
Optimisation d'un service d'autopartage de véhicules électriques

Amine Ait-Ouahmed¹, Fen Zhou¹, Didier Josselin^{2,1}

1. Laboratoire Informatique d'Avignon, Université d'Avignon, (S)FR
Agor@ntic

mohammed-amine.ait-ouahmed@univ-avignon.fr, fen.zhou@univ-avignon.fr

2. UMR ESPACE 7300 CNRS, Université d'Avignon, UNSA, (S)FR
Agor@ntic

didier.josselin@univ-avignon.fr

RÉSUMÉ. Dans un système d'autopartage dit «à un seul sens», les utilisateurs peuvent prendre une voiture dans une station et la laisser ensuite à une autre station. Ce comportement conduit généralement à une situation pour laquelle certaines stations sont pleines et d'autres vides. Le système est d'autant plus contraint que les véhicules sont électriques et qu'un temps de recharge minimal est nécessaire. Dans ce travail, nous proposons des heuristiques qui optimisent la redistribution des voitures et la gestion du service. Ces heuristiques permettent de calculer le nombre de voitures électriques, le nombre d'agents nécessaire et les différentes opérations de redistribution à réaliser pour une journée donnée. Les algorithmes proposés sont appliqués sur le réseau Auto Bleue de la ville de Nice et une cartographie est réalisée sur QuantumGIS à partir des demandes estimées.

ABSTRACT. In the so called "one way" electric carsharing system, users can take a car at a station, use and leave it at another station. This process usually leads to a situation where some stations are full while others are empty. The system is especially compelling because vehicles are electrical and a minimal charging time is required. Therefore, a balanced system requires the optimal distribution of the vehicles. In this work, we propose a heuristics algorithm that optimizes the redistribution of cars and their service management. This algorithm calculates the number of electric cars, the number of agents required and the redistribution operations to perform in a given day. The algorithms are applied to the Auto Bleue network in the surrounding of Nice (France) and a map is provided within QuantumGIS using estimated demands.

MOTS-CLÉS : autopartage de véhicules électriques, redistribution de voitures, algorithme génétique, QuantumGIS, Auto Bleue à Nice

KEYWORDS: electric carsharing, car redistribution, genetic algorithm, QuantumGIS

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEomatics conference, SAGEO 2015.

1. Introduction

L'autopartage est le partage d'une flotte de voitures entre abonnés. Derrière cette définition se cache un principe simple : l'utilisation occasionnelle en libre service d'une voiture sans en être le propriétaire. Selon une enquête nationale française (Louvet, Godillon, 2013), l'autopartage représente une solution de substitution à la voiture privée. Avant d'être abonnés, environ un tiers des ménages ne possède pas leur propre voiture, tandis que 75 % d'entre eux ne possèdent pas de voiture après la souscription. En outre, une grande proportion des répondants de l'enquête a déclaré que l'autopartage leur a permis de renoncer à l'achat d'une première voiture (34,4 % des répondants) ou d'une voiture supplémentaire de (8,8 %). La voiture est moins possédée, mais aussi moins utilisée : le nombre de kilomètres parcourus en voiture par an diminue de 41 %, suite à l'adhésion à l'autopartage. En ce qui concerne l'autopartage exploité par les autorités locales ou par des entreprises privées, nous distinguons deux catégories :

- l'autopartage "en boucle" : cette forme de partage de voiture est la plus classique car à ses débuts, l'autopartage a utilisé un système de boucle où les utilisateurs étaient obligés de restituer leur voiture à la station de départ ;
- l'autopartage "à un seul sens" : il est différent du système en boucle dans la mesure où l'utilisateur peut restituer sa voiture dans toutes les stations et pas nécessairement à la station de départ. Ce système présente des avantages pour les clients, car la restitution de la voiture est beaucoup moins contraignante et peut être effectuée dans une station proche de la destination du client. C'est ce système que nous étudions dans cet article.

Bien que l'autopartage dans un seul sens ait plusieurs avantages, il génère un problème majeur de relocalisation des véhicules. En effet, le fait que l'utilisateur ne ramène pas le véhicule dans sa station d'origine peut générer un déséquilibre dans la distribution des véhicules à travers la ville. Les opérateurs d'autopartage résolvent ce problème par l'introduction d'agents mobiles qui déplacent les véhicules entre les stations afin d'équilibrer leur répartition. Ce type de problème apparaît également dans des services similaires comme le partage de vélos (bike-sharing) (Schuijbroek *et al.*, 2013). Ces opérations peuvent représenter un coût important, d'où le besoin de les optimiser. Au cours de ces dernières années, plusieurs études ont traité des plans de déploiement du service d'autopartage et de sa gestion. (Jorge, Correia, 2013) proposent notamment une revue de la littérature assez complète à ce sujet.

Les décisions concernant le nombre de stations d'autopartage, leur localisation et la taille de la flotte des voitures représentent des éléments stratégiques du problème. Parmi les travaux d'optimisation qui se sont intéressés à ce niveau de décision, nous pouvons citer les modèles de Programmation Linéaire en Nombres Entiers (PLNE) avec deux niveaux de décision (Correia *et al.*, 2012) ou la localisation des dépôts et la sélection des tournées pour maximiser les bé-

néfices de l'organisation de l'autopartage dans un seul sens (cas de Lisbonne). D'autres auteurs (George, Xia, 2011) traitent le problème de la détermination de la taille optimale de la flotte. (Ion *et al.*, 2009) utilisent un algorithme flou basé sur un ensemble de règles pour sélectionner les stations électriques.

