractérise par une flotte de véhicules opérant à partir d'un dépôt et par un ensemble d'emplacements. Laisser M

$\{1, \dots, M\}$ être un ensemble de M véhicules identiques initialement situés au dépôt à l'heure 0. Les clients demandent un service tout au long de la journée jusqu'à une heure fixe L .

L'emplacement et l'heure de cette demande sont indiqués par y et r , respectivement. Les arrivées de demandes sont décrites par un taux d'arrivée et une répartition connus.

Bien que notre méthode de solution ne repose pas sur elle, pour simplifier l'exposition, nous supposons que l'ensemble $\{0, 1, \dots, JE\}$ représente tous les emplacements de clients potentiels dans le problème. Dans l'ensemble JE , le dépôt est représenté par 0 et les emplacements des clients sont représentés par $1, \dots, JE$. Chaque emplacement client peut représenter un client individuel ou une agrégation de clients. Par exemple, dans un scénario de ville, les clients peuvent être regroupés par blocs de ville. Nous supposons que le temps de trajet entre deux emplacements $i, j \in JE$ est déterministe et est représenté par τ_{ij} . Le taux d'arrivée des demandes sur place je est représenté par λ_{je} . Plusieurs demandes peuvent arriver du même endroit au cours de la journée.

Associé à chaque demande n est un temps de service connu μ_n et un délai de livraison $[e_n, l_n]$ chez le client. Le temps de service est supposé être une fonction de l'emplacement faisant la demande et est connu à l'avance, mais toutes les autres informations associées à une demande deviennent connues une fois que la demande est réalisée. La livraison de la demande n doit se produire à l'heure ou après e_n , mais au plus tard l_n . La restriction de commencer le service après e_n signifie qu'un véhicule peut devoir attendre à l'emplacement de la demande. Lorsque seul un délai de livraison est respecté, e_n est réglé sur l_n , le temps de la n e demande. Dans ce travail, nous ne modélisons pas de temps de chargement pour chaque demande n ni l'inclure dans notre calcul

expériences. Nous le faisons car le temps de chargement est probablement négligeable dans notre application motivante. Néanmoins, le modèle peut facilement être adapté pour inclure un temps de chargement, et l'approche de solution proposée n'est pas affectée par l'inclusion d'un tel temps.

Les demandes peuvent être servies par la flotte de véhicules existante ou par un tiers. Nous supposons qu'il est plus coûteux de traiter une demande via le tiers que via la flotte existante, mais nous ne modélisons pas explicitement les coûts ou les décisions d'acheminement pour le tiers. Nous modélisons uniquement l'attribution des demandes au tiers, mais nous reportons cette décision jusqu'à ce que nous déterminions qu'il est impossible pour la livraison d'être effectuée par la flotte de véhicules existante. Pour les demandes servies par la flotte existante, nous ne permettons pas que les contraintes de temps d'une demande soient violées. De plus, nous supposons qu'une fois qu'une demande est chargée sur un véhicule, la demande sera livrée par ce véhicule.

Une fois les demandes faites, un véhicule $m \in M$ au dépôt peut être assigné des demandes et quitter le dépôt immédiatement. Alternativement, un véhicule peut attendre au dépôt avant de recevoir des demandes. Une fois qu'un véhicule quitte le dépôt, l'itinéraire pour ce véhicule est fixé et le véhicule retourne au dépôt lorsqu'il a effectué toutes les livraisons qui lui ont été attribuées. Les itinéraires des véhicules sont limités de telle sorte que le véhicule retourne au dépôt avant la date limite du dépôt $L + \Delta$. L'objectif du problème est de maximiser le nombre attendu de demandes qui peuvent être satisfaites par la flotte existante dans les délais impartis. Cet objectif est un proxy pour offrir le meilleur niveau de service client.

Nous formulons le SDDP comme un MDP, qui modélise les décisions prises à chaque époque de décision.

3.1. Territoire de l'État

L'état du système contient toutes les informations nécessaires pour prendre des décisions de chargement et de routage à l'époque de la décision k , ainsi que les décisions concernant l'attribution des demandes au tiers. L'état comprend l'heure, les emplacements des demandes, les fenêtres horaires des demandes, les heures d'arrivée des véhicules au dépôt et un indicateur identifiant si le véhicule se trouve au dépôt. Soit l'ensemble N_k

représenter l'ensemble des demandes réalisées à l'époque k qui n'ont pas été chargés sur un véhicule M ou attribué à un tiers. Nous représentons les attributs de la demande n

dans N_k par le tuple (γ_n, e_n, l_n) . Laisser γ_n être l'emplacement de demande n . Laisser e_n être le début de la fenêtre temporelle et l_n être la date limite de la fenêtre de temps pour la demande n . Laisser $(\gamma_n, e_n, l_n)_{n \in N_k}$ désigne le vecteur de demande les attributs.

Nous représentons les attributs du véhicule $m \in M$ à l'époque k par le tuple (ρ_m, t_m) . L'attribut ρ_m représente le prochaine arrivée prévue du véhicule m au dépôt ou la dernière heure jusqu'à laquelle un véhicule au dépôt doit attendre avant d'être reconsidéré pour l'affectation. L'attribut t_m

est mis à jour lorsqu'un véhicule m quitte le

dépôt ou attend au dépôt. De plus, lorsqu'un véhicule attend au dépôt, l'indicateur 1_{mk} est défini sur 1; sinon, il est mis à 0. Laisser $(\rho_m, t_m)_{m \in M}$

dénoter la vecteur des attributs du véhicule. La combinaison de ces informations dans le tuple $(t_k, (\gamma_n, e_n, l_n)_{n \in N_k}, (\rho_m, t_m)_{m \in M})$, nous avons l'état du système s_k .

3.2. Espace d'action

Une action est sélectionnée à chaque époque de décision. Pour chaque demande en N_k , il est décidé de l'affecter à un véhicule M , attendre pour affecter la demande ou l'attribuer à un tiers. Les demandes ne peuvent être attribuées qu'aux véhicules du dépôt et ne peuvent pas être attribuées aux véhicules en route. Nous supposons que les demandes affectées à un véhicule sont traitées dans l'ordre de l'itinéraire de durée minimale qui satisfait les fenêtres de temps de demande et la date limite du dépôt, et que le véhicule part immédiatement. Nous discutons de notre politique de routage dans la section 5.2. Les véhicules qui ne sont pas affectés à des demandes attendent au dépôt jusqu'à une période de temps fixe η avant d'être reconsidéré pour affectation.

une action une_k à l'époque de la décision k est un M -vecteur dimensionnel où le m élément une_m

est l'action associée avec véhicule m dans M . Une action spécifique au véhicule une_m représente le sous-ensemble de demandes N_k être servi par véhicule m et peut être l'ensemble vide. Laisser $\phi(une_m)$

être une fonction qui renvoie l'heure à laquelle le véhicule m retourne au dépôt compte tenu de l'ensemble des clients spécifiés par l'action

une_m et le temps t_k de l'époque actuelle. Pour plus de commodité, laissez $M_k = \{m \in M \mid 1_{mk} = 1\}$ être l'ensemble des véhicules

des au dépôt à l'époque k . L'ensemble des actions disponibles dans l'état s_k pour véhicule m est

$$\{une_m \mid m \in M_k\} \quad (1)$$

$$une_m \subseteq N_k \text{ pour } m \in M_k \quad (2)$$

$$\bigcap_{m \in M_k} une_m = \emptyset \quad (3)$$

$$\phi(une_m) - t_k \leq L + \Delta \quad (4)$$

État (1) indique que si le véhicule m est en route, aucune demande supplémentaire n'est affectée à ce véhicule. État (2) autorise un sous-ensemble de demandes dans N_k à affecter au véhicule m si véhicule m est au dépôt. Cependant, il est possible que le sous-ensemble sélectionné soit l'ensemble vide. État (3) exige que chaque demande soit affectée, au plus, à un véhicule. État (4) exige que ce véhicule

m retourne au dépôt avant la date limite du dépôt. Nous désignons par $UNE(s_k)$ l'ensemble des actions disponibles dans l'état s_k .

Donc, $UNE(s_k)$ se compose de tous les vecteurs d'action réalisables.

3.3. Transition vers un état post-décision

État donné s_k et l'action sélectionnée $une_k \in UNE(s_k)$, une transition déterministe est effectuée vers un état post-décision

$s_{k+1} = (t_{k+1}, (\gamma_n, e_n, l_n)_{n \in N_{k+1}}, (\rho_m, t_m)_{m \in M_{k+1}})$. Dans cette transition, le temps reste inchangé. Donc, $t_{k+1} = t_k$. L'ensemble N_{k+1} comptes

pour les demandes de transition N_k . Une requête n transitions hors de l'ensemble N_k lorsqu'il est chargé sur un véhicule M' ou attribué à un tiers. Une demande est attribuée à un tiers lorsqu'il n'est plus possible que la demande soit servie par un véhicule au dépôt ou l'un des véhicules en route. Pour les demandes qui restent N_{ak} ,

les demandes non attribuées à un véhicule ou à un tiers au moment t_k , les emplacements des demandes et les fenêtres horaires restent inchangés.

Si l'action choisie une_m pour véhicule $m \in M'_k$ à la décision époque k , contient les demandes, puis l'heure d'arrivée au dépôt pour le véhicule m est déterminé par la fonction $\varphi(une_m$

$t_k)$. Si la place une_m ne contient pas de demandes pour véhicule m dans M'_k , puis véhicule m attendra au dépôt pendant un maximum de temps η avant d'être reconsidéré pour affectation; Donc ρ_{ma}

$t_k + \eta$. Si véhicule $m \in \{M \setminus M'_k\}$, l'heure d'arrivée du dépôt reste inchangée. Ainsi, les heures d'arrivée du dépôt sont mises à jour

$$\rho_{ma} = \begin{cases} \varphi(une_m, t_k), & m \in M'_k \text{ tel que } une_m \text{ } \\ t_k + \eta, & m \in M'_k \text{ tel que } \\ \rho_{ma}^m, & m \in M \setminus M'_k. \end{cases} \quad (5)$$

Chaque véhicule auquel des demandes sont attribuées doit quitter le dépôt. Par conséquent, I_{mk} est mis à jour en tant que

$$I_{mk} = \begin{cases} 0, & \text{si } (I_{mk} \\ I_{mk}, & \text{autrement.} \end{cases} \quad (6)$$

Nous reconnaissons que la spécification $I_{mk} = 1$ dans les premières conditions de redondance étant donné que une_m ne peut être vrai que pour véhicules au dépôt. Cependant, l'inclusion de cette redondance indique clairement qu'une transition de 1 à 0 est en cours.

3.4. Transition vers un État pré-décisionnel

Le MDP passe de l'état post-décision S_{une} à l'état de la décision $S_{k+1} = (t_{k+1}, \Psi_{k+1}, \Theta_{k+1}, I_{k+1}, \rho_{k+1}, I_{k+1})$ lorsque la prochaine époque est déclenchée. La $k+1$ La première époque se produit au moment t_{k+1} à la suite d'au moins l'une des situations suivantes:

1. un véhicule arrive au dépôt ou un véhicule termine sa temps d'attente.
2. une nouvelle demande arrive et au moins un véhicule est en attente au dépôt.

Ainsi, le temps de la prochaine époque de décision est probabiliste. Laisser B être une variable aléatoire qui représente l'interarrivée de la prochaine demande.

Laisser b être une réalisation de B . ensuite t_{une}

$$t_{k+1} = \begin{cases} \min_{m \in M} \{ \rho_{ma}^k, t_{une} + b \}, & \text{si } M'_k \neq \emptyset, \\ \min_{m \in M} \{ \rho_{ma}^k \}, & \text{si } M'_k = \emptyset, \end{cases} \quad (7)$$

où $M'_k = \{ m \mid I_{mk} = 1, \forall m \in M \}$.

