



HAL
open science

Le problème multi-agents de la patrouille

Yann Chevaleyre

► **To cite this version:**

| Yann Chevaleyre. Le problème multi-agents de la patrouille. 2006. hal-00115783

HAL Id: hal-00115783

<https://hal.science/hal-00115783>

Preprint submitted on 23 Nov 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Le problème multi-agents de la patrouille

Yann Chevaleyre*

Résumé

Le problème multi-agents de la patrouille a été récemment introduit dans [2]. Ce problème consiste à faire se déplacer un ensemble d'agents ou de robots sur un territoire prédéfini de telle sorte, informellement, que chaque partie de ce territoire soit visitée par des agents le plus souvent possible. Pour résoudre ce problème, des approches basées sur des architectures d'agents réactifs et cognitifs avaient été décrites dans [2]. Ce problème est ici abordé comme une tâche d'optimisation combinatoire. Pour cela, le territoire est modélisé sous forme d'un graphe. Différentes stratégies d'exploration du graphe sont envisagées, en particulier basées sur des cycles issus du voyageur de commerce. Avec ces dernières et dans le cas où le graphe est métrique, nous obtenons en temps polynomial une stratégie d'exploration dont la valeur est bornée par $3 \times opt + 4 \times \max_{ij} \{c_{ij}\}$, avec c_{ij} désignant le poids de l'arête (i, j) . On montre enfin qu'avec une autre approche basée sur un partitionnement du graphe en régions distinctes, le problème devient 15-approximable, même dans le cas de graphes non métriques.

Mots-clefs : Systèmes Multiagents, Problème de la Patrouille, Optimisation combinatoire, Approximation polynomiale

1 Introduction

Le problème multi-agents de la patrouille consiste à faire parcourir à des agents un territoire, de telle sorte que les différentes parties du territoire soient visitées le plus souvent possible par ces agents. Ce problème se pose typiquement dans les jeux vidéos [2, 3] lorsqu'une équipe de créatures virtuelles a pour mission de patrouiller sur un territoire déterminé, dans certaines applications internet, ainsi que pour le déplacement d'une équipe de robots [15].

*LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. chevaley@lamsade.dauphine.fr

Ce problème avait été initialement introduit dans le cadre des systèmes multi-agents (SMA) par Geber Ramalho [2]. Ce dernier avait abordé le problème en utilisant le paradigme multi-agents classique. Mentionnons deux des algorithmes représentatifs de cette approche : une première solution avait consisté à développer des agents réactifs, ne possédant aucune capacité de planification ou de raisonnement, chaque agent se déplaçant en direction de la zone de son voisinage immédiat qui soit restée sans visite le plus longtemps. Dans une deuxième approche basée sur des agents cognitifs, chaque agent se dirigeait vers la partie du territoire - pas nécessairement voisine - ayant été sans visite d'agent le plus longtemps possible, en utilisant un algorithme du *plus court chemin*. Un mécanisme de communication entre agents leur permettait de se coordonner en se partageant les régions à visiter, afin que deux agents ne se dirigent pas vers le même lieu. Les différents algorithmes développés par Machado et Ramalho ont été implémentés dans un simulateurs, pour permettre la visualisation du comportement de chaque agent (figure 1).

Cet article aborde le problème sous l'angle très différent de l'optimisation combinatoire. Notons que de nombreux problèmes d'optimisation combinatoire ont des points communs avec le problème de la patrouille : citons les *Vehicle Routing Problem*, et en particulier le *min-max-VRP*. Il est peut être possible d'établir d'autres résultats en utilisant une réduction de notre problème vers ce dernier. Cependant, les problèmes de la famille des *VRP* supposent l'existence d'un dépôt, auquel chaque agent se retourne à la fin de sa tournée. A cause de cette différence, nous n'avons pu trouver de réduction efficace de notre problème vers un *VRP*. La contribution principale de cet article a consisté à mettre en évidence le lien entre notre problème et le problème du voyageur de commerce. En exploitant ce lien et en introduisant certains biais sur l'espace de recherche, on peut aboutir à des solutions de bonnes qualités.

Des expérimentations ont été mises en oeuvre pour comparer certaines des solutions proposées dans cet article et les algorithmes multiagents de [2]. Des résultats préliminaires, qui sont détaillés dans [1], ont montré sur divers graphes que les stratégies obtenues par les méthodes décrites (en section 6) étaient jusqu'à deux fois meilleures que les algorithmes multiagents classiques de [2].