En ce qui concerne le niveau de décision tactique et opérationnelle, (Jorge *et al.*, 2012) testent un modèle de PLNE pour résoudre le problème de relocalisation des voitures. (Kek *et al.*, 2009) présentent un algorithme d'aide à la décision pour les opérateurs d'autopartage de Singapour. (Jorge *et al.*, 2014) comparent les relocalisations optimales calculées avec un modèle PLNE, avec deux politiques de relocalisation simulée.

Dans le domaine de l'optimisation et de l'aide à la décision pour l'autopartage à un seul sens, (Boyaci *et al.*, 2013) ont tenu compte des contraintes de recharge électrique des voitures. Dans ce travail, le problème de relocalisation des véhicules a été traité comme un problème d'échange de flux de voitures entre les stations. Les auteurs supposent qu'après chaque utilisation, les véhicules doivent se recharger pendant la même durée (2 heures) indépendamment de la distance parcourue.

Dans le cadre de notre étude, nous proposons une prise en compte plus réaliste des contraintes de recharge électrique et modélisons le problème comme un problème de routage des véhicules au lieu d'un problème de flux général, ce qui nous permet de suivre les véhicules et leurs affecter des temps de recharge correspondants aux distances parcourus. Nous tenons également compte du temps d'utilisation du véhicule par un client, d'où découle un temps de recharge proportionnel. La demande est également estimée à partir de données géographiques de densité de population et de bassins de chalandise des stations.

Après cette introduction, le reste de l'article est organisé comme suit. Nous présentons le problème d'autopartage de véhicules électriques dans un seul sens dans la section 2. Ensuite, une heuristique ainsi qu'un algorithme génétique sont proposés pour l'optimisation du problème dans la section 3. Une hypothèse pour la génération d'instances sur le site de Nice est fixée dans la section 4 et des simulations sont menées dans la section 5 pour comparer les différents algorithmes et illustrer l'effet de la relocalisation des voitures sur les coûts. Enfin, le papier est conclu dans la section 6.

2. Le modèle : définition du problème et notations

Pour définir plus formellement la relocalisation et le routage dans l'autopartage de voitures électriques dans un seul sens, nous allons préciser les données manipulées, les règles, les contraintes et les objectifs. Nous présentons aussi un petit exemple pour faciliter la compréhension du problème.

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEomatics conference, SAGEO 2015.

2.1. Les données

Les données sont composées d'un ensemble $N = \{1, 2, \dots, nb_{Stat}\}$ de stations pour le stationnement et la recharge électrique des véhicules, chaque station $i \in N$ ayant une capacité Q_i maximale de places. L'ensemble $T = \{1, 2, \dots, nb_T\}$ des périodes de temps divise le temps d'une journée de service en des périodes de temps $t \in T$ de même durée. Le temps du trajet entre les stations k et i est donné par t_{ki} . L'ensemble des demandes des clients est $D = \{d^{(kt')(it)}\}$ où $d^{(kt')(it)}$ représente un départ de la station k à t' vers la station i à t . L'ensemble $F = \{1, 2, \dots, nb_{Veh}\}$ réunit les véhicules électriques. Chaque voiture doit se recharger au minimum pendant une durée $Tr_{d_{ki}}$ après un trajet de la station k vers i . On dispose également d'un ensemble d'agents (salariés de l'opérateur du service) $E = \{1, 2, \dots, nb_{Agt}\}$ capables de déplacer un véhicule à la fois entre les stations.

2.2. Les règles et les contraintes du système

Le problème est soumis à des contraintes de temps et de ressources :

- Un client peut demander une location de voiture, caractérisée par la station et la date de départ, conjointement avec la station et la date d'arrivée ;
- Une voiture peut satisfaire des clients, être déplacée par un agent ou bien être garée dans une station de recharge ;
- Un agent peut déplacer une voiture d'une station à une autre ;
- Chaque voiture ou chaque agent peut commencer la journée dans une station et la terminer dans une autre ;
- Le nombre de voitures garées dans une station ne doit pas dépasser la capacité de la station ;
- Aucun trajet ne doit dépasser 100 km, distance qui correspond approximativement à l'autonomie des voitures électriques
- Chaque voiture doit être rechargée après chaque voyage. Le temps de recharge est proportionnel à la durée du trajet effectué (pour chaque 15 km effectué, 30 min de recharge sont nécessaires).

Une implémentation de ces contraintes avec un programme linéaire en nombres entiers est utilisée pour une résolution exacte de petites instances du problème, mais cette implémentation n'est pas détaillée dans cet article.

2.3. Objectif

Nous minimisons un coût global C obtenu par la somme pondérée de 4 objectifs.

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEomatics conference, SAGEO 2015.

$$\min \{C = \alpha \cdot C_C + \beta \cdot C_V + \gamma \cdot C_A + \delta \cdot C_D\}$$

où :

- C_C est le nombre de demandes de clients non satisfaites ;
- C_V est le nombre de voitures utilisées;
- C_A est le nombre d'agents utilisés;
- C_D est la somme des distances parcourus.