L'arrivée des véhicules au dépôt dépend de l'heure de l'époque. Par conséquent, I_{mk+1} est mis à jour pour les véhicules qui arrivent à l'époque $k+1$

$$I_{mk+1} = \begin{cases} 1, & \text{si } (I_{ma}^k = 0) \wedge (\rho_{ma}^k < t_{k+1}), \\ I_{ma}^k, & \text{autrement.} \end{cases} \quad (8)$$

Nous reconnaissons que la spécification $I_{ma}^k = 0$ dans le premier est redondant étant donné que $\rho_{ma}^k < t_{k+1}$ implique que $I_{ma}^k = 0$. Cependant, inclure clairement cette redondance identifie qu'une transition de 0 à 1 est en cours.

Pendant la transition également, les attributs des demandes dans N_{ak} restent inchangés lors de leur transition vers N_{k+1} .

Toute nouvelle demande est observée et ces demandes sont ajoutées à l'ensemble N_{k+1} .

3.5. Contribution, critère et objectif

Choisir l'action une_k quand en état s_k produit une récompense. Laisser

$N_{une} = \{ n \mid n \in U, n \in M \setminus M'_k \}$ être l'ensemble des demandes attribuées à un véhicule M à l'époque de la décision k . Ensuite, la fonction de contribution est

$$C_k(s_k, une_k) = \sum_{n \in N_{une}} r_n \quad (9)$$

Laisser K être une variable aléatoire qui représente le nombre total d'époques de décision. Une règle de décision à l'époque k est une fonction $\delta(s_k)$ qui sélectionne une action disponible $une_k \in \text{UNE}(s_k)$. Une police d'écriture π est une séquence de règles de décision. Laisser Π représenter l'ensemble de toutes les politiques possibles. Le critère est

$$V_{\pi}(s_0) = \mathbb{E}_{\pi} \left[\sum_{k=0}^{K-1} C_k(s_k, \delta_{\pi}(s_k)) \mid s_0 \right] \quad (\text{dix})$$

L'objectif est de trouver une politique optimale π^* tel que

$V_{\pi^*}(s_0) \geq V_{\pi}(s_0)$ pour tous $\pi \in \Pi$.

4. Retarder le départ du dépôt

Dans la section 5, nous présentons une méthode de solution heuristique pour le SDDP. Cependant, comme c'est souvent le cas, il peut être possible d'identifier la structure dans la solution ou la politique optimale qui peut améliorer la qualité des méthodes de solution heuristique. Dans cette section, nous identifions une telle structure.

Dans le SDDP, l'une des décisions cruciales est de savoir quand un véhicule doit quitter le dépôt. D'une part, le travail est souvent préféré au temps d'inactivité (voir Thomas 2007). D'un autre côté, le fait d'attendre des véhicules au dépôt peut se traduire par des itinéraires plus efficaces car plus de demandes peuvent s'accumuler avant d'être affectées à un itinéraire. Dans cette section, nous montrons que, pour un ensemble de demandes donné, il est possible de retarder le départ d'un véhicule du dépôt sans impact sur la valeur objective.

La durée pendant laquelle un véhicule peut attendre au dépôt est appelée temps de retard maximum (MDT). Le MDT est calculé

sur la base de l'itinéraire le plus court possible pour un ensemble de demandes donné. En raison de l'existence de fenêtres horaires, un véhicule peut arriver à un emplacement de demande avant le début de la fenêtre horaire et devra donc attendre avant de commencer le service. Compte tenu d'un itinéraire proposé, le calcul MDT utilise ces temps d'attente pour déterminer combien de temps un véhicule peut rester au dépôt sans respecter les délais et sans modifier le temps de retour au dépôt. La formule MDT est présentée dans la proposition 1. Notre résultat principal est présenté dans la proposition 2. Le résultat indique qu'en retardant l'heure de départ par le MDT, au moins autant de clients peuvent être servis sans retarder l'heure de départ. Les résultats de cette section suggèrent que, si un véhicule a une valeur positive pour le MDT, les demandes sur l'itinéraire proposé ne devraient pas être attribuées et le véhicule ne devrait pas partir.

Avant de présenter des propositions 1 et 2, nous définissons notation et quantités supplémentaires. Nous définissons un itinéraire comme une commande de demandes de telle sorte que la commande commence et se termine au dépôt. Associer le numéro $0, 1, \dots, J, J + 1$ avec la position d'une demande sur le circuit, où le dépôt est en position 0 et $J + 1$. Soit $[e_j, l_j]$ et μ_j être la fenêtre de temps et le temps de service associés à la demande en position j en tournée. Ces valeurs sont connues à l'avance pour les emplacements des clients. De même, laissez $\tau_{j,j+1}$ être le temps de trajet à partir de la demande en position j sur le tour à la demande en position $j + 1$ sur la tournée. Ces quantités sont également connues à l'avance. Nous définissons également les quantités suivantes:

$ré_j$ l'heure de départ du véhicule à la demande position j ,

UNE_j l'heure d'arrivée du véhicule à la demande position j ,

W_j le temps qu'un véhicule passe à attendre à la demande en position j pour commencer le service,

CW_j le temps d'attente cumulé d'un véhicule desservant demandes en position 0 à j en tournée, et

DS_j le délai à la demande en place j , qui est la quantité de temps UNE_j peut être retardé sans violer l_j .

Nous supposons $W_0 = 0$, $ré_0 = t_k$, et $CW_0 = 0$. Les valeurs UNE_j et $ré_j$ sont calculés récursivement en utilisant les formules suivantes:

$$UNE_j = ré_{j-1} + \tau_{j-1,j}, \quad 1 \leq j \leq J + 1, \quad \text{et} \quad (11)$$

$$ré_j = \max \{ UNE_j, e_j + \mu_j \}, \quad 1 \leq j \leq J. \quad (12)$$

Le temps d'attente est calculé à l'aide de la formule suivante:

$$W_j = \max \{ e_j - UNE_j, 0 \}, \quad 1 \leq j \leq J. \quad (13)$$

Alternativement, l'heure de départ peut être réécrite en utilisant le temps d'attente comme

$$ré_j = UNE_j + W_j + \mu_j, \quad 1 \leq j \leq J. \quad (14)$$

Les temps d'attente cumulés et les délais sont calculés avec les formules suivantes:

$$CW_j = \sum_{k=0}^j W_k, \quad 1 \leq j \leq J, \quad \text{et} \quad (15)$$

$$DS_j = l_j - UNE_j, \quad 1 \leq j \leq J. \quad (16)$$

Compte tenu de cette notation, nous présentons la proposition 1. La preuve détaillée est fournie dans l'annexe en ligne.

Proposition 1. *Étant donné une visite possible, laissez $ré_0$ être l'heure de départ du dépôt au début de la visite et UNE_{J+1} être l'heure d'arrivée au dépôt à la fin de la tournée. La visite peut être retardée de quitter le dépôt par g unités de temps de sorte que la faisabilité de la fenêtre de temps soit maintenue et que le nouveau temps de retour au dépôt soit $UNE_{Nouveau}$*

$g \leq UNE_{J+1} - ré_0$. Le MDT est le maximum tel g et est calculé par la formule suivante:

$$MDT = \min \{ \min_{1 \leq j \leq J} \{ DS_j + CW_{j-1} \}, CW_J \}. \quad (17)$$

Formule (17) indique que le MDT est le minimum du temps d'attente cumulé jusqu'à J et le temps que UNE_j peut être déplacé vers l'avant pour $j, 1 \leq j \leq J$, sans dépassement de délai.

Il découle directement de la proposition 1 que, étant donné une politique optimale qui attribue un ensemble de demandes à un véhicule, il existe également une politique optimale qui retarde le départ de ce véhicule de MDT servir au moins autant de clients que la politique actuelle. Le résultat est formalisé dans la proposition 2.

Proposition 2. *Étant donné une tournée réalisable qui laisse le dépôt à $ré_0$, il existe une politique qui sert au moins autant de clients au départ à $ré_0 + MDT$ par rapport à la tournée au départ de $ré_0$.*

Un inconvénient de la proposition 2 est que l'optimalité du retard par MDT repose sur l'utilisation d'une politique optimale. Comme indiqué précédemment et discuté en détail dans la section suivante, nous devons résoudre le problème de manière heuristique et ne pouvons donc pas garantir que nous prenons des décisions optimales à chaque époque de décision. Ainsi, la valeur de l'utilisation de la proposition 2 avec une approche heuristique ne peut être déterminée que par des expériences de calcul. Nous présentons les résultats de ces expériences dans la section 6.

5. Approche de la solution

Traditionnellement, l'équation (dix) a été résolu en utilisant l'équation de Bellman et la programmation dynamique en arrière. En raison de la taille des espaces d'état et d'actions, cependant, la résolution du SDDP en utilisant la programmation dynamique arrière traditionnelle est impossible pour tous, sauf les plus petits problèmes. En conséquence, nous nous tournons vers une approche de programmation dynamique approximative vers l'avant. L'approche est approximative ou heuristique

parce que nous ne connaissons pas le coût futur exact d'une décision prise à l'époque de la décision actuelle. Un aperçu de ces approches est présenté dans Powell (2011). Dans notre approche de programmation dynamique vers l'avant, nous sélectionnons les actions en utilisant l'approche par exemple de scénario dont nous discuterons dans la section 5.1. Les récompenses et les transitions suivent comme décrit dans le modèle de la section 3.

5.1. Approche de planification basée sur des scénarios

Comme indiqué ci-dessus, dans notre approche de programmation dynamique avancée, nous sélectionnons les actions en utilisant une approche de planification de scénario exemple. Dans la planification basée sur des scénarios d'échantillonnage, à chaque époque de décision, différents scénarios sont construits sur la base de l'état de pré-décision actuel et d'échantillons aléatoires de l'ensemble de paramètres stochastiques. Pour chaque scénario, nous résolvons le problème de routage déterministe résultant. Dans l'ensemble des solutions, une pour chaque scénario, nous choisissons ensuite une solution, le plan distingué, qui guide notre sélection d'actions pour l'époque de décision.

Notre algorithme de planification basé sur des scénarios est présenté dans Algorithm 1. Algorithm 1 prend plusieurs informations en entrée. Le premier paramètre qu'il prend est l'ensemble N_k qui est l'ensemble des demandes réalisées mais non affectées à l'époque de décision actuelle k . Il prend également un vecteur de paramètres appelé $params_k$. Le vecteur comprend les paramètres associés aux distributions d'arrivée et de fenêtre temporelle qui seront utilisés pour générer des données de scénario à l'époque de la décision k . Il comprend également les paramètres suivants: l'horizon d'échantillonnage, l'ensemble des emplacements des clients JE , le temps de l'époque t_k , le délai pendant lequel les demandes peuvent continuer d'arriver (L),

et la fin de la journée ($L + \Delta$). Le paramètre d'entrée $nbr_scenarios$ est le nombre de scénarios à générer. Le vecteur $partir_k$ stocke la première heure à laquelle chaque véhicule peut quitter le dépôt. Laisser $partir_{mk}$ représente la première heure à laquelle un véhicule $m \in M$ peut partir du dépôt. Les premières heures de départ sont calculées comme

$$\begin{aligned} & \{ t_k, m \in M' \cdot k, \\ & partir_{mk} \quad P_{\mathbb{R}} \quad m \in M \setminus M' \cdot k. \end{aligned} \quad (18)$$

Équation (18) indique que pour les véhicules au dépôt à l'époque k , l'heure de départ la plus ancienne est égale à l'heure de l'époque t_k . Pour les véhicules en route à l'époque k , l'heure de départ la plus ancienne est égale à l'heure de retour du véhicule au dépôt. Le dernier paramètre d'entrée pour l'algorithme 1 est $M' \cdot k$, qui est l'ensemble des véhicules actuellement au dépôt à l'époque k . Le résultat de l'algorithme 1 est le plan distingué $\rho +_k$.

