
Auto-organisation spatio-temporelle pour le VRPTW dynamique

Besma Zeddini^{*,**} — Mahdi Zargayouna^{**}

* IRSEEM-ESIGELEC,
Saint Etienne du Rouvray 76801, France
besma.zeddini@ifsttar.fr

** Université Paris-Est, IFSTTAR, GRETTIA,
2 rue de la Butte Verte,
93166 Noisy Le Grand, France,
hamza-mahdi.zargayouna@ifsttar.fr

RÉSUMÉ. Les problèmes de tournées de véhicules sont des problèmes de grande complexité pour lesquels différentes techniques d'intelligence artificielle ont été utilisées. Dans ce papier, nous proposons un modèle d'auto-organisation orienté-agent pour la version dynamique du problème de tournées de véhicules avec fenêtres temporelles. Notre proposition est fondée sur une représentation spatio-temporelle des zones d'action des agents, capable de maintenir une bonne répartition des véhicules sur l'environnement. Cette répartition répond à l'objectif du problème dynamique, puisqu'elle permet aux agents de prendre leurs décisions tout en anticipant les éventuels changements futurs des paramètres du système.

ABSTRACT. Vehicle Routing problems are highly complex problems for which different Artificial Intelligence techniques have been used. In this paper, we propose an agent-oriented self-organization model for the dynamic version of the problem with time windows. Our proposal is based on a space-time representation of the agents' action zones, which is able to maintain a good distribution of the vehicles on the environment. This distribution answers the objective of the dynamic problem, since it allows the agents to take their decisions while anticipating future changes in the system's parameters.

MOTS-CLÉS: auto-organisation, systèmes multi-agents, problèmes de tournées de véhicules.

KEYWORDS: Self-organization, Multi-agent systems, Vehicle routing problems.

1. Introduction

De multiples problèmes opérationnels de distribution, tels que la livraison de marchandises à des magasins, la livraison d'argent aux banques et distributeurs automatiques, l'ordonnancement de bus scolaires, la distribution de courriers ou de journaux, etc. sont des instanciations de problèmes théoriques NP-difficiles, extensions du problème de tournées de véhicules (VRP¹). Dans sa version originale, le VRP est un problème de voyageur de commerce multivéhicule : il existe un certain nombre de nœuds à visiter une et une seule fois par un nombre limité de véhicules (Dantzig *et al.*, 1959). L'objectif est de trouver un ensemble de tournées des véhicules qui minimise la distance totale parcourue. Au-delà de leur intérêt pratique, le VRP et ses extensions sont des problèmes d'optimisation stimulants et d'un intérêt académique significatif. La variante qui a reçu le plus d'attention dans la littérature, et qui modélise le plus fidèlement les problèmes opérationnels, est le VRP avec contraintes de capacité et fenêtres temporelles (VRPTW²) (Solomon, 1987), dans lequel les demandes à traiter ne sont plus simplement des nœuds, mais des clients. Pour chaque client, les informations suivantes sont prises en compte : le nœud concerné par la demande, deux bornes temporelles entre lesquelles il doit être visité et une quantité (nombre de produits, de personnes, etc.) à transporter. Chaque véhicule a une capacité limitée, qui ne doit pas être dépassée par les quantités associées aux clients qu'il visite. L'ajout de fenêtres temporelles restreint considérablement l'espace de solutions valides.

Le VRP et ses extensions sont divisés en deux catégories (Larsen, 2000) : les problèmes statiques et les problèmes dynamiques. La distinction entre ces deux catégories repose traditionnellement sur la connaissance (problème statique) ou la non connaissance (problème dynamique) avant le début de la résolution de tous les clients qui doivent être visités. Les problèmes opérationnels de tournées de véhicules sont rarement statiques, et nous pouvons dire qu'aujourd'hui, un système statique n'a pas de chance de satisfaire les besoins en mobilité des utilisateurs. En effet, dans un contexte opérationnel, et même si l'ensemble des clients est connu avant l'exécution, il existe toujours quelque élément qui rend le problème dynamique. Ceci englobe les pannes, les retards, les voyageurs ne se présentant pas, etc. Il est ainsi toujours utile de considérer un problème qui n'est pas totalement statique.

Nous nous fondons sur le paradigme multi-agent pour résoudre le VRPTW dynamique. Les systèmes multi-agents (SMA) sont un paradigme ayant pour objectif de canaliser l'intelligence artificielle distribuée dans le cadre d'une approche conceptuelle simple dans l'appréhension de problèmes complexes (Adams, 2001). Un agent est un système informatique qui est situé dans un environnement et qui est capable d'appliquer des actions autonomes dans le but de satisfaire ses buts (Wooldridge *et al.*, 1995), et un SMA est un réseau d'agents faiblement couplés qui interagissent afin de résoudre des problèmes qui dépassent les capacités ou les connaissances de chacun (Sycara, 1998). Le VRPTW dynamique se prête bien à une modélisation multi-

1. *Vehicle Routing Problem*

2. *VRP with Time Windows*

agent pour les raisons suivantes. D'abord, le VRPTW est un problème difficile, pour lequel les méthodes exactes ne traitent que des versions de petite taille, qui ne peuvent représenter des applications réelles. Le choix d'une modélisation permettant une distribution des traitements peut être une solution afin de proposer des temps de réponses courts face à des demandes clients. Ensuite, il s'agit d'un problème qui suppose une gestion dynamique des données. Par définition, toutes les données ne sont pas disponibles avant le démarrage de l'exécution et de plus, le système doit pouvoir s'adapter à la réalité du réseau de transport : perturbation, panne, etc. les SMA sont connus pour être bien adaptés à ces contextes dynamiques (Ferber, 1995). Par ailleurs, avec le développement technologique, il est raisonnable de considérer des véhicules avec des capacités calculatoires embarquées. Dans ce contexte, le problème est, de fait, distribué et nécessite une modélisation adaptée afin de tirer profit des capacités embarquées des véhicules. Enfin, la considération d'un point de vue multi-agent permet d'imaginer de nouvelles mesures, de nouvelles heuristiques, non envisagées par les approches centralisées. Même si l'approche multi-agent ne garantit pas de solution optimale, elle est souvent capable de trouver des solutions satisfaisantes en des temps d'exécution raisonnables (Wooldridge *et al.*, 1995).

Plusieurs approches multi-agents ont été adoptées dans la littérature proposant de distribuer des « heuristiques d'insertion ». Les heuristiques d'insertion consistent à insérer les clients selon leur ordre d'arrivée. Étant donnée une flotte de véhicules, chaque nouveau client est inséré dans la tournée du véhicule ayant à effectuer le détour minimal pour visiter le nouveau client. La décision d'allouer tel client à tel véhicule est irrévocable. Cette non remise en cause de l'insertion des clients fait que ces heuristiques présentent les temps d'exécution les plus rapides, et a fait qu'elles ont été parmi les premières heuristiques qui ont été proposées pour résoudre les problèmes de tournées (Solomon, 1987). Si les approches multi-agents de la littérature proposent différentes étapes additionnelles pour améliorer le rendement des heuristiques d'insertion, rares sont celles qui proposent des mesures du coût d'insertion d'un client dans la tournée d'un véhicule, autre que le détour effectué par le véhicule. Dans ce papier, nous proposons une nouvelle mesure de ce coût d'insertion, fondée sur une représentation spatio-temporelle des zones d'action des agents véhicules. L'objectif est de permettre au SMA de s'auto-organiser en affichant une répartition spatio-temporelle équilibrée de ses agents véhicules. Cette répartition équilibrée des véhicules dans le temps et dans l'espace permet de diminuer le nombre de véhicules mobilisés pour la desserte des clients. En effet, en affichant une répartition équilibrée, le SMA est plus réactif aux demandes des clients, qui surviennent d'une manière non déterministe dans le temps et dans l'espace.

La suite du papier est structurée comme suit. Dans la section 2, nous discutons des analogies et des différences entre notre approche et celles de la littérature et nous présentons formellement le problème VRPTW. Dans la section 3, nous présentons l'architecture retenue du SMA que nous proposons, résultant d'une comparaison entre trois architectures possibles. Dans la section 4, nous détaillons la représentation spatio-temporelle des zones d'action des agents véhicules et son utilisation comme mesure

pour la décision d'insertion des *Client*. Nous présentons les résultats expérimentaux de notre proposition au niveau de la section 5 avant de conclure.

2. État de l'art

2.1. Optimisation combinatoire

Les approches exactes de résolution du problème VRPTW ne peuvent traiter des configurations opérationnelles. En effet, sur l'ensemble des problèmes *benchmark* de Solomon (1987) – relativement petit, 56 problèmes de 100 clients euclidiens³ chacun –, seuls 45 disposent d'une solution optimale connue à ce jour (Nagata *et al.*, 2010). Cependant, les lecteurs intéressés par les approches d'optimisation peuvent se référer à, e.g. (Jepsen *et al.*, 2008) pour un état de l'art. En effet, une grande quantité de solutions proposées dans la littérature concernent des heuristiques, qui donnent de bons résultats en des temps raisonnables, et qui ont présenté de très bons résultats avec les problèmes *benchmark*. Par exemple, la recherche locale à voisinage large (Bent *et al.*, 2004a; Pisinger *et al.*, 2007), la recherche locale itérative (Ibaraki *et al.*, 2005; Ibaraki *et al.*, 2008), la recherche locale multidépart (Lim *et al.*, 2007), le recuit simulé (Czech *et al.*, 2002), les stratégies évolutives (Mester *et al.*, 2005; Homberger *et al.*, 2005), et les colonies de fourmis (Gambardella *et al.*, 1999). Ces approches donnent les meilleures performances avec les problèmes statiques (où l'ensemble des demandes est connu au départ). Pour une revue extensive des approches de la littérature pour le VRPTW, le lecteur intéressé est prié de se référer à, e.g. (Golden *et al.*, 2008; Desaulniers *et al.*, 2002).