Dans cet article, nous n'introduisons pas encore d'optimisation multi-critère. La pondération des quatre critères nous permettrait en effet d'adapter la fonction d'objectifs selon les besoins. Présentement, les poids α , β , γ et δ ont été respectivement fixés à : 1000, 200, 100 et 10. Les valeurs des poids correspondent à l'importance des différents objectifs dans notre problème d'optimisation. Cet article présentant la méthode dans son ensemble, nous fixons ici arbitrairement un gradient net d'importance entre les critères : la principale priorité est la satisfaction du plus grand nombre de clients, puis vient la minimisation du nombre de voitures, du nombre d'agents et enfin de la somme des distances parcourues. D'autres scénarios de fonctions objectif orientées davantage vers l'optimisation des ressources seraient envisageables, mais ici nous nous focalisons sur une fonction objectif résolument orientée vers la qualité de service, premier critère d'un service public.

2.4. Exemple

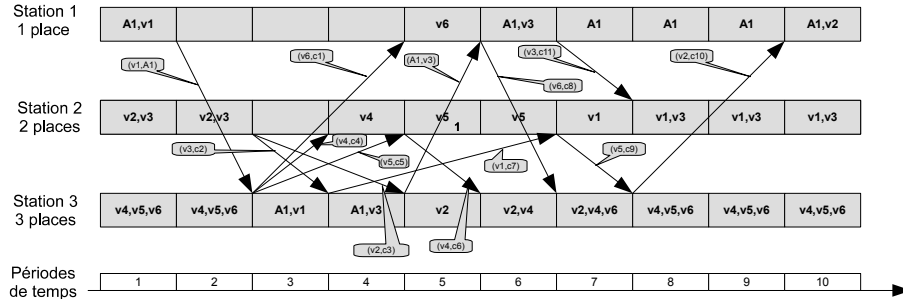


FIGURE 1. Exemple d'autopartage avec 6 véhicules et 3 stations distinctes (v = véhicule, A = agent et c = client); par exemple, $(v1, A1)$ indique que l'agent $A1$ a déplacé la voiture 1, $(v6, c1)$ indique que le client 1 a utilisé la voiture 6.

La Figure 1 décrit un schéma logistique simplifié associé à un scénario possible pour une journée de service. Dans ce scénario, le service d'autopartage est

TABLE 1. Exemple de onze instances de demandes d'utilisateurs

Clients	Départ		Arrivée	
	Station	Période de Temps	Station	Période de Temps
client 1 (c1)	3	2	1	4
client 2 (c2)	2	2	3	3
client 3 (c3)	2	2	3	4
client 4 (c4)	3	2	2	3
client 5 (c5)	3	2	2	4
client 6 (c6)	2	4	3	5
client 7 (c7)	3	3	2	6
client 8 (c8)	1	5	3	6
client 9 (c9)	2	6	3	7
client 10 (c10)	3	7	1	9
client 11 (c11)	1	6	2	7

basé sur 6 voitures électriques et 3 stations : la station 1 (1 place de parking et 1 place de recharge), la station 2 (2 parkings et 2 recharges) et la station 3 (3 parkings, 3 recharges). La journée de service est divisée en dix périodes de temps. Le temps nécessaire pour recharger une voiture après un voyage est égal à 1 période de temps et le temps du trajet entre toutes les stations est égal à 1 période de temps également. Onze demandes de clients sont caractérisées par des couples (station, période de temps) de départ et d'arrivée indiquées dans la Table 1. On peut observer que grâce aux relocalisations opérées par l'agent (A1), les 11 clients ont pu être satisfaits dans l'exemple de la Figure 1.

3. Solutions approchées et algorithmes heuristiques

Les techniques heuristiques permettent un compromis entre vitesse d'exécution et qualité de solution. Dans un premier temps, nous introduisons une nouvelle Heuristique Gloutonne (HG) permettant la construction rapide d'une solution réalisable pour le problème d'autopartage traité. En utilisant cette HG, nous définissons ensuite un Algorithme Génétique (AG).

3.1. Heuristique gloutonne

Nous développons une heuristique spécifique à notre problème, de complexité polynomiale, qui prend en compte les objectifs, les règles et les contraintes évoquées précédemment. Vue la complexité du problème traité, les expérimentations que nous avons réalisées (cf Table 2) montrent qu'il est impossible de trouver une solution exacte dans un temps raisonnable pour des instances de taille réelle (soit environ 30). Nous nous tournons donc vers des solutions approchées et efficaces. On utilise pour cela une fonction de routage pour les véhicules et les agents, dans un graphe qui permet de modéliser et d'optimiser le partage des ressources (capacité d'accueil des stations).

Chaque véhicule est ainsi associé à un chemin qui commence au début d'une journée de service dans une station et visite d'autres stations jusqu'à la fin du service. Un tel chemin est fondamentalement caractérisé par ses traces dans deux dimensions : l'espace et le temps. Il est donc naturel d'utiliser un graphe spatio-temporel $G(V, A, C, R)$ composé comme suit :

- L'ensemble des sommets $V = \{v_t^i : t \in T, i \in N\} \cup \{v_s, v_d\}$ est composé de tous les nœuds v_t^i où i représente une station et t une période de temps, les nœuds v_s et v_d représentant respectivement le début et la fin de la journée de service ;

- L'ensemble des arcs est $A = A_1 \cup A_2 \cup A_3$ où :

- $A_1 = \{(v_t^i, v_{t+1}^i), (v_s, v_i^1), (v_i^{nbT}, v_d) : i \in N, t \in T\}$. Chaque arc (v_t^i, v_{t+1}^i) représente un lien au sein de la même station entre deux périodes de temps consécutives.