Algorithme 1 (Planification basée sur des scénarios)

Contribution: $N_k, params_k, nbr_scenarios, départ_k, M' \cdot k$

Production: $\rho +_k$

1: $ré_k \leftarrow GenerateScenarioData (N_k, params_k, nbr_scenarios)$

2: $S_k \leftarrow GenerateScenarioPlans (ré_k, partir_k)$

3: $\rho +_k \leftarrow Consensus(S_k, M' \cdot k)$

Un élément clé de la planification basée sur les scénarios est la génération de nouveaux plans de routage pour les scénarios qui incluent les demandes existantes et les demandes futures potentielles. Par conséquent, la première étape de l'algorithme 1 est de générer des scénarios. La fonction $GenerateScenarioData (N_k, params_k$

$nbr_scenarios)$ est responsable de la tâche. Un scénario contient l'ensemble des demandes réalisées dans N_k et également les demandes futures potentielles qui sont générées à partir d'une distribution de probabilité. Le $params_k$ vector contient toutes les informations nécessaires pour générer de futures demandes potentielles. Un élément important de $params_k$ est l'horizon d'échantillonnage. L'horizon d'échantillonnage spécifie jusqu'où les futures demandes potentielles devraient être générées. Nous appelons les demandes futures potentielles des demandes échantillonnées. Le paramètre $nbr_scenario$ spécifie le nombre de scénarios à générer. Laisser $ré_k$ représente l'ensemble des scénarios générés renvoyés par $GenerateScenarioData$.

La prochaine étape de l'algorithme 1 est de générer des plans de scénario.

Cela est possible grâce à la fonction $GenerateScenarioPlans (ré_k, partir_k)$. Un

plan de scénario est un plan de routage qui contient les demandes réalisées et échantillonnées. Étant donné un scénario de l'ensemble $ré_k$ et heures de départ du véhicule $partir_k$, un plan de scénario spécifie un itinéraire pour chaque véhicule à partir de l'heure actuelle t_k à la fin de la journée. Un itinéraire pour un véhicule est représenté par un ensemble ordonné de demandes, tandis qu'un plan de scénario est représenté par un ensemble d'itinéraires. Nous utilisons les expressions «plan de scénario» et «plan de routage de scénario» de manière interchangeable. Laisser S_k représente l'ensemble des plans de scénario renvoyés par la fonction. La fonction

$GenerateScenarioPlans$ résout un sous-problème déterministe du SDDP. Ce sous-problème et notre méthode de solution sont décrits dans la section 5.2.

La dernière fonction dans l'algorithme 1 est le consensus. Cette fonction prend un ensemble de plans de scénario et renvoie un seul plan $\rho +_k$, que nous appelons le plan distingué. Ce plan distingué guide la sélection des actions à l'époque k . Nous présentons la fonction de consensus plus en détail dans la section 5.3.

5.2. Optimisation des sous-problèmes

La fonction $GenerateScenarioPlans$ dans Algorithm 1

est responsable de la génération des plans de routage des scénarios pour chaque scénario généré à l'époque de décision. Chaque scénario représente un avenir possible et la génération d'un plan de routage pour cet avenir se traduit par un problème d'optimisation déterministe que nous résolvons à l'aide d'une méthode heuristique. Étant donné que chaque scénario contient des demandes réalisées qui n'ont pas été attribuées à un véhicule dans la flotte ou à un tiers ainsi que des demandes échantillonnées, il est peu probable qu'un plan de routage de scénario réalisable qui achemine tous les clients existe en ce qui concerne les fenêtres de temps. Par conséquent, le sous-problème déterministe que nous

chercher à résoudre est un problème d'orientation dans lequel le plus grand sous-ensemble de demandes pouvant être satisfaites est choisi. Nous appelons ce sous-problème le problème d'orientation déterministe multi-voyages en équipe avec fenêtres de temps (MTTOPTW).

L'optimisation du sous-problème crée des itinéraires pour tous les véhicules et permet plusieurs déplacements pour chaque véhicule. Les véhicules en route à l'époque de décision actuelle doivent retourner au dépôt avant de commencer leur premier voyage dans le sous-problème. La première heure à laquelle chaque véhicule peut commencer son premier voyage est spécifiée par $partir_{mk}$ comme défini dans la section 5.1. Les véhicules doivent terminer leur dernier voyage à temps $L + \Delta$.

Le MTTOPTW appartient à la classe des problèmes d'orientation (OP) et est une extension du problème d'orientation d'équipe avec des fenêtres temporelles (TOPTW). Une étude de la littérature sur l'OP et ses variantes est présentée à Vansteenwegen, Souffriau et Oudheusden (2011). À notre connaissance, le MTTOPTW n'a pas été présenté dans la littérature. Nous fournissons une formulation de programmation entière du MTTOPTW et décrivons comment le sous-problème SDDP peut être transformé en une instance du MTTOPTW dans l'annexe en ligne.

Parce que les PO en général sont difficiles à résoudre exactement et que notre problème introduit le défi supplémentaire de plusieurs trajets, nous résolvons un assouplissement du sous-problème en permettant des violations de délais avec des pénalités. Si les violations de délais n'étaient pas autorisées, la méthode de solution choisie devrait rechercher un espace de solution avec des plans de routage contenant de nombreuses combinaisons différentes de sous-ensembles de l'ensemble complet des requêtes. Un seul sous-ensemble de demandes peut également avoir de nombreuses affectations de véhicules et des ordres d'itinéraire différents. Par conséquent, il serait difficile de trouver le plus grand sous-ensemble de demandes pouvant être satisfaites. Avec cette relaxation, une fonction objectif à deux niveaux est mise en œuvre. L'objectif principal est de minimiser le nombre de demandes avec des contraintes de temps violées. L'objectif secondaire est de minimiser la durée totale de l'itinéraire.

Nous implémentons *variable neighborhood search* (VNS) comme méthode d'optimisation de nos sous-problèmes. Une implémentation standard de VNS est décrite dans Voccia, Campbell et Thomas (2013). Voccia, Campbell et Thomas (2013) présentent un VNS pour un seul véhicule. Pour résoudre le MTTOPT en utilisant le VNS, nous codons tous les itinéraires de véhicules comme un seul itinéraire. Dans l'encodage, nous séparons chaque route par une requête factice. Tomodel voyages multiples, le calcul de la durée de l'itinéraire intègre implicitement les retours de dépôt. Le temps de trajet est ajouté pour les retours de dépôt lorsqu'une demande échantillonnée se produit sur un itinéraire, mais le temps de réalisation de la demande se produit après l'heure du véhicule.

visite de dépôt la plus récente. Le VNS se termine si la solution n'a pas été mise à jour au cours des cinq dernières itérations.

5.3. Fonction consensus

Dans la planification basée sur des scénarios, la fonction de consensus est utilisée pour identifier la meilleure solution parmi un ensemble de solutions possibles. Cette meilleure solution, après avoir été réparée pour la faisabilité, représente l'ensemble des actions recommandées pour l'époque de la décision. Pour le SDDP, cela signifie définir l'ensemble des requêtes à affecter aux véhicules du dépôt. Dans certains cas, l'ensemble des demandes peut être vide pour des véhicules particuliers. Cela indique que ces véhicules devraient attendre au dépôt jusqu'à ce qu'une affectation soit effectuée à une époque ultérieure. Un ensemble vide pourrait indiquer qu'aucune demande disponible n'entraînerait une affectation possible, ou pourrait indiquer la possibilité d'une opportunité de routage plus efficace à l'avenir. Cependant, le report des décisions peut entraîner l'envoi de demandes à être signifiées par un tiers pour remplir l'engagement de la fenêtre de temps.

La fonction de consensus que nous mettons en œuvre pour le SDDP est fondamentalement différente de l'exemple classique d'une fonction de consensus dans Bent et Van Hentenryck (2004). Notamment, dans Bent et VanHentenryck (2004), la fonction de consensus est utilisée pour déterminer le prochain client pour chaque véhicule. Pour ce faire, il identifie les clients les plus fréquemment sélectionnés comme prochaine destination dans les plans de scénario. L'idée est que le prochain choix du client devrait pouvoir s'adapter à un ensemble robuste de scénarios futurs. Le SDDP requiert une fonction de consensus fondamentalement différente pour prendre des décisions concernant les demandes à traiter avec la flotte donnée et celles à sous-traiter, quand les demandes et les véhicules doivent attendre au dépôt et l'ordre dans lequel les demandes doivent être traitées. Cependant, nous maintenons que la meilleure solution devrait s'adapter à un ensemble robuste de scénarios futurs.

Algorithme 2 (Consensus)

Contribution: S_k, M'_k

Production: $\rho + k$

- 1: $P_k \leftarrow \text{ConstructPartialScenarioPlans}(S_k, M'_k)$
- 2: $\rho^k \leftarrow \text{SelectConsensusPlan}(P_k)$
- 3: $\rho + k \leftarrow \text{ConstructFeasiblePlan}(\rho^k)$

Nous présentons notre fonction de consensus dans l'algorithme 2.

La partie principale de la fonction de consensus se produit dans `SelectConsensusPlan`, où un seul plan est sélectionné à partir d'un ensemble de plans. La fonction `ConstructPartialScenarioPlans` effectue une tâche de pré-traitement sur tous les plans de scénario, et `ConstructFeasiblePlan` effectue une tâche de post-traitement sur le plan sélectionné par `Select-ConsensusPlan`.

La fonction `ConstructPartialScenarioPlans` modifie chaque plan de scénario dans S_k . La fonction exécute deux étapes. Premièrement, il supprime les routes associées aux véhicules en route. Cela signifie que le consensus

La fonction ne peut affecter des demandes qu'aux véhicules du dépôt. Deuxièmement, pour chaque itinéraire qui contient au moins une demande échantillonnée, il supprime la demande échantillonnée qui se produit à la position la plus ancienne sur l'itinéraire et toutes les demandes échantillonnées et réalisées qui sont placées à des positions ultérieures sur l'itinéraire. La deuxième étape est effectuée car les demandes échantillonnées représentent des demandes futures potentielles. Si la demande échantillonnée dans la position la plus ancienne devait être réalisée, le véhicule devrait alors retourner au dépôt avant de signifier la demande. Par conséquent, la demande échantillonnée indique un moment approprié pour planifier un retour au dépôt. De plus, les décisions concernant les demandes réalisées dans des positions ultérieures sur l'itinéraire peuvent être reportées. Nous appelons le plan modifié un **plan de scénario partiel**. La fonction `ConstructPartialScenarioPlans` renvoie P_k , le multiset de plans de scénarios partiels.

La fonction `SelectConsensusPlan` prend en entrée le multi-ensemble de plans de scénario partiels et sélectionne un seul plan de scénario partiel. Nous appelons cela le **plan consensuel** p^*_k . La sélection du plan est guidée par une fonction qui associe un nombre réel à chaque plan de scénario partiel dans P_k . Nous appelons cette fonction **Score**. Pour décrire le mappage de la fonction `Score`, nous devons d'abord définir les concepts de routes identiques et de plans de scénarios partiels identiques:

1. Deux routes sont identiques si et seulement si les deux contiennent le même ensemble ordonné de demandes.
2. Deux plans de scénario partiels sont identiques si et seulement si les deux contiennent le même ensemble de routes.

On laisse la fonction `Identique` (ρ, ρ') 1 si partiel des plans ρ et ρ' sont identiques et 0 sinon. Notez que tous les véhicules au dépôt à l'époque k sont mathématiquement identiques car ils ont la même capacité à répondre à un ensemble de demandes. Par conséquent, notre définition de plans de scénarios partiels identiques ne prend pas en compte les véhicules individuels.

Pour chaque plan partiel $\rho \in P_k$, la fonction `Score` compte le nombre de plans dans P_k qui sont identiques ρ .

La fonction `Score` attribue ensuite ce nombre à ψ_ρ comme le score du plan partiel ρ . Plus formellement, pour chacun $\rho \in P_k$

$$\psi_\rho = \sum_{\rho' \in P_k, \rho' \neq \rho} \text{Identique}(\rho, \rho') \quad (19)$$

`SelectConsensusPlan` sélectionne ensuite le plan de scénario partiel avec la valeur la plus élevée. Mathématiquement, `SelectConsensusPlan` est défini comme

$$p^*_k = \operatorname{argmax}_{\rho \in P_k} \{ \psi_\rho \} \quad (20)$$

Nous supposons que les liens sont rompus arbitrairement.

