
Patrouille multi-agent dynamique, application robotique pour le service de personnes mobiles

Jacques Saraydaryan^{1,2}, Fabrice Jumel^{1,2}, Olivier Simonin^{1,3}

1. Laboratoire CITI-Inria, équipe Chroma, INSA de Lyon,
Inria Grenoble Rhône-Alpes
2. CPE Lyon, Domaine scientifique de la Doua, 69100 Villeurbanne, France
jacques.saraydaryan@cpe.fr, fabrice.jumel@cpe.fr
3. INSA Lyon, Université de Lyon, 20 Avenue Albert Einstein, 69100 Villeurbanne, Fr.
olivier.simonin@insa-lyon.fr

RÉSUMÉ. Dans le contexte de la robotique de service, déployer une flotte de robots mobiles au service d'humains qui évoluent dans de vastes espaces est un défi important. Nous nous intéressons ici au problème de servir régulièrement et équitablement des personnes se déplaçant librement avec un nombre limité de robots. Nous montrons qu'il s'agit d'un problème de patrouille multi-agent dynamique, pour lequel nous formalisons des critères de performance. Nous proposons et comparons quatre stratégies de patrouille, deux étant fondées sur des solutions standard de la patrouille multi-agent, deux autres prenant en compte la mobilité et l'oisiveté des personnes à servir. Nous présentons un simulateur combinant un modèle de piétons (PedSim) et un environnement robotique. Les stratégies sont évaluées et comparées, en faisant varier les paramètres du problème, comme le nombre de robots, la topologie de l'environnement et la dispersion des individus dans l'environnement, montrant l'efficacité des nouvelles approches.

ABSTRACT. In this paper, we address the challenge of serving people by a set of mobile robots. As people move we can define the problem as a dynamic multi-agent patrolling. We propose different metrics by considering not only the time to patrol all the people but also the equity of the service delivery. We propose and compare four algorithms, two are based on standard solutions to multi-agent patrolling and two are defined according to the mobility and idleness of the persons. We present a simulator combining a pedestrian model and a robotic model. Strategies are compared on problem settings changing the number of robots, the topology of the environment and the people dispersion. Results show the efficiency of the new approaches.

MOTS-CLÉS : patrouille multi-agent, robotique de service, simulation, environnement peuplé.

KEYWORDS: multi-agent patrolling, service robotics, simulation, populated environment.

DOI:10.3166/RIA.31.379-400 © 2017 Lavoisier

1. Introduction

Dans cet article, nous visons à définir le comportement de robots mobiles chargés de fournir des services à des personnes en déplacement. Ces services peuvent concerner le transport d’objets ou de nourritures, la prise d’informations auprès de personnes évoluant dans de vastes environnements (comme un hôpital, des salles de réception) ou encore des environnements dégradés après une catastrophe naturelle ou industrielle. Dans ce papier, nous illustrons nos travaux sur un contexte de **robots serveurs**, chargés de délivrer, par exemple, des mets à des personnes évoluant dans un environnement de type salles de réception/musée, cf. illustration figure 1.

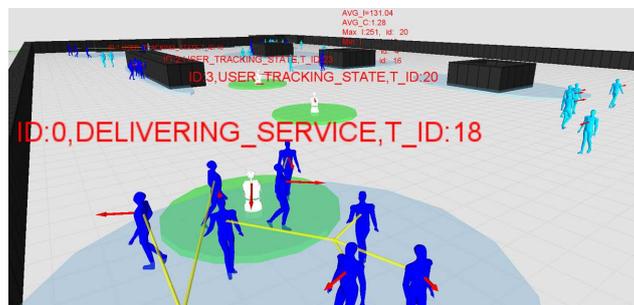


Figure 1. Simulation sous Rviz de la patrouille dynamique

Organiser la livraison de services (par des agents/robots) est traditionnellement traité comme un problème d’optimisation des distances à parcourir. Mais notre contexte se différencie fortement par la mobilité des “cibles” à servir, car les trajectoires des individus ne sont pas connues/prévisibles a priori. Cependant, demander à des robots de visiter aussi souvent que possible des personnes en déplacement est un problème proche de la patrouille multi-agent. Celle-ci concerne la visite régulière d’espaces connus (Chevalyere, 2007) ou inconnus (Glad *et al.*, 2009) par un ensemble d’agents/robots coopératifs. Les variantes dépendent de la présence d’obstacles dynamiques, d’intrus à intercepter, ou si le système est ouvert (Poulet *et al.*, 2012).

Nous proposons de définir une variante *dynamique* de la patrouille, où les “lieux” à visiter sont les personnes en déplacement. Nous nommons cette variante le problème des “robots serveurs”. Il s’agit de répondre simultanément à deux contraintes : comment minimiser le temps entre les visites consécutives de chaque personne et comment assurer une équité de service sur l’ensemble des personnes ? Dans cet article nous visons à définir et comparer des solutions répondant à ces contraintes. Nous considérons que les robots disposent en continu de deux informations sur les personnes à servir : leur position et le temps écoulé depuis leur dernière visite (nommé oisiveté dans la patrouille).

Nous explorons trois types d'approches pour guider les robots dans l'environnement (vers les personnes à servir). La première se focalise sur la couverture de l'espace, comme dans la patrouille standard, sans tenir compte des informations courantes sur les personnes. La deuxième exploite les informations d'oisiveté (à minimiser). Enfin, la troisième étend cette dernière en tenant compte des positions des personnes et en particulier de leur organisation possible en groupes.

L'article est organisé comme suit. En section 2 nous évoquons les travaux référencés sur la patrouille multi-agent classique et nous discutons des différences avec notre problématique. Puis en section 3 nous définissons formellement le problème de la patrouille dynamique. La section 4 est dédiée à la définition d'algorithmes fondés sur des approches standard de la patrouille statique ainsi qu'à des solutions tenant compte de la mobilité (en groupe) des personnes. La section 5 présente un outil de simulation et les scénarios nous permettant de comparer les stratégies. Nous analysons l'influence de paramètres du problème comme le nombre de robots, le type d'environnement et la dispersion des individus. Enfin, en section 6, nous concluons et présentons les perspectives de ce travail.

2. Etat de l'art

Le problème de la patrouille multi-agent est défini comme la visite répétée d'un espace par une flotte d'agents/robots, ayant généralement des caractéristiques identiques (Chevalyre, 2007). Dans le cas des robots mobiles (considérés ici), les zones à visiter correspondent à des points d'intérêts. Le coût du déplacement entre ces points correspond à la distance (ou temps de navigation) les séparant. Une représentation standard du problème est un graphe où les noeuds représentent les zones à visiter et les arrêtes les chemins les reliant et valués de leur coût. L'objectif usuel est de minimiser le temps entre deux visites successives d'un même point d'intérêt, cette durée est appelée **oisiveté** (Elmaliach *et al.*, 2010).

Dans le cas d'un unique robot, la solution optimale, si elle existe, est un cycle hamiltonien du graphe (c'est-à-dire passant une et une seule fois par chaque noeud). Dans le cas d'une flotte de robots, une solution optimale est de faire parcourir ce cycle hamiltonien par tous les robots en les séparant d'une distance maximum (Portugal, Rocha, 2011). D'autres algorithmes de patrouilles ont été proposés comme le partitionnement en sous graphes (hamiltoniens) patrouillés par des sous-ensembles de la flotte (Glad *et al.*, 2009)(Pasqualetti *et al.*, 2010).