Cet article commence par une présentation du problème multi-agents de la patrouille, et du formalisme employé. Ensuite, le lien avec le problème du voyageur de commerce (TSP) est établi dans le cas mono-agent. Puis, des biais sont introduits sur l'espace de recherche sous la forme de *classes de stratégies*. Ensuite, ces classes sont comparées les unes aux autres afin de déterminer dans lesquelles on peut trouver de bonnes stratégies. Enfin, les deux dernières sections montrent des résultats d'approximation obtenus à l'aide de certaines de ces classes : la section 6 montre que dans le cas métrique, en utilisant des algorithmes d'approximation pour TSP, on peut obtenir dans le cas multi-agents des stratégies dont le critère est borné par $3 \times opt + 4 \times \max_{ij} \{c_{ij}\}$, et la dernière section montre que le problème de la patrouille est 15-approximable y compris dans le cas non métrique.

2 Le problème de la patrouille

Le territoire que patrouillent les agents peut-être décrit par un graphe, comme dans [2]. Soit un graphe $G = (V, E)$ représentant le territoire, V étant l'ensemble des n sommets du graphe et E l'ensemble des arêtes. A l'instant initial, chacun des r agents se trouve sur l'un des noeuds du graphe. Notons que le nombre d'agents est constant, et ne varie pas durant le processus de patrouille. On associe à chaque arête (i, j) de E un coût c_{ij} , représentant la *distance* entre i et j . On suppose dans la suite de cet article que la fonction c_{ij} est positive, symétrique. Pour se déplacer de i à j , un agent nécessitera donc un temps égal à c_{ij} . On souhaite pour chacun des r agents à disposition élaborer une stratégie de parcours dans ce graphe de telle sorte que chaque noeud du graphe soit visité le plus souvent possible.



FIG. 1 – Simulateur d'agent de patrouille, développé par Aydano *et al.*

Définition 1 Nous appellerons *stratégie d'un agent* la fonction $\pi : \mathbb{N} \rightarrow V$ telle que $\pi(j)$ désigne le $j^{\text{ème}}$ noeud visité par l'agent. Nous désignerons par $\Pi = \{\pi_1 \dots \pi_r\}$ la *stratégie multiagent*, définie comme étant l'ensemble des stratégies $\pi_1 \dots \pi_r$ des r agents. $\Pi^r(G)$ désigne l'ensemble des stratégies multiagents possibles avec r agents sur le graphe G . Lorsqu'il n'y a pas d'ambiguïté sur le nombre d'agents impliqués ni sur le graphe considéré, nous noterons cet ensemble Π .

Connaissant la stratégie d'un agent i , nous pouvons déterminer les instants auxquels chaque noeud est visité. Ainsi, l'agent i ayant la stratégie π_i visitera le noeud $\pi_i(0)$ à l'instant 0. De même, il visitera le noeud $\pi_i(j)$ à l'instant égal au poids du chemin $\pi_i(0) \dots \pi_i(j)$, c'est à dire $\sum_{k=0}^{j-1} c_{\pi_i(k)\pi_i(k+1)}$. Pour alléger l'écriture, nous définirons le poids d'un chemin (et donc d'un cycle) $s_0 \dots s_m$ comme étant la somme des poids des arêtes présentes

dans ce chemin $c(s_0 \dots s_m) = \sum_{i=0}^{m-1} c_{s_i s_{i+1}}$. Par extension, nous désignerons le poids d'un ensemble d'arêtes E' avec la même fonction $c(E')$. Notre objectif étant de créer des bonnes stratégies d'exploration du territoire, il nous faut décider de critères d'évaluation pour ces stratégies. Pour cela, nous introduisons la notion d'*oisiveté d'un noeud* :

Définition 2 *L'oisiveté d'un noeud i au temps t , notée $Idle_t(i)$ est l'intervalle de temps écoulé depuis la dernière visite d'un agent sur ce noeud. Par convention, à l'instant 0, l'oisiveté de chaque noeud est nulle. A partir de l'oisiveté d'un noeud, on peut construire divers critères pour mesurer l'oisiveté du graphe à un instant donné. En voici trois : l'oisiveté maximale, moyenne, et polynomiale : $AvgIdle_t(G) = \frac{1}{n} \sum_{i=1}^n Idle_t(i)$, $MaxIdle_t(G) = \max_{i \in V} Idle_t(i)$, $PolyIdle_t^\alpha(G) = \sqrt[\alpha]{\frac{1}{n} \sum_{i=1}^n Idle_t(i)^\alpha}$*

L'oisiveté polynomiale, inspirée de [8], offre un compromis entre les mesures moyennes et maximales. En effet, selon que le paramètre α est proche de 1 ou de l'infini, l'oisiveté polynomiale tendra vers l'oisiveté moyenne ou maximale. Nous lui préférons néanmoins des critères plus simples à analyser, comme par exemple l'oisiveté maximale. A partir de ces mesure instantanées, on peut définir des critères qui nous serviront à comparer les différentes stratégies multiagents.