La fonction finale de l'algorithme 2 est `ConstructFeasiblePlan`. L'apport à cette fonction est le plan de consensus p^*_k , ce qui peut ne pas être possible en ce qui concerne les fenêtres temporelles. L'impossibilité peut survenir parce que le

méthode de solution de sous-problème présentée dans la section 5.2 résout une relaxation du problème d'origine. Si nécessaire, la fonction `ConstructFeasiblePlan` construit un plan de scénario partiel réalisable à partir d'un plan irréalisable. Le plan partiel du scénario réalisable qu'il renvoie est le plan distinct p^*_k . Alors que `SelectConsensusPlan` sélectionne un plan partiel de scénario unique parmi l'ensemble de tous les plans partiels de scénario, c'est le plan distinct qui fournit l'ensemble d'actions recommandé pour l'époque de décision. Si cette procédure supprime toutes les demandes d'un itinéraire, le résultat est que l'itinéraire est un ensemble vide.

Nous décrivons maintenant comment `ConstructFeasiblePlan` construit un plan réalisable à partir d'un plan infaisable. Nous nous concentrons sur la suppression des demandes des itinéraires car l'optimisation des sous-problèmes cherche à minimiser les violations de fenêtre de temps. Ainsi, le réarrangement des clients sur les itinéraires est susceptible de créer une infaisabilité ailleurs dans la solution. Nous commençons au début de l'itinéraire et vérifions chaque demande pour une violation du délai. La première demande avec un dépassement de délai est supprimée. Une fois la demande supprimée, nous mettons à jour les heures d'arrivée aux arrêts suivants et continuons la procédure en recherchant le prochain arrêt avec violation. Nous continuons à supprimer les clients dont les fenêtres horaires ont été violées et à mettre à jour les heures d'arrivée jusqu'à la fin de la visite. Une fois cette procédure terminée, une procédure similaire peut être utilisée pour une violation de délai de dépôt. cependant,

6. Expériences de calcul

Dans cette section, nous présentons des expériences de calcul utilisant des ensembles de données avec une variété de géographies de localisation et de contraintes de temps différentes. Section 6.1 décrit la construction des ensembles de données utilisés dans les expériences, et la section 6.2 détaille les tests exécutés pour chaque ensemble de données. Nous présentons notre procédure de paramétrage dans la section 6.3 et nos résultats dans la section 6.4.

6.1. Ensembles de données

Les ensembles de données pour le SDDP sont adaptés des ensembles de données bien connus de Salomon (1987) pour le VRPTW ainsi que pour les ensembles de données étendus de Solomon de Gehring et Homberger (1999). Nous utilisons uniquement les informations de localisation de ces ensembles de données. Chaque ensemble de données VRPTW contient des zones géographiques des clients qui sont dispersées de façon aléatoire (R), groupées (C) ou à la fois dispersées de façon aléatoire et groupées (RC). Pour chaque zone géographique, nous créons neuf ensembles de données de localisation SDDP avec 100 emplacements clients. Nous décrivons comment les neuf ensembles de données de localisation SDDP sont créés pour la géographie R. Les ensembles de données d'emplacement C et RC sont créés de manière similaire. Nous commençons par l'ensemble de données Solomon R qui contient 100 clients. Nous appelons cet ensemble de données R_1 . Comme il n'existe qu'un seul ensemble de données Solomon R unique en termes de localisation des clients, nous utilisons également les ensembles de données étendus Solomon de

Tableau 1. Sources pour les ensembles de données de localisation SDDP R

Ensemble de données SDDP	Benchmark VRPTW		
	Auteur	Base de données	No. de cust.
R_1	Salomon	R	100
R_2	G & H	R1_2	200
R_3	G & H	R1_4	400
R_4	G & H	R1_6	600
R_5	G & H	R1_8	800
R_6	G & H	R2_2	200
R_7	G & H	R2_4	400
R_8	G & H	R2_6	600
R_9	G & H	R2_8	800

Gehring et Homberger (1999). Nous créons huit ensembles de données d'emplacement SDDP à partir des ensembles de données étendus en échantillonnant 100 emplacements clients de chacun des deux ensembles de données avec 200, 400, 600 et 800 emplacements clients. Nous appelons ces ensembles de données R_2 à R_9. Table 1 résume les ensembles de données source pour les ensembles de données de localisation SDDP R. Y compris les géographies R, C et RC, cela donne un total de 27 géographies.

Nous utilisons les emplacements des clients pour déterminer les temps de trajet entre les clients. Nous veillons à ce qu'aucun client ne soit à plus d'une heure du dépôt. Pour ce faire, nous calculons d'abord la distance de Manhattan entre les emplacements. Les distances de Manhattan reflètent les distances parcourues lors de l'utilisation de la structure routière dans une ville. Pour obtenir les temps de trajet dans un ensemble de données, nous ajustons toutes les distances en fonction d'un facteur spécifique à l'ensemble de données de sorte que le client le plus éloigné du dépôt dans l'ensemble de données puisse être atteint en une heure.

Nous supposons que les heures des demandes de chaque site client suivent des processus de Poisson indépendants. Parce que nous utilisons un modèle à temps discret, les temps interarrivaux générés sont arrondis à la minute la plus proche. Une discussion détaillée des probabilités de transition est fournie dans l'annexe en ligne.

Nous générons les demandes comme suit: Premièrement, les heures d'arrivée des demandes sont générées. Les heures interarrivales sont réparties $\exp(\lambda)$, où λ est déterminé par la somme des taux d'arrivée pour tous les sites clients:

$\lambda = \sum_{j \in E} \lambda_j$. Ensuite, l'emplacement associé à une heure d'arrivée de la demande est sélectionné en tirant un nombre aléatoire u de la distribution uniforme $U[0, 1]$. Les probabilités cumulées sont calculées sur la base de la formule suivante:

$$P_{je} = \begin{cases} 0, & \text{si } i = 0, \\ \sum_{h=1}^i \lambda_h / \lambda, & \text{si } i = 1, \dots, JE, \\ 1, & \text{si } i = JE, \end{cases}$$

où $\lambda_h = \sum_{j \in E} \lambda_j$ pour $h = 1, \dots, JE$. Ainsi, l'emplacement sélectionné est je si u est telle que $P_{j-1,e} \leq u < P_{j,e}$.

Dans nos expériences, nous considérons cinq types de fenêtres temporelles. Deux types sont des délais de 1 et 2 heures, tandis que les trois autres types sont des fenêtres de temps d'une heure. Pour

demande n , nous construisons les cinq types de fenêtres temporelles comme suit:

- TW.d1: un délai de A1 heure est créé en définissant e_n et t_n à $e_n + 1$ heure.
- TW.d2: le délai A2 heures est créé en définissant e_n et t_n à $e_n + 2$ heures.
- TW.f: En fonction de l'heure de la demande, la fenêtre temporelle commence à une heure fixe dans le futur. Une fenêtre de temps d'une heure est créée en définissant e_n et t_n à $e_n + 1$ heure.
- TW.h: Basé sur l'heure de la demande et L , une distribution uniforme est utilisée pour sélectionner e_n à partir de l'une des heures restantes de la journée. Les fenêtres horaires ne commencent qu'à l'heure. Une fenêtre horaire d'une heure est créée en définissant t_n à $e_n + 1$ heure. Par exemple, si une demande arrive six heures et 51 minutes après une journée de neuf heures (avec $L = 8$ et $\Delta = 1$), il n'y a que deux options pour le début de la fenêtre de temps, soit 7 ou 8. On attribue e_n à 7 ou 8 avec une probabilité égale. Comme la fenêtre de temps ne dure qu'une heure, si 8 est sélectionné, alors $e_n = 8$ et $t_n = 9$. La construction de ces fenêtres temporelles désigne les tomates Azi, Gendreau et Potvin (2012).

- TW.r: Basé sur l'heure de la demande et L , une distribution uniforme est utilisée pour sélectionner e_n . Contrairement à TW.h, e_n n'est pas tenu d'être à l'heure. Cet ensemble de données peut être considéré comme ayant dispersé aléatoirement e_n fois. Une fenêtre horaire d'une heure est créée en définissant t_n à $e_n + 1$ heure.

Les demandes arrivent dans les huit premières heures ($L = 8$) et les véhicules doivent retourner au dépôt à la neuvième heure ($L + \Delta = 9$). Lors de la construction des fenêtres horaires, nous exigeons que toutes les demandes aient une largeur de fenêtre horaire minimale d'une heure. Cela signifie que si une requête TW.f est générée à 7,5 heures, nous décalons la fenêtre de temps de sorte que $e_n = 8$ et $t_n = 9$. Nous suivons cette procédure pour TW.f, TW.h et TW.r. Pour le cas du délai TW.d2, les demandes qui arrivent en fin de journée ont un délai d'au moins une heure.

Les ensembles de données de livraison le jour même sont nommés en fonction du type de fenêtre de temps et de l'ensemble de données de localisation. Par exemple, TW.f_R_1 fait référence au type de fenêtre temporelle TW.f et à l'ensemble de données de localisation R_1.

Pour chaque type de fenêtre géographique et temporelle, 25 flux de demandes aléatoires sont générés avec les heures et les emplacements des demandes. Lorsque nous discutons des résultats pour un ensemble de données particulier, TW.f_R_1 par exemple, nous rapportons les résultats moyens sur les 25 flux de demandes.

Les expériences qui incluent l'échantillonnage nécessitent des essais supplémentaires pour chaque type d'ensemble de données pour construire les scénarios d'échantillonnage. Nous appelons les essais utilisés dans les procédures d'échantillonnage des échantillons de scénario. Les échantillons de scénario sont générés à partir des mêmes distributions que celles utilisées pour générer les 25 flux de demandes. Pour chaque ensemble de données, les 25 demandes et exemples de flux sont affichés sur: <http://ir>

6.2. Les tests

Pour chaque ensemble de données, nous expérimentons des solutions de recherche avec et sans l'utilisation du MDT (appelées avec et sans retard). Nous expérimentons également avec et sans l'échantillonnage. Pour implémenter MDT avec un exemple de planification de scénario, lorsque la fonction de consensus sélectionne une action qui affecte un ensemble de demandes non vide à un véhicule, nous appliquons la formule MDT pour déterminer si l'attente affectera notre capacité à servir l'ensemble de clients affecté. S'il y a du temps d'attente disponible, nous faisons attendre le véhicule jusqu'à la prochaine période de décision, moment auquel nous pouvons changer l'action associée à ce véhicule. Nous appelons l'approche combinée «les deux». Pour chaque variante d'approche de solution,

Dans notre implémentation, η équivaut à une minute. Dans les expériences dans lesquelles nous combinons l'échantillonnage et le MDT, nous supposons que η est égal au maximum d'une minute et le MDT.

Notre approche de solution est implémentée en Python 2.7.5 et exécuté sur un cluster informatique à l'Université de l'Iowa en utilisant une combinaison de processeurs 2,6 GHz et 2,9 GHz exécutant CentOS 6.3. Nous n'avons pas parallélisé notre mise en œuvre.

6.3. Réglage des paramètres

Nous avons mené une série d'expériences complètes pour identifier des réglages de paramètres raisonnables. Nous nous intéressons à deux paramètres, à savoir la durée de l'horizon d'échantillonnage et le nombre d'échantillons de scénario. La durée de l'horizon d'échantillonnage indique dans quelle mesure dans le futur les demandes échantillonnées sont générées pour les échantillons du scénario. Nous avons considéré des horizons d'échantillonnage de 1/2 heure, 1 heure et 2 heures, et 10, 25 et 50 échantillons de scénario. Aux fins des expériences de paramétrage, nous avons utilisé trois véhicules et défini λ_{je}

0,002

les demandes sont permanentes pour tous les emplacements des clients et ont utilisé un jour abrégé de quatre heures. Cela correspond à un taux d'arrivée global de 0,2 requêtes par minute, ce qui signifie que le nombre attendu de requêtes était de 36. Nos expériences ont indiqué qu'un horizon d'échantillonnage d'une demi-heure avec 10 échantillons de scénario fonctionne bien sur une variété d'ensembles de données. Par conséquent, nous exécutons les expériences restantes avec ces deux paramètres.

Au cours des expériences de paramétrage, nous avons découvert une tendance notable. Nous avons remarqué que les horizons d'une heure et de deux heures sont moins performants que l'horizon d'échantillonnage plus court d'une demi-heure. Nous avons mené des expériences supplémentaires qui indiquent que des horizons d'échantillonnage plus longs nécessitent un plus grand nombre d'échantillons de scénarios. Plus d'échantillons de scénario sont nécessaires car un horizon d'échantillonnage plus long signifie que le nombre de trajets d'échantillonnage possibles augmente. Par conséquent, un plus grand nombre d'échantillons de scénario sont nécessaires pour trouver une bonne distinction

plan. La création de plans de routage pour ce plus grand nombre de plans de scénario augmente évidemment le temps de solution. Nous nous concentrons plutôt sur un horizon plus court et moins de scénarios d'échantillonnage.