Le problème de la patrouille est modélisé par un graphe car l'environnement à visiter est supposé statique ou quasi-statique. En effet, les seuls éléments dynamiques généralement considérés sont la présence éventuelle d'intrus sur les noeuds (Portugal, Rocha, 2011). Cependant, en condition réelle, l'hypothèse d'un environnement statique est peu réaliste. C'est pourquoi différentes variantes ont été proposées pour étendre l'énoncé initial. Ces variantes concernent le graphe/la carte à patrouiller. En effet, la connectivité des points d'intérêts peut changer au cours du temps en raison de la présence d'obstacles dynamiques, comme des personnes, des portes ou des objets

déplacés (Poulet *et al.*, 2012)(Aaron *et al.*, 2014). Ces modèles se limitent toutefois à activer/désactiver les nœuds ou les arêtes du graphe au cours du temps.

Le problème général du transport d'objets par des robots dans un environnement industriel a motivé de nombreux travaux depuis une dizaine d'années. A l'opposé, le transport de biens/services vers des personnes mobiles, c'est-à-dire dans un contexte très dynamique, a été très peu considéré. On peut mentionner des compétitions de robots de service (Lu *et al.*, 2013) qui focalisent sur les aspects d'ingénierie plus que sur la question de la coopération multi-robots.

Un problème connexe important est celui de la localisation et du suivi d'objets ou de personnes dans un espace donné. Les travaux menés se concentrent sur les critères de robustesse et de précision du suivi (Bernardin, Stiefelhagen, 2008). Or, dans notre problématique, servir de nombreuses personnes avec peu de robots nécessite surtout de changer souvent de cible et de coordonner les choix des robots. Notons que ces différents contextes d'interaction avec l'humain nécessitent de doter les robots de capacité à identifier les personnes, reposant sur des techniques de fusion multi-capteurs (Volkhardt *et al.*, 2013). Plus généralement, l'interaction entre robots et humains requière des modalités adaptées (Rios-Martinez *et al.*, 2014). Nous ne les étudions pas dans cet article, nous supposons qu'un service peut être fourni à une personne lorsque le robot est suffisamment proche d'elle.

3. Patrouille dynamique

Comme nous venons de le voir, à notre connaissance, le problème de la patrouille dynamique (de personnes) n'a pas été adressé par la communauté multi-agent ou robotique. Ainsi, nous commençons par le définir formellement ci-après.

3.1. Définition du problème

Nous modélisons les différentes entités en interaction comme un SMA contenant des personnes et des robots mobiles, évoluant dans un environnement fini, potentiellement complexe (présence de murs, d'obstacles et d'agents mobiles). La patrouille multi-agent dynamique est composée de :

- un ensemble de personnes mobiles, noté $H = \langle h_1, \dots, h_k \rangle$
- un ensemble de robots mobiles, noté $R = \langle r_1, \dots, r_n \rangle$
- une carte continue ou discrète 2D de l'environnement d'évolution des agents, notée W .

Les personnes sont considérées comme identifiables par les robots. En d'autres termes, les robots disposent d'une liste des personnes évoluant dans l'environnement et ils sont capables de les reconnaître lorsqu'ils se trouvent dans leur champ de perception. De plus, les robots disposent d'une estimation en continue de la position, la vitesse et du cap des personnes, comme cela peut être obtenu à partir de caméras fixes placées en hauteur (eg. voir (Gomez *et al.*, 2014)).

Chaque personne h_i est représentée, à l'instant t , par un tuple $\{x_i, y_i, o_i, v_i, I_i, C_i\}_t$ où

- $\{x_i, y_i\}_t$ est sa coordonnée dans W ,
- o_i est son orientation,
- v_i est son vecteur vitesse,
- $I_i \in \mathbb{N}$ est sa valeur d'oisiveté, égale au temps écoulé depuis le dernier service reçu.
- $C_i \in \mathbb{N}$ est le nombre de services reçus depuis $t = 0$.

L'objectif de chaque robot est de se rapprocher des personnes présentes pour leur proposer un/des services. Pour naviguer en toute sécurité les robots disposent d'un planificateur de chemin tenant compte de l'espace personnel des agents (comme dans (Papadakis *et al.*, 2014)).

Chaque robot r_i est représenté, à l'instant t , par un tuple $\{x_i, y_i, o_i, v_i, g_i\}_t$ où

- $\{x_i, y_i\}_t$ est sa coordonnée dans W ,
- o_i est son orientation,
- v_i est son vecteur vitesse,
- g_i est son but courant (personne à servir): $g_i \in \{H \cup \emptyset\}$, $g_i = \emptyset$ si le robot n'est assigné à aucune personne.

Les services fournis (non définis dans cet article) par les robots peuvent être de nature différentes mais partagent les **paramètres** suivants :

- $durée$ est la durée nécessaire d'interaction pour permettre au robot de fournir son service.
- D_{min} est la durée minimale qui doit séparer deux services consécutifs fournis à la même personne.

Fonctions objectifs. Le problème de la patrouille dynamique (de personnes) diffère de la patrouille traditionnelle dans la mesure où l'oisiveté (ou idleness en anglais) n'est pas considérée sur des positions de l'environnement mais est attachée aux personnes en déplacement. Le but est de satisfaire les personnes en proposant un service le plus souvent possible, c'est-à-dire en minimisant leur oisiveté notée I_i . En complément, nous introduisons un second objectif qui est l'équité de service envers les personnes. Les robots doivent minimiser la dispersion du nombre de services fournis par personne. Nous choisissons d'utiliser la variance normalisée (permettant de comparer des séries hétérogènes), que nous souhaitons minimiser :

$$N_{Variance}(C_1, \dots, C_k) = \frac{Variance(C_1, \dots, C_k)}{Moyenne^2(C_1 \dots C_k)} \quad (1)$$

Concernant l'oisiveté, nous utilisons les fonctions standard d'évaluation (Pasqualetti *et al.*, 2010). Nous les adaptons à notre contexte en remplaçant les nœuds

/ cellules généralement patrouillés par les personnes à servir. Ainsi nous obtenons les deux **critères** suivants :

$$\bar{I}_t = \frac{1}{|H|} \sum_{i=1}^{|H|} I_{i(t)} \tag{2}$$

qui est l'oisiveté moyenne instantanée, sur l'ensemble des personnes.

$$I_{max_t} = \max(\forall h_i \in H, I_{i(t)}) \tag{3}$$

qui est l'oisiveté maximum instantanée.

Les fonctions d'optimisation sont alors définies comme la minimisation de ces critères, ou la minimisation de leur moyenne dans le temps.

4. Stratégies multi-robots

Dans la suite du document, nous considérerons que le nombre de robots n est plus petit que le nombre de personnes à servir k . Dans le cas contraire, si $n \geq k$, assigner un robot à chaque personne est une solution simple à ce problème.

Dans un premier temps, nous définissons une matrice contenant la position courante de chaque robot et personne. Nous prenons comme hypothèse que cette matrice est partagée entre tous les robots. Cette matrice, nommée *MLoc*, contient les coordonnées courantes de tous les agents, comme suit :

r_1 .. r_n	h_1 .. h_k
(x_{r1}, y_{r1}) .. (x_{rn}, y_{rn})	(x_{h1}, y_{h1}) .. (x_{hk}, y_{hk})

tels que $r_1..r_n$ représentent les robots et $h_1..h_k$ les humains.

Pour représenter les données de la patrouille et les assignations des robots aux personnes, nous définissons une seconde matrice, *MAssign*. Elle contient l'oisiveté des personnes (Idleness), le nombre de services qu'elles ont reçu (Count) et le robot qui leur est assigné :

Personnes	h_1	h_2	..
idleness	I_{h_1}	I_{h_2}	..
count	C_{h_1}	C_{h_2}	..
robot	$r_i \emptyset$	$r_j \emptyset$..