Définition 3 *Soit $\Pi = \{\pi_1 \dots \pi_r\}$, une stratégie multiagents de parcours du graphe par r agents. A partir des oisivetés instantanées, nous définissons les critères suivants :*

- $MaxMaxIdle_\Pi(G) = \max_{t \in \mathbb{R}^+} (\liminf_{t' \rightarrow t} MaxIdle_{t'}(G))$, voir figure 2.
- $MaxAvgIdle_\Pi(G) = \max_{t \in \mathbb{R}^+} (\liminf_{t' \rightarrow t} AvgIdle_{t'}(G))$,
- $AvgMaxIdle_\Pi(G) = \lim_{t' \rightarrow \infty} \frac{1}{t'} \int_{t=0}^{t'} MaxIdle_t(G) dt$,
- $AvgAvgIdle_\Pi(G) = \lim_{t' \rightarrow \infty} \frac{1}{t'} \int_{t=0}^{t'} AvgIdle_t(G) dt$.

Dans un souci de simplicité, nous nous limiterons essentiellement à l'étude du critère $MaxMaxIdle$, que l'on peut formuler de façon plus simple : La valeur de $MaxMaxIdle$ est égale au *plus grand intervalle de temps durant lequel un noeud ne reçoit la visite d'aucun agent*. Notons que pour toute stratégie Π , nous aurons : $MaxMaxIdle_\Pi \geq AvgMaxIdle_\Pi \geq AvgAvgIdle_\Pi$ et $MaxMaxIdle_\Pi \geq MaxAvgIdle_\Pi \geq AvgAvgIdle_\Pi$. Le critère choisi $MaxMaxIdle$ étant une borne supérieure des autres critères, sa minimisation produira vraisemblablement des solutions acceptables, relativement aux autres critères. Remarquons par ailleurs que si le poids des arêtes $c_{ij} \in \mathbb{N}$, alors on peut simplifier l'expression du critère de la façon suivante : $MaxMaxIdle_\Pi(G) = \max_{t \in \mathbb{R}^+} \lceil MaxIdle_t(G) \rceil$.

Pour finir cette section, donnons une définition formelle du problème de la patrouille.

Définition 4 (le problème de la patrouille) *Soit un graphe $G = (V, E)$ connexe, ainsi qu'un nombre r d'agents. Le problème de la patrouille consiste à déterminer une stratégie multiagents $\Pi = \{\pi_1 \dots \pi_r\}$ qui minimise $MaxMaxIdle_\Pi(G)$.*

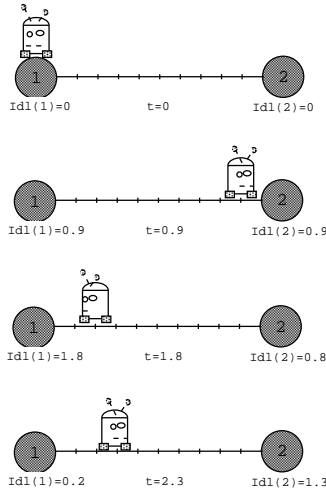


FIG. 2 – Un agent patrouillant un graphe composé de deux noeuds reliés par une arête de poids 1. La stratégie de l’agent est $\pi = 1, 2, 1, 2, 1, \dots$. Après un aller-retour effectué par l’agent (symbolisé par un robot), l’oisiveté de chaque noeud tend vers 2 à mesure que l’agent s’en approche, mais n’atteint jamais 2, puisque l’oisiveté d’un noeud sur lequel l’agent se trouve devient nulle. Donc, $\liminf_{t \rightarrow 2} Idl_t(1) = 2$, et donc $MaxMaxIdl_\pi(G) = 2$.

3 Liens entre le problème du voyageur de commerce et le problème mono-agent de la patrouille

Dans cette section, après avoir rappelé la définition et quelques propriétés du problème du voyageur de commerce (TSP), nous montrerons en quoi ce problème peut être utile pour comprendre et résoudre le problème multi-agents de la patrouille. Nous montrerons en particulier que lorsqu’il n’y a qu’un seul agent, le problème de la patrouille (avec $MaxMaxIdl$ et $AvgAvgIdl$) est équivalent à une certaine variante du TSP.