6.4. Résultats

Section 6.4.1 présente des résultats détaillés à l'aide de trois véhicules. Dans les sections 6.4.2 et 6.4.3, nous décrivons les effets de l'utilisation de différents nombres de véhicules et de différents taux d'arrivée homogènes, respectivement. Dans la section 6.4.4, nous présentons les résultats en utilisant des taux d'arrivée de demandes hétérogènes.

6.4.1. Résultats détaillés - trois véhicules. Dans cette section, nous présentons les résultats de trois véhicules pour observer les modèles de base. Nous utilisons un taux d'arrivée de demande de base de

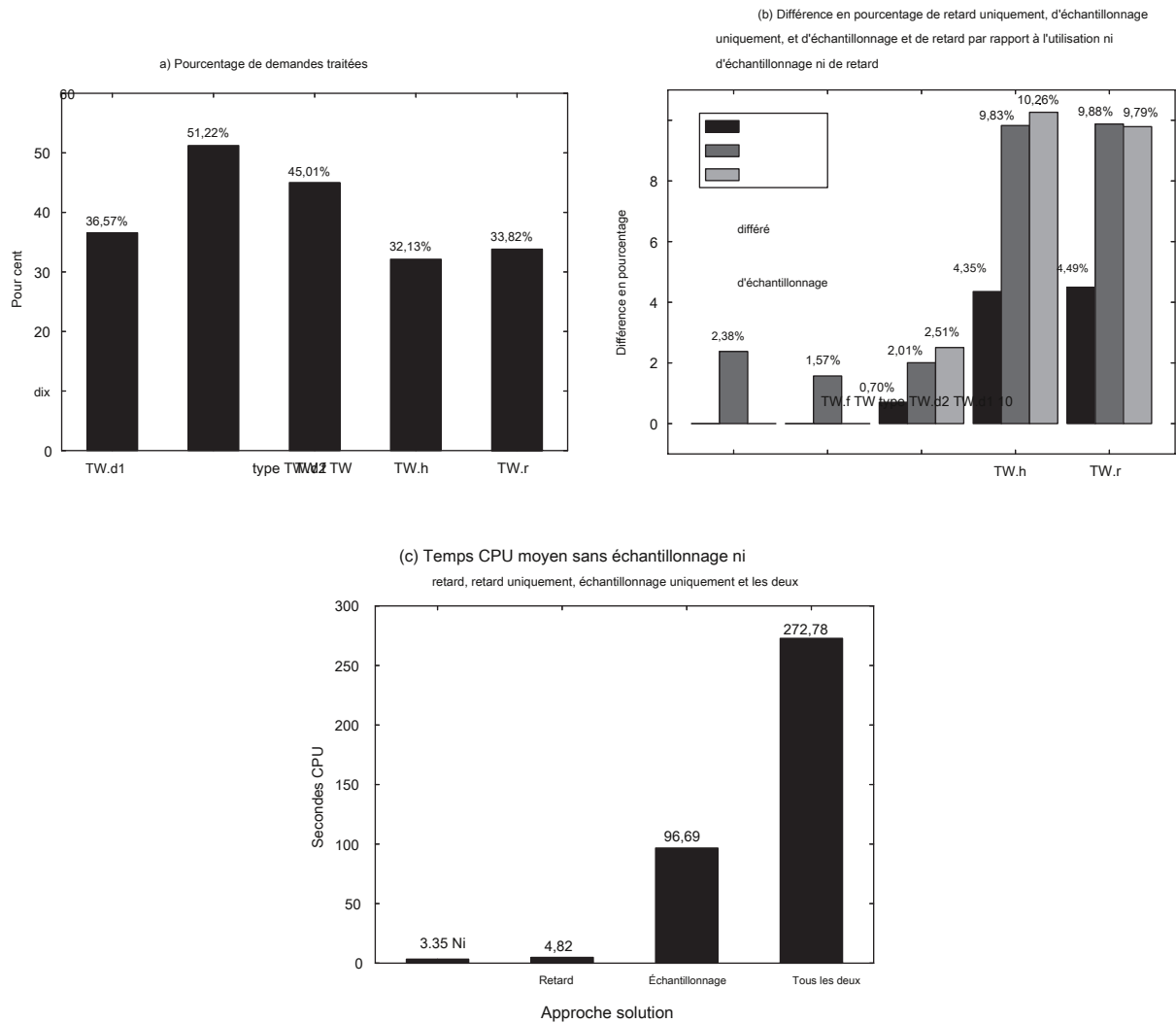
$\lambda_{je} = 0,002$ requêtes par minute pour chaque emplacement client

tion. Ce taux d'arrivée correspond à un taux d'arrivée global de $\lambda = 0,2$ et un nombre prévu de demandes de 96 pour la journée complète d'opérations. Pour les trois résultats du véhicule, nous utilisons les 27 ensembles de données de localisation.

Figure 1 fournit un résumé visuel des trois résultats du véhicule. Des tableaux détaillés sont présentés dans l'annexe en ligne. Moyenne de toutes les méthodes de test, figure 1 (a) montre que le pourcentage rempli pour les types de fenêtre temporelle TW.d1, TW.d2 et TW.f est plus grand que pour TW.h et TW.r. Le type TW.d2 a des délais de 2 heures tandis que les autres types de fenêtre de temps ont des délais de 1 heure ou des fenêtres de temps. Les ensembles de données avec les types TW.h et TW.r sont construits de telle manière que de nombreuses fenêtres temporelles se produisent vers la fin de la journée. Par conséquent, le sous-ensemble de demandes pouvant être traitées de manière satisfaisante est petit par rapport au nombre total de demandes. Par conséquent, les taux de réalisation optimaux pour TW.h et TW.r sont inférieurs à ceux des autres types. Dans ces résultats et ceux qui suivent, il ne semble pas y avoir de différence notable entre les géographies R, C et RC.

Dans tous les cas, le pourcentage rempli est plus important lorsque l'échantillonnage est utilisé par rapport à la non-utilisation de l'échantillonnage, comme l'indique la figure 1 (b). L'échantillonnage a un effet plus important sur les types de fenêtre temporelle TW.h et TW.r par rapport aux types TW.d1, TW.d2 et TW.f. L'échantillonnage fournit une amélioration moyenne de 1,99% pour les types TW.d1, TW.d2 et TW.f, tandis que pour les types TW.h et TW.r, l'échantillonnage fournit une amélioration moyenne de 9,86%. Pour les types TW.d1, TW.d2 et TW.f, les délais et les plages horaires correspondent directement à l'heure d'arrivée des demandes et sont répartis uniformément sur la journée. Pour les types de fenêtre temporelle TW.h et TW.r, de nombreuses fenêtres temporelles de demandes se produisent vers la fin de la journée. Cela signifie que de nombreuses demandes ont des fenêtres de temps qui se chevauchent, ce qui rend difficile de traiter toutes les demandes dans leurs fenêtres de temps respectives. Sans l'échantillonnage, les véhicules quittent le dépôt avec toutes les demandes qui peuvent être satisfaites pendant le trajet.

Figure 1. Résultats moyens pour les trois véhicules



cependant, autorisez les véhicules à partir lors du voyage en cours avec un nombre plus restreint de demandes. Les trajets plus courts sont choisis en raison de l'anticipation de la possibilité d'un meilleur acheminement dans la dernière partie de la journée. L'échantillonnage fournit des informations sur le moment où les véhicules doivent retourner au dépôt en fonction des arrivées de demandes attendues; cela se traduit par une augmentation du pourcentage de remplissage.

Pour voir comment l'utilisation de l'échantillonnage contribue à améliorer la qualité de la solution, nous présentons un exemple détaillé. Figure 2

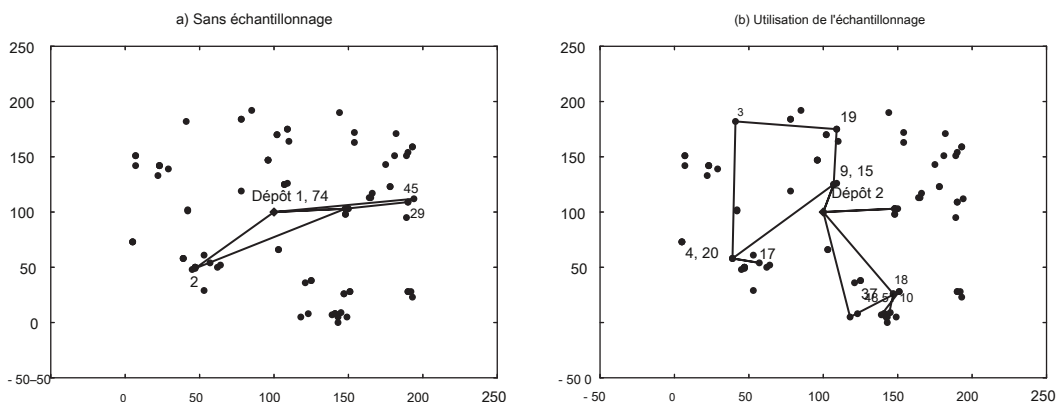
et table 2 afficher les résultats pour un seul véhicule avec et sans échantillonnage. Figure 2 affiche la géographie de l'itinéraire, tandis que le tableau 2 montre la séquence des demandes en utilisant les numéros de demande de nos ensembles de données. L'exemple est tiré de TW.r_C_3, essai 5. Au temps 6, l'algorithme utilisant l'échantillonnage part pour la demande 2. Cependant, l'algorithme sans échantillonnage part pour les demandes 1 et 2. Cette différence subtile affecte la capacité de servir les demandes futures. Le plan de routage sans échantillonnage revient au temps 416, ne laissant que 124 minutes pour répondre aux demandes supplémentaires. Le routage complet

le plan sans échantillonnage ne sert que cinq demandes, tandis que le plan d'acheminement complet avec consensus sert 13 demandes.

Les gains réalisés par le retard (MDT) en conjonction avec l'échantillonnage sont proches de zéro et parfois négatifs sur la figure 1 (b). Par conséquent, il semble que l'échantillonnage retarde bien quand il est précieux. De plus, le temps CPU moyen par époque (sur tous les types de fenêtres temporelles) est de 272,78 secondes en utilisant l'échantillonnage et le retard par rapport à 96,69 secondes pour l'échantillonnage seul, comme indiqué sur la figure 1 (c). Avec le retard, la taille des sous-problèmes augmente en raison de l'accumulation des demandes réalisées. L'augmentation de la taille du problème crée un temps d'optimisation plus long.