- $A_2 = \{(v_{t'}^k, v_t^i) : i \in N, k \in N, t \in T, t' \in T, (t = t' + d_{ki} + Tr_{d_{ki}} \wedge i \neq k)\}$. Chaque arc $(v_{t'}^k, v_t^i)$ représente un lien entre deux stations différentes i et k à deux moments différents t' et t ; $t = t' + d_{ki} + Tr_{d_{ki}}$ assure que les voitures respectent le temps du trajet d_{ki} entre les deux stations, ainsi que le temps de recharge $Tr_{d_{ki}}$ nécessaire après le trajet.

- $A_3 = \{(v_{t'}^k, v_t^i) : i \in N, k \in N, t \in T, t' \in T, d^{(kt')}(i(t-Tr_{d_{ik}})) \in D\}$. Chaque arc $(v_{t'}^k, v_t^i)$ représente une demande de location de voiture à partir de la station k à t' vers la station i à $t - Tr_{d_{ik}}$.

- L'ensemble des coûts est $C' = C_1 \cup C_2 \cup C_3$ où :

- $C_1 = \{c_a : a \in A_1\}$; cet ensemble modélise les coûts d'immobilisation des véhicules dans une station pendant une période de temps. La valeur de chaque $c_a \in C_1$ a été fixée à 0 ; une valeur plus grande inciterait l'algorithme à déplacer les véhicules par les agents sans réel intérêt (sans augmentation du nombre de clients satisfaits).

- $C_2 = \{c_a : a \in A_2\}$; cet ensemble modélise les coûts de relocalisation des véhicules entre les stations. La valeur de chaque $c_{(v_{t'}^k, v_t^i)} \in C_2$ a été fixée à la valeur de distance d_{ki} séparant les stations k et i .

- $C_3 = \{c_a : a \in A_3\}$; cet ensemble de coûts négatifs permet de rendre attractifs certains liens dans le processus de calcul des plus courts chemins pour satisfaire les clients ; effectivement, plus un parcours de véhicule satisfait des clients, plus son coût décroît grâce aux arcs de coût négatif. Dans un premier temps, la valeur de tous les $c_a \in C_3$ a été fixée de manière homogène à -100 ; cette valeur permet de faire prévaloir l'importance de satisfaction des clients par rapport aux autres coûts. Dans la deuxième partie de cette section, la méta-heuristique se basera sur des valeurs hétérogènes de l'ensemble C_3 générées par l'heuristique gloutonne HG.

- L'ensemble des ressources (parking et lieux de recharge) est $R = \{r_{t,t+1}^i : r_{t,t+1}^i = Q_i, t \in T, i \in N\}$; ici, chaque ressource $r_{t,t+1}^i$ représente le nombre de

places disponibles dans la station i entre t et $t + 1$. Nous associons à chaque ressource $r_{t,t+1}^i$ un ensemble d'arcs : $U_{r_{t,t+1}^i} = \{(v_t^i, v_{t+1}^i), (v_{t'}^k, v_{t''}^i) : (v_t^i, v_{t+1}^i) \in A_1, (v_{t'}^k, v_{t''}^i) \in A_2 \cup A_3, (t \leq t') \wedge (t'' - Tr_{d_{ki}} \leq t)\}$, qui utilisent cette ressource.

Le principe de la méthode proposée consiste à chercher à chaque itération le meilleur chemin de véhicule qui peut être ajouté à la solution partielle sans violer les contraintes de capacités des stations. Après le calcul d'un tel chemin, on calcule un graphe résiduel en supprimant les ressources utilisées par le véhicule. Plus précisément, pour chaque véhicule, nous calculons le meilleur chemin $Path_j$ à partir du nœud source v_s vers le nœud destination v_d , en utilisant l'algorithme de Dijkstra. Une fois le chemin $Path_j$ calculé, nous mettons à jour l'ensemble A_3 par la suppression des arcs représentant les demandes des utilisateurs desservis par la voiture j . Si $Path_j$ traverse un arc appartenant à l'ensemble $U_{r_{t,t+1}^i}$, une unité de ressource est soustraite à partir de $r_{t,t+1}^i$. Une fois $R_{t,t+1}^i$ égal à zéro, nous supprimons l'ensemble des arcs $U_{r_{t,t+1}^i}$. L'heuristique est décrite dans l'Algorithme 1. Dans cet algorithme, la fonction *routeAgents* prend comme entrée les opérations de relocalisation des voitures et calcule le nombre d'agents nécessaire pour les effectuer. Les trajets des agents durant la journée de service sont tracés de manière itérative en utilisant l'algorithme de plus court chemin de Dijkstra.

3.2. Meta-heuristique

Les algorithmes génétiques (Fonseca, Fleming, 1993) représentent une méthode d'optimisation intelligente qui exploite des solutions générées aléatoirement dans l'espace de recherche et utilise des informations historiques sur les solutions générées pour diriger la recherche dans l'espace des solutions possibles. Une adaptation adéquate de l'algorithme génétique est proposée. Le fonctionnement général est introduit dans l'Algorithme 2 et ses différentes étapes sont détaillées dans le reste de cette section.