Chaque personne peut avoir un robot assigné ou non (troisième ligne). Une personne non prise en charge reçoit une assignation notée \emptyset (nulle). Nous notons \mathcal{A}_{h_i} le robot assigné à une personne h_i . La personne dont le robot r_i est en charge est notée $\mathcal{A}_{r_i}^{-1}$.

Quatre algorithmes traitant du problème de la patrouille dynamique sont présentés dans cette section. Tout d'abord nous définissons deux approches spatiales connues pour la patrouille multi-agent d'espaces, l'une fondée sur une marche aléatoire et l'autre sur un cycle hamiltonien précalculé couvrant l'environnement. Puis nous définissons une approche exploitant l'information d'oisiveté des personnes. Enfin, la dernière approche est une extension qui considère que les personnes peuvent former des groupes permettant de limiter les déplacements des robots.

Les approches spatiales sont décentralisées mais les autres algorithmes requièrent de partager les matrices $MLoc$ et $MAssign$ pour permettre leur exécution distribuée sur les robots. Chaque algorithme est exécuté simultanément sur tous les robots, mais chaque robot ne retient que la personne à laquelle il est assigné.

4.1. Exploration aléatoire

Le premier algorithme consiste à appliquer une marche aléatoire gaussienne (Camp *et al.*, 2002) sur chaque robot. L'algorithme Exploration Aléatoire est présenté dans l'algorithme 1. Il est appelé à une fréquence f , par chaque robot. Il fournit une coordonnée aléatoire gaussienne à une distance prédéfinie du robot JUMP_DISTANCE. La fonction *gaussianRandomPoint* sélectionne un angle aléatoire suivant l'orientation courante du robot ($r.o$). Le tirage de l'angle aléatoire suit une fonction gaussienne définie par la moyenne μ et la variance σ . S'il y a un obstacle entre la position du robot et l'objectif, le robot reçoit un nouvel objectif valide utilisant un nouveau tirage de coordonnées suivant une loi uniforme.

Algorithme 1 Exploration aléatoire du robot r

```

 $rand_p = gaussianRandomPoint(r.x, r.y, r.o)$ 
while  $isObstacleOnTrajectory(r.x, r.y, rand_p.x, rand_p.y)$  do
     $rand_p = uniformRandomPoint(r.x, r.y)$ 
robot  $r$  go to  $rand_p$ 

```

Algorithme 2 Service opportuniste du robot r

```

for each human  $h_d$  detected by robot  $r$  do
    if  $I_{h_d} > D_{min}$  then
         $r$  delivers service to  $h_d$ 
         $I_{h_d} = 0$ 
         $C_{h_d} ++$ 

```

Lorsqu'un robot perçoit une personne h_d dont l'oisiveté est supérieure à D_{min} la livraison du service est émulée par une immobilisation du robot de quelques secondes (*duree*). Ensuite, l'oisiveté de la personne I_{h_d} est remise à 0 et le nombre de services reçus C_{h_d} est incrémenté. L'algorithme 2 spécifie ce processus de livraison.

Algorithme 3 Initialisation de l'unicycle

phase de pré-calcul

Discretiser l'environnement sous forme d'une grille
 Calculer un cycle \sim Hamiltonian couvrant l'espace accessible
 Calculer n positions de départ sur le cycle à égale inter-distance

phase de positionnement

Assigner à chaque robot une position de départ différente
 Chaque robot se déplace jusqu'à sa position de départ
 Attendre que tous les robots aient atteint leur position de départ

Algorithme 4 Patrouille unicycle du robot r

Follow the Hamiltonian cycle (always clockwise)

if robot r meets a human **then**
 deliver service using Algo. 2

4.2. Couverture par unicycle

Cette stratégie consiste à appliquer une solution optimale pour la patrouille d'un graphe donné (statique). Elle consiste à calculer un cycle hamiltonien couvrant l'environnement qui est un chemin visitant chaque noeud une et une seule fois (ou un cycle le plus proche possible d'un cycle hamiltonien s'il n'existe pas), cf. illustration figure 2b. L'application de cette solution à la patrouille multi-agent nécessite de placer les robots à équidistance maximale sur ce cycle et de le suivre dans le même sens. Cela garantit de (re)visiter chaque nœud à la même fréquence (Portugal, Rocha, 2011). Dans notre contexte, nous adaptons cette stratégie en pré-calculant le parcours hamiltonien couvrant une représentation discrète de l'environnement, puis en déterminant les n positions de départs, cf. algorithme 3. La patrouille consiste alors pour chaque robot à suivre le cycle et à servir les personnes rencontrées, cf. algorithme 4.

Les robots délivrent le service comme dans la précédente approche, lorsqu'ils rencontrent des personnes vérifiant la condition d'oisiveté supérieure à D_{min} , cad. l'algorithme 2. Notons que dans cette version aucune attente supplémentaire des robots n'est réalisée afin de maintenir les robots à inter-distances égales.

4.3. Affectation gloutonne sur l'oisiveté

Dans cette approche, nous prenons en considération l'oisiveté individuelle des personnes pour déterminer la stratégie de service. Comme l'objectif est de minimiser l'oisiveté des personnes, nous assignons les robots aux personnes possédant la plus grande oisiveté.

Ainsi nous définissons un algorithme glouton : à chaque étape d'assignation, on sélectionne la personne non assignée avec la plus grande oisiveté et on lui assigne le robot libre le plus proche. Si plusieurs personnes possèdent la même valeur d'oisiveté, elles sont ordonnées suivant leur valeur d'index i . Cette condition garantit que l'ensemble des personnes sélectionnées ne changera pas avant que l'un d'entre eux

Algorithme 5 Affectation gloutonne individuelle du robot r (IIG)

Input : set R of n robots, set P of p humans, robot r
 Remove the robots R from M_{Assign}
 $pers_to_serve$ = extract from P the n persons with highest Idleness
for each h_i in $pers_to_serve$ ordered by decreasing idleness **do**
 assign to h_i the nearest free robot q in R : $\mathcal{A}_{h_i} = q$
 // robot q is now not free
 Robot r move to $M_{Loc}[\mathcal{A}_r^{-1}]$

soit servi. Cette approche, nommée IIG (Individual Idleness Greedy), est donnée en détail dans l'Algorithme 5. L'ensemble des personnes étant en mouvement, la totalité des assignations est recalculée à une fréquence f permettant notamment de redéfinir le robot le plus proche des personnes à servir.

Quand un robot atteint l'humain dont il est en charge, h_r , il lui délivre le service (pause de *duree* sec.). Ensuite, M_{Assign} est mis à jour : $\mathcal{A}_{h_d} = \emptyset$, $I_{h_d} = 0$ et C_{h_d} est incrémenté. Ce mécanisme est précisé dans l'Algorithme 6.

Algorithme 6 Service du robot r à sa cible \mathcal{A}_r^{-1}

for each human h_d detected by the robot r **do**
 if $\mathcal{A}_{h_d} = r$ AND $I_{h_d} > D_{min}$ **then**
 r delivers service to h_d
 $I_{h_d} = 0$, $C_{h_d} ++$
 $\mathcal{A}_{h_d} = \emptyset$

4.4. Clusters de proximité pour affectation gloutonne locale

L'approche par Clusters de proximité est définie afin de prendre en compte l'organisation spatiale des humains pouvant se déplacer en groupes. L'idée est de minimiser le temps de déplacement des robots en servant les personnes regroupées plutôt que de traverser l'environnement pour atteindre les personnes possédant les plus hautes oisivetés.