L'utilisation du délai sans échantillonnage pour le type de fenêtre temporelle TW.f offre une amélioration significative du pourcentage rempli par rapport à l'utilisation ni de l'échantillonnage ni du retard, mais avec un temps de calcul nettement inférieur à l'échantillonnage. Avec l'échantillonnage seul pour TW.f, le temps processeur moyen est de 52,25 secondes contre 1,40 secondes

Figure 2. Un exemple de routes avec et sans échantillonnage



avec retard seul. Pour les types TW.h et TW.r, l'utilisation de l'échantillonnage seul se traduit par une amélioration moyenne en pourcentage de 9,86% tandis que le temps CPU moyen par époque est de 183,98 secondes. Nous notons que, bien que ces temps CPU de l'approche d'échantillonnage soient beaucoup trop élevés pour la prise de décision en temps réel, notre algorithme peut être paramétré pour réduire le temps de près d'un facteur de 10. L'utilisation du retard sans échantillonnage offre un pourcentage moyen amélioration de 4,42% et un temps processeur moyen de 6,54 secondes. Par conséquent, le retard offre des améliorations raisonnables en termes de pourcentage rempli sans la charge de calcul de l'échantillonnage. Cela suggère que le retard pourrait être utilisé à la place de l'échantillonnage lorsque des temps de calcul plus rapides sont nécessaires. L'annexe en ligne fournit un exemple détaillé de la manière dont l'utilisation du délai permet d'améliorer la qualité de la solution par rapport à la non-utilisation du délai. Essentiellement, sans délai et sans échantillonnage, un véhicule peut être envoyé en début de journée pour répondre aux demandes avec des plages horaires qui surviennent vers la fin de la journée. Les fenêtres temporelles tardives forcent les véhicules à tourner au ralenti aux emplacements des clients, empêchant ainsi le véhicule de répondre à d'autres demandes. En utilisant le retard, cependant, le véhicule est autorisé à attendre à l'emplacement du dépôt jusqu'à ce que les demandes qui ont des fenêtres de temps antérieures arrivent. Par conséquent, le retard retarde les décisions d'acheminement pour les demandes qui ne peuvent être traitées que plus tard. Il en résulte que des demandes plus urgentes peuvent être traitées à leur arrivée, augmentant ainsi le pourcentage global de demandes traitées. un véhicule peut être envoyé en début de journée pour répondre aux demandes avec des plages horaires qui surviennent vers la fin de la journée. Les fenêtres temporelles tardives forcent les véhicules à tourner au ralenti aux emplacements des clients, empêchant ainsi le véhicule de répondre à d'autres demandes. En utilisant le retard, cependant, le véhicule est autorisé à attendre à l'emplacement du dépôt jusqu'à ce que les demandes qui ont des fenêtres de temps antérieures arrivent. Par conséquent, le retard retarde les décisions d'acheminement pour les demandes qui ne peuvent être traitées que plus tard. Il en résulte que des demandes plus urgentes peuvent être traitées à leur arrivée, augmentant ainsi le pourcentage global de demandes traitées. un véhicule peut être envoyé en début de journée pour répondre aux demandes avec des plages horaires qui surviennent vers la fin de la journée. Les fenêtres temporelles tardives forcent les véhicules à tourner au ralenti aux emplacements des clients, empêchant ainsi le véhicule de répondre à d'autres demandes.

6.4.2. Nombre de véhicules. Dans cette section, nous examinons comment les résultats changent en fonction du nombre de véhicules

Tableau 2. Un exemple de routes avec et sans échantillonnage

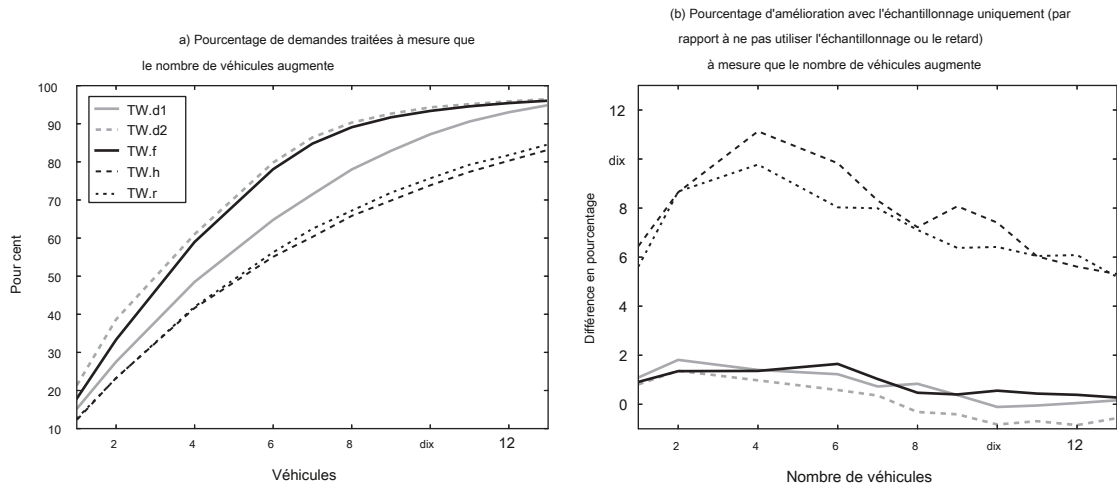
Type d'échantillonnage	Route	Nombre de demandes remplies
Sans échantillonnage	0, 2, 1, 0, 74, 29, 45, 0	5
Avec échantillonnage	0, 2, 0, 9, 4, 17, 20, 3, 19, 15, 0, 10, 18, 51, 48, 37, 0	13

augmente de un à 13. Nous utilisons les mêmes paramètres de paramètres et exécuteurs R_1 et C_1 pour tous les types de fenêtres temporelles. Le taux d'arrivée des demandes pour chaque site client, $\lambda_{i,t}$, est de nouveau réglé à 0,002 requêtes par minute. Fig- ure 3 (a) indique le pourcentage moyen de demandes satisfaites pour toutes les méthodes d'essai à mesure que le nombre de véhicules augmente. Pour atteindre un taux de satisfaction de 95%, cela montre qu'une entreprise a besoin de 11 véhicules utilisant le TW.d2 et de 12 véhicules pour les types de fenêtre de temps TW.f. Le nombre de véhicules nécessaires pour atteindre un niveau de service de 95% pour les types TW.h et TW.r est beaucoup plus important par rapport au nombre nécessaire pour les types TW.d2 et TW.f. Les résultats démontrent qu'avec 13 véhicules, un taux de remplissage légèrement inférieur à 85% peut être atteint pour les types TW.h et TW.r.

Poursuivre l'examen des courbes de réalisation de la figure 3 (a), les courbes des types de fenêtres temporelles TW.d2 et TW.f sont très similaires, tandis que les courbes des types de fenêtres temporelles TW.h et TW.r sont également assez similaires. Pour les types TW.d2 et TW.f, y compris un véhicule supplémentaire a un gros pay-o around jusqu'à environ huit véhicules. Lors de l'inclusion d'un véhicule supplémentaire au-delà de ce point, la loi des rendements décroissants prend effet. Un praticien n'inclurait pas de véhicule supplémentaire lorsque le coût de l'ajout de ce véhicule est supérieur aux retours. Pour les types de fenêtres temporelles TW.h et TW.r, la courbe apparaît plus linéaire. Cela suggère qu'il n'y a pas de rendement marginal décroissant pour chaque véhicule supplémentaire jusqu'à 13 véhicules. La prévalence de périodes de temps qui se chevauchent, ce qui rend difficile de servir les clients de manière réalisable, est la raison probable d'un retour relatif non fixe, ce qui signifie qu'un véhicule supplémentaire est inclus dans les clients, empêchant ainsi le véhicule de répondre à d'autres demandes.

Sur la base de ces résultats, un schéma de fenêtre temporelle similaire aux types de fenêtre temporelle TW.d2 et TW.f permet aux entreprises d'atteindre des taux de réalisation plus élevés avec moins de véhicules. Dans certaines situations, il peut être nécessaire d'utiliser les types de fenêtre de temps TW.h et TW.r. Dans de telles situations, une option consiste à louer des véhicules supplémentaires pour répondre à la demande de fin de journée. Les véhicules supplémentaires à la fin de la journée peuvent aider à répondre aux demandes

Figure 3. Résultats moyens à mesure que le nombre de véhicules augmente pour les ensembles de données R et C



serait autrement impossible à servir, mais sans le coût supplémentaire de la location d'un véhicule à temps plein.

Figure 3 (b) montre ce qui se passe avec l'échantillonnage lorsque le nombre de véhicules augmente avec les différents types de fenêtres temporelles. La valeur de l'échantillonnage augmente initialement à mesure que le nombre de véhicules augmente pour la plupart des types de fenêtres temporelles. À un moment donné (par exemple, quatre véhicules pour TW.h et TW.r), la valeur de l'échantillonnage commence à diminuer à mesure que le nombre de véhicules augmente. Cette tendance est observée parce qu'au départ, la capacité des véhicules est limitée, et donc l'amélioration de l'échantillonnage est limitée. Avec le bon rapport demandes / véhicules, l'échantillonnage permet de construire des itinéraires efficaces. Cependant, à mesure que de plus en plus de véhicules sont ajoutés mais que le nombre de demandes reste fixe, le besoin d'itinéraires efficaces diminue car il y a plus de souplesse dans la manière dont ces demandes sont traitées. Donc,

6.4.3. Taux d'arrivée homogènes. Nous considérons également l'effet de l'échantillonnage lorsque des taux d'arrivée homogènes différents sont utilisés. Nous avons utilisé les ensembles de données de localisation R et C pour ces expériences et créé de nouveaux flux de demandes pour chaque taux d'arrivée.

Nous fixons λ_j à 0,001,

0,002, 0,003 ou 0,004 requêtes par minute pour chaque emplacement client dans l'ensemble de données. Ces taux d'arrivée correspondent au nombre global attendu de 48 demandes,

96, 144 et 192, respectivement. Table 3 décrit ces taux d'arrivée. Nous utilisons les réglages des paramètres précédents avec trois véhicules.

Figure 4 affiche les résultats des expériences de taux d'arrivée homogène avec échantillonnage uniquement. Figure 4 (a) indique le pourcentage de demandes traitées en utilisant uniquement l'échantillonnage. À mesure que le taux d'arrivée augmente, le pourcentage de demandes traitées diminue. Cela se produit car le nombre de demandes augmente alors que le nombre de véhicules reste fixe. Par conséquent, la proportion de demandes

qui peuvent être desservis avec la capacité donnée, car le taux d'arrivée augmente a tendance à diminuer.

Figure 4 (b) affiche les pourcentages de différences en utilisant uniquement l'échantillonnage par rapport à l'utilisation ni d'échantillonnage ni de retard. Il existe un modèle distinct qui apparaît pour les types de fenêtre de temps TW.d, TW.f et TW.r. Au départ, le pourcentage de différence augmente puis commence à diminuer à mesure que le taux d'arrivée augmente. Avec le

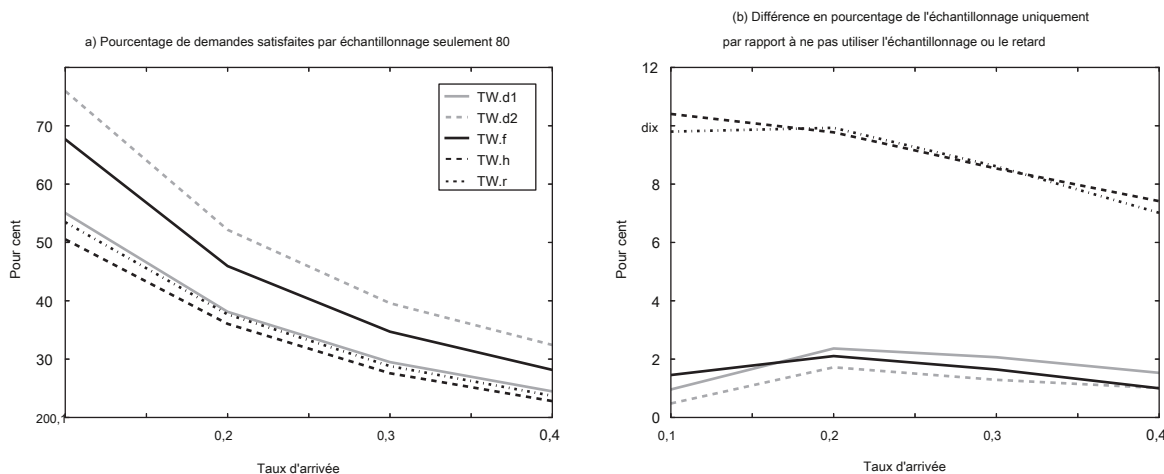
0,1 taux d'arrivée global, l'échantillonnage n'améliore les solutions que de peu car il y a peu de demandes par rapport au nombre de véhicules disponibles. Avec un taux d'arrivée global élevé de 0,4, le pourcentage d'amélioration par rapport à l'échantillonnage est également faible, mais le faible impact est dû au grand nombre de demandes par rapport aux véhicules. Lorsque beaucoup plus de demandes sont disponibles, la façon dont le routage peut se produire est plus flexible; cela diminue la valeur de l'inclusion de l'échantillonnage dans l'approche de la solution. Ces résultats corroborent les résultats de la section 6.4.2 où nous avons discuté des expériences pour une gamme de tailles de flottes. Le type de fenêtre de temps TW.h montre un modèle de pourcentage de différences décroissantes à mesure que le taux d'arrivée augmente. Ce schéma différent se produit probablement en raison de la structure des fenêtres temporelles. Le type TW.h est le seul type de fenêtre horaire avec des heures de début de fenêtre horaire qui se produisent à l'heure.