Voici les principales étapes de la méta-heuristique développée :

- Modélisation du chromosome (solution) : toute instance d'un problème traité est associée à un ensemble fini de solutions réalisables, dont chacune peut être caractérisée par une modélisation sous forme de chromosome, ce qui permet la distinction entre les différentes solutions. Dans l'algorithme génétique proposé, nous construisons toutes les solutions utilisant l'algorithme HG. Pour obtenir les différentes solutions, nous associons à chaque solution s l'ensemble des coûts négatifs C_3^S générés aléatoirement entre 0 et -100 , au lieu de l'ensemble de valeurs homogènes C_3 présenté dans la section précédente. En effet, l'ensemble de valeurs hétérogènes C_3^S permet de donner un ordre d'importance entre les différentes demandes des clients, ce qui donne des solutions différentes selon les ordres générés. Une modélisation sous forme de chromosomes de la solution s est une représentation vectorielle de l'ensemble C_3^S .

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEOMatics conference, SAGEO 2015.

Algorithme 1 : Algorithme de l'heuristique gloutonne

```

Data :  $G(V, A, C, R, U)$  ;
/* Le graphique spatio-temporel modélisant le problème */
Result :  $Paths_{Cars}$ ,  $Relocation$ ,  $SatisfiedDemands$ ,  $Paths_{Agents}$ 
1 initialization;
2  $Paths_{Cars} \leftarrow \emptyset$  /* l'ensemble de chemins qui constituent les trajectoires
   des voitures */ ;
3  $SatisfiedDemands \leftarrow \emptyset$  /* l'ensemble de demandes satisfaites */ ;
    $Relocation \leftarrow \emptyset$  /* l'ensemble de relocalisation des voitures */ ;
4  $Paths_{Agents} \leftarrow \emptyset$  /* l'ensemble de chemins qui constituent les
   trajectoires des agents */ ;
5  $j \leftarrow 1$  ;
6  $costPath_j \leftarrow 0$  ;
7 while ( $j \leq nb_{veh}$ )  $\wedge$  ( $costPath_j \leq 0$ ) do
8    $path_j \leftarrow Dijkstra(G(V, A, C, R))$  ;
9    $costPath_j \leftarrow Cost(path_j)$  ;
10  forall  $(v_{t'}^k, v_t^i) \in path$  do
11    if  $(v_{t'}^k, v_t^i) \in A_3$  then
12       $A_3 \leftarrow (v_{t'}^k, v_t^i)$  ;
13       $SatisfiedDemands \leftarrow SatisfiedDemands \cup (v_{t'}^k, v_t^i)$  ;
14    forall  $U_{r_{t'', t''+1}^i}$  do
15      if  $(v_{t'}^k, v_t^i) \in U_{r_{t'', t''+1}^i}$  then
16         $r_{t'', t''+1}^i \leftarrow r_{t'', t''+1}^i - 1$  ;
17        if  $r_{t'', t''+1}^i = 0$  then
18           $A \leftarrow A \setminus U_{r_{t'', t''+1}^i}$  ;
19    if  $(v_{t'}^k, v_t^i) \in A_2$  then
20      /*  $(v_{t'}^k, v_t^i)$  est un arc de relocalisation */
       $Relocation \leftarrow Relocation \cup (v_{t'}^k, v_t^i)_j$  ;
21     $Paths_{Cars} \leftarrow Paths_{Cars} \cup path_j$  ;
22   $j \leftarrow j + 1$  ;
23  $Paths_{Agents} \leftarrow routeAgents(Relocation)$ ;

```

– Fonction *fitness* : cette fonction d'évaluation permet la sélection ou le rejet d'un individu, pour ne garder que les individus ayant les plus bas coûts dans la population actuelle. Cette méthode garantit que les individus formant l'élite de la population seront conservés, tandis que les individus mal adaptés seront éliminés de la population. Pour calculer la valeur fitness d'une solution, nous utilisons la fonction objectif C définie dans la section 2.

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEOMatics conference, SAGEO 2015.

Algorithme 2 : Algorithme génétique

Générer des solutions aléatoires pour la population initiale :

– créer des ensembles de coûts négatifs différents (voir 3.2-Modélisation du chromosome) : $C_3^{S_0}, C_3^{S_1}, C_3^{S_2}, \dots, C_3^{S_{taillePopulation}}$ (taillePopulation= 50 individus)

– construire les solutions initiales avec HG

Évaluer la population initiale (fonction fitness)

REPEAT

– sélectionner les parents de la population précédente

– croisement des parents pour créer une nouvelle génération (taux=70%)

– effectuer la mutation sur la nouvelle population (taux de 1%)

– construire les solutions de la nouvelle population avec HG

– évaluer la nouvelle population (fonction fitness)

JUSQU'À un critère d'arrêt (temps limite = 30 min)

– Sélection : elle consiste à choisir des solutions parents pour la reproduction de la population. Dans cette procédure, les meilleures solutions sont favorisées en utilisant une sélection stochastique par roulette. Dans ce schéma de sélection, la probabilité de sélectionner un individu est proportionnelle à la valeur de sa fonction fitness.