Définition 5 Soit G , un graphe complet, et c_{ij} un poids associé à chaque couple de sommet (i, j) de G . Le problème du voyageur de commerce consiste à trouver un cycle élémentaire $s_0 \dots s_n$ (c’est à dire ne passant pas deux fois par le même sommet) qui parcourt tous les sommets du graphe une fois et une seule, et tel que $c(s_0 \dots s_n)$ soit minimale.

Le problème du voyageur de commerce ainsi défini suppose que tous les sommets du graphe sont reliés entre eux (graphe complet). Afin de nous rapprocher du problème de

la patrouille, pour lequel les graphes ne sont pas supposés complets, nous utiliserons une variante du problème du voyageur de commerce, connu sous le nom de *graph-TSP* [12], qui porte sur des graphes $G(V, E)$ non nécessairement complets : dès lors, il ne s'agit plus de trouver un cycle élémentaire mais un cycle $s_1 \dots s_m$ passant *au moins une fois* par chaque sommet tel que $\cup_{i=1}^m s_i = V$, et que $c(s_1 \dots s_m)$ soit minimal. Il est montré dans [13] que toute instance du problème de *graph-TSP* peut être efficacement réduite à une instance d'un problème TSP.

Soulignons l'intérêt de mettre en évidence des liens entre les problèmes qui nous intéressent et le TSP. D'abord, il existe de très nombreux algorithmes disponibles publiquement permettant de résoudre des instances du TSP de taille importante. Nous avons pour notre part utilisé le logiciel open-source "Concorde"¹ qui implémente plusieurs algorithmes efficaces, dont l'heuristique *lin-kernighan* [9]. Il est par ailleurs capable de fournir la solution optimale sur des instances comptant jusqu'à plusieurs centaines de noeuds. Un autre intérêt d'une telle réduction provient des connaissances théoriques accumulées sur le TSP. Ainsi, il est connu que sur des graphes respectant l'inégalité triangulaire, le TSP est $\frac{3}{2}$ -approximable, ce qui signifie qu'il existe un algorithme (décrit par Christofides [7] et en $O(n^3)$) capable de générer une solution inférieure à $\frac{3}{2}$ fois la solution optimale. Ce résultat est naturellement valable pour le *graph-TSP*. Nous montrons ci-après que lorsqu'il n'y a qu'un agent, la stratégie de patrouille optimale pour le critère *MaxMaxIdl* consiste à parcourir le cycle correspondant à la solution au *graph-TSP* sur G .

Mentionnons par ailleurs qu'une variante du TSP, connue sous le nom de "*minimum latency TSP*" [4], s'occupe de minimiser le temps moyen d'attente de chaque noeud. Il semble donc que cette variante, qui pose des problèmes très différents des TSP classiques, s'approche du problème mono-agent de la patrouille avec le critère *AvgAvgIdl*. Nous nous focaliserons dans la suite sur le critère *MaxMaxIdl*, du fait qu'il constitue une borne supérieure aux autres critères, et qu'il permet d'établir un lien fort avec les TSP. Ce lien peut être formalisé par le théorème 1. Pour montrer ce théorème, nous aurons besoin du lemme suivant, qui relie dans le cas monoagent le poids d'un cycle parcouru par l'agent et la valeur de la *MaxMaxIdl*.

Lemme 1 *Soit $G = (V, E)$, un graphe. Soit $s_1 \dots s_m$, un cycle sur G . Si $\cup_{i=1}^m s_i = V$ et s'il existe un sommet $x \in V$ présent une seule fois dans le cycle, alors la stratégie π consistant à faire parcourir ce cycle indéfiniment par un unique agent sera telle que $MaxMaxIdl_\pi(G) = c(s_1 \dots s_m)$.*

Preuve. (lemme 1). Pour tout sommet $y \in V$, le temps nécessaire pour partir de y et y revenir au plus égal à $c(s_1, s_2) + c(s_2, s_3) + \dots + c(s_{m-1}, s_m) = c(s_1 \dots s_m)$. Quant au sommet x par lequel le cycle ne passe qu'une seule fois, le temps nécessaire à l'agent

¹logiciel disponible à l'adresse www.math.princeton.edu/tsp/concorde.html

pour partir de x et y revenir est exactement de $c(s_1 \dots s_m)$. Donc, $MaxMaxIdl_\pi(G) = c(s_1 \dots s_m)$. \square

Trivialement, pour tout graphe connexe, un cycle ayant les propriétés requises par ce lemme existe toujours. En particulier le cycle induit par le *graph-TSP* ainsi que le cycle obtenu par l'algorithme de Christofides [7] qui approxime le *graph-TSP* vérifient ces propriétés.