Afin de construire une stratégie, nous identifions tout d'abord des groupes de personnes en fonction de leur proximité spatiale. Nous utilisons l'algorithme de clustering k-means pour déterminer autant de groupes que de robots. Puis, chaque robot est assigné au cluster qui lui est le plus proche. La seconde étape de la stratégie mène chaque robot à servir toutes les personnes de son groupe. Cette étape est réalisée simplement en appelant l'algorithme 5 (IIG) sur le sous-ensemble des personnes du cluster.

Les différentes étapes de la stratégie sont données dans l'algorithme 7. Celui-ci est exécuté à une fréquence f .

Algorithme 7 Clusters de proximité & glouton

```

robot_list =  $\cup\{r \in R\}$ 
c_list = ClusterCreation(H)/c_list =  $\cup c_i: \{h_{ij}, \dots, h_{im}\}$ 
for each c  $\in$  c_list do
  robot_o = robot_list ordered by distance to cluster c
  r = first element of robot_o
  remove r from robot_list
  IIG( $\{r\}, c, r$ )

```

5. Comparaison des stratégies**5.1. Simulateur**

Afin d'évaluer les différentes stratégies, nous avons développé un outil basé sur l'implémentation ROS (*Robot Operating System*) du simulateur PedSim (Gloor *et al.*, 2004)¹. Dans ce simulateur, chaque humain (agent) suit un chemin prédéfini grâce à un planificateur local capable de prendre en compte des champs de forces répulsifs (obstacles) et attractifs (liens sociaux et zone d'attraction) suivant un modèle de déplacement d'individus défini dans (Helbing, Molnár, 1995). Ce modèle assure l'anti-collision des robots et l'évitement d'obstacles dynamiques (personnes). La figure 1 illustre ce simulateur.

De manière à simuler une flotte de robots autonomes et leur permettre d'interagir, nous avons enrichi le simulateur. Nous avons ajouté des services permettant aux robots de calculer une trajectoire de déplacement (A^*) dans le simulateur. Les robots peuvent aussi transmettre des informations sur les agents (humains ou robots) perçus dans leur champ de vision/détection.

Le simulateur collecte les différentes données perçues par les robots (obstacles proches, position des agents) et met à jour les matrices de mesures. Selon l'algorithme d'assignation utilisé, les informations doivent être partagées entre robots (cas des algorithmes 5 et 7).

Concernant la marche aléatoire, les paramètres de la gaussienne sont $JUMP_DISTANCE = 10\ m$, $\mu = 1$ and $\sigma = 0.1$ (ces valeurs limitent la fréquence de changement d'orientation du robot).

La fréquence de prise de décision f est de 1 Hz pour les différents algorithmes sauf pour l'approche par cluster où la prise de décision est diminuée à 1/3 Hz pour éviter des oscillations dans les affectations des robots liées au recalcul des différents clusters auxquels ils sont affectés.

Afin de comparer les différentes stratégies, deux types d'expérimentations sont considérés dans la suite de l'article. Le premier considère un scénario unique corres-

1. (ROS packages for PedSim (Pedestrian Simulator) based on social force model of Helbing *et. al.*, https://github.com/srl-freiburg/pedsim_ros, s. d.)

pondant à un espace de networking ou de musée où évoluent plus de 50 personnes ainsi que 4 robots en charge de distribuer des services. Il est utilisé pour comparer en détail les résultats obtenus par les différentes stratégies. Le second type d'expérimentations vise à comparer les performances des différentes stratégies en changeant la configuration du scénario, à savoir la topologie de l'environnement (sans obstacle, musée cad. avec peu d'obstacles, et bureaux) et la dispersion des personnes dans l'environnement.

5.2. Analyse des résultats sur le scénario musée

5.2.1. Description du scénario

Nous considérons un ensemble de robots évoluant dans un environnement clos et contenant des obstacles. Ce peut être un espace de networking ou à un lieu culturel comme un musée. On considère que les robots doivent servir des mets de manière équitable à l'ensemble des personnes présentes. Le scénario de déplacement des personnes repose sur deux parcours possibles dans l'environnement (cf. figure 2a circuits rouge et bleu). Des personnes sont générées à l'entrée de ces parcours avec des liens éventuels (permettant de rester groupés).

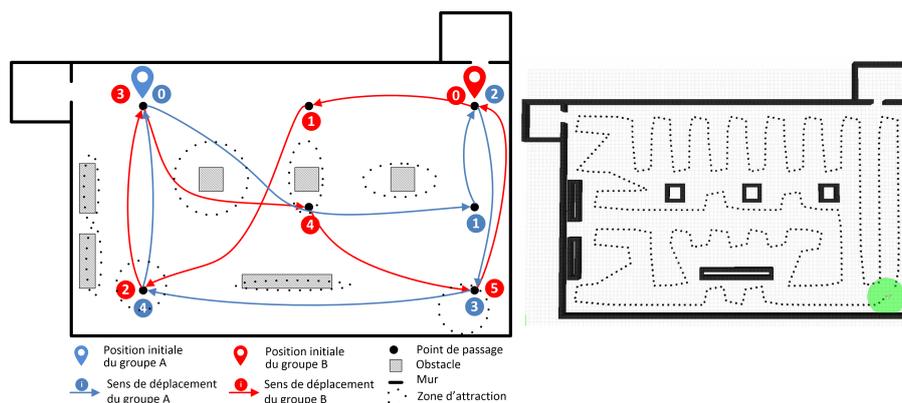


Figure 2. a. Scénario musée : waypoints des chemins suivis par les personnes et zones d'attractions. b. Solution unicycle hamiltonien précalculée

Ce scénario a les propriétés suivantes :

- Les robots (2 à 8) démarrent dans le coin inférieur gauche.
- 50 personnes se déplacent suivant 2 chemins prédéfinis (25 sur chaque circuit).
- 20 personnes, parmi les 50, ont des liens sociaux et forment des groupes de 2 à 5 personnes se déplaçant ensemble.
- Les robots et les personnes se déplacent à une vitesse maximum de 2 m.s^{-1} .

- La limite de détection d'un humain autour d'un robot est de 4 mètres.
- Le temps nécessaire pour délivrer le service est de 2 secondes.
- L'environnement a une surface de 5200 m^2 .

La planification des déplacements est réalisée avec l'algorithme A*. Lors du suivi d'un chemin, un évitement des obstacles, fondé sur des champs de forces, est appliqué en complément. Concernant l'algorithme 3, l'unicycle pré-calculé pour ce scénario est présenté en figure 2b.

5.2.2. Résultats avec 4 robots

Dans cette section nous évaluons les quatre algorithmes proposés sur le scénario proposé avec 4 robots serveurs. Chaque algorithme a été testé 10 fois pour une durée simulée de 600 secondes.

Les critères suivants ont été mesurés :

- l'oisiveté moyenne instantanée,
- l'oisiveté maximum instantanée,
- la variance normalisée du nombre de services délivrés par personne.

Une vidéo illustrant les résultats peut être consultée au lien suivant : <https://www.youtube.com/watch?v=oUbl1uVTmnw>.

Les figures 3, 4 et 5 présentent les valeurs moyennées sur 10 exécutions. Sur ces figures, les résultats des algorithmes Exploration aléatoire, Patrouille unicycle, Affectation gloutonne individuelle (IIG) et Clusters de Proximité IIG sont tracés respectivement en rouge, noir, bleu et vert.