– Opérateurs de croisement et de mutation : la création d'un nouvel individu (fils) est le produit du croisement de deux chromosomes parents via un processus d'échange de parties de chromosomes choisies aléatoirement entre les deux parents. Ce processus permet une intensification dans l'espace de recherche grâce à l'échange d'informations entre les chromosomes (individus). Pour assurer une large diversification de la population, on utilise un opérateur de mutation simple qui consiste à modifier aléatoirement la valeur d'une section du chromosome. Les taux de croisement et de mutation ont été fixés respectivement à 70 % et 1 % par rapport à la taille de la population, fixée elle à 50 individus.

4. Simulation d'autopartage : instances pour le site de Nice (Auto Bleue)

L'autopartage de la métropole de Nice (AVEM, 2015) permet de louer des véhicules en libre service à Nice et dans ses environs proches. Tous les véhicules sont électriques et disposent donc d'une autonomie limitée et nécessitent des rechargements fréquents. 64 stations sont actuellement localisées sur ce territoire par le service Auto Bleue, accessible à 4000 clients inscrits.

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEomatics conference, SAGEO 2015.

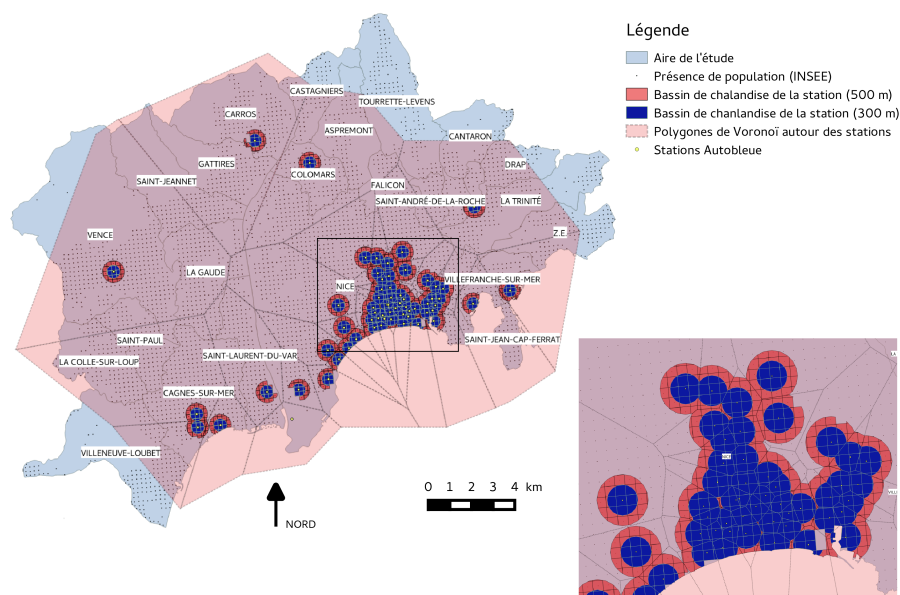


FIGURE 2. Données géographiques utilisées pour la création des instances

Il nous est actuellement impossible de disposer des informations clients sur le service Auto Bleue. De ce fait, et également afin d'étalonner et d'évaluer la méthode proposée, nous avons généré des instances aléatoires pour simuler les flux origines-destinations (FIGURE 2). Pour ce faire, nous avons tout d'abord créé une partition à partir de polygones de Voronoï, où chaque station possède une aire de chalandise bornée par celles de ses voisines. Un autre intérêt est l'obtention d'une partition spatiale où ne persiste aucun trou, ni aucune superposition d'aires de chalandise contiguës. Considérant ensuite deux distances d'accès aux stations à pied (300 et 500 mètres), nous avons réalisé des zones tampons à vol d'oiseau autour des stations, qui nous donnent une estimation suffisante des distances relatives, même si l'usage de distances sur le réseau aurait été plus précis dans ce contexte. Avec cette approche, nous simulons seulement un accès de proximité du service, faisant abstraction d'un éventuel usage de modes intermodaux (véhicule personnel, bus, etc.) pour rejoindre les stations. L'intersection des zones tampons avec les polygones de Voronoï nous fournit une zone de chalandise théorique par station. Chacune de ces aires est alors croisée avec les données de population issues du carroyage de l'INSEE (côté de 200 m). Finalement, par une requête spatiale à chaque station, nous calculons la somme des populations incluses dans l'aire de chalandise. Cette

valeur permet d'estimer une probabilité de départ et d'arrivée de véhicules d'autopartage pour une station donnée et sert de base à la création des instances simulées.

5. Résultats

Les simulations ont été effectuées sur un PC Intel Core équipé d'un CPU de 3.3GHz et 8GBytes de RAM. Les algorithmes ont été implémentés en Java.

5.1. Comparaison des différentes méthodes en fonction de la montée en charge

Pour comparer nos solutions avec les solutions optimales, nous avons utilisé le solveur CPLEX 12.2 qui se base sur une Programmation Linéaire en Nombres Entiers (PLNE) du problème. Comme la capacité de résolution du solveur se restreint à de petites instances, nous avons effectué les comparaisons sur des instances générées aléatoirement avec moins de 20 stations. En effet, nous avons pu constater que l'obtention d'une solution exacte est impossible pour des instances de plus de 30 stations simulées et 60 clients, ou 20 stations et 150 clients. Dans la Table 2 des résultats, on constate que l'heuristique gloutonne HG permet de trouver des solutions avec un gap (écart) maximal d'environ 20 % par rapport à la solution optimale et que l'algorithme génétique permet de réduire encore cet écart.

