

Modélisation distribuée pour une résolution dynamique du problème de transport à la demande

Moncef Temani, Besma Zeddini, Adnan Yassine

Laboratoire LMAH
Université des sciences et techniques
76058 le Havre Cedex
FRANCE
moncef.temani@fst.rnu.tn, besmile.zed@gmail.com, adnan.yassine@univ-lehavre.fr

RÉSUMÉ. Dans les systèmes de transports à la demande (TAD), l'offre de transport doit s'adapter à la demande. Dans cet article, nous nous intéressons à la modélisation des systèmes de TAD en tant que systèmes multi-agents (SMA) et proposons un modèle d'auto-organisation afin de maintenir une bonne répartition des véhicules sur le réseau de transport dans un système de TAD purement dynamique, i.e. où aucun client n'est connu avant le démarrage de l'exécution. Nous fondons notre proposition sur une approche décentralisée et une modélisation particulière de l'environnement dans un SMA. Notre modèle permet aux agents véhicules, face à l'apparition d'un nouveau client, de prendre leur décision grâce à une négociation qu'ils entreprennent entre eux, et de calculer le surcoût engendré par l'insertion du client, et ce suivant une heuristique d'insertion originale.

ABSTRACT. In Demand responsive transportation systems (DRTS), the offer must be adapted to the demand. In this paper, we are interested in modeling DRTS as multiagent systems (MAS) and we are proposing an auto-organization model in order to maintain a good dispatching of the vehicles on the network in a purely dynamic DRTS, i.e. where no customers are known before the execution starts. We ground our proposal on a decentralized approach and a particular modeling of the environment in a MAS. Our model allows for Vehicle agents, when a new customer appears, to take their decisions thanks to a negotiation that they launch, and to calculate the incurred overcost by the insertion of the customer, done following an original insertion heuristic.

MOTS-CLÉS : Transport à la demande, auto-organisation, réponse à seuil, intelligence collective, agents réactifs.

KEYWORDS : Transport on demand, self-organization, response threshold, collective intelligence, reactive agents.

1. Introduction

Les nuisances environnementales liées à l'excès de déplacements automobiles (émissions polluantes : CO, NOx et particules, etc.), conjuguées aux nuisances citées, le plus souvent, par les citadins, en zone urbaine (bruit généré par les transports, insécurité routière, etc.) rendent obligatoire pour les villes de revoir leurs systèmes d'organisation du transport afin de réduire à la fois le nombre de kilomètres produits par l'automobile utilisée en tant que moyen individuel de déplacement et le volume de véhicules en circulation dans un espace donné à un instant donné. Ceci nécessite de revoir l'usage des modes de transport en fonction d'une hiérarchie dans le repère espace-temps alors qu'aujourd'hui le réflexe mobilité est d'abord automobile. Dans ce contexte on parle de systèmes de transport à la demande (TAD) dont l'objectif est de satisfaire à des demandes de transport des personnes avec le minimum de coût. Dans la littérature, les TAD font références au Dial A ride Problem (DARP) qui dérive du problème générale de construction de tournées des véhicules connu sous le nom de Vehicle Routing Problem (VRP). Le VRP représente un problème d'optimisation combinatoire multi-objectif, il a fait l'objet de nombreux travaux et de nombreuses variantes dans la littérature [4, 9]. Il appartient à la catégorie NP-difficile [3]. Un état de l'art complet des problèmes de VRP dans un contexte statique, en particulier, dynamique (DVRP) et de leurs applications est donné dans [7]. Quand il s'agit du problème de VRP avec des collectes et livraison de marchandises on parle de PDP (Pickup & Delivery Problem ou Problème de Ramassage & Livraison). Un DARP (Dial-a-Ride Problem) est une extension du PDP dans lequel le transport de marchandises est remplacé par le transport des personnes [5]. Puisqu'il s'agit d'un problème de transport de personnes, le DARP se focalise plus sur la satisfaction de ces personnes.