Généralement, la plupart des travaux traitant du VRPTW dynamique sont plus ou moins des adaptations directes des méthodes statiques. Par exemple, la recherche locale à voisinage large est adaptée pour travailler dans un contexte dynamique dans (Gendreau *et al.*, 2006). Dans (Housroum *et al.*, 2006), les auteurs proposent d'adapter les algorithmes génétiques pour traiter des VRPTW dynamiques. L'algorithme génétique proposé commence par créer une population de solutions initiales et essaie continuellement d'en améliorer la qualité. Lorsqu'un nouveau client apparaît, il est inséré dans toutes les solutions courantes dans les positions minimisant le coût additionnel. Cependant, les heuristiques d'insertion sont les plus utilisées pour travailler dans un environnement dynamique (e.g. (Madsen *et al.*, 1995; Fu *et al.*, 1999; Horn, 2002; Diana, 2006)). Comme énoncé dans l'introduction, les heuristiques d'insertion sont, dans leur version originale, des algorithmes gloutons, dans la mesure où une décision d'allouer tel client à tel véhicule est irrévocable. Les approches multi-agents proposées dans la littérature adoptent globalement ce même principe général de fonctionnement que nous adoptons également pour notre système. Les heuristiques d'insertion sont également combinées avec des méta-heuristiques afin d'améliorer la qualité des solutions. Zhu *et al.* (2000) proposent une approche pour

3. les clients euclidiens ont des coordonnées cartésiennes, et la distance et le temps de parcours entre chaque couple de clients sont calculés selon la métrique euclidienne.

un VRP dynamique, dans laquelle un solveur central constitué de réacteurs gère les événements survenant dans le réseau. Lorsqu'un client apparaît, il est inséré dans la tournée d'un véhicule à la manière des heuristiques d'insertion. Après chaque insertion, une procédure d'optimisation est lancée tentant de réduire le nombre de véhicules et la distance totale parcourue, elle est répétée jusqu'à ce que la solution courante ne s'améliore plus. Les clients sont traités séquentiellement selon un ordre de priorité décroissant, fonction de leurs distances respectives et de l'ordre décroissant de leur fenêtre temporelle ouvrante.

Le problème des heuristiques d'insertion quand elles sont appliquées dans un contexte dynamique, c'est que leur résolution est myope. La myopie des heuristiques d'insertion résulte de leur optimisation de la situation actuelle, sans projection dans le futur. En effet, le système ne sait pas quels clients vont apparaître lorsque tels clients sont affectés à tels véhicules. L'apparition d'un nouveau client pourrait rendre cette affectation sous-optimale, et une remise en cause des anciennes décisions devrait être effectuée. En empêchant cette remise en cause, les heuristiques d'insertion présentent les temps d'exécution les plus rapides mais s'infligent un sérieux handicap.

Pourtant, dans leur grande majorité, les approches orientées-agent de la littérature reposent, au moins en partie, sur des heuristiques d'insertion.

2.2. Systèmes multi-agents

Dans (Thangiah *et al.*, 2001), les auteurs proposent une architecture multi-agent pour résoudre un VRP et un VRP à dépôts multiples. Dans (Kohout *et al.*, 1999), les auteurs proposent une architecture multi-agent pour résoudre un problème de transport à la demande. Le principe de ces deux systèmes est le même, celui de distribuer une heuristique d'insertion, suivi d'une étape de post-optimisation. Dans (Thangiah *et al.*, 2001), les clients sont traités séquentiellement, diffusés à tous les véhicules, ces derniers proposent des offres et le meilleur est retenu. Dans la seconde étape, les véhicules échangent des clients afin d'améliorer leurs solutions, chaque véhicule connaissant tous les autres agents véhicules du système. Les véhicules s'exécutant en parallèle, les auteurs envisagent d'appliquer des heuristiques et des méta-heuristiques différentes pour chaque véhicule, sans changer d'architecture. In-Time (Kohout *et al.*, 1999) est un système réalisé dans le cadre d'une application réelle d'ordonnement de véhicules d'une société de transport en aéroport. Le système est composé d'agents clients et d'agents véhicules. L'agent client s'annonce et tous les agents véhicules calculent son coût d'insertion dans leurs itinéraires et renvoient des offres à l'agent client. Ce dernier sélectionne l'offre la moins chère. Le mécanisme proposé par les auteurs représente une recherche locale distribuée. En effet, leur système permet à un client de demander d'une manière stochastique d'annuler son ordonnancement actuel et de se ré-annoncer au système dans l'objectif d'avoir un meilleur marché. MARS (Fischer *et al.*, 1995) modélise un ordonnancement coopératif dans une compagnie maritime sous la forme d'un SMA. La solution du problème de l'ordonnement global émerge de la prise de décision locale et des stratégies de ré-

solution de problèmes. Le système utilise une extension du *Contract Net Protocol* (CNP) (Smith, 1980) et montre qu'il peut être utilisé pour obtenir de bonnes solutions initiales aux problèmes complexes d'allocation de ressources. Le SMA profite d'une structuration organisationnelle *a priori*, puisque chaque véhicule est associé à une société particulière et ne peut traiter que les clients de cette société.

Pour les raisons explicitées dans l'introduction, nous choisissons une modélisation multi-agent pour résoudre le problème VRPTW dynamique. Pour leur rapidité d'exécution et leur adaptation à une configuration dynamique, nous privilégions une résolution fondée sur les heuristiques d'insertion, face aux autres heuristiques et méta-heuristiques de la littérature. Nous adoptons ainsi un processus de résolution similaire aux approches multi-agents que nous venons de présenter. Néanmoins, dans ces propositions, la myopie des heuristiques d'insertion est contrecarrée par des modifications du protocole d'interaction, en permettant une remise en cause partielle des décisions d'insertion des clients, mais aucune ne s'est intéressée à la redéfinition du coût d'insertion de ces clients. Nous proposons de garder le même protocole d'interaction – sans remise cause des décisions d'insertion des clients – privilégiant ainsi une résolution rapide. En revanche, nous proposons une nouvelle manière de calculer le coût d'insertion d'un client dans la tournée d'un véhicule qui, à la différence du calcul du détour infligé au véhicule, se concentre sur la couverture spatio-temporelle des agents véhicules. Ainsi, ce nouveau coût permet de contrebalancer la myopie des heuristiques d'insertions en privilégiant un processus d'insertion centré sur le futur.

2.3. Définition du problème

Dans ce qui suit, nous donnons une définition formelle du problème VRPTW afin de définir les paramètres et les contraintes du problème d'une manière non ambiguë. Il est important de noter que même si dans cette définition l'objectif à minimiser est le coût global des tournées, il existe traditionnellement un objectif qui lui est hiérarchiquement supérieur (Nagata *et al.*, 2010), il s'agit de la taille de la flotte de véhicules mobilisés par le système. En effet, la flotte de véhicules n'est pas fixée lorsque le système ne propose pas une résolution exacte du problème, et est également un critère à minimiser.

Définition 1 (VRPTW) Une instance $I = (G, D, T, S, F, R, \kappa)$ du problème VRPTW est définie comme suit. Soit $G = (V, E)$ un graphe avec un ensemble de nœuds $V = \{(v_i)\}$, $i = \{0, \dots, N\}$ (le nœud v_0 est le dépôt) et un ensemble d'arcs $E = \{(v_i, v_j) | v_i \in V, v_j \in V, v_i \neq v_j\}$. Soient deux matrices $D = \{(d_{ij})\}$ et $T = \{(t_{ij})\}$ de coûts, de dimensions $N \times N$ (l'arc (v_i, v_j) a un coût-distance d_{ij} et un coût-temps t_{ij}), un M -tableau F de véhicules, et un N -tableau R de tuples (R pour requêtes) $(q_i, s_i, [e_i, l_i])$ (le nœud v_i a une demande q_i , un temps de service s_i et une fenêtre temporelle $[e_i, l_i]$, $q_1 = (0, 0, [e_0, l_0])$). La fenêtre $[e_0, l_0]$ est « l'horizon d'ordonnancement » du problème. Toutes les fenêtres temporelles doivent être comprises entre ces deux bornes. Un véhicule doit être au niveau de i avant l_i mais peut être au

niveau de i avant e_i et auquel cas il doit attendre que le service commence. Chaque demande client est supposée être inférieure à κ .