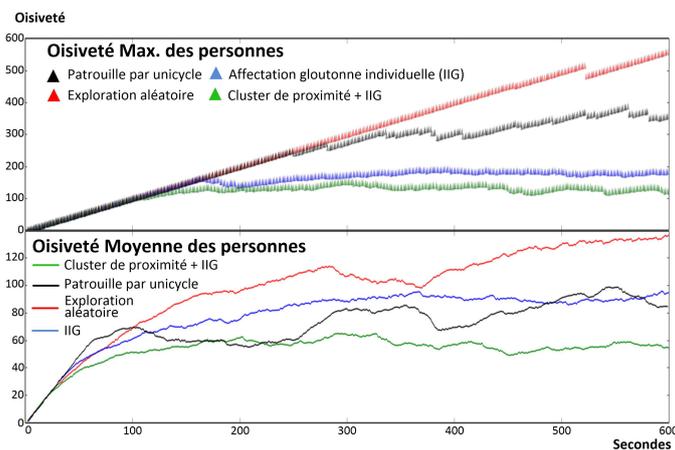


Figure 3. Oisiveté moyenne et maximale durant 600 secondes

La figure 3 présente la moyenne instantanée de l'oisiveté (trait fin) et l'oisiveté maximale (trait épais) pour les quatre algorithmes. Comme précisé en section 3.1, le problème de patrouille standard vise à minimiser l'oisiveté des cibles. Des valeurs d'oisiveté (maximale et moyenne) faibles correspondent à une meilleure qualité. Cependant, pour garantir l'équité du service entre les personnes, il est pertinent que l'oisiveté maximale (personne la moins bien servie) soit la plus proche possible de l'oisiveté moyenne.

Concernant l'oisiveté maximale de l'approche Exploration aléatoire, sa progression est quasi régulière sur les 600s de la simulation. Cela signifie qu'au moins un individu n'a jamais été servi sur cette durée. Dans le cas de la stratégie basée sur l'unicycle, bien qu'il n'y ait aucune garantie que les individus soient tous servis, les résultats sont meilleurs que la marche aléatoire. Mais pour ces deux algorithmes, l'oisiveté maximale reste élevée, car ces stratégies spatiales ne prennent pas en compte les valeurs d'oisiveté. A l'opposé, les algorithmes IIG et CP-IIG, fondés sur ce critère, maintiennent une oisiveté maximale basse, avec un léger avantage pour l'approche par cluster (CP-IIG) qui sert plus de personnes.

Les résultats sur l'oisiveté moyenne sont un peu différents. L'approche aléatoire garde les plus mauvais résultats, mais l'unicycle et IIG ont des résultats proches. On peut constater que l'unicycle subit de fortes variations, ce qui est logique car les personnes ne parcourent pas la totalité de l'environnement.

La mesure de la moyenne d'oisiveté confirme que l'approche par Clusters de Proximité-IIG est clairement la meilleure sur ce scénario. L'oisiveté moyenne est meilleure durant toute la durée avec un gain croissant au cours du temps. En effet, l'approche CP IIG limite les déplacements des robots entre des personnes proches contrairement à IIG. Ainsi les robots servent plus de personnes entraînant une diminution de l'oisiveté moyenne.

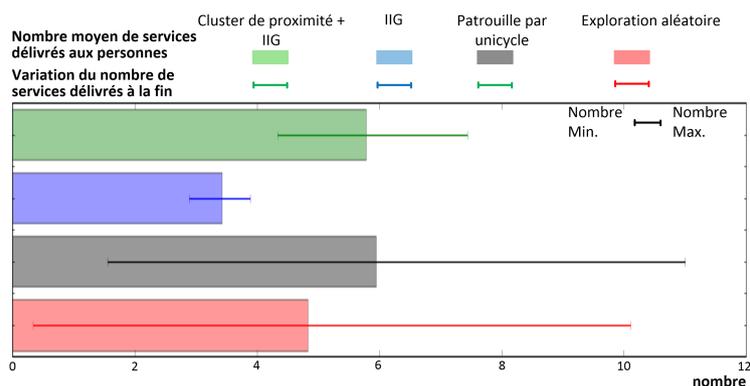


Figure 4. Valeur moyenne du nombre de services délivrés par individu

La figure 4 montre le nombre moyen de services délivrés par individus à la fin des expérimentations (le maximum est le mieux). Les valeurs minimum et maximum obtenues sont aussi présentées. La meilleure moyenne est obtenue par l'unicycle mais avec une grande iniquité (minimum de 2 et maximum à 11). Nous obtenons aussi une grande iniquité avec l'approche Exploration aléatoire. L'Affectation gloutonne individuelle (IIG) et Clusters de Proximité IIG offrent une bien meilleure équité, ce qui est la conséquence logique de servir en premier les personnes ayant la plus grande oisiveté. La stratégie Clusters de Proximité IIG offre le meilleur compromis entre la quantité de services délivrés et l'équité du service.

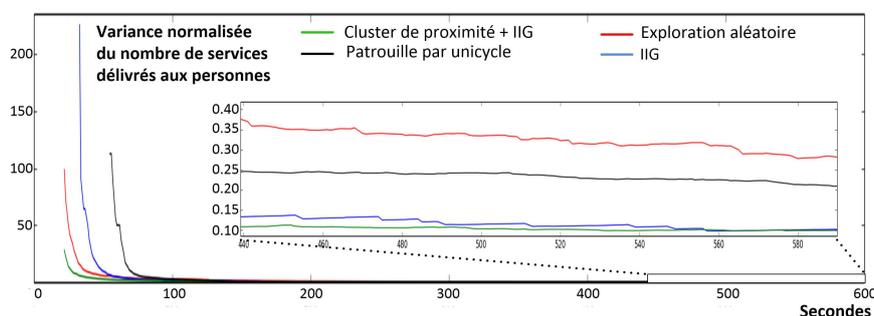


Figure 5. Variance normalisée du nombre de services rendus par individu

La figure 5 montre la variance normalisée du nombre de services délivrés durant une expérimentation. Une bonne équité correspond à une variance faible. La variance est normalisée pour être comparable malgré les différences en moyenne sur le nombre de services délivrés par les différents algorithmes. La valeur obtenue n'a de sens que lorsque suffisamment de personnes ont été servies. On constate que la variance se stabilise à partir de 100s.

Sur la figure 5, nous faisons un zoom sur la fin de la simulation (450s-600s) pour comparer les valeurs d'équité. Les algorithmes Exploration aléatoire et Patrouille unicycle présentent des valeurs plus hautes, confirmant que les deux approches spatiales ne sont pas équitables. Les algorithmes IIG et Clusters de Proximité IIG ont des valeurs plus faibles (entre 0.1 et 0.15) qui tendent à se confondre. Cela confirme que ces approches sont équitables entre les individus, comme supposé à partir de la figure 4.

5.2.3. Synthèse des performances

Le tableau 1 montre les résultats obtenus sur tous les algorithmes du scénario musée avec 4 robots en fin de simulation (à 600s). Les cases grisées représentent les meilleurs résultats pour chaque critère (ligne). L'approche Clusters de Proximité IIG est meilleure que les autres, si l'on regarde les critères de minimisation de l'oisiveté maximum et d'équité (la variance du nombre de services délivrés par personne).

Si nous considérons la capacité des robots à maximiser le nombre de services délivrés, les stratégies Patrouille unicycle et Clusters Proximité-IIG ont les meilleurs résultats (5.9 and 5.8), mais CP-IIG est la meilleure en prenant en compte l'équité.