Le DARP se décline à la fois de façon statique et dynamique. Souvent, ce cloisonnement n'est pas toujours strict, car par exemple, dans le cas statique, des réservations peuvent être annulées au dernier moment, ce qui implique un certain degré de dynamisme tandis que dans le cas dynamique, beaucoup de réservations sont connues à priori avant la planification [1]. Le problème est alors en général traité comme une succession de sous problèmes statiques [4]. Une application dans le contexte des transports urbains concerne le bus "à la demande" : les clients peuvent donner une fenêtre de temps au lieu de départ ou au lieu d'arrivée, mais pas les deux en même temps. Des demandes sont connues par avance et d'autres peuvent arriver pendant le jour même. Les auteurs ont adopté un algorithme d'insertion qui a donné des résultats intéressants avec des durées d'exécution courtes [9]. Ce problème est connu sous le nom de DDARP : Dynamic Dial-A-Ride Problem dans un contexte Many-to-Many (i.e plusieurs lieux de collectes et de dépôts). Dans [8], le problème de DARP a été traité en ligne en considérant une flotte de véhicules homogène de capacité unitaire, i.e un véhicule ayant un passager à bord, ne peut servir un autre passager que s'il est arrivé à la destination du premier.

Le problème de DARP dynamique est étudié d'une façon marginale faute d'applications et il n'existe pas, à notre connaissance, en version purement dynamique. De même, les TAD actuels sont déterministes et peu adaptatifs à une demande très changeante. Dans de tels systèmes, c'est la demande qui s'adapte à l'offre. Le problème qui se pose dans cet article est original et innovant, du fait que nous traitons un cas purement dynamique : les demandes sont traitées en temps réel. Cette dynamique vient du fait qu'aucune réservation n'est connue à l'avance (la veille ou même avant). En outre, les techniques classiques supposent d'avoir un central qui connaît en permanence la position, les trajectoires et les dessertes prévues de toute la flotte (le central reçoit la demande, choisit le transporteur et

l'informe de sa nouvelle trajectoire). De plus, ces techniques supposent une connaissance parfaite par le central de l'actualité des véhicules (y compris pauses, pannes, problèmes de communication . . .), ce qui est peu réaliste. Ajoutons que ce fait peut entraîner des calculs coûteux en temps. Ce qui nous pousse à adopter une méthode de résolution décentralisée pour faire face à ce genre de problèmes.

Dans cet article, nous proposons un système de transport à la demande qui doit à l'opposé adapter l'offre, en temps réel, à la demande. Il doit permettre à terme de répondre à de nombreuses exigences difficiles à concilier : qualité de service, minimisation des coûts, impact environnemental et sociétal. Enfin, il constituera un outil d'aide à la décision permettant de prendre en compte dynamiquement et en temps réel des problèmes de construction de tournées de véhicules, ce qui n'est pas offert par les systèmes de transport existants. L'enjeu de cet article est d'étudier la possibilité de développer des méthodes robustes et réactives qui permettent de faire face rapidement aux aléas et de prendre en compte l'arrivée de nouvelles informations (ajout d'un nouveau client dans une tournée de véhicule de transport à la demande etc.). Le scénario d'exécution commence avec la première demande d'un usager qui apparaît aléatoirement dans un endroit de la ville. Il envoie une requête en désignant son point de départ et son point d'arrivée. La résolution consiste à choisir le véhicule le mieux placé pour satisfaire les demandes de déplacement des passagers déjà existants dans ce véhicule ainsi que la nouvelle demande ; et ceci en optimisant son taux de remplissage, son temps et coût de déplacement.

Le reste de ce papier est organisé comme suit : nous présenterons notre approche dédiée au problème étudié. Puis, nous concluons sur les résultats et les perspectives possibles d'amélioration de l'approche.