Deux variables de décision sont définies : $X = (x_{ijk})$ de dimension $N \times N \times M$ et $B = (b_i)$ de dimension N ayant le sens suivant :

$$x_{ijk} = \begin{cases} 1 & \text{si le véhicule } k \text{ dessert le nœud } v_i \text{ immédiatement après le nœud } v_j \\ 0 & \text{sinon} \end{cases}$$

$$b_i = t \Leftrightarrow v_i \text{ est desservi à } t$$

La fonction objectif est :

$$\min \sum_{i,j=0}^N d_{ij} \sum_{k \in F} x_{ijk} \quad [1]$$

Résoudre un problème VRPTW consiste à trouver X et B optimisant la fonction objectif pour toute instance de I sujette aux contraintes suivantes :

$$\sum_{k \in F} \sum_{j=1}^N x_{ijk} = 1 \quad \forall v_i \in V \setminus v_0 \quad [2]$$

$$\sum_{j=1}^N x_{0jk} = 1 \quad \forall k \in F \quad [3]$$

$$\sum_{i=0}^N x_{ijk} - \sum_{i=0}^N x_{jik} = 0 \quad \forall k \in F, \forall v_j \in V \setminus v_0 \quad [4]$$

$$\sum_{j=1}^N x_{j0k} = 1 \quad \forall k \in F \quad [5]$$

$$\sum_{i=0}^N q_i \sum_{j=0}^N x_{ijk} \leq \kappa \quad \forall k \in F \quad [6]$$

$$x_{ijk}(b_i + s_i + t_{ij} - b_j) \leq 0 \quad \forall k \in F, \forall (v_i, v_j) \in E \quad [7]$$

$$e_i \leq b_i \leq l_i \quad \forall k \in F, \forall v_i \in V \quad [8]$$

$$x_{ijk} \in \{0, 1\}, \forall (v_i, v_j) \in E \quad \forall k \in F \quad [9]$$

La fonction [1] exprime l'objectif du système : la minimisation du coût total. Les contraintes [2] restreignent l'affectation de chaque client (sauf le dépôt v_0) à exactement un seul véhicule. Les contraintes [3] à [5] caractérisent le chemin à suivre par un véhicule k : k doit quitter le dépôt une seule fois [3], s'il dessert un autre client, il doit le quitter [4] et finalement retourner au dépôt une seule fois [5]. Les contraintes [6] garantissent la non-violation des limites de capacité de tous les véhicules. Les contraintes [7] – [8] s'assurent de la non-violation des contraintes temporelles.

3. Système multi-agent pour le VRPTW dynamique

Notre système est composé d'un ensemble dynamique d'agents interagissant pour aboutir à une solution au problème. Une solution consiste en un ensemble de tournées de véhicules, chaque tournée étant constituée d'une séquence de clients avec leurs temps de passage associés. Nous définissons deux catégories d'agents. Les agents clients représentant les clients du système (marchandises, objets, personnes, etc.) et les agents véhicules représentant des véhicules dans le SMA. Nous supposons également l'existence d'une borne d'accès au système (serveur web, interface graphique, générateur automatique de clients, etc). Lors de la connexion d'un utilisateur au SMA, les données fournies par l'utilisateur concernant le client sont vérifiées (nœud existant dans le réseau, fenêtres de temps valides, etc.) et le cas échéant, un agent client le représentant dans le SMA est créé.

Dans (Zeddini *et al.*, 2008), nous avons conçu, implémenté et comparé trois architectures possibles pour modéliser le problème VRPTW dynamique : une architecture centralisée, une architecture décentralisée et une architecture hybride. Les trois approches simulent rigoureusement le même comportement quant à la résolution du problème : une heuristique d'insertion distribuée. La différence entre elles concerne le processus d'interaction et la localisation des traitements. Nous les présentons dans ce qui suit. En fin de section, nous décrivons une comparaison expérimentale de ces trois architectures, prélude au choix de l'architecture la plus pertinente pour les développements ultérieurs.

3.1. Architecture centralisée

Dans cette architecture, l'ensemble des demandes est traité par un même « agent ». Il dispose de toutes les informations nécessaires sur chaque véhicule et sur chaque client : l'occupation des véhicules, leurs positions actuelles ainsi que les demandes des clients en cours et en attente de traitement. Ayant toutes ces informations, il attribue

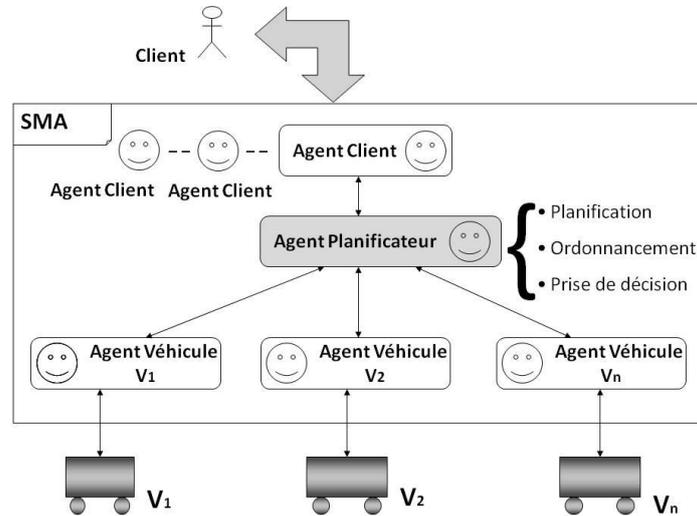


Figure 1. Architecture centralisée

à chaque client le véhicule le plus approprié pour le desservir, *i.e.* celui pour lequel le coût d'insertion est minimal.

La figure 1 illustre cette architecture dans laquelle, outre les deux agents définis plus haut, nous ajoutons un agent « planificateur » qui représente le centre de prise de décision, il a en charge les calculs d'itinéraires des véhicules et la notification aux agents de sa décision.

Le scénario que nous avons proposé d'étudier est le suivant : à un instant t donné, un utilisateur apparaît au SMA et un agent client le représentant est créé. Dès sa création, l'agent client envoie sa demande à l'agent planificateur, qui essaie de l'insérer dans la tournée de chaque véhicule du SMA, et retient celui qui a le coût additionnel minimal. Si aucun agent véhicule ne peut insérer le client, un nouvel agent véhicule est créé. Enfin, l'agent planificateur envoie la tournée modifiée à l'agent véhicule concerné et informe l'agent client de son véhicule et de son temps de passage. Les agents véhicules n'effectuent donc aucun calcul et se contentent d'accuser réception de leur tournée courante, et mettent à jour leurs informations.

L'approche centralisée pose deux problèmes principaux. D'une part, il n'est pas possible de distribuer les traitements sur différents hôtes de manière à limiter le temps de réponse du système, ce qui va à l'encontre de l'exigence de réponse rapide à des clients en ligne. D'autre part, la panne de l'agent planificateur entraîne une panne générale du système. Néanmoins, l'architecture centralisée offre l'avantage de minimiser les communications entre agents, qui se limitent à la notification du résultat du calcul aux agents clients et *Véhicule* concernés.

3.2. Architecture décentralisée

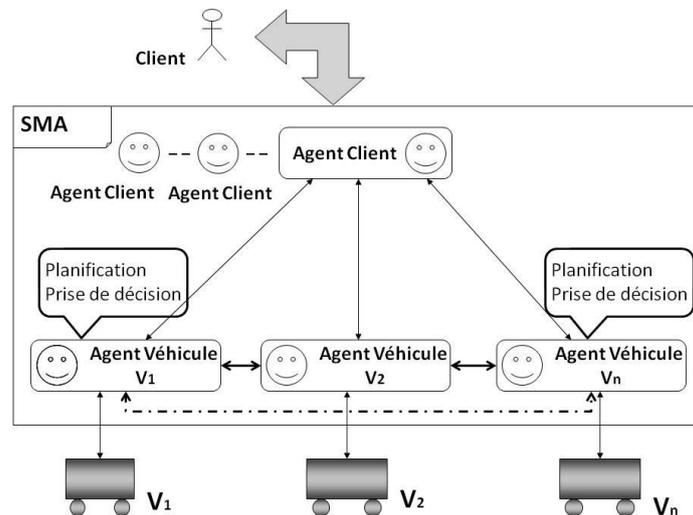


Figure 2. Architecture décentralisée

L'architecture décentralisée est illustrée dans la figure 2. Dans cette architecture, il n'existe pas de goulet d'étranglement pour le calcul d'itinéraires. À chaque apparition d'un nouveau client, ce dernier diffuse sa demande à tous les véhicules du système. Les agents véhicules s'envoient des messages avec leur coût d'insertion de ce client. Chaque agent véhicule compare son propre coût par rapport à celui reçu de la part des autres agents, et se désiste si le coût qui lui est proposé est meilleur que le sien. L'agent véhicule ayant le coût minimal se présente enfin à l'agent client et tous deux mettent à jour leurs informations. Cette architecture offre l'avantage de distribuer les traitements et d'être tolérante aux pannes. Néanmoins, le nombre de messages échangés entre les agents véhicules étant de complexité quadratique⁴, les coûts en bande passante sont prohibitifs.