5.2. Apport de la relocalisation des véhicules sur le site de Nice

Nous avons également utilisé l'algorithme génétique pour la relocalisation des véhicules et nous l'avons comparé sur les mêmes instances à un service d'autopartage sans relocalisation sur le site de Nice (64 stations). La Figure 3 compare les performances des deux types de service sur des demandes de clients générées selon l'hypothèse fixée dans la section précédente. On commence avec une demande faible (50 clients) et on augmente la demande jusqu'à la saturation du service (700 clients). Chaque résultat est représenté dans la figure sous forme de l'intervalle de confiance autour de la moyenne au risque de 5% avec un échantillon de 10 simulations pour une instance avec la même quantité de demandes. On constate que les résultats sont dans leur ensemble discriminants. 3(a) montre l'amélioration du pourcentage de satisfaction des clients avec la relocalisation des véhicules. 3(b) montre une diminution du nombre de voitures utilisées quand la demande est faible et leur augmentation quand le service est proche de la saturation, car la relocalisation permet une plus grande exploitation de la capacité des stations. 3(c) montre une nette augmentation de la fréquence d'utilisation de chaque véhicule en cas de relocalisation. 3(d) fournit le nombre d'agents nécessaire pour effectuer les relocalisations dans les différentes instances ; ce nombre s'amortit lors de la montée en charge.

Copyright © by the paper's authors. Copying permitted for private and academic purposes. Proceedings of the Spatial Analysis and GEomatics conference, SAGEO 2015.

TABLE 2. Résultats des simulations (*NbS* = nombre de stations, *NbD* = nombre de demandes, *HG* = Heuristique Gloutonne, *AG* = Algorithme Génétique)

Instances		Comparaison des algorithmes						
NbS	NbD	Algo	Coût total <i>C</i>	Nb clients non satisfaits	Nb de voitures utilisées	Nb d'agents utilisés	Gap par rapport à l'optimal	Temps de calcul
5	30	CPLEX	4140	0	19	3	0%	6.1h
		HG	4250	0	19	4	2.17%	0.1s
		AG	4140	0	19	3	0%	22s
	40	CPLEX	7840	3	22	4	0%	6.3h
		HG	9430	5	20	3	20%	0.1 s
		AG	7950	3	22	5	1.4%	30s
	60	CPLEX	16570	11	25	5	0%	6.7h
		HG	16770	11	25	7	1.2%	0.1s
		AG	16720	11	25	6	1.026%	36s
10	40	CPLEX	5750	0	27	3	0%	+10h
		HG	6550	1	26	3	13.91%	0.3s
		AG	5760	0	27	3	0.001%	2.6m
	60	CPLEX	10350	2	35	12	0%	+10h
		HG	12810	5	34	9	21.6%	0.3s
		AG	11340	3	35	12	9.5%	2.7m
	100	CPLEX	27110	16	50	9	0%	+10h
		HG	30430	19	50	12	12.2%	0.4s
		AG	30170	19	47	15	11.2%	3.7m
20	60	CPLEX	8330	0	38	11	0%	+10h
		HG	9230	0	39	13	10.8%	1.1s
		AG	9230	0	39	13	10.8%	10.8m
	100	CPLEX	14900	1	65	8	0%	+10h
		HG	18420	4	63	17	23.6%	1.2s
		AG	16240	1	65	12	8.9%	11.5m
	150	CPLEX	17310	0	72	26	0%	+10h
		HG	20340	3	69	32	17.5%	1.2s
		AG	19820	1	74	30	14.5%	12.5m
30	60	CPLEX			Out of memory			
		HG	8790	0	37	13	-	2s
		AG	8660	0	37	12	-	27 min

6. Conclusion et perspectives

Un algorithme génétique basé sur une heuristique gloutonne efficace est proposé pour optimiser une journée de service d'autopartage de voitures électriques dans un seul sens. Notre objectif est de maximiser le nombre d'utilisateurs et de réduire au minimum le coût logistique (nombre d'agents, nombre de voitures). La pondération de chacun de ces critères est modifiable dans la fonction d'objectif et permet ainsi d'explorer potentiellement plusieurs types de solutions. Dans notre approche, nous traçons le chemin de chaque véhicule à l'aide d'un graphe spatio-temporel qui permet de modéliser toute une journée de service. À titre de comparaison, nous utilisons un solveur exact qui ne résout que de petites instances. Les résultats des simulations confirment que notre heuristique permet d'obtenir une solution proche de l'optimale dans des temps de calcul tout à fait raisonnables et parfaitement compatibles avec des systèmes d'autopartage de villes moyennes comme Nice. Cette méthode nous permet