2. Modèle dynamique proposé

Les techniques de l'intelligence en essaim démontrent toute leur puissance dans les cas très dynamiques, c'est à dire des problèmes dont les données ou les paramètres varient en permanence et où on a besoin d'un recalcul en temps réel [2]. Considérer un collectif soulève alors le problème d'organiser les diverses activités des agents dans le but que ce collectif se comporte dans son ensemble comme un tout cohérent. Dans notre travail, l'organisation est obtenue selon une approche dynamique décentralisée et d'une façon réactive. Notre modèle est basé sur un système constitué d'agents réactifs, dans lequel l'importance des interactions excède celle de l'architecture des agents, ces agents n'ont pas ou peu de représentation des autres, possèdent des mécanismes de décision de type stimulus-réponse [2] et interagissent via leur environnement avec les autres.

Nous proposons une approche agentifiée. Le système est composé des agents suivants : agent Véhicule, agent Client, agent Interface, agent Requête de transport, agent Courtier et agent Ordonnanceur. Nous représentons le réseau urbain par un graphe dynamique $G(t) = (V(t), E(t))$ où $V(t)$ est l'ensemble de noeuds où chacun peut bien désigner un carrefour, un point de départ ou d'arrivée d'un client, un établissement comme une école, une gare, un cinémas, un centre commercial, etc. et $E(t)$ l'ensemble des arcs possibles de la ville. Dans notre problème de transport à la demande, le poids sur un arc correspond au temps nécessaire pour traverser cet arc. La dynamique porte sur la valuation des arcs, qui est susceptible d'évoluer dans le temps, en fonction de l'évolution de la fluidité de la circulation par exemple. Les clients sont attribués à un noeud ou à un véhicule. Ces derniers sont attachés à des noeuds ou des arcs.

3. Architecture du système

Pour l'architecture du système nous avons exploré trois possibilités :

(i) Architecture centralisée : L'ensemble des demandes sont recueillies dans un même centre et attribuées aux véhicules appropriés (i.e. envoi de la requête à un véhicule) (ii) Architecture décentralisée : Envoi de la requête à tous les véhicules (iii) Architecture semi centralisée : Si l'approche centralisée mentionnée ci-dessus vise essentiellement l'optimisation globale du système, et si l'approche décentralisée vise une meilleure souplesse et adaptabilité, l'approche semi décentralisée tente d'être à mi-chemin entre les deux.

Pour notre problème, Nous avons choisi une architecture décentralisée parce qu'elle permet d'adapter le système en fonction de ses besoins dynamiques dans un environnement en changement perpétuel.

3.1. Architecture décentralisée

- Architecture décentralisée : Ce qui diffère dans notre modèle des systèmes de transport à la demande (TAD) dynamiques existants, c'est que les demandes ne sont pas traitées par lot mais en "temps réel". L'objectif de cette approche est de réaliser une adaptabilité globale capable de maintenir une bonne répartition des véhicules sur l'ensemble de la zone à couvrir. Dans cette approche, l'agent interface joue le rôle de coordinateur entre les clients et les véhicules. Étant donnée que le traitement est trop lourd dans un contexte strictement dynamique et que l'efficacité du système décroîtrait d'une façon très rapide. La solution est donc de décentraliser l'interaction, mais d'avoir la possibilité néanmoins d'effectuer un prétraitement sur les requêtes provenant des agents Clients. Ceci ne peut être réalisé par un agent particulier à la manière des agents intermédiaires [11] (dans notre système on pourrait penser à l'agent Interface et à l'agent Courtier). L'agent Interface n'aurait pas à maintenir des accointances, et toutefois ne diffuse pas ses requêtes à tous les agents. Seuls les véhicules sélectionnés par l'agent courtier et dont le profil correspond à la requête seront notifiés (i.e. ceux dont la zone de couverture ou le sphère de faisabilité comporte ces noeuds par exemple).