Tableau 1. Oisiveté moyenne, oisiveté maximale, variance du nombre de services délivrés et nombre de services délivrés en fin de simulation (Musée, 4 robots)

	Explo. Aléa.	Patrouille unicycle	IIG	Clusters Proxim.+IIG
Idleness Moy.	98.4s	70.1s	77.5s	52.9s
Idleness Max.	559.1s	386.7s	191.4s	152s
Norm. Count Variance	0.28	0.21	0.10	0.09
Count	4.8	5.9	3.4	5.8

5.2.4. Stratégies sur les individus vs. les groupes

Une question adressée dans cet article est de déterminer si le fait de considérer les individus par groupes a un impact sur les performances de la patrouille. Le tableau 1 et les résultats présentés en section 5.2.2 montrent que l'approche Clusters de Proximité est bien meilleure que la version individuelle, pour les critères liés à l'oisiveté et l'équité (variance du nombre de services délivrés).

Nos scénarios expérimentaux ont été définis afin d'assurer que certaines personnes puissent évoluer en petits groupes. Les personnes peuvent être liées explicitement dès le départ de l'expérimentation ou implicitement et temporairement autour d'une zone d'attraction. Cette hypothèse a motivé la définition de la stratégie Clusters de Proximité IIG. Comme les groupes ne peuvent généralement pas être connus à l'avance et peuvent apparaître dynamiquement, cet algorithme se base sur k-means, où k est le nombre de robots. Chaque robot se retrouve ainsi associé à un groupe de personnes localisé à proximité. Une perspective sera de comparer d'autres algorithmes d'identification des groupes, incluant des techniques fondées sur des modèles sociologiques comme (Rios-Martinez *et al.*, 2014).

Nous complétons cette analyse, en section 5.3, en faisant varier le scénario des chemins suivis par les personnes, c'est-à-dire en les forçant à rester sur une partie plus ou moins étendue de l'environnement.

5.2.5. Influence du nombre de robots sur les stratégies

Les résultats présentés précédemment ont été réalisés en déployant 4 robots. Nous avons procédé à d'autres mesures pour déterminer l'influence du nombre de robots. Nous avons ainsi comparé les algorithmes avec 2, 4, 6 et 8 robots (dans des conditions identiques de simulation).

La figure 6 présente la valeur maximale d'oisiveté à la fin de la simulation (moyennée sur 10 exécutions). On peut constater que, de manière logique, le fait d'augmenter le nombre de robots améliore les résultats de toutes les stratégies. On constate aussi

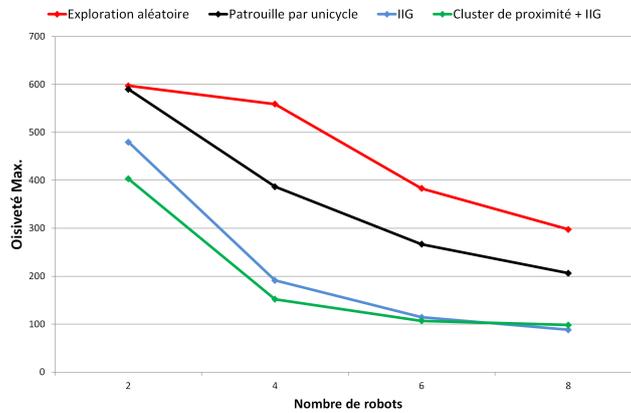


Figure 6. Oisiveté maximale à 600s en fonction du nombre de robots

que les stratégies basées sur l’oisiveté ont toujours de meilleurs résultats que les approches spatiales/aléatoires.

Si l’on compare l’approche IIG et cluster IIG, leur différence de performance décroît avec l’augmentation du nombre de robots. En effet, un nombre croissant de robots réduit automatiquement la distance moyenne du plus proche robot d’une personne donnée, diminuant l’intérêt de calculer des clusters de proximité.

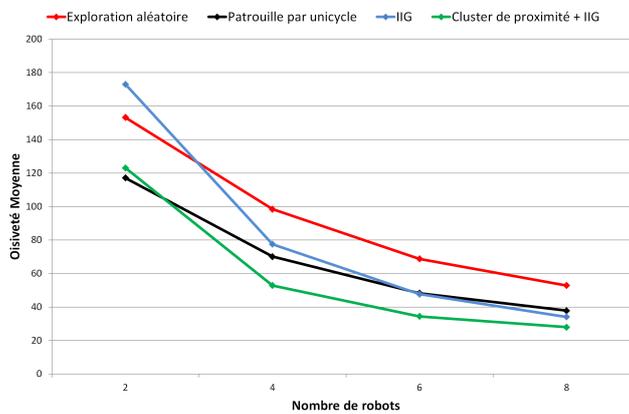


Figure 7. Oisiveté moyenne à 600s en fonction du nombre de robots

Ces résultats sont confirmés sur la figure 7 où nous comparons l’oisiveté moyenne finale. A partir de 6 robots, les algorithmes fondés sur l’oisiveté sont meilleurs que

les approches spatiales et aléatoires. La stratégie basée sur les clusters est la plus performante de 4 à 8 robots.

5.3. Analyse des performances avec des environnements différents et des dispersions spatiales variables

5.3.1. Description des scénarios

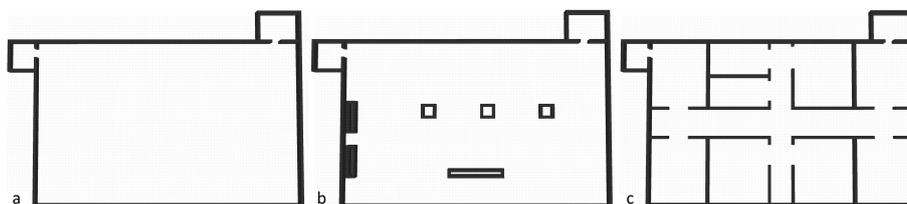


Figure 8. Environnements expérimentaux : a) Vide, b) Musée, c) Bureaux

Tout comme dans le scénario précédent, les robots évoluent dans un environnement clos et doivent servir des personnes de manière équitable. Par contre, nous considérons différentes topologies d'obstacles pour évaluer leur impact sur les performances des algorithmes.

Les paramètres liés aux humains et aux robots sont inchangés : 50 personnes évoluant selon deux chemins prédéfinis, ainsi que 4 robots évoluant à une vitesse maximale de $2m.s^{-1}$ avec un rayon de perception de 4 mètres. Le temps de livraison d'un service reste de 2s et la durée totale d'expérimentation est de 600 s.

Trois types d'environnements ont été étudiés (cf. figure 8) :

- aucun obstacle : **environnement vide**
- quelques obstacles : **environnement musée**
- pièces et couloirs : **environnement bureaux**.

Cette variation de la configuration des obstacles va influencer, d'une part, sur la facilité de déplacement des robots, et d'autre part, sur le parcours des personnes. Le déplacement des robots se base sur un calcul de plus court chemin A*. Les obstacles génèrent de la complexité pour la traversée de l'environnement, donc impacte la capacité à délivrer des services.

Un second paramètre est étudié, la dispersion spatiale des individus. Plus la dispersion est grande, plus les individus seront répartis équitablement dans l'environnement. A contrario, plus une dispersion est faible, plus les individus seront concentrés dans une zone de l'environnement. Nous avons testé 3 types de dispersion des individus (pour chaque scénario) :

- une **dispersion basse** (B), les individus sont regroupés autour d’une zone représentant un quart de l’environnement total,
- une **dispersion moyenne** (M), les individus sont regroupés autour d’une zone représentant la moitié de l’environnement total,
- une **dispersion élevée** (E), les individus sont repartis sur la totalité de l’environnement.