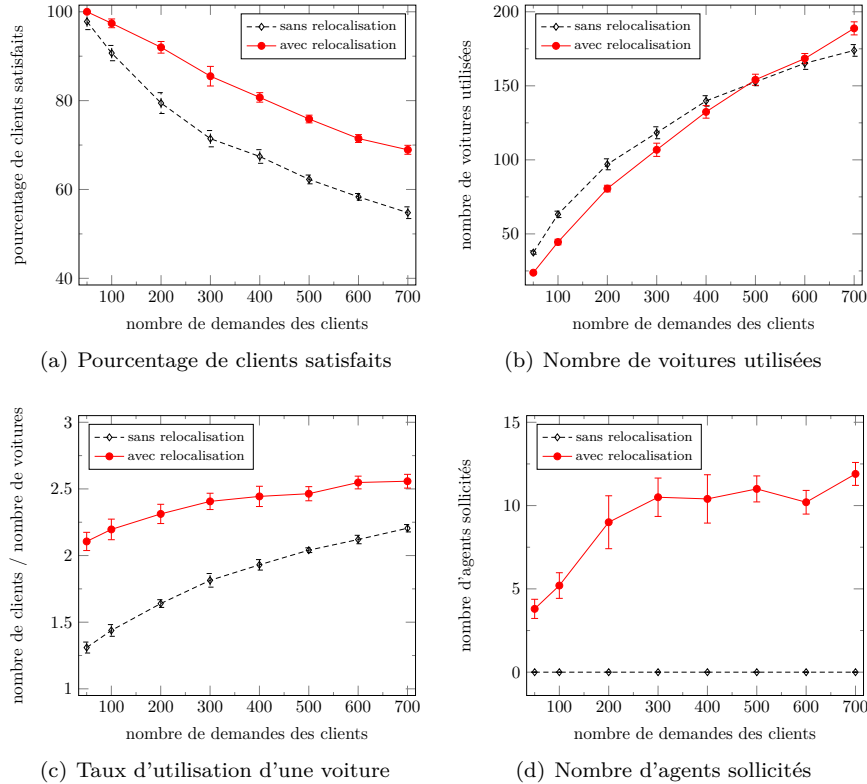


FIGURE 3. Comparaison d'autopartages avec relocalisation et sans relocalisation des véhicules sur le site de Nice; les valeurs sont représentées avec leur intervalle de confiance au risque α de 5%

également de valider l'apport de la relocalisation des voitures sur le service d'auto partage à Nice.

En termes de perspectives, un raffinement de la fonction d'objectif pour l'obtention d'un coût uniformisé, une approche multi-critère permettant de viser l'un ou l'autre des objectifs ou de rechercher des compromis, ainsi que des simulations sur des instances de taille supérieure (jusqu'à 100 demandes ou plus) afin de tester la résistance à la montée en charge des algorithmes développés, constituent des pistes intéressantes de recherche. Par ailleurs, une montée en charge jusqu'à 700 clients journaliers constitue un plafond maximal (actuellement, Auto Bleue compte environ 4000 abonnés avec une flotte de 200 voitures réparties sur une soixantaine de stations). Toutefois, la méthode proposée possède une capacité importante de scalabilité qui est actuellement testée dans un travail complémentaire.

Bibliographie

- AVEM. (2015). La voiture électrique en libre-service et en autopartage. Consulté sur http://www.avem.fr/index.php?page=libre_service_ve&cat=appli_det&id=1
- Boyaci B., Geroliminis N., Zografos K. (2013). An optimization framework for the development of efficient one-way car-sharing systems. *13th Swiss Transport Research Conference*, vol. 240, p. 718–733.
- Correia G., Homem De Almeida G., Antunes A. P. (2012). Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, n° 1, p. 233–247.
- Fonseca M. C., Fleming J. P. (1993, July). *Genetic algorithms for multiobjective optimization: formulation, discussion and generalization*. in Genetic Algorithms: Proceeding of the Fifth International Conferencen, San Mateo, CA: Morgan Kaufmann.
- George D. K., Xia C. H. (2011). Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *European Journal of Operational Research*, vol. 211, n° 1, p. 198–207. Consulté sur <http://dx.doi.org/10.1016/j.ejor.2010.12.015>
- Ion L., Cucu T., Boussier J., Teng F., Breuil D. (2009). Site selection for electric cars of a car-sharing service. *World Electric Vehicle Journal*, vol. 3, n° 1, p. 1–10.
- Jorge D., Correia G. (2013). Carsharing systems demand estimation and defined operations: A literature review. *European Journal of Transport and Infrastructure Research*, vol. 13, n° 3, p. 201–220.
- Jorge D., Correia G., Barnhart C. (2012). Testing the Validity of the MIP Approach for Locating Carsharing Stations in One-way Systems. *Procedia - Social and Behavioral Sciences*, vol. 54, p. 138–148. Consulté sur <http://dx.doi.org/10.1016/j.sbspro.2012.09.733>
- Jorge D., Correia G., Barnhart C. (2014). With Simulated Relocation Policies in One-Way Carsharing Systems. *Transactions on Intelligent Transportation Systems*, vol. 15, n° 4, p. 1667–1675.
- Kek A. G. H., Cheu R. L., Meng Q., Fung C. H. (2009). A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, n° 1, p. 149–158. Consulté sur <http://dx.doi.org/10.1016/j.tre.2008.02.008>
- Louvet N., Godillon S. (2013). Enquête nationale sur l'autopartage. *6t bureau de recherche*.
- Schuijbroek J., Hampshire R., Hoeve W.-J. van. (2013). Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems. , n° 1491. Consulté sur <http://repository.cmu.edu/tepper/1491>