6.4.4. Taux d'arrivée hétérogènes. Nous présentons maintenant les résultats d'expériences utilisant une arrivée hétérogène

Tableau 3. Taux d'arrivée des demandes et nombre total prévu de

les taux (λ_j) (Demandes par min.)	Arrivée du lieu	
	Arrivée générale taux (λ) (Demandes par min.)	Attendu nombre total des demandes
0,001	0,1	48
0,002	0,2	96
0,003	0,3	144
0,004	0,4	192

Figure 4. Résultats moyens pour des taux d'arrivée homogènes avec échantillonnage uniquement



paramètres de taux. Nous choisissons des paramètres de taux d'arrivée hétérogènes tels que le taux d'arrivée global, λ , est égal à 0,2 requêtes par minute. Ce taux d'arrivée global est cohérent avec les expériences précédentes, à l'exception de celles présentées dans la section 6.4.3. Pour toutes les expériences hétérogènes, nous attribuons à 50 emplacements clients des taux d'arrivée de

λ_{j0} 0,001 requêtes par minute (faible) et 50 emplacements d'avoir des taux d'arrivée de λ_{j0} 0,003 requêtes par minute (haute). Nous utilisons les ensembles de données R et C.

Nous attribuons les taux d'arrivée aux emplacements de quatre façons. Nous appelons chacun de ces moyens des types hétérogènes 1, 2, 3 et 4. Pour le type hétérogène 1, nous attribuons au hasard les taux d'arrivée élevés et faibles aux emplacements des clients. Dans les expériences restantes, nous créons deux clusters de 50 emplacements clients chacun. Des exemples de grappes de types 2, 3 et 4 sont présentés dans la figure 5. Dans les panneaux (a) - (d), le dépôt est représenté par un diamant près du centre de l'espace. Pour les types hétérogènes 2 et

3, un cluster est formé avec les emplacements les plus proches du dépôt et l'autre cluster est formé avec les emplacements les plus éloignés du dépôt. Pour le type 2, nous attribuons les taux d'arrivée faibles aux emplacements proches du dépôt et les taux d'arrivée élevés aux emplacements éloignés du dépôt. Pour le type 3, nous inversons les taux d'arrivée afin que les emplacements proches du dépôt aient des taux d'arrivée élevés et que les emplacements éloignés du dépôt aient des taux d'arrivée bas. Pour le type hétérogène de type 4, nous divisons les clients en deux groupes en utilisant un algorithme de balayage (voir Gillett et Miller

1974). Par conséquent, un cluster est approximativement situé au-dessus de la y -coordonnées du dépôt, et l'autre groupe est situé approximativement en dessous de la y -coordonnées du dépôt. Nous attribuons des taux d'arrivée élevés aux emplacements du cluster supérieur et des taux d'arrivée faibles aux emplacements du cluster inférieur. Pour plus de commodité, nous résumons les types hétérogènes dans le tableau 4 et inclure le cas homogène pour comparaison.

Nous présentons un résumé des résultats dans la figure 6. Figure 6 (a) indique le pourcentage de demandes remplies

échantillonnage uniquement. Pour le cas homogène, le pourcentage de remplissage est de 41,99%. Par comparaison, le pourcentage moyen rempli pour les types hétérogènes 1 à 4 est de 41,21%, 36,66%, 46,51% et 42,31%, respectivement. Les résultats pour les types hétérogènes 1 et 4 ne diffèrent pas de manière significative des expériences homogènes. Cependant, le type hétérogène de type 2 a un pourcentage plus faible de demandes remplies, tandis que le type 3 a un pourcentage plus élevé de demandes remplies. Cela est dû au fait que le type 2 a des taux d'arrivée élevés pour les emplacements éloignés du dépôt tandis que le type 3 a des taux d'arrivée élevés pour les emplacements près du dépôt. Plus de demandes peuvent être traitées lorsque les emplacements des demandes sont proches du dépôt, car moins de temps de déplacement est nécessaire.

Nous considérons également le pourcentage moyen de différence entre l'utilisation de l'échantillonnage seul par rapport à la non-utilisation de l'échantillonnage ou du délai. Les résultats sont présentés sur la figure 6 (b). Pour les expériences homogènes, la différence moyenne est de 5,18%. Pour les types hétérogènes 1 à 4, les améliorations moyennes sont de 5,29%, 4,78%, 5,63% et 5,17%, respectivement. Nous voyons que le type hétérogène 2 présente le moins d'amélioration par rapport à l'échantillonnage tandis que le type hétérogène 3 présente le plus d'amélioration par rapport à l'échantillonnage. Ces résultats indiquent que la localisation des clients à forte probabilité peut jouer un rôle important dans l'impact de l'échantillonnage sur les solutions.

Tableau 4. Résumé des types de taux d'arrivée homogènes et hétérogènes

Type	La description
Homogène	Tous les tarifs sont égaux pour chaque emplacement
Type hétérogène 1	Aléatoire
Hétérogène type 2	Taux d'arrivée élevés pour emplacements loin du dépôt
Type hétérogène 3	Taux d'arrivée élevés pour emplacements à proximité du dépôt
Hétérogène type 4	Taux d'arrivée élevés «Au-dessus» du dépôt

Graphique 5. Clusters pour les types hétérogènes 2 à 4

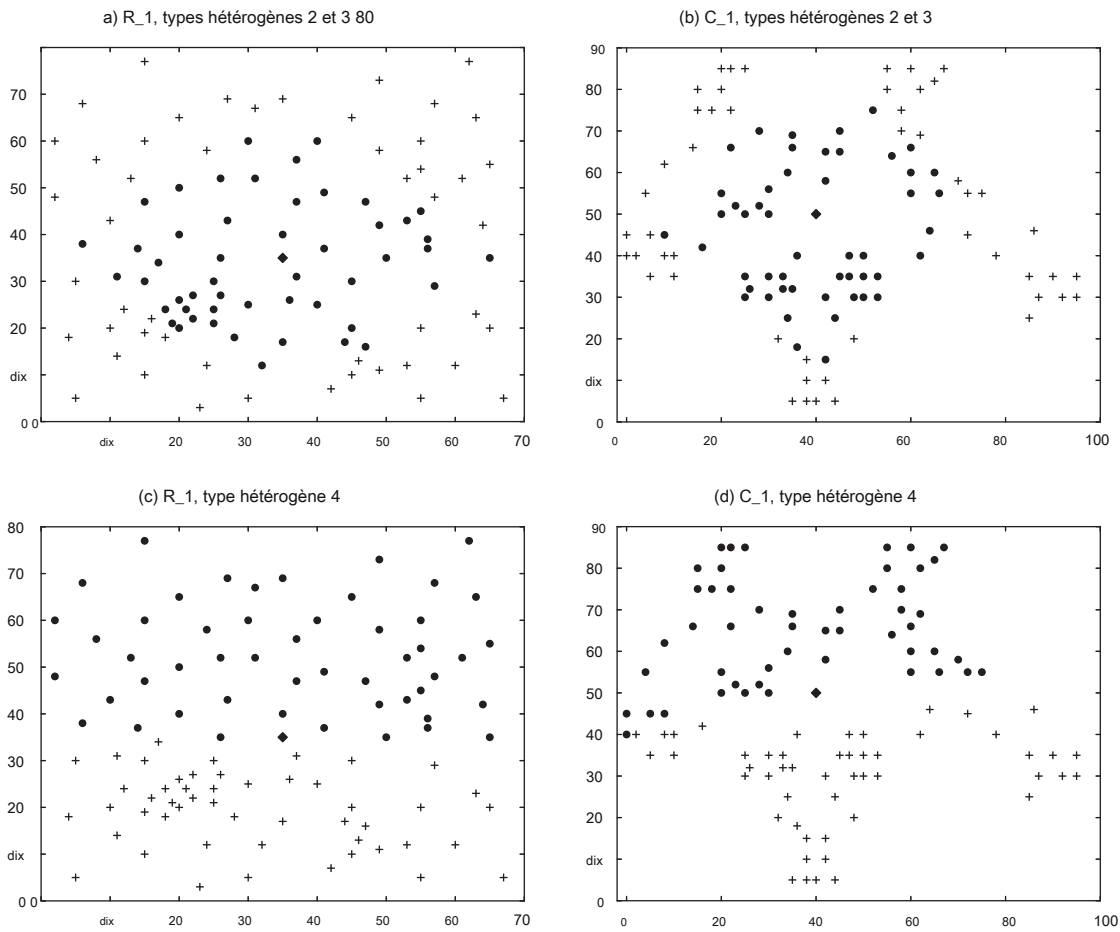
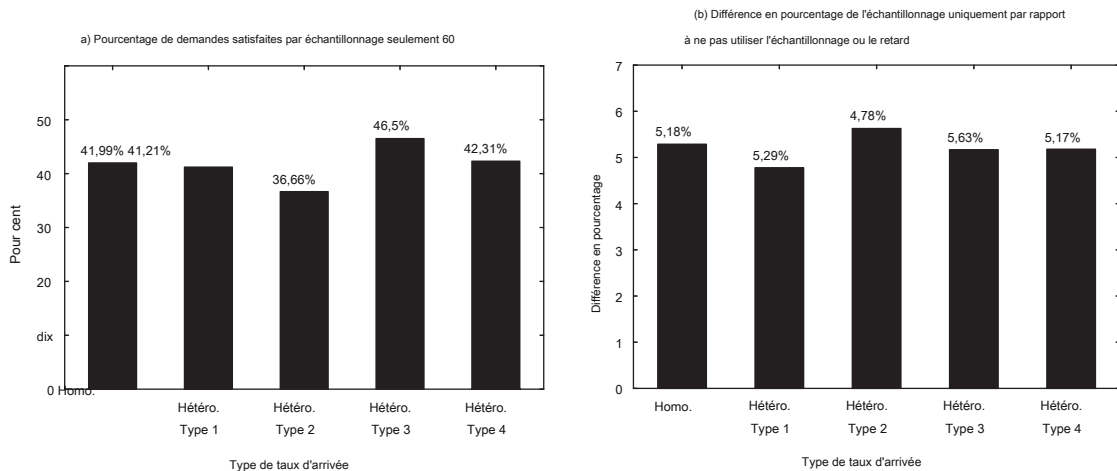


Figure 6. Résultats moyens pour les taux d'arrivée homogènes et hétérogènes avec échantillonnage uniquement



7. Conclusions

Dans cet article, nous présentons le SDDP, un problème dynamique de collecte et de livraison motivé par les pratiques commerciales actuelles. Nous présentons un modèle MDP formel pour le SDDP. A chaque époque de décision, un sous-problème du SDDP est résolu. Ce sous-problème est appelé

MTTOPTW et, à notre connaissance, est également un nouveau problème dans la littérature. Nous incorporons deux façons de prendre des décisions mieux informées aux époques de décision. Une règle implicite courante dans la littérature est de retarder les départs de véhicules lorsque cela est possible. En retardant les départs de véhicules, les décisions sont reportées jusqu'à ce que plus d'informations

sur le problème est réalisé. Nous introduisons le calcul MDT, qui détermine combien de temps un véhicule peut attendre au dépôt sans diminuer la capacité de répondre aux demandes actuelles et futures. Nous utilisons une méthode de planification basée sur un exemple de scénario comme un autre moyen de prendre des décisions mieux informées. Dans la méthode de planification des scénarios d'échantillonnage, différents scénarios sont construits sur la base d'échantillons aléatoires des arrivées de demandes des clients. Sur la base de ces scénarios, les décisions de routage sont prises. Nous introduisons une fonction de consensus spécialement conçue pour les besoins uniques du SDDP qui tire parti des informations échantillonnées pour guider la construction de l'itinéraire des véhicules. Notre fonction de consensus s'identifie lors de l'attente au dépôt en prévision des demandes futures est bénéfique, ainsi que la sélection des demandes qui doivent être affectées à des véhicules spécifiques à traiter immédiatement tout en retardant les décisions concernant d'autres demandes. Cela permet aux véhicules de répondre à plus de demandes totales.