Dans cette section les stratégies Patrouille par unicycle (noir), IIG (bleu) et Clusters de Proximité IIG (vert) sont comparés. Chaque scénario (1 type d’environnement, 1 type de dispersion, 1 stratégie) est évalué pendant une période de 10 minutes et ce 10 fois afin de limiter les biais de mesures.

5.3.2. Résultats et analyses

Pour l’ensemble des scénarios évalués, deux critères sont étudiés : l’**oisiveté moyenne** des individus, représentant le temps d’attente moyen entre deux services, et l’**oisiveté maximum** des individus représentant le temps d’attente maximum entre deux services. Les résultats portant sur ces critères d’oisiveté sont présentés sur la figure 9. Concernant la lecture des résultats, plus les valeurs sont basses (oisiveté moyenne, oisiveté maximum) plus les stratégies sont efficaces.

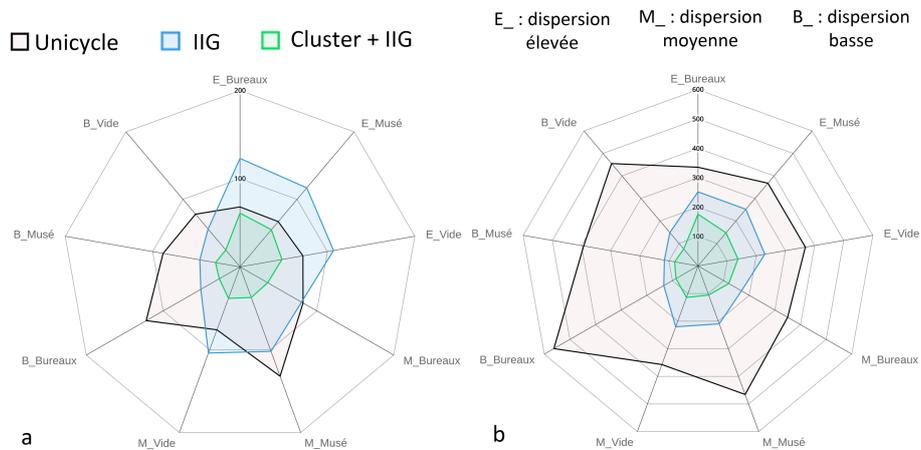


Figure 9. Performances des algorithmes a) oisiveté moyenne des individus en fin d’expérimentation, b) oisiveté maximum des individus en fin d’expérimentation

Nous examinons aussi **Le nombre de services rendus** qui permet de fournir une mesure quantitative de la livraison des services par les robots. Afin de prendre en compte également le **critère d’équité**, la variance normalisée du nombre de services vient compléter les résultats d’expérimentations. Ces informations sont présentées en figure 10. Pour la lecture des résultats, plus le nombre de services rendus est élevé,

meilleurs sont les résultats (figure 10a). Quant à l'équité, plus les valeurs sont petites plus la stratégie est équitable (figure 10b).

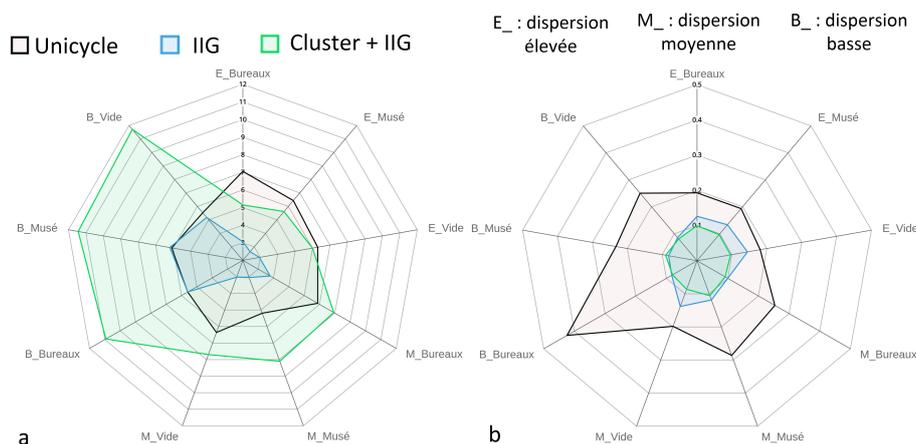


Figure 10. Performances des algorithmes a) nombre de services rendus en fin d'expérimentation, b) équité de service en fin d'expérimentation

Analyse stratégie unicycle. La stratégie de patrouille par unicycle vise à optimiser une couverture spatiale de l'environnement. Ainsi, plus les personnes sont dispersées dans l'environnement, plus les robots ont une chance de les rencontrer. C'est pourquoi cette approche obtient de meilleurs résultats sur le nombre de services rendus que les autres approches pour des valeurs de dispersion élevée (figure 10a). La stratégie de patrouille par unicycle est opportuniste. Les robots servent des personnes dès qu'ils les croisent. L'oisiveté n'étant pas un critère discriminant il n'est pas étonnant de voir des valeurs élevées pour cette stratégie concernant les critères d'oisiveté maximale (figure 9b) et d'équité (figure 10b). Concernant l'oisiveté moyenne, l'unicycle présente des mesures plus élevées que les autres stratégies à l'exception des scénarios où la dispersion est élevée et le scénario d'un environnement vide et d'une dispersion moyenne (M_vider).

Terminons par une remarque sur le scénario d'un environnement vide et d'une dispersion d'individu moyenne (M_vider). La stratégie unicycle possède une oisiveté moyenne plus basse que la stratégie IIG, et sert également plus de personnes dans ce cas de figure. La construction de l'unicycle dans ce scénario planifie le passage des robots plusieurs fois dans les zones de déplacement des individus (chemins prédéfinis). Tout comme dans les scénarios possédant une dispersion basse, une fois arrivée dans une zone fortement peuplée, les robots servent de façon opportuniste les personnes rencontrées. Ils se trouvent bloqués dans ces zones pendant une durée importante, mais cela s'avère utile pour une tâche de service.

Analyse stratégie IIG. La stratégie IIG se concentre uniquement sur l'oisiveté pour sélectionner les personnes à servir. Ainsi les mesures d'oisiveté maximum (figure 9b) et d'équité (figure 10b) restent bonnes malgré un nombre de services rendus faible. Pour des scénarios présentant une dispersion des individus importante (Élevée E et Moyenne M), la stratégie IIG sert peu de personnes. Dans cette approche, les robots doivent parcourir de grandes distances avant de servir des individus entraînant une perte d'efficacité. L'oisiveté moyenne de la stratégie IIG (figure 9a) reste dépendante du nombre de services rendus, aussi lorsque le nombre de services rendus est faible l'oisiveté moyenne augmente (dispersion élevée).

Analyse stratégie Cluster Proximité IIG. Cette stratégie porte à la fois sur des critères spatiaux (clusters basés sur les distances euclidiennes entre les individus) et l'oisiveté. Ce compromis présente de meilleurs résultats pour la plupart des critères : oisiveté moyenne, oisiveté maximum, nombre de services rendus, équité. Toutefois, cette stratégie sert moins de personnes que l'unicycle pour des scénarios où leur dispersion est forte. Sur une dispersion élevée les clusters associés aux robots sont étirées nécessitant de long trajets entre les personnes d'un même cluster. Nous pouvons également noter que plus la dispersion est faible plus les critères d'oisiveté maximum, d'oisiveté moyenne et d'équité se resserrent entre la stratégie IIG et la stratégie Cluster de Proximité. En effet, plus les personnes sont regroupées, moins les clusters de proximités présentent un intérêt. Néanmoins, la stratégie par clusters reste très efficace pour servir les personnes (nombre de services rendus, figure 9b) et ce même pour une dispersion faible.