3.1.1. Processus Dynamique

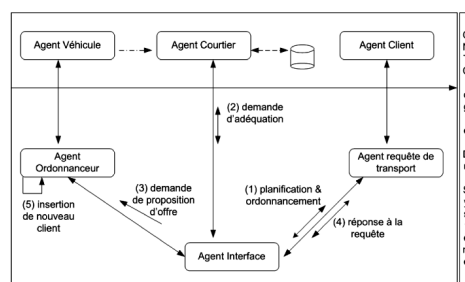


Figure 1. Processus d'interaction

Le processus dynamique de planification est comme suit (Voir la figure 1) : les Agents Ordonnanceurs sont créés par les agents véhicules quand ces derniers annoncent leurs disponibilités à l'agent Courtier. Un utilisateur envoie une demande de transport par l'intermédiaire de l'agent Client, qui génère l'agent Requête de transport. Il désigne son point de départ et son point d'arrivée comme indiqué précédemment. Ces informations seront

traduites en des données convenables à l'ontologie du service. (1) Tout agent requête de transport demande à l'agent Interface de traiter la requête. L'agent Interface traite la requête (2) en premier lieu par obtenir, à travers l'agent Courtier, la liste des véhicules qui correspondent au profil recherché, puis (3) en faisant un appel de propositions d'offre à l'ensemble des Agents Ordonnanceurs. Ces derniers calculent chacun son surcoût ΔC ou effort supplémentaire. Pour connaître cet effort supplémentaire, on calcule la longueur (en temps) de l'itinéraire actuel (le coût) d'une part, celle de l'itinéraire pour décharger les actuels passagers plus la prise et la desserte de l'usager ayant fait une demande d'une autre part, et on soustrait la première à la seconde. Cette différence est l'effort supplémentaire ou surcoût. (4) et lance un processus de négociation avec les autres agents Ordonnanceur, en s'échangeant les propositions. (5) Chaque agent Ordonnanceur ordonne les réponses reçues et une fois ils arrivent à faire un compromis, le vainqueur ajoute le nouveau client dans sa tournée actuelle et indique aux restes des agents Ordonnanceurs et à l'agent Interface qu'il a remporté la demande. Enfin, l'agent Interface répond au client en lui envoyant l'identificateur du véhicule qui le prendra en charge.

4. Algorithme d'ordonnancement

Les agents Ordonnanceurs effectuent le travail principal de planification par évaluation de l'insertion d'un voyage (source et destination) de telle façon à respecter le plus possible les échéances des passagers existants dans les véhicules. La politique d'insertion est similaire à ADARTW [6]. Pour chaque agent Ordonnanceur, un bloc d'ordonnancement commence toujours par le premier client sur son chemin et se termine quand le dernier client arrive à destination. La figure suivante illustre l'ajout d'un client dans un bloc d'ordonnancement d'un véhicule contenant déjà trois clients ($C1, C2, C3$) dont l'itinéraire est déjà planifié. Chacun a un lieu de départ (+) et de destination (-). A un bloc relatif à un