Théorème 1 *Soit G , un graphe et s^{opt} le cycle qui est une solution optimale au problème du graph-TSP sur G . La stratégie consistant pour un seul agent à parcourir indéfiniment ce cycle est la stratégie mono-agent optimale au sens du critère $MaxMaxIdl$.*

Preuve. Soit π , une stratégie monoagent quelconque, dont la $MaxMaxIdl_\pi(G)$ est bornée², ce qui implique que chaque noeud de G est visité infiniment souvent. Par définition, il existe un noeud x et un réel $t_x > 0$ tels que $Idl_{t_x}(x) = Idl_{t_x + MaxMaxIdl_\pi(G)}(x) = 0$ et que durant l'intervalle de temps $]t_x, t_x + MaxMaxIdl_\pi(G)[$, un ensemble de noeuds $V_x \subseteq V \setminus \{x\}$ soient visités. Montrons par l'absurde que $V_x = V \setminus \{x\}$, c'est à dire que tous les noeuds hormis x sont visités durant cet intervalle de temps.

Supposons que $V_x \subset V \setminus \{x\}$. Forcément, il y aura un noeud $z \notin V_x$ qui aura été visité à $t_z < t_x$ puis³ à $t'_z > t_x + MaxMaxIdl_\pi(G)$ et non visité durant $]t_x, t_x + MaxMaxIdl_\pi(G)[$. Par conséquent, l'oisiveté maximale sur le noeud z sera supérieure à $MaxMaxIdl_\pi(G)$, ce qui est une contradiction. On a donc par l'absurde $V_x = V \setminus \{x\}$.

Nous venons donc de montrer que pour toute stratégie π , il existe un instant t_x à partir duquel l'agent positionné sur un noeud x visitera l'ensemble des autres noeuds avant de revenir sur x , et que la durée de cette exploration sera égale à $MaxMaxIdl_\pi(G)$. Cela revient donc à dire que dans toute stratégie π , il existe un cycle (correspondant à l'ensemble des noeuds visités durant $]t_x, t_x + MaxMaxIdl_\pi(G)[$) incluant tous les sommets de G , et dont le poids est égal à $MaxMaxIdl_\pi(G)$.

Donc, pour toute stratégie monoagent π dont la $MaxMaxIdl$ est bornée, il existe une stratégie π' consistant à parcourir indéfiniment un cycle s et tel que $MaxMaxIdl_\pi = MaxMaxIdl_{\pi'} = c(s)$.

Dès lors, une stratégie optimale π^{opt} consistera à parcourir indéfiniment le plus petit cycle incluant tous les sommets du graphe, ce qui correspond exactement à la définition du *graph-TSP*. De plus, par l'application du lemme 1, $MaxMaxIdl_{\pi^{opt}}(G)$ sera égal au poids du cycle induit par le *graph-TSP*. \square

²trivialement, une telle stratégie existe toujours

³Un noeud est visité lorsqu'un agent sur trouve dessus. Néanmoins, du fait que pour tout noeud i on ait $Idl_0(i) = 0$, on considère qu'à l'instant zéro, l'ensemble des noeuds sont visités simultanément.

4 Quatre classes de stratégies multi-agents pour le problème de la patrouille

Nous avons vu dans la section précédente comment utiliser un algorithme de TSP pour résoudre le problème mono-agent de la patrouille. Comment étendre ce résultat au cas multi-agent ? Notons tout d’abord qu’il existe un grand nombre de variantes du problème du voyageur qui puissent être considérées comme multi-agents. Les plus connus d’entre eux sont les *Vehicule Routing Problem* (VRP) [10] et le *multiple-salesman traveling problem* (m-TSP) [6]. Ces deux problèmes supposent l’existence d’un “dépot” dans lequel les agents doivent revenir après leurs visites, ce qui permet dans certains cas de réduire le problème à un simple TSP [6], mais les rend difficilement utilisables pour notre étude. Un autre problème plus récent, nommé “multi-agent TSP” a récemment été introduit dans [17, 16], et traité par des méthodes génétiques dans [14]⁴. Ce problème est très proche du notre, mais il n’existe pas à notre connaissance résultat théorique important le concernant, ni d’algorithme ayant fait ses preuves sur des graphes importants. Afin d’établir des résultats formels sur notre problème, nous introduisons dans cette section des biais importants sur les stratégies envisagées. Plus précisément, nous définissons d’abord deux classes de stratégies principales (illustrées par la figure 3), dite “stratégies à cycle unique” et “stratégies régionalisées”, ainsi que quelques résultats préliminaires sur ces classes. Ces deux classes ont un grand intérêt en elles-mêmes, car elles formalisent des stratégies simples et intuitives. Nous définissons aussi deux autres classes moins intéressantes en elles-mêmes mais qui nous serviront pour démontrer par la suite des résultats d’approximation. En particulier, la dernière stratégie dite “mixte”, permettra d’améliorer les résultats obtenus avec des stratégies à cycle unique.

Définition 6 Soit P , un partitionnement de G en $\{P_1, \dots, P_q\}$ régions connexes (avec $q \leq r$). Dans la suite de ce papier, on désignera par $G_i(P_i, E \cap P_i^2)$ les sous-graphes de G induits par les noeuds de la région i . Toute stratégie multi-agents de parcours de G pour laquelle chaque agent i ne visite les noeuds que d’une région P_j sera appelée stratégie multi-agents régionalisée selon P . On notera $\Pi_{P_1 \dots P_q}$, la classe de toutes les stratégies régionalisées selon $P_1 \dots P_q$, et $\Pi_{\langle P_1, r_1 \rangle \dots \langle P_q, r_q \rangle}$ la classe des stratégies multi-agents régionalisées selon $P_1 \dots P_q$, dans lesquelles chaque région P_i est patrouillée par exactement r_i agents, avec bien sûr $r = \sum_{i=1}^q r_i$. On notera aussi $\Pi_{q-Regions}$ la classe des stratégies régionalisées selon un partitionnement en q régions. Enfin, si le nombre d’agents est égal au nombre de régions ($r = q$), on dit que la stratégie est individuellement régionalisée.

⁴Les auteurs de l’article cité confondent multi-agent TSP et m-TSP : ils traitent le premier et non le dernier.

Ainsi, avec les stratégies individuellement régionalisées, chaque noeud n'est visité que par un seul agent (cf fig. 3). Le lemme suivant nous indique ce qu'est une bonne stratégie individuellement régionalisée.

Lemme 2 Soit $G(V, E)$, un graphe connexe métrique, et $P = \{P_1, \dots, P_r\}$ un partitionnement de ce graphe en r régions chacune connexe. Soit Π_P^{gTSP} , la stratégie individuellement régionalisée consistant à faire parcourir à chaque agent i le cycle induit par le graph-TSP du sous-graphe G_i . Alors, Π_P^{gTSP} est la stratégie individuellement régionalisée selon P qui soit optimale au sens du critère $MaxMaxIdl$.

Preuve. (lemme 2). Par définition, pour toute stratégie individuellement régionalisée $\Pi_P = \{\pi_1, \dots, \pi_r\}$, on a $MaxMaxIdl_{\Pi_P}(G) = \max_i \{MaxMaxIdl_{\pi_i}(G_i)\}$. Il s'ensuit que si la stratégie Π_P est composée des stratégies mono-agent $\pi_1^{opt}, \dots, \pi_r^{opt}$ chacune optimale sur $G_1 \dots G_r$, alors, Π_P est optimale au sens du critère $MaxMaxIdl$. D'après le théorème 1, le cycle induit par le graph-TSP est la stratégie mono-agent optimale, et donc parmi toutes les stratégies de Π_P , celle formée des cycles induits par les graph-TSP sur les sous-graphes G_i est optimale. \square

Introduisons maintenant une nouvelle classe de stratégies, qui nous permettra d'établir un pont avec les TSP, sans avoir à recourir à un partitionnement.

Définition 7 Soit $s_0 \dots s_m$, un cycle (pas nécessairement élémentaire) visitant tous les noeuds de G . Une stratégie multi-agents $\Pi = \{\pi_1 \dots \pi_r\}$ est dite à cycle unique $s_0 \dots s_m$ ssi il existe $d_1 \dots d_r \in \mathbb{N}$ tels que $\pi_i(k) = s_{(k+d_i) \bmod m}$. On notera Π_{Cycle} la classe de toutes les stratégies à cycle unique.

Intuitivement, une stratégie est à cycle unique si tous les agents parcourent le même cycle dans le même sens, mais avec un certain décalage (cf fig. 3). Grâce au lemme suivant, nous montrons qu'à un terme près, r agents parcourant un cycle aboutiront à une $MaxMaxIdl$ r fois plus petite qu'un seul agent parcourant ce même cycle, à la condition que les agents soient espacés de manière régulière sur le cycle.

Lemme 3 Soit $s_0 \dots s_m$, un cycle passant par tous les sommets de G et tel qu'il existe au moins un sommet $x \in V$ par lequel le cycle ne passe qu'une fois. Il existe alors une stratégie $\Pi = \{\pi_1 \dots \pi_r\}$ à cycle unique $s_0 \dots s_m$ telle que $\frac{MaxMaxIdl_{\pi_1}(G)}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq MaxMaxIdl_{\Pi}(G) \leq \frac{MaxMaxIdl_{\pi_1}(G)}{r} + \max\{c_{ij} \mid (i, j) \in E_S\}$. Ici, E_S représente l'ensemble des arêtes du cycle $s_0 \dots s_m$. Ce résultat peut aussi s'écrire $\frac{c(s_0 \dots s_m)}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq MaxMaxIdl_{\Pi}(G) \leq \frac{c(s_0 \dots s_m)}{r} + \max\{c_{ij} \mid (i, j) \in E_S\}$.

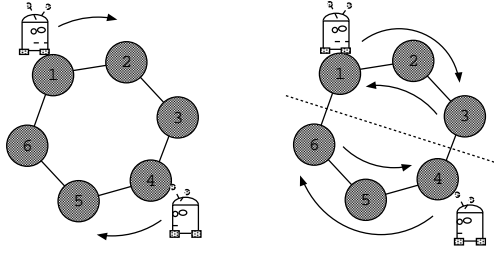


FIG. 3 – Exemples de stratégies à cycle unique $\Pi_{cyc} = \{\pi_{cyc1}, \pi_{cyc2}\}$ (à gauche) et de stratégies régionalisées $\Pi_{Reg} = \{\pi_{reg1}, \pi_{reg2}\}$ (à droite) avec deux agents. On a donc $\pi_{cyc1} = 1, 2, 3, 4, 5, 6, 1, 2, 3, \dots$ et $\pi_{cyc2} = 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, \dots$. Quant à la stratégie régionalisée, on a $\pi_{reg1} = 1, 2, 3, 2, 1, 2, \dots$ et $\pi_{reg2} = 4, 5, 6, 5, 4, 5, \dots$. Par conséquent, $MaxMaxIdl_{\Pi_{cyc}}(G) = 3$ et $MaxMaxIdl_{\Pi_{reg}}(G) = 4$. Ici, la stratégie à cycle unique est donc meilleure que la stratégie régionalisée, mais c'est parfois l'inverse.

Preuve. Soit $s_0 \dots s_m$, le cycle pas nécessairement élémentaire sur G . Considérons deux agents parcourant ce cycle et dont les positions à l'instant 0 soient respectivement les noeuds s_0 et s_d . Soit $l(i, j) = \sum_{k=i}^{j-1} c_{s_k s_{k+1}}$. La valeur $l(0, d)$ est donc le temps nécessaire au premier agent pour rejoindre le noeud s_d en partant de s_0 . Quel que soit le noeud visité par l'agent 2 à l'instant t , ce noeud sera au moins visité par l'agent 1 à l'instant $t + l(0, d)$. Par conséquent, nous aurons $MaxMaxIdl_{\{\pi_1, \pi_2\}}(G) \leq \max\{l(0, d), l(d, m)\}$. Par ailleurs, si à l'instant t_x , l'agent 2 se trouve sur le noeud x , alors du fait que x n'apparaît qu'une seule fois dans le cycle et par application du lemme 1, on sait que la prochaine visite de l'agent 2 à ce noeud x surviendra à l'instant $t_x + l(0, m)$. Par conséquent, après avoir été visité à l'instance t_x , le noeud x ne sera pas visité avant l'instant $t_x + l(0, d)$. Donc, $MaxMaxIdl_{\{\pi_1, \pi_2\}}(G) = \max\{l(0, d), l(d, m)\}$.

Si l'on généralise à r agents, on obtient $MaxMaxIdl_{\Pi}(G) = \max\{l(0, d_1), l(d_1, d_2), \dots, l(d_{r-1}, m)\}$. En choisissant chaque d_k afin qu'il soit le plus grand entier vérifiant $l(0, d_k) \leq k \times \frac{l(0, m)}{r}$, on a en particulier $d_0 = 0$ et $d_r = m$, ce qui nous permet d'écrire $MaxMaxIdl_{\Pi}(G) = \max_{k=0..r-1} l(d_k, d_{k+1})$. Il nous reste à borner $l(d_k, d_{k+1}) = l(0, d_{k+1}) - l(0, d_k)$ de la façon suivante :

$$k \times \frac{l(0, m)}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq l(0, d_k) \leq k \times \frac{l(0, m)}{r}$$

$$(k + 1) \times \frac{l(0, m)}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq l(0, d_{k+1}) \leq (k + 1) \times \frac{l(0, m)}{r}$$

En réunissant ces deux équations, on obtient :

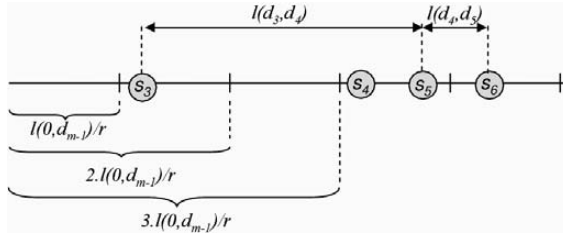


FIG. 4 – Noeuds d’un cycle $s_0 \dots s_m$ sur une échelle représentant la distance qu’un agent doit parcourir pour les visiter.

$$\frac{l(0, m)}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq l(d_k, d_{k+1}) \leq \frac{l(0, m)}{r} + \max\{c_{ij} \mid (i, j) \in E_S\}$$

Le graphique 4 illustre cette situation. Soit $\Pi = \{\pi_1 \dots \pi_r\}$, la stratégie à cycle unique $s_0 \dots s_m$ suivie par r agents et telle que chaque agent i soit positionné à l’instant $t = 0$ sur le noeud s_{d_i} . Du fait que $l(0, m)$ soit le temps nécessaire à un agent pour parcourir le cycle entier, on a pour tout agent i , $MaxMaxIdl_{\pi_i}(G) = l(0, m)$. Du fait que $MaxMaxIdl_{\Pi}(G) = \max_{k=0..m-1} l(d_k, d_{k+1})$, on aura donc le résultat du lemme. \square

Remarquons que le terme $\max\{c_{ij} \mid (i, j) \in E_S\}$ présent dans la borne du critère $MaxMaxIdl$ exposée dans le lemme 3 provient du fait qu’à l’instant 0, chaque agent est obligatoirement positionné sur l’un des noeuds de graphe. Si l’on permettait aux agents d’être positionnés à *mi-chemin* sur les arêtes du graphes au temps 0, alors on aurait naturellement $MaxMaxIdl_{\Pi}(G) = \frac{MaxMaxIdl_{\pi_1}(G)}{r}$.

Pour finir, nous définissons deux autres classes de stratégies : tout d’abord, la classe des stratégies d’aller-retour, qui sera utilisée dans les démonstrations de la section suivante pour obtenir des résultats d’approximation généraux. Ensuite, la classe des stratégies mixtes, qui combine les aspects des stratégies à cycle unique et régionalisées.

Définition 8 Une stratégie monoagent π est dite stratégie d’aller-retour selon un chemin $s_1 \dots s_m$ ssi elle consiste à parcourir ce chemin d’avant en arrière. Concrètement, $\pi = s_1, s_2, \dots, s_{m-1}, s_m, s_{m-1}, \dots, s_2, s_1, s_2, \dots$. Dès lors, $MaxMaxIdl_{\pi} \leq 2c(s_1 \dots s_m)$. Une stratégie multiagents est dite d’aller-retour si elle est composée de stratégies mono-agents d’aller-retour et que tous les noeuds de G sont visités. La classe correspondante sera notée Π_{AR} .

Lemme 4 *Pour toute stratégie multiagents Π , il existe une stratégie $\Pi' = \{\pi'_1 \dots \pi'_r\} \in \Pi_{AR}$ telle que $MaxMaxIddl_{\Pi'}(G) \leq 2 \max_i c(s_0^i \dots s_{m_i}^i) \leq 2 \times MaxMaxIddl_{\Pi}(G)$. Ici, $s_0^i \dots s_{m_i}^i$ désigne le chemin associé à la stratégie monoagent π'_i .*

Preuve. Soit $t_{\Pi} = MaxMaxIddl_{\Pi}(G)$. Soit $s