5.3.3. Synthèse

Cet ensemble d'expérimentations confirme l'intérêt d'utiliser une stratégie utilisant à la fois des critères de distance spatiale et des critères basés sur l'oisiveté. La dispersion des personnes dans l'environnement influe significativement sur les critères d'oisivetés et de nombre de services rendus. Le type d'environnement a un impact plus mesuré. On constate néanmoins que plus l'environnement contraint le déplacement des robots, plus l'oisiveté moyenne est élevée.

Malgré une variation de la dispersion des individus dans l'environnement ou du type d'environnement, l'approche de Cluster de proximité IIG présente le meilleur compromis entre nombre de services rendus, équité, oisiveté moyenne des individus et oisiveté maximum. Même si pour des dispersions élevées, la patrouille par unicycle sert plus de personnes, l'approche par Cluster de Proximité IIG reste proche en nombre de services rendus et possède des critères d'oisiveté moyenne et maximum bien meilleurs. De plus, les différences de mesures entre IIG et Clusters de Proximité IIG sont moins importantes lorsque la dispersion des individus est basse. Tout comme l'augmentation du nombre de robots rapproche ces deux stratégies, la concentration des individus dans une zone entraîne une diminution de l'impact des clusters sur l'oisiveté moyenne, maximum et de l'équité. Malgré cela, la stratégie de Cluster de Proximité IIG obtient un nombre de services rendus bien supérieur à IIG (facteur supérieur à 2) limitant au maximum les déplacements des robots.

6. Conclusion

Dans cet article, l'analyse d'une problématique de robots serveurs nous a conduits à définir le problème de la patrouille multi-agent dynamique (d'humains ou d'agents mobiles en général). Ceci nous a mené à poser deux critères à satisfaire : i) minimiser l'oisiveté maximum ou moyenne (le temps écoulé entre deux services à une personne) ii) minimiser la différence du nombre de services délivrés entre personnes (i.e. optimiser l'équité). Considérant ces critères, nous avons défini quatre algorithmes de patrouille multi-robot, deux fondés sur des solutions standard de la patrouille d'un espace (cycle hamiltonien couvrant l'environnement et marche aléatoire) et deux fondés sur les valeurs d'oisiveté (visite des personnes par ordre décroissant d'oisiveté). Le dernier calcule des clusters de personnes selon leur proximité, afin de réduire les distances parcourues par les robots.

Pour évaluer ces approches nous avons proposé un simulateur intégrant le modèle de piétons PedSim et le framework robotique ROS. Les résultats expérimentaux montrent que les approches fondées sur l'oisiveté présentent de meilleurs résultats que les approches standard de patrouille spatiale lorsque l'on considère les critères d'oisiveté maximum et d'équité (variance du nombre de services reçus). L'avantage de l'approche par cluster a aussi été mesuré. Ces résultats sont confirmés lorsque le nombre de robots utilisés est augmenté, mais dans ce cas, l'intérêt d'utiliser des clusters diminue. De plus, même si l'impact de la dispersion des individus sur les mesures est significatif, la stratégie de clusters de proximité IIG reste la plus pertinente pour notre problématique.

Ces résultats ouvrent plusieurs perspectives. Tout d'abord, nous visons à généraliser l'évaluation de la patrouille multi-agent dynamique en fournissant un ensemble de benchmarks. Cela doit permettre l'évaluation et la proposition de nouveaux algorithmes. Ensuite, nous visons à réaliser des expérimentations réelles associant robots mobiles et humains. Ceci nécessite d'adresser des problématiques opérationnelles et de définir des comportements de robots adaptés à la présence humaine, tout particulièrement en ce qui concerne la navigation. Enfin, La gestion des ressources (en énergie notamment) des robots pourrait présenter un nouveau critère d'évaluation pertinent.

Bibliographie

- Aaron E., Krizanc D., Meyerson E. (2014). DMVP: foremost waypoint coverage of time-varying graphs. *CoRR*, vol. abs/1407.7279. Consulté sur <http://arxiv.org/abs/1407.7279>
- Bernardin K., Stiefelwagen R. (2008). Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, vol. 2008, n° 1, p. 246309. Consulté sur <http://jivp.eurasipjournals.com/content/2008/1/246309>
- Camp T., Boleng J., Davies V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, vol. 2, n° 5, p. 483-502. Consulté sur <http://dblp.uni-trier.de/db/journals/wicomm/wicomm2.html#CampBD02>

- Chevaleyre Y. (2007). The patrolling problem: Theoretical and experimental results. In W. O. Library (Ed.), *Combinatorial optimization and theoretical computer science*, p. 161-174.
- Elmaliach Y., Agmon N., Kaminka G. A. (2010). Multi-robot area patrol under frequency constraints. *AMAI*, vol. 57, n° 3–4, p. 293–320.
- Glad A., Buffet O., Simonin O., Charpillet F. (2009). Self-organization of patrolling-ant algorithms. In *Third IEEE international conference on self-adaptive and self-organizing systems, SASO 2009, san francisco, california, usa, september 14-18, 2009*, p. 61–70. Consulté sur <http://dx.doi.org/10.1109/SASO.2009.39>
- Gloor C., Stucki P., Nagel K. (2004). Hybrid techniques for pedestrian simulations. In P. M. A. Sloot, B. Chopard, A. G. Hoekstra (Eds.), *Acrid*, vol. 3305, p. 581-590. Springer. Consulté sur <http://dblp.uni-trier.de/db/conf/acri/acri2004.html#GloorSN04>
- Gomez J. V., Mavridis N., Garrido S. (2014). Fast marching solution for the social path planning problem. In *2014 IEEE international conference on robotics and automation, ICRA 2014, hong kong, china, may 31 - june 7, 2014*, p. 1871–1876. Consulté sur <http://dx.doi.org/10.1109/ICRA.2014.6907105>
- Helbing D., Molnár P. (1995). Social force model for pedestrian dynamics. *Physical Review E*, p. 4282–4286.
- Lu Q., Lu G., Bai A., Zhang D., Chen X. (2013). An intelligent service system with multiple robots. In *Robot competition of international joint conference on artificial intelligence (ijcai 2013)*.
- Papadakis P., Rives P., Spalanzani A. (2014, septembre). Adaptive Spacing in Human-Robot Interactions. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'14.*, Chicago, United States. Consulté sur <https://hal.inria.fr/hal-01010224>
- Pasqualetti F., Franchi A., Bullo F. (2010, 12/2010). On optimal cooperative patrolling. In *49th IEEE conference on decision and control*, p. 7153-7158. Atlanta, GA, USA.
- Portugal D., Rocha R. (2011). A survey on multi-robot patrolling algorithms. *IFIP Advances in Information and Communication Technology*, vol. 349, p. 139-146.
- Poulet C., Corruble V., Seghrouchni A. E. F. (2012). Auction-based strategies for the open-system patrolling task. In *Proc. prima - 15th international conference on principles and practice of multi-agent systems*.
- Rios-Martinez J., Spalanzani A., Laugier C. (2014, septembre). From Proxemics Theory to Socially-Aware Navigation: A Survey. *International Journal of Social Robotics*. Consulté sur <https://hal.inria.fr/hal-01067278>
- ROS packages for pedsim (pedestrian simulator) based on social force model of helbing et al.*, https://github.com/srl-freiburg/pedsim_ros. (s. d.). Consulté sur https://github.com/srl-freiburg/pedsim_ros
- Volkhardt M., Weinrich C., Gross H.-M. (2013, Sept). Multi-modal people tracking on a mobile companion robot. In *Mobile robots (ecmr), 2013 european conference on*, p. 288-293.