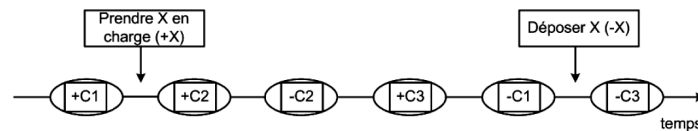


Figure 2. Algorithme d'ordonnancement

véhicule et contenant déjà n clients, correspond $k = 2n$ arrêts (prise en charge et dépôt) et il y a $(k + 1)(k + 2)/2$ possibilités d'insertion en admettant qu'une prise en charge d'un client doit précéder son dépôt. Dans l'exemple précédent, on a $(6 + 1)(6 + 2)/2$, soit 28 possibilités d'insérer le client X . Pour un véhicule V ayant une capacité maximale Q_v , le nombre de possibilités $nbPossibil$ d'insertion d'un nouveau client au pire des cas s'élève à $((Q_v - 1) * 2 + 1)((Q_v - 1) * 2 + 2)/2 = Q_v(2Q_v - 1)$. A part le nombre d'insertions possibles pour un nouveau client, la complexité dépend aussi du calcul d'itinéraires entre les noeuds (origines et destinations) des clients formant le bloc d'un véhicule. Dans le pire cas, la complexité d'insertion d'une nouvelle demande s'élève à $nbPossibil * (2Q_v - 1) * \Theta(m + n \log(n)) = Q_v(2Q_v - 1)^2 * \Theta(m + n \log(n))$; n et m étant le nombre de noeuds et des arcs du graphe et $\Theta(m + n \log(n))$ la complexité de l'algorithme de Dijkstra pour le calcul du plus court chemin dans un graphe. La fonction objectif $MinZ = \Delta C$ mesure le surcoût minimal dû à l'insertion d'un nouveau client. Pour le terme ΔC , plusieurs variables ont été considérées : temps supplémentaire, capa-

cit   actuelle, proximit   du client. Si deux ou plusieurs v  hicules ont la m  me valeur du surco  t pour un m  me client    un instant t , celui qui a la capacit   minimale doit prendre en charge ce client. S'ils ont aussi la m  me capacit  , celui qui est plus proche (en distance) va ramener le client.

5. Auto-organisation

La nature dynamique et non d  terministe du probl  me peut conduire    des concentrations de l'offre dans certaines zones plus attractives et de ce fait    la cr  ation de zones de famines. En effet le centre-ville peut   tre, par exemple, une zone d'attraction forte    certaines heures de la journ  e alors que certaines zones p  riurbaines d  tes dortoirs seront d  laiss  es. Le temps d'attente dans une telle zone sera alors tr  s important, nous avons donc choisi d'  ventuellement d  grader les performances dans les zones attractives et d'avoir une meilleure couverture du domaine par la flotte de v  hicules afin d'  viter des zones orphelines.

Plusieurs solutions sont envisageables : (i) un surdimensionnement de la flotte de v  hicules mais cela va, entre autre,    l'encontre des objectifs environnementaux fix  s et la volont   d'avoir le meilleur taux de remplissage possible des v  hicules ; (ii) une r  solution centralis  e hi  rarchis  e, ceci s'oppose au mod  le d  centralis   retenu ; (iii) l'utilisation de m  canismes d'auto-organisation. C'est cette derni  re voie que nous avons choisie pour ses caract  ristiques distribu  es, locales et adaptatives.

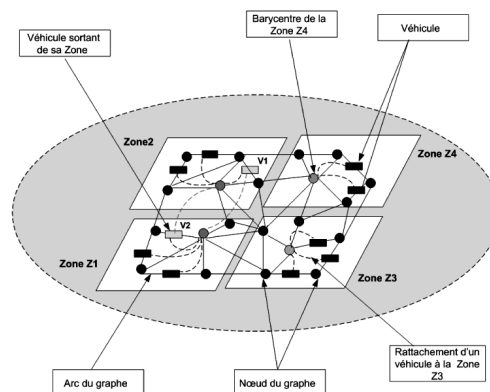


Figure 3. Sp  cialisation et zones d'attraction

On d  finit donc des zones d'attraction au sein du graphe auxquelles les v  hicules sont rattach  s, ces zones peuvent   voluer par la suite soit au niveau de leur couverture du graphe, soit au niveau du nombre de v  hicules rattach  s. Lorsqu'un v  hicule quitte sa zone il est p  nalis   dans une fonction de co  t qui joue comme une force de rappel exerc  e par un ressort accroch      sa zone (voir figure 2). Si le v  hicule n'est plus dans la couverture g  ographique de sa zone, il a la possibilit   de n  gocier un changement de zone de rattachement avec d'autres. Dans la figure 2, les v  hicules V1 et V2   changent de zones. Ainsi, on maintient une r  partition spatiale de l'offre, conduisant    am  liorer globalement le volume de v  hicules attach  s    des zones. Cette partie de la mod  lisation est en cours d'impl  mentation