Nous menons des expériences pour diverses géographies de localisation de clients, contraintes de temps, nombre de véhicules et taux d'arrivée des demandes. Nos résultats offrent plusieurs perspectives. Dans l'ensemble, davantage de demandes peuvent être remplies lorsque des fenêtres horaires se produisent près de l'heure à laquelle les demandes sont effectuées et sont réparties uniformément tout au long de la journée (types de fenêtres horaires TW.d2 et TW.f) par rapport à lorsque de nombreuses fenêtres horaires se produisent tard dans la journée (TW types de fenêtre de temps .h et TW.r). Cependant, l'échantillonnage et le délai MDT sont plus précieux lorsque de nombreuses fenêtres temporelles se produisent tard dans la journée. À partir de nos trois expériences de référence de véhicule pour les types de fenêtre temporelle TW.h et TW.r, l'échantillonnage sans délai MDT offre une augmentation moyenne des demandes remplies de

9,85%, tout en utilisant le délai MDT sans échantillonnage offre une augmentation moyenne de 4,42%. Bien que la qualité des solutions avec le délai MDT ne soit pas aussi élevée par rapport à l'échantillonnage, le délai MDT peut être utilisé lorsque des temps de calcul plus rapides sont nécessaires. L'utilisation de l'échantillonnage conjointement avec le délai MDT n'entraîne pas de solutions meilleures que l'utilisation de l'échantillonnage seul, mais elle nécessite des temps de calcul considérablement plus longs. Les résultats suggèrent que l'échantillonnage reporte les décisions d'acheminement lorsqu'il est bénéfique de le faire.

Des expériences supplémentaires démontrent que la valeur de l'échantillonnage et le délai MDT diminuent lorsque le nombre de véhicules ou le taux d'arrivée des demandes augmente. Ceci est le résultat d'une flexibilité accrue. Lorsque le nombre de véhicules est important, il y a plus de capacité, et donc plus de flexibilité dans le choix de la manière de traiter les demandes. Lorsque le taux d'arrivée des demandes est élevé, il y a plus de flexibilité dans le choix des demandes à servir, sur la base d'une capacité limitée. Ces augmentations de flexibilité réduisent la valeur des informations futures. Des expériences hétérogènes démontrent que la localisation de

les clients à forte probabilité peuvent jouer un rôle important dans l'impact de l'échantillonnage sur les solutions.

Il existe plusieurs possibilités de travaux futurs. Premièrement, ce travail considère un seul mode de livraison. Plusieurs entreprises envisagent des expériences qui mélangent des fourgonnettes, des voitures, des coursiers à vélo et des conducteurs tiers de type Uber. Les travaux futurs pourraient explicitement considérer chacun de ces modes et leurs coûts dans la décision de quand et par quel mode envoyer une demande du dépôt. De plus, ce travail suppose que toutes les demandes sont acceptées et servies par la flotte donnée ou un tiers. Les travaux futurs pourraient envisager la possibilité que les demandes puissent être acceptées ou rejetées à l'arrivée. Les travaux futurs pourraient également envisager des systèmes d'incitation pour déplacer les délais de demande ou les fenêtres temporelles vers des moments où il existe une capacité excédentaire disponible pour répondre à la demande, comme dans Campbell et Savelsbergh (2006).

Remerciements

Les auteurs sont reconnaissants aux commentaires utiles de deux arbitres anonymes et de l'éditeur associé.

Références

- Albareda-Sambola M, Fernández E, Laporte G (2014) La dynamique problème de routage de véhicules sur plusieurs périodes avec des informations probabilistes. *Comput. Oper. Res.* 48: 31–39. Amazon (2017) Prime Now. <https://primenow.amazon.com>
- Angelilli E, Bianchessi N, Mansini R, Speranza MG (2009) Court terme stratégies pour un problème de routage dynamique multi-période. *Transport Res. Partie C: Technologies émergentes.* 35 (6): 713–721. Arslan A, Agatz N, Kroon LG, Zuidwijk RA (2016) Crowdsourced livraison: Un problème dynamique de ramassage et de livraison avec des pilotes ad-hoc. *Rapport technique, ERIM Report Series Reference.* <http://srm.com/abstract/2726731>
- Azi N, Gendreau M, Potvin JY (2012) Un itinéraire dynamique des véhicules problème avec plusieurs itinéraires de livraison. *Ann. Oper. Res.* 199 (1): 103–112.
- Bent RW, Van Hentenryck P (2004) Planification basée sur des scénarios pour routage de véhicules essentiellement dynamique avec des clients stochastiques. *Oper. Res.* 52 (6): 977–987.
- Bent RW, Van Hentenryck P (2007) Stratégies d'attente et de réinstallation dans le routage stochastique de véhicules en ligne. Veloso M, éd. *Proc. 20 e Inter-nat. Conf. Conf. Intelligence artificielle (AAAI Press, Menlo Park, CA),* 1816–1821.
- Campbell AM, Savelsbergh M (2005) Aide à la décision pour le consommateur initiatives d'épicerie directes. *Transport Sci.* 39 (3): 313–327. Campbell AM, Savelsbergh M (2006) Programmes d'incitation pour les services de livraison à domicile. *Transport Sci.* 40 (3): 327–341. Cliford S, Miller CC (2012) Instantanément à vous, moyennant des frais. *New York Times* (28 décembre), B1. <http://www.nytimes.com/2012/12/28/business/fear-of-amazon-pushes-retailers-to-take-on-risk-of-same-day-shipping.html>
- Cordeau JF, Laporte G (2007) Le problème de l'accès à distance: modèles et algorithmes. *Ann. Oper. Res.* 153 (1): 29–46. Ehmke JF, Campbell AM (2014) Mécanismes d'acceptation des clients pour les livraisons à domicile dans les zones métropolitaines. *EUR. J. Oper. Res.* 233 (1): 193–207.
- Gehring H, Homberger J (1999) Une évolution hybride parallèle métaheuristique pour le problème de routage des véhicules avec les fenêtres de temps. *Proc. EUROGEN 1999- Short Course Evolutionary Algorithms Engrg. Comput. Sci.*, Rapports Département Math. Informer. Tech., Université de Jyväskylä, Vol. A 2/1999, 57–64. Ghiani G, Manni E, Thomas BW (2012) Une comparaison d'anticipation algorithmes pour le problème du vendeur voyageur dynamique et stochastique. *Transport Sci.* 46 (3): 374–387.

- Ghiani G, Manni E, Quaranta A, Triki C (2009) Algo anticipation rithms pour l'envoi de courrier le jour même. *Transport Res. Partie E: Logist. Transport Rev.* 45 (1): 96-106.
- Gillett BE, Miller LR (1974) Un algorithme heuristique pour le véhicule-problème de répartition. *Oper. Res.* 22 (2): 340–349.
- Howard R (2014) Livraison le jour même: liste de contrôle pour les détaillants à la recherche d'un antidote à Amazon: Première partie. <http://www.parcelindustry.com/article-4003-same-day-delivery-a-checklist-for-retailers-cherche-un-antidote-a-amazon-part-one.html>.
- HvattumLM, Løkketangen A, Laporte G (2007) Une branche et un regret heuristique pour les problèmes de routage de véhicules stochastiques et dynamiques. *Réseaux* 49 (4): 330-340.
- Instacart (2016) Foire aux questions. <https://www.instacart.com/faq>.
- Kiechle G, Doerner KF, Gendreau M, Hartl RF (2009) En attente stratégies de transport régulier et d'urgence des patients. Fleischmann B, Borgwardt KH, Klein R, Tuma A, éd. *Oper. Res. Proc. 2008 (Springer-Verlag, Berlin Heidelberg)*, 271–276.
- Klapp M, Erera A, Toriello A (2018) La dynamique unidimensionnelle problème de vagues de répartition. *Transport Sci.* 52 (2): 402–415. Pillac V, Gendreau M, Guéret C, Medaglia AL (2013) des problèmes de routage dynamique des véhicules. *EUR. J. Oper. Res.* 225 (1): 1–11. PowellW (2011) *Programmation dynamique approximative: résoudre les malédictions de dimensionnalité*, 2e éd. (John Wiley & Sons, Hoboken, NJ). Pureza V, Laporte G (2008) Stratégies d'attente et d'achat pour le problème de ramassage et de livraison dynamique avec les fenêtres de temps. *INFOR: Informer. Systèmes op. Res.* 46 (3): 165-176. Sáez D, Cortés CE, Núñez A (2008) Contribution prédictive adaptative hybride trol pour le problème de ramassage et de livraison dynamique multi-véhicules basé sur des algorithmes génétiques et un clustering flou. *Comput. Oper. Res.* 35 (11): 3412–3438.
- Sheridan PK, Gluck E, Guan Q, Pickles T, Balcioglu B, Benhabib B (2013) La politique dynamique du plus proche voisin pour le problème du ramassage et de la livraison de plusieurs véhicules. *Transport Res. Partie A: Pratique politique* 49: 178-194.
- Solomon MM (1987) Algorithmes pour le routage des véhicules et problèmes de planification avec des contraintes de fenêtre temporelle. *Oper. Res.* 35 (2): 254-265.
- Srouf FJ, Agatz N, Oppen J (2018) Stratégies de gestion temporelle incertitude des problèmes de ramassage et de livraison avec les fenêtres horaires. *Transport Sci.* 52 (1): 3–19. Thomas BW (2007) Stratégies d'attente pour anticiper le service les demandes des emplacements clients connus. *Transport Sci.* 41 (3): 319–331.
- Thomas BW (2011) *Routage dynamique des véhicules*. Cochran J, éd. *Wiley Encyclopédie de la recherche opérationnelle et des sciences de la gestion* (Wiley, Hoboken, NJ).
- Ulmer MW, Mattfeld DC, Köster F (2018) Temps de budgétisation pour routage dynamique des véhicules avec demandes stochastiques des clients. *Transport Sci.* 52 (1): 20–37.
- Ulmer MW, Mattfeld DC, Soeffker N (2016) Dynamic multi routage de véhicule de période: itération de valeur approximative basée sur des tables de recherche dynamiques. Document de travail, TU Braunschweig, Braunschweig, Allemagne, http://web.winforms.phil.tu-bs.de/paper/ulmer/WP_Ulmer_Dynamic_Lookup.pdf.
- Ulmer MW, Thomas BW, Mattfeld DC (2016) Dépôt préemptif revient pour un problème de livraison dynamique le jour même. Document de travail, TU Braunschweig, Braunschweig, Allemagne, https://www.researchgate.net/publication/305433476_Preemptive_Depot_Returns_for_a_Dynamic_Same-Day_Delivery_Problem.
- Ulmer MW, Goodson JC, Mattfeld DC, Hennig M (2019) O ffl in-programmation dynamique approximative en ligne pour le routage dynamique de véhicules avec des requêtes stochastiques. *Transport Sci.* 53 (1): 185-202.
- Ulmer MW, Mattfeld DC, Hennig M, Goodson JC (2015) Un déploiement algorithme de routage des véhicules avec les demandes stochastiques des clients. Mattfeld D, Spengler T, Brinkmann J, Grunewald M, éd. *Gestion de la logistique*, Journal Notes Logist. (Springer, Cham, Suisse), 217–227.
- Vansteenwegen P, Souffriau W, Oudheusden DV (2011) L'orientation problème de teering: une enquête. *EUR. J. Oper. Res.* 209 (1): 1–10. Voccia SA, Campbell AM, Thomas BW (2013) Le voyage probabiliste problème de vendeur avec les fenêtres temporelles. *EURO J. Logistique des transports.* 2 (1–2): 89–107.
- Wen M, Cordeau JF, Laporte G, Larsen J (2010) La dynamique problème de routage de véhicule sur plusieurs périodes. *Comput. Oper. Res.* 37 (9): 1615-1623.

Copyright 2019, par INFORMS, tous droits réservés. Le droit d'auteur de Transportation Science est la propriété d'INFORMS: Institute for Operations Research et son contenu ne peut pas être copié ou envoyé par courrier électronique à plusieurs sites ou publié sur une liste de diffusion sans l'autorisation écrite expresse du titulaire du droit d'auteur. Cependant, les utilisateurs peuvent imprimer, télécharger ou envoyer par courrier électronique des articles pour un usage individuel.