3.3. Architecture hybride

L'approche hybride (cf. figure 3) est un compromis entre l'approche centralisée et décentralisée. Un nouvel agent « dispatcheur » est inséré entre les agents clients et véhicules et a pour rôle de diffuser la demande du client, de collecter les offres des agents véhicules et de choisir celui proposant le coût minimal. Le processus décrit un

4. Chacun des N véhicules diffusant $N - 1$ messages aux autres agents véhicules, la complexité est de $\mathcal{O}(N^2)$

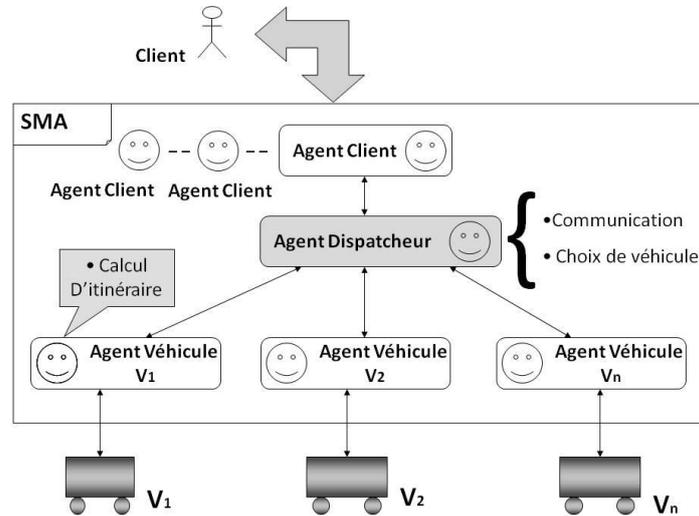


Figure 3. Architecture hybride

CNP (*Contract Net Protocol*) (Smith, 1980) où, à chaque apparition d'un agent client, l'agent dispatcheur reçoit un ensemble de propositions des agents véhicules et choisit celui dont le coût associé est minimal. L'utilisation de l'agent dispatcheur permet aux N véhicules du SMA d'envoyer un seul message chacun, au lieu des $N - 1$ messages dans l'architecture décentralisée.

Dans (Zedini, 2009), nous avons implémenté les trois architectures multi-agents proposées, et nous les avons comparé par rapport au temps moyen de réponse à un client en ligne. Nous définissons le temps de réponse comme le temps depuis la création de l'agent client jusqu'à sa réception d'une affectation à un véhicule. Nous avons effectué des expérimentations sur un réseau de quatre ordinateurs. Nous avons créé des classes de 100, 200, 300 et 400 clients. Les lieux d'origine et de destination des clients sont générés d'une manière aléatoire, avec des fenêtres de temps aléatoires. Pour chaque client, outre ses nœuds de départ, d'arrivée, sa fenêtre de temps, on doit également spécifier le moment où il est connu par le système, son temps d'apparition. Pour chaque client, nous avons créé dix temps d'apparition générés aléatoirement, de manière à tourner nos simulations avec les mêmes clients, mais avec dix ordres d'apparition différents. Les résultats ont montré que l'architecture hybride tire profit de la distribution des traitements, sans souffrir d'un nombre trop important de messages échangés. L'architecture distribuée vient en seconde position en termes de performances, tirant profit de la distribution mais souffrant de sa grande consommation de bande passante. L'architecture centralisée vient en dernière position, puisque son gain en termes de nombre de messages échangés ne contrebalance pas sa séquentialisation des traitements.

Pour ces raisons, nous adoptons dans la suite de cet article l'architecture hybride comme référence. Les SMA détaillés dans ce qui suit, adhèrent tous à cette architecture. En effet, c'est dans le cadre du processus décrit par l'architecture hybride que se situe notre proposition des zones d'action spatio-temporelle des agents véhicules. L'agent dispatcheur fonde sa décision d'insérer tel client dans la tournée de tel véhicule sur le coût additionnel de l'insertion d'un client dans la tournée du véhicule. Traditionnellement, ce coût correspond à la mesure du détour effectué par le véhicule pour intégrer le nouveau client. Nous proposons une nouvelle mesure du coût d'insertion fondée sur une modélisation particulière des zones d'action des agents véhicules. Il s'agit d'une mesure alternative à celle du détour effectué par le véhicule qui, contrairement à celle-ci, intègre l'impact de la décision courante sur l'insertion des clients futurs, encore inconnus.

4. Modèle d'auto-organisation

Dans les méthodes heuristiques et multi-agents de la littérature, l'objectif hiérarchique de minimisation du nombre de véhicules mobilisés est considéré en priorité par rapport à la distance totale parcourue par l'ensemble des véhicules. L'immense majorité des heuristiques de la littérature sont par conséquent fondées sur une approche à deux étapes : la minimisation du nombre de véhicules, suivie de la minimisation de la distance parcourue (Nagata *et al.*, 2010). Le modèle d'auto-organisation que nous proposons dans cette section a pour objectif de minimiser le nombre de véhicules mobilisés, tout en gardant l'utilisation d'une heuristique d'insertion « pure », *i.e.* sans aucune amélioration ultérieure. Bien que ce modèle n'intègre pas explicitement la minimisation de la distance parcourue par les véhicules, nous nous intéressons à son comportement par rapport à ce critère (cf. section 5), en vue d'une éventuelle deuxième étape de minimisation de la distance.

Pour parvenir à cet objectif, notre modèle permet aux agents véhicules de couvrir une zone spatio-temporelle maximale du réseau de transport, évitant ainsi la mobilisation de nouveaux véhicules lors de l'apparition d'un nouveau client. Un couple espace-temps $\langle i, t \rangle$ – avec i un nœud et t un temps – est dit « couvert » par l'agent véhicule v si v peut être au niveau du nœud i à l'instant t . Dans le cadre du VRPTW dynamique, maximiser la couverture spatio-temporelle des agents véhicules revient à se donner le maximum de chances de satisfaire la demande d'un nouveau client dans le futur. Cette mesure rompt avec la logique des mesures traditionnelles qui se focalisent sur l'accroissement de la distance parcourue, en négligeant l'impact de la décision courante sur les insertions futures.

Suivant la description présentée dans la section précédente, l'agent dispatcheur choisit entre plusieurs agents véhicules celui dont le coût d'insertion proposé est minimal. Les systèmes fondés sur cette heuristique d'insertion utilisent généralement la mesure de Solomon (1987) comme coût d'insertion. Cette mesure consiste à insérer le client qui entraîne une augmentation minimale du coût général du véhicule (fonction du détour à effectuer par le véhicule). Cette mesure est simple et est la plus intuitive

mais elle présente un inconvénient sérieux. En effet, l'insertion du client courant pourrait rendre irréalisables un grand nombre de dessertes de clients futurs. Son problème est qu'elle génère des plans de véhicules qui sont très contraints dans le temps et dans l'espace, *i.e.* des plans offrant peu de possibilités d'insertion entre chaque couple de clients adjacents dans la tournée d'un véhicule. Par conséquent, l'apparition de nouveaux clients risque fort de mobiliser un nouveau véhicule pour les servir. À travers la modélisation des zones d'action des véhicules, nous proposons un nouveau calcul du coût d'insertion d'un client dans la tournée d'un agent véhicule, et ainsi un nouveau critère de choix entre eux. Nous proposons un calcul dont l'objectif est de choisir l'agent véhicule en « maximisant sa probabilité de participation à des insertions futures ».

4.1. Modélisation de l'environnement

La « zone d'action » spatio-temporelle d'un agent véhicule est composée d'un sous-ensemble des nœuds du réseau de transport avec les temps qui lui sont associés. Ainsi, nous modélisons l'environnement du SMA sous la forme d'un réseau espace-temps, déduit à partir du graphe de réseau de transport. Chaque nœud du graphe est désormais un couple $\langle espace, temps \rangle$, qui représente « l'état » d'un nœud à une période de temps discrète. Le réseau espace-temps se compose de h sous-graphes, où chaque sous-graphe $sg \in SG$ est une copie du graphe statique, et correspond à l'état du graphe à un certain moment dans le temps (cf. figure 4). Nous indexons les nœuds d'un sous-graphe sg du graphe spatio-temporel correspondant à l'instant qu'ils décrivent comme suit : $\langle 1, t \rangle, \dots, \langle N, t \rangle$, avec t appartenant à $\{1, \dots, h\}$. Le nombre total des nœuds dans le graphe spatio-temporel est alors égal à $h \times N$. Les arcs reliant les nœuds d'un sous-graphe sg sont ceux du graphe statique, et les coûts sont les coûts de la matrice C .

4.2. Intuition du modèle

Soit un agent véhicule v qui n'a aucun client dans son plan. Afin que cet agent puisse insérer un nouveau client c décrit par n un nœud, $[e, l]$ une fenêtre temporelle, s un temps de service, et q une quantité, il faut que l soit assez grand pour permettre à v d'être au niveau de n sans violer ses contraintes temporelles. Plus précisément, il faut que le temps courant $t_{courant}$, plus le temps de parcours entre le dépôt et n soit inférieur ou égal à l (cf. figure 5). Partant de cette observation, nous définissons la zone d'action d'un agent véhicule comme le nombre de clients potentiels qui satisfont à cette contrainte. Pour ce faire, nous définissons la zone d'action d'un agent véhicule comme le nombre de couples $\langle n, t \rangle$ du réseau espace-temps qui restent valides (n est accessible par le véhicule à l'instant t) étant donné son plan courant. La zone d'action d'un agent véhicule avec un plan vide est illustrée par une ombre conique dans la figure 6.

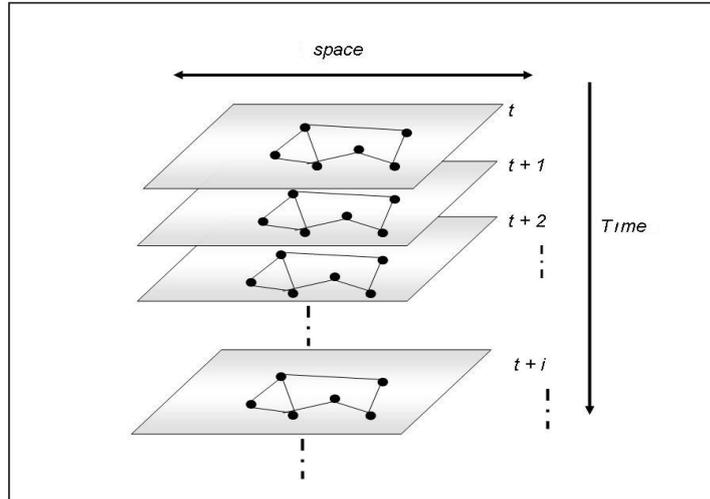


Figure 4. Réseau espace-temps

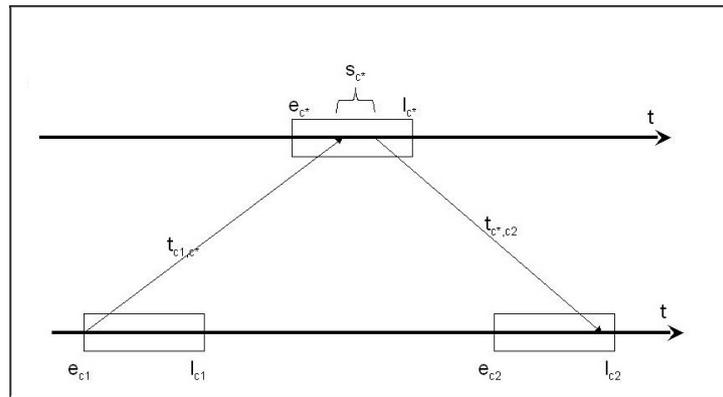


Figure 5. Insertion valide de c^* entre c_1 et c_2

Lorsqu'un client est inséré dans la tournée d'un agent véhicule, la zone d'action de ce dernier est recalculée, puisqu'un certain nombre de couples $\langle n, t \rangle$ deviennent non valides suite à son insertion. Dans la figure 7, un nouveau client est inséré dans la tournée du véhicule. La zone d'action d'un agent véhicule après l'insertion d'un client est représentée par l'intérieur du contour des lignes noires, qui représente les nœuds espace-temps qui demeurent accessibles après l'insertion du client (le calcul de la nouvelle zone d'action est explicité plus bas).

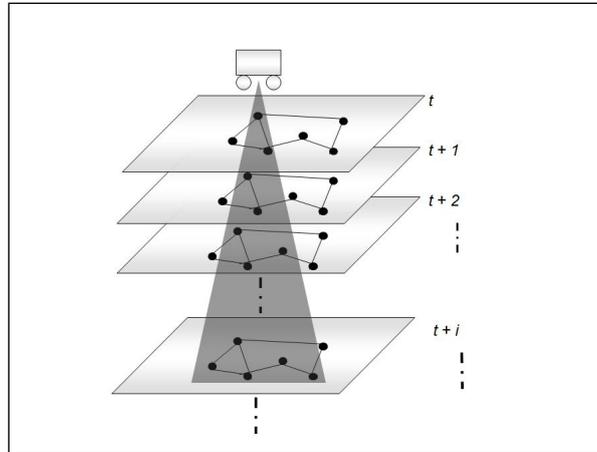


Figure 6. *Zone d'action spatio-temporelle initiale*

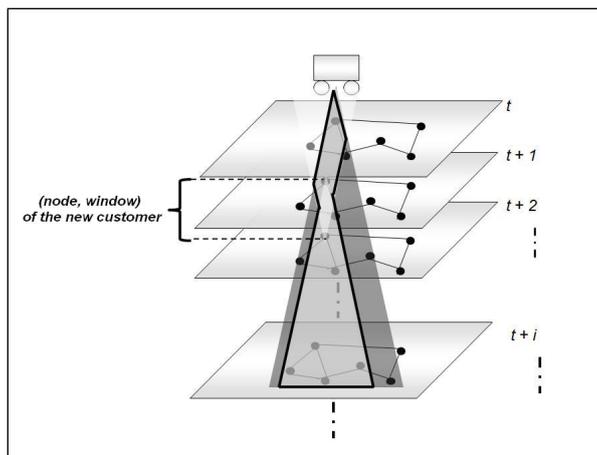


Figure 7. *Zone d'action après insertion d'un nouveau nœud*

Le coût associé à une offre d'un agent véhicule v pour l'insertion d'un agent client c correspond à la diminution hypothétique de la zone d'action de v suite à l'insertion de c dans son plan. L'idée est que l'agent véhicule choisi pour l'insertion d'un client est celui qui perd le moins de chances d'être candidat pour l'insertion de clients futurs. Ainsi, le critère maximisé par l'ensemble des agents véhicules est la somme de leurs zones d'action, *i.e.* la capacité qu'a le SMA de réagir à l'apparition d'agents clients, sans mobiliser de nouveaux véhicules.

À titre d'illustration, nous présentons dans ce qui suit la version de la mesure relative à un problème euclidien, *i.e.* où les temps de parcours sont calculés suivant la métrique euclidienne. Les sections suivantes détaillent la mesure ainsi que sa dynamique.

4.3. Calcul des zones d'action

Dans le cas euclidien, le réseau de transport est un plan, et le temps de parcours entre deux points i (décrit par (x_i, y_i)) et j (décrit par (x_j, y_j)) est égal à

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Par conséquent, si un véhicule se trouve au point i à l'instant t_i , il peut être au plus tôt au point j à l'instant $t_i + \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

Nous pouvons calculer à tout moment, à partir de la position d'un véhicule, l'ensemble des triplet (x, y, t) où il peut être dans le futur. En effet, en considérant un plan avec un axe des abscisses dans $[x_{min}, x_{max}]$ et un axe des ordonnées dans $[y_{min}, y_{max}]$, l'ensemble des positions spatio-temporelles est l'ensemble des points dans le cube délimité par $[x_{min}, x_{max}], [y_{min}, y_{max}]$ et $[e_0, l_0]$. Soit un véhicule se trouvant au niveau du dépôt (x_0, y_0) à l'instant t_0 . L'ensemble des points (x, y, t) accessibles par ce véhicule sont décrits par l'inéquation suivante :

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} \leq (t - t_0)$$

Les (x, y, t) satisfaisant cette inéquation sont ceux qui sont positionnés à l'intérieur du cône \mathcal{C} de sommet (x_0, y_0, t_0) et d'équation $\sqrt{(x - x_0)^2 + (y - y_0)^2} = (t - t_0)$ (cf. figure 8). Ce cône représente la zone d'action d'un agent véhicule dans le cas euclidien. Il représente toutes les positions spatio-temporelles possibles que cet agent véhicule est susceptible d'avoir dans le futur.

Nous utilisons la zone d'action des agents véhicules lors du choix entre les coût d'insertion proposés par plusieurs agents véhicules à un agent client. Il faut donc pouvoir comparer des zones d'action de différents agents véhicules. Pour ce faire, nous proposons de quantifier cette zone d'action, en calculant le volume du cône \mathcal{C} représentant les positions futures possibles du véhicule :

$$Volume(\mathcal{C}) = \frac{1}{3} \times \pi \times (l_0 - e_0)^3$$

Il s'agit d'une quantification de la zone d'action initiale de tout nouvel agent véhicule créé. Lors de l'apparition d'un *Client*, un agent véhicule calcule sa nouvelle zone d'action, le coût qu'il propose à l'agent dispatcheur est la différence entre son ancienne zone d'action et sa nouvelle, que nous détaillons dans le paragraphe qui suit.

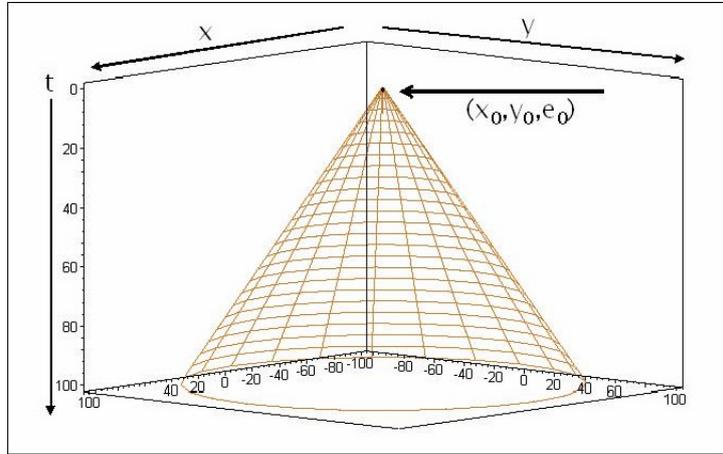


Figure 8. Zone d'action spatio-temporelle initiale

4.4. Dynamique des zones d'action

Considérons un client c_2 (de coordonnées (x_2, y_2) et avec une fenêtre temporelle $[e_2, l_2]$) qui rejoint le système, et supposons qu'il soit dans la zone d'action de v , avec v le seul agent véhicule du système, avec un plan vide. L'agent v doit déduire sa nouvelle zone d'action *i.e.* ses nouveaux couples $\langle \text{espace}, \text{temps} \rangle$ qu'il peut toujours atteindre sans violer les contraintes temporelles de c_2 . La nouvelle zone d'action est définie en répondant aux deux questions suivantes : « si v devait être au niveau de (x_2, y_2) à l_2 , où est ce qu'il aurait pu être avant ? Et s'il devait y être à e_2 où est-ce qu'il pourrait être après $e_2 + s_2$? ». Les triplets (x, y, t) où l'agent véhicule peut être avant la desserte de c_2 sont décrits par l'inéquation $[a]$, et les triplets (x, y, t) où l'agent véhicule peut être après la desserte de c_2 sont décrits par l'inéquation $[b]$.

$$\sqrt{(x - x_2)^2 + (y - y_2)^2} \leq (l_2 - (t + s)) \quad [a]$$

$$\sqrt{(x - x_2)^2 + (y - y_2)^2} \leq (t - (e_2 + s_2)) \quad [b]$$

La nouvelle zone d'action est illustrée par la figure 9 : la nouvelle mesure consiste en l'intersection du cône initial \mathcal{C} avec l'union des deux nouveaux cônes décrits par les inéquations $[a]$ et $[b]$ (dénotés respectivement par \mathcal{C}_1 et \mathcal{C}_2). La nouvelle mesure de la zone d'action est égale au volume de l'intersection de \mathcal{C} avec l'union de \mathcal{C}_1 et \mathcal{C}_2 .

Afin de calculer le nouveau volume, nous calculons d'abord le volume de l'intersection de \mathcal{C} avec \mathcal{C}_2 , qui est égale au volume de \mathcal{C}_2 , car il est totalement inclus dans \mathcal{C} . Le volume du cône \mathcal{C}_2 de sommet $(x_2, y_2, e_2 + s_2)$ est égal à $\frac{1}{3} \times \pi \times (l_0 - (e_2 + s_2))^3$. Ensuite, nous calculons l'intersection du cône initial avec le cône de sommet (x_2, y_2, l_2^*) ⁵, qui donne deux cônes identiques de base elliptique. Le calcul du volume de l'intersection de ces deux cônes est décrit dans (Zargayouna, 2007).

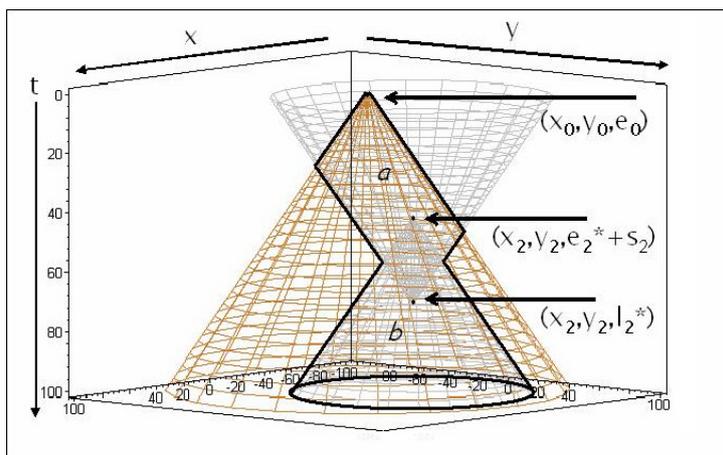


Figure 9. Zone d'action spatio-temporelle après l'insertion de c_2

Le coût de l'insertion d'un client dans la tournée d'un véhicule est égal à la mesure associée à l'ancienne zone d'action du véhicule moins la mesure de la nouvelle zone d'action, après l'insertion du client. La quantité ainsi mesurée représente les positions spatio-temporelles que le véhicule ne peut désormais plus avoir, s'il avait à insérer ce client dans son plan. L'agent véhicule retenu pour la desserte d'un client est celui dont l'insertion du client lui diminue le moins sa zone d'action spatio-temporelle. Cela correspond à choisir le véhicule qui perd le moins de possibilités d'être candidat pour des clients futurs.

L'environnement physique dans le cas non euclidien n'est plus un cube espace-temps, mais un réseau espace-temps. Dans (Zargayouna, 2005), nous proposons une méthode pour l'auto-organisation du SMA dans le cas général, où chaque agent véhicule associe à chaque nœud n du réseau l'ensemble des instants où il peut être à son niveau, ce qui représente sa zone d'action. La quantité associée à la zone d'action d'un agent véhicule est définie comme le cardinal de ces nœuds espace-temps. À chaque tentative d'insertion d'un client, l'agent véhicule calcule les nœuds espace-temps qu'il ne peut plus couvrir désormais. Le coût d'insertion proposé pour l'insertion du nouveau client est le cardinal de ces nœuds.

5. Le calcul de l^* dépasse le cadre de cet article et est détaillé dans (Zargayouna, 2007).

4.5. Coordination des zones d'action

L'objectif du modèle d'auto-organisation est de permettre une meilleure couverture spatio-temporelle du réseau de transport. Cette amélioration est matérialisée par une mobilisation minimale de véhicules face à l'apparition de nouveaux clients. Avec le mécanisme décrit jusque-là, chaque agent véhicule essaie de maximiser sa propre zone d'action indépendamment du comportement des autres agents du SMA. Or, il serait plus intéressant d'inciter la société d'agents véhicules dans son ensemble de couvrir le réseau d'une manière complémentaire. Plus précisément, qu'un véhicule perde des nœuds espace-temps qu'il est le seul à couvrir devrait être plus coûteux que de perdre des nœuds que d'autres agents couvrent.

Pour ce faire, à chaque nœud du réseau espace-temps, nous commençons par associer la liste des véhicules qui le couvrent. Ensuite, à chaque création d'un nouvel agent véhicule, l'ensemble des nœuds espace-temps faisant partie de sa zone d'action est calculé. Le véhicule procède ensuite à la notification de ces nœuds qu'ils font partie de sa zone d'action. Chaque nœud pour sa part, met à jour la liste des véhicules qui le couvrent à chaque notification d'un agent véhicule. De même, lorsque la zone d'action d'un agent véhicule perd un nœud, le nœud est également notifié, et sa liste d'agents véhicules mise à jour.

Maintenant, lors du calcul du coût d'insertion d'un client, chaque agent véhicule commence par calculer les nœuds espace-temps qu'il perdrait s'il venait à insérer le nouveau client. Ensuite, il interroge chacun de ces nœuds du « prix à payer » s'il venait à ne plus les couvrir. Ce prix est inversement proportionnel au nombre de véhicules qui couvrent ce nœud. Plus précisément, le prix à payer est égal à

$$\frac{1}{\text{card}(v_{\langle n,t \rangle})}$$

avec $v_{\langle n,t \rangle}$ dénotant les agents véhicules qui couvrent le nœud espace-temps $\langle n, t \rangle$.

Ainsi, le réseau espace-temps étant le seul à connaître les zones d'action des agents véhicules (grâce aux listes de véhicules associées aux nœuds), il associe plus ou moins de pénalité aux décisions de non-couverture du réseau par les véhicules à travers le temps. Ainsi, les agents véhicules sont incités à couvrir l'ensemble du réseau d'une manière coordonnée, améliorant ainsi la réactivité du SMA.

5. Expérimentations

Il existe un problème de validation des différentes approches de résolution des problèmes de tournée de véhicules. En effet, les différentes heuristiques, afin de pouvoir être comparées entre elles, doivent être testées avec les mêmes données. Ce problème de standardisation de l'évaluation des problèmes de tournées de véhicules avec fenêtres temporelles a été résolu en partie grâce à un travail de Marius M. Solomon (1987), qui a créé un ensemble de fichiers de clients différents pour ce problème. Il est

admis que ces problèmes sont assez divers et assez nombreux pour pouvoir comparer avec une confiance suffisante les différentes approches proposées. Ces problèmes sont difficiles, puisqu'il n'existe aucune heuristique donnant les meilleurs résultats pour tous les problèmes en même temps. Dans les benchmarks de Solomon, six différents ensembles de problèmes sont définis : C1, C2, R1, R2, RC1 et RC2. Les clients sont uniformément distribués géographiquement dans les problèmes du type R, regroupés dans les problèmes du type C, et un mélange de clients uniformément distribués et regroupés est utilisé dans les problèmes de type RC. Les problèmes du type 1 ont des fenêtres temporelles étroites, ainsi très peu de clients peuvent coexister dans le même plan, et les problèmes de type 2 ont des fenêtres temporelles larges. Enfin, un temps de service constant est associé à chaque client, il est de 10 dans les problèmes de types R et RC, et de 90 dans les problèmes de type C.

Le problème de validation pour le cas dynamique reste entier, puisqu'il n'existe pas de *benchmark* pour le problème dynamique. Dans notre cas, où le caractère dynamique provient du fait que tous les clients ne sont pas connus à l'avance, plusieurs choix doivent être effectués. En effet, combien de clients sont connus à l'avance ? Quelle est la cadence d'arrivée des clients ? Dans la littérature, les auteurs ont fait des choix différents. Certains auteurs partent des problèmes de Solomon et introduisent la dynamique en cachant une partie des clients et en les diffusant au fur et à mesure (e.g. (Bent *et al.*, 2004b)). D'autres auteurs considèrent de nouveaux problèmes créés à partir de données réelles, les paramètres relatifs à la dynamique des clients sont ceux du problème réel (e.g. (Diana, 2006)). Or, si les données du problème ne sont pas rigoureusement les mêmes, il faut implémenter les autres approches pour les comparer avec une nouvelle approche. Et comme tous les paramètres de calibrage des systèmes proposés ne sont souvent pas publiés, il devient très difficile de pouvoir comparer différentes propositions.

Nous concernant, nous avons fait le choix de n'avoir aucun client connu à l'avance. Par ailleurs, nous ne disposons pas de données réelles qui peuvent nous renseigner sur la manière avec laquelle les clients sont révélés au système. Nous utilisons les *benchmark* de Solomon en suivant le processus de modification proposé par Gendreau *et al.* (1999) pour rendre le problème dynamique. Soit $[0, T]$ le temps de la simulation. Toutes les données relatives au temps (fenêtres temporelles, temps de service et temps de parcours) sont multipliés par $\frac{T}{l_0 - e_0}$, avec $[e_0, l_0]$ l'horizon d'ordonnancement du problème. Les auteurs divisent l'ensemble des clients en deux sous-ensembles, le premier sous-ensemble définissant les clients connus à l'avance, et le second ceux qui sont révélés durant l'exécution. Nous ne procédons pas à cette division, puisque nous ne considérons aucun client connu à l'avance. Pour l'ensemble de nos clients, un temps d'occurrence est associé à chaque client définissant le moment où le client est connu par le système. Étant donné un client i , le temps d'occurrence qui lui est associé est généré d'une manière aléatoire entre $[0, \bar{e}_i]$, avec :

$$\bar{e}_i = e_i \times \frac{T}{l_0 - e_0}$$

Il est connu que le comportement des heuristiques d'insertion est fortement sensible à l'ordre d'apparition des clients. Pour cette raison, nous ne considérons pas un seul ordre d'arrivée. Nous effectuons le précédant processus dix fois avec chaque ensemble de clients, pour créer ainsi dix versions différentes de chaque fichier. Nous tournons ainsi nos systèmes avec dix ordres d'apparition différents pour chaque problème.

Nous avons implémenté deux SMA avec un comportement identique, suivant l'architecture hybride. La seule différence concerne la mesure utilisée par un agent véhicule pour calculer son coût d'insertion d'un client. Pour le premier SMA implémenté, elle repose sur la mesure de Solomon (Δ Distance). Pour le second, le coût d'insertion d'un client est fondé sur la mesure de la variation des zones d'action spatio-temporelles (Δ Zone Espace-Temps). Nous avons exécuté nos expérimentations sur un PC équipé d'un processeur Core 2 DuoTM à 2,77 GHZ, avec une mémoire vive de 4 Go.

Afin d'alléger la présentation des résultats, nous choisissons de faire tourner nos expérimentations avec les classes de problèmes les plus représentatives : R et C, celles de type RC ne sont qu'un mélange de ces deux classes. Nous menons nos expérimentations avec les classes de type 1, qui sont des instances très contraintes dans le temps (fenêtres temporelles étroites). En effet, les classes de type 2 (R2, C2 et RC2) présentent des fenêtres de temps larges. Dans ces problèmes, peu de véhicules sont nécessaires pour servir l'ensemble des clients (du fait du relâchement des contraintes temporelles). Par conséquent, l'aspect ordonnancement des clients dans la tournée d'un véhicule prime dans les classes de ce type sur l'aspect affectation des clients aux tournées. Or, notre objectif premier est la minimisation du nombre de véhicules, et c'est sur les classes de problèmes de type 1 que l'heuristique d'insertion classique a le plus de lacunes. Nous considérons donc les classes de problèmes R1 et C1.

Le tableau 1 synthétise les résultats obtenus en reportant les sommes des meilleurs résultats pour chaque classe de problème et chaque nombre de clients considérés. Les résultats détaillés peuvent être trouvés dans (Zeddini, 2009). Chaque ligne du tableau 1 correspond à un fichier de clients particulier. Les quatre colonnes du tableau correspondent à la version du SMA (avec la mesure de Solomon et avec la mesure spatio-temporelle) et le critère considéré (le nombre de véhicules mobilisées par le système (| Flotte |) ainsi que la distance totale parcourue par les véhicules). Entre les dix ordres d'apparition de clients par fichier, nous retenons celui qui donne le meilleur résultat. Chaque cellule du tableau reporte la somme de ces résultats.

Les résultats montrent la supériorité du modèle d'auto-organisation spatio-temporelle. En effet, quelle que soit la classe de problèmes, et quel que soit le nombre de clients considérés, le SMA se fondant sur le modèle d'auto-organisation spatio-temporelle mobilise moins de véhicules. La mesure atteint donc les objectifs pour lesquels nous les avons conçus, ceux de réaliser une meilleure couverture du réseau, matérialisée par un moindre nombre de véhicules pour desservir les clients.

Une fois ce résultat validé, il est intéressant de regarder le comportement du système par rapport à la distance totale parcourue par l'ensemble des véhicules. Rappe-

Problème	Δ Distance		Δ Zone Espace-temps	
	Flotte	Distance	Flotte	Distance
R1 25 clients	64	6372	53	6561
C1 25 clients	34	3167	31	3152
R1 50 clients	107	12036	92	12089
C1 50 clients	60	6712	53	7093
R1 100 clients	181	17907	150	17348
C1 100 clients	121	16011	108	16512

Tableau 1. Résultats synthétiques

lons que ce critère n'est considéré pour la validation des résultats qu'en cas d'égalité dans le nombre de véhicules mobilisés, ce qui n'est majoritairement pas le cas dans nos expérimentations. Les conclusions que nous tirons de ces résultats sont que la nouvelle mesure est compétitive par rapport à la mesure de Solomon relativement à la distance totale parcourue par les véhicules. Le SMA fondé sur le modèle d'auto-organisation donne, en moyenne, de meilleurs résultats que la mesure de Solomon pour les problèmes C1 avec 25 clients et R1 avec 100 clients, et il donne des résultats de moindre qualité pour les autres. Il n'en reste pas moins que les résultats restent très compétitifs si on considère que ce modèle ne prend pas du tout en compte la distance comme critère de choix entre clients. Ce dernier résultat contredit les conclusions reportées dans (Zargayouna, 2007), qui concluent qu'une mesure spatio-temporelle n'est pas compétitive par rapport à la distance parcourue par les véhicules. Cette différence est à imputer à deux facteurs. Le premier c'est que nous considérons un espace discret, sous forme d'un graphe, alors que dans (Zargayouna, 2007), nous avons considéré un problème euclidien avec un espace continu. Notre mesure est de ce fait plus précise. Le second facteur est que nous utilisons des zones d'action coordonnées (cf. paragraphe 4.5) qui améliorent la couverture spatio-temporelle du réseau par les véhicules en les empêchant de se concentrer sur certaines zones spatiotemporelles au détriment d'autres.

Le modèle développé dans cet article offre au décideur le choix entre deux modèles à utiliser, selon la configuration opérationnelle de son application. Dans les cas où l'opérateur de transport dispose d'une flotte de véhicules limitée, et où le coût de mobilisation d'un nouveau véhicule est très coûteux, il est préférable d'utiliser le modèle d'auto-organisation spatio-temporelle, qui mobilise le minimum de véhicules. En revanche, dans certains systèmes, l'opérateur dispose d'une flotte de véhicules virtuellement illimitée, et les coûts en termes de distance parcourue sont plus importants. En effet, certains systèmes reposent sur une flotte de taxis mobilisables à volonté qui, par ailleurs, effectuent un service traditionnel de taxis individuels. Dans ce genre de système, il est plus intéressant de faire reposer son système sur le modèle traditionnel.

6. Conclusion et perspectives

Dans ce papier, nous proposons un modèle d'auto-organisation orienté-agent pour le VRPTW dynamique se fondant sur les zones d'action des agents. Ces dernières reflètent la couverture spatio-temporelle des agents véhicules de l'environnement. Nous utilisons la variation de ces zones d'action comme une nouvelle métrique afin de réduire le comportement myope des mesures traditionnelles. En optimisant la couverture spatio-temporelle de l'environnement par les agents véhicules, notre modèle permet au SMA de s'auto-organiser en affichant une répartition spatio-temporelle équilibrée de ses agents véhicules, et de diminuer ainsi le nombre de véhicules mobilisés pour la desserte des clients.

Nous projetons pour le travail reporté dans cet article plusieurs pistes de réflexion que nous réservons pour nos travaux futurs. D'abord, nous prévoyons d'observer le comportement du SMA selon des critères plus qualitatifs, tels que l'existence de phénomènes émergents et leur éventuelle utilité ou nocivité au processus d'optimisation des tournées. Nous envisageons également de travailler sur le prépositionnement des véhicules sur le réseau de manière à anticiper l'apparition de clients et améliorer ainsi la réactivité du système et sa qualité de service (temps d'attente réduits). Ensuite, comme la quasi-totalité des travaux de la littérature, les temps de parcours dans les SMA développés dans cet article sont statiques. Si les systèmes sont mis en place dans des zones urbaines sujettes aux congestions, les tournées calculées risquent fortement de devenir non valides en cours d'exécution. Pour pallier cette limite, nous envisageons d'utiliser un système d'« options » sur les requêtes des clients. Une option est une réservation d'un client, avec une possibilité de désistement ultérieur, moyennant une pénalité. Le retrait d'une option par un véhicule pourrait survenir quand les prévisions de trafic, qui deviennent de plus en plus précises à travers le temps, engendrent une violation des fenêtres temporelles d'un client qu'il aurait « réservé ».

En continuation des travaux sur les temps de parcours dynamiques, nous envisageons d'appliquer nos propositions et de les approfondir dans le contexte de la gestion de crise. Plus précisément, nos SMA peuvent être étendus pour permettre la gestion et l'optimisation des évacuations d'une zone sinistrée, avec les transports en commun. Il s'agit dans ce contexte de trouver une nouvelle gestion des flottes de véhicules à disposition, ainsi que de leurs itinéraires. Le temps de parcours dans ce genre de systèmes est évidemment très dynamique et il doit être intégré dans le processus de résolution d'une manière bien plus systématique que la gestion d'options.

Enfin, dans les SMA que nous proposons dans cet article, un même véhicule traite une requête d'un client de bout en bout, *i.e.* depuis le dépôt jusqu'à sa destination. Nous envisageons de considérer la possibilité pour deux véhicules de se porter candidats à un même client, qui effectuerait une partie de son parcours à bord du premier véhicule, et une autre partie à bord du second. Les deux véhicules doivent donc synchroniser leurs tournées de manière à permettre ce transfert. Les formations de coalition nous semblent être une piste intéressante à investiguer dans ce contexte.

7. Bibliographie

- Adams J. A., « Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence », *AI Magazine*, vol. 22, n° 2, p. 105-108, 2001.
- Bent R., Hentenryck P., « A two-stage hybrid local search for the vehicle routing problem with time windows », *Transportation Science*, vol. 38, n° 4, p. 515-530, 2004a.
- Bent R. W., Hentenryck P. V., « Scenario-Based Planning for Partially Dynamic Vehicle Routing Problems with Stochastic Customers », *Operations Research*, vol. 52, n° 6, p. 977-987, 2004b.
- Czech Z. J., Czarnas P., « A Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windows », *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, Canary Islands (Spain), p. 376-383, 2002.
- Dantzig G. B., Ramser J. H., « The Truck Dispatching Problem », *Management Science*, vol. 6, n° 1, p. 80-91, 1959.
- Desaulniers G., Desrosiers J., Solomon M., Soumis F., Cordeau J.-F., « The VRP with Time Windows », in D. Vigo, P. Toth (eds), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, p. 157-193, 2002.
- Diana M., « The Importance of Information Flows Temporal Attributes for the Efficient Scheduling of Dynamic Demand Responsive Transport Services », *Journal of advanced Transportation*, vol. 40, n° 1, p. 23-46, 2006.
- Ferber J., *Les systèmes multiagents : Vers une intelligence collective*, Inter-Editions, 1995. 522 pages.
- Fischer K., Muller J., Pischel M., Schier D., « A Model for Cooperative Transportation Scheduling », in V. R. Lesser, L. Gasser (eds), *Proceedings of the First International Conference on Multiagent Systems (ICMAS'95)*, AAAI Press / MIT Press, Menlo park, CA (USA), p. 109-116, 1995.
- Fu L., Tepley S., « On-Line and Off-Line Routing and Scheduling of Dial-a-Ride Paratransit Vehicles », *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, Blackwell Publishers, Oxford (UK), p. 309-319, 1999.
- Gambardella L. M., Taillard E. D., Agazzi G., « MACS-VRPTW : A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows », in M. D. D. Corne, F. Glover (eds), *New Ideas in Optimization*, McGraw-Hill (London), p. 63-76, 1999.
- Gendreau M., Guertin F., Potvin J.-Y., Séguin R., « Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries », *Transportation Research Part C*, vol. 14, p. 157-174, 2006.
- Gendreau M., Guertin F., Potvin J.-Y., Taillard E. D., « Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching », *Transportation Science*, vol. 33, n° 4, p. 381-390, 1999.
- Golden B., Raghavan S., Wasil E., *The vehicle routing problem, latest advances and new challenges*, vol. 43 of *Operations research/computer science interfaces*, Springer Verlag, 2008.
- Homberger J., Gehring H., « Two-phase hybrid metaheuristic for the vehicle routing problem with time windows », *European Journal of Operational Research*, vol. 162, p. 220-238, 2005.
- Horn M. E., « Fleet scheduling and dispatching for demand-responsive passenger services », *Transportation Research C*, vol. 10, n° 1, p. 35-63, 2002.

- Housroum H., Hsu T., Dupas R., Goncalves G., « A hybrid GA approach for solving the Dynamic Vehicle Routing Problem with Time Windows », in I. C. Society (ed.), *Proceedings of the IEEE Conference on Information and Communication Technologies : from Theory to Applications*, Damascus, Syria, p. 3347-3352, 2006.
- Ibaraki T., Imahori S., Nonobe K., Sobue K., Uno T., Yagiura M., « An iterated local search algorithm for the vehicle routing problem with convex time penalty functions », *Discrete Applied Mathematics*, vol. 156, n° 11, p. 2050-2069, 2008.
- Ibaraki T., Kubo M., Masuda T., Uno T., Yagiura M., « Effective local search algorithms for the vehicle routing problem with general time window constraints », *Transportation Science*, vol. 39, n° 2, p. 206-232, 2005.
- Jepsen M., Petersen B., Spoorendonk S., Pisinger D., « Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows », *Operations Research*, vol. 56, n° 2, p. 497-511, 2008.
- Kohout R., Erol K., « In-Time agent-based vehicle routing with a stochastic improvement heuristic », *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence (AAAI'99/IAAI'99)*, AAAI Press, Menlo Park, CA (USA), p. 864-869, 1999.
- Larsen A., *The Dynamic Vehicle Routing Problem*, PhD thesis, University of Denmark, 2000.
- Lim A., Zhang X., « A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows », *INFORMS Journal on Computing*, vol. 19, n° 3, p. 443-57, 2007.
- Madsen O. B., Ravn H. F., Rygaard J. M., « A Heuristic Algorithm for a Dial-a-Ride Problem with Time Windows, Multiple Capacities, and Multiple Objectives », *Operations Research*, vol. 60, p. 193-208, 1995.
- Mester D., Bräysy O., « Active guided evolution strategies for large scale vehicle routing problems with time windows », *Computers & Operations Research*, vol. 32, n° 6, p. 1593-1614, 2005.
- Nagata Y., Bräysy O., Dullaert W., « A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows », *Computers & Operations Research*, vol. 37, p. 724-737, April, 2010.
- Pisinger D., Ropke S., « A general heuristic for vehicle routing problems », *Computers & Operations Research*, vol. 34, n° 8, p. 2403-2435, 2007.
- Smith R. G., « The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver », *IEEE Trans. on Comp.*, vol. C-29, n° 12, p. 1104-1113, December, 1980.
- Solomon M., « Algorithms for the vehicle routing and scheduling with time window constraints », *Operations Research*, vol. 15, p. 254-265, 1987.
- Sycara K. P., « Multiagent Systems », *AI Magazine*, vol. 19, n° 2, p. 79-92, 1998.
- Thangiah S. R., Shmygelska O., Mennell W., « An agent architecture for vehicle routing problems », *Proceedings of the 2001 ACM symposium on Applied computing (SAC '01)*, ACM Press, New York, NY (USA), p. 517-521, 2001.
- Wooldridge M., Jennings N. R., « Intelligent Agents : Theory and Practice », *Knowledge Engineering Review*, vol. 10, n° 2, p. 115-152, 1995.

- Zargayouna M., « Une représentation spatio-temporelle de l'environnement pour le Transport À la Demande », *Atelier : Représentation et raisonnement sur le temps et l'espace, Plate-forme AFIA 2005*, Nice (France), 2005.
- Zargayouna M., Modèle et langage de coordination pour les systèmes multi-agents ouverts. Application au problème du transport à la demande, Thèse de doctorat, Université Paris-Dauphine, Paris (France), 2007.
- Zeddini B., Modèles d'auto-organisation multi-agent pour le transport à la demande, Thèse de doctorat, Université du Havre, Le Havre (France), 2009.
- Zeddini B., Yassine A., Temani M., Ghédira K., « Collective intelligence for demand-responsive transportation systems : a self organization model », *Proceedings of the 8th international conference on New technologies in distributed systems (NOTERE'08)*, ACM Press, New York, NY (USA), p. 1-8, 2008.
- Zhu K. Q., Ong K.-L., « A reactive method for real time dynamic vehicle routing problem », *12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'00)*, IEEE Computer Society, Vancouver (Canada), p. 176-179, 2000.