6. Simulation et résultats

L'architecture multi-agent proposée a été développée en utilisant la plateforme d'agent REPAST Symphony (Recursive Porus Agent Simulation Toolkit) écrite en Java [10]. En plus de modèle décentralisé, nous avons développé, les deux autres modèles présentés ci-avant (voir section 3), il s'agit du modèle centralisé et du modèle semi-décentralisé. Pour chaque modèle, le même algorithme d'ordonnancement a été utilisé pour l'ensemble des véhicules, mais le processus de fonctionnement des agents est modifié :

- **Modèle centralisé** : Dans ce modèle, l'agent interface est considéré comme un central qui connaît en permanence la position et les trajectoires et dessertes prévues de toute la flotte. L'agent interface reçoit la requête du client, choisit le véhicule le plus approprié et l'informe de sa nouvelle trajectoire.

- **Modèle semi-décentralisé** : ce modèle tire profit du rôle de l'agent interface qui sélectionne l'offre la plus appropriée parmi les propositions reçues par les agents Ordonnanceurs en appliquant des politiques de filtrage. Le reste de la procédure est similaire au modèle décentralisé expliqué ci-dessus.

6.1. Étude comparative

Nous avons effectué les simulations des trois modèles sur un graphe à 50 sommets, sur 7200 pas de temps, avec des véhicules à 4 places. Les clients apparaissent à des lieux et heures aléatoires, avec des sources et des destinations aléatoires. Dans le tableau ci-dessus

Paramètres / Approches	Nb véhicules	Nb clients	Coût optimal en temps	Coût réel en temps	Écart
Décentralisée	4	45	7334	7932	8,15%
	4	225	4419	4888	10,61%
	8	200	16076	17253	7,32%
	8	350	26755	29391	9,85%
Semi-décentralisée	4	45	6038	6589	9,1%
	4	225	5022	5610	11,7%
	8	200	16984	18419	8,4%
	8	350	31761	35001	10,2%
Centralisée	4	45	7246	7987	10,22%
	4	225	5434	6139	12,97%
	8	200	15130	16542	9,33%
	8	350	28681	31968	11,46%

Figure 4. Quelques résultats de la simulation

(voir table 1), la colonne coût optimal indique le temps total minimal obtenu à l'aide du plus court chemin pour emmener tous les clients de leur lieu de prise en charge jusqu'à leur destination. La colonne Coût réel indique le temps effectivement mis en prenant en compte la contrainte de remplissage. Écart est le pourcentage de différence entre les deux coûts. Les tests réalisés sur les trois modèles ont montré que le modèle semi décentralisé est capable de fournir des résultats meilleurs que le modèle centralisé et que notre modèle (décentralisé) a donné de meilleurs résultats par rapport au modèle semi décentralisé et au modèle centralisé. Concernant les résultats obtenus avec le modèle décentralisé, les écarts entre le coût réel et le coût optimal ne sont pas importants si on prend en compte le temps nécessaire pour garer un véhicule dans le cas réel et le prix moins coûteux proposé aux clients dans le transport collectif. Nous avons remarqué aussi que pour un nombre donné de véhicules et à une période donnée de la simulation (7200 pas de temps), notre modèle donne de meilleurs résultats avec un nombre limité de clients ayant des itinéraires longs qu'avec un grand nombre de clients ayant des itinéraires courts. Ceci est expliqué par le temps supplémentaire dû à la satisfaction de tous les clients, et qui est plus important

dans le cas d'un grand nombre de clients. On remarque que les résultats sont très encourageants. La simulation doit être raffinée et le mécanisme d'auto-organisation est en cours de développement.

7. Conclusions et perspectives

Dans cet article, nous avons présenté un système de transport à la demande purement dynamique dans un environnement en changement perpétuel. Pour cela, nous avons adopté une approche décentralisée basée sur l'optimisation et la négociation entre les véhicules. Pour faire face aux zones de famine ou, au contraire, de concentration de véhicules dans certaines régions, nous avons proposé un modèle d'auto-organisation pouvant s'adapter aux changements environnementaux. Les perspectives de notre travail sont de disposer d'un Système d'Information Géographique (SIG) pour une meilleure représentation spatiale précise et concrète de la ville. En plus, nous pouvons bénéficier de ce modèle afin d'intégrer un modèle probabiliste plus fin pour la population d'utilisateur, meilleures statistiques pour le remplissage etc. Enfin, Les résultats sont encourageants et il nous reste à compléter notre travail par la validation complète du modèle proposé.

8. Bibliographie

- [1] A. ATTANASIO, J. F. CORDEAU, G. GHIANI, G. LAPORTE « Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. », *Parallel Computing*, vol. 30, 2004.
- [2] E. BONABEAU, E. DORIGO, G. THÉRAULAZ, « Swarm Intelligence : from natural to artificial systems. », *Oxford university press*, 1999.
- [3] N. CHRISTOFIDES, A. MINGOZZI, P. TOTH, « The vehicle routing problem. », *John Wiley*, 1979.
- [4] J. F. CORDEAU, G. LAPORTE, J. Y. POTVIN, M. W. SAVELSBERGH, « Transportation on demand », *Centre de Recherche sur les transports (CRT), Montréal*, 2004.
- [5] D. HAUPTMEIER, S. O. KRUMKE, J. RAMBAU, « The online dial-a-ride problem under reasonable load », *In Proceedings of the 4th Italian Conference on Algorithms and Complexity, Springer*, 2000.
- [6] J. J. JAW, A. R. ODoni, H. N. PSARAFTIS, N. H. M. WILSON, « A heuristic algorithm for the multi-vehicle many-to-many advance request dial-a-ride problem with time windows », *Pergamon Journals Ltd.*, vol. 20B, 1986.
- [7] A. LARSEN, « The dynamic vehicle routing problem », *Technical University of Denmark*, 2000.
- [8] M. LIPMANN, X. LU, W. E. DE PAEPE, R. A. SITTERS, L. STOUGIE, « On-line dial-a-ride problems under a restricted information model », *Algorithmica, Springer*, 2004.
- [9] O. B. G. MADSEN, K. TOSTI, J. VAELDS, « A heuristic method for dispatching repair men », *Annals of operations research*, vol. 61, 1995.
- [10] M. J. NORTH, T. R. HOWE, N. T. COLLIER, R. J. VOS, « The repast symphony runtime system », *Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms*, 2005.
- [11] K. SYCARA, H. C. WONG, « A taxonomy of middle-agents for the internet, robotics institute carnegie mellon », *Proceedings of the Fourth International Conference on MultiAgent Systems*, 2000.

8.1. Biographie

M. Moncef Temani est actuellement directeur de l'Institut Supérieur d'Informatique de Tunis. Il a des activités qui s'articulent autour de l'intelligence Artificielle, en particuliers les SMA : systèmes Multi-Agents, les problèmes liés à la logistique et la productique (ordonnancement, VRP, dial-a-ride problem), et il s'intéresse, également, au domaine du Web service et sécurité.

Melle. Besma Zeddini est doctorante à l'université des sciences et techniques du havre-France, au sein du laboratoire LMAH. ses activités concernent les systèmes multi-agents, les problèmes liés au transport et à la logistique, la planification et la gestion des transports à la demande Temps réels et l'optimisation multi-critères.

M. Adnan Yassine est directeur du laboratoire de Mathématique Appliquée du Havre (LMAH) en France, ses activités d'enseignement et de recherche sont centrées sur l'analyse numérique (algorithmique numérique, calcul scientifique, programmation mathématique), analyse convexe et non convexe, optimisation convexe et non convexe (différentiable et non différentiable), optimisation globale, optimisation financière et la recherche opérationnelle (optimisation combinatoire, optimisation dans les réseaux, ordonnancement et théorie de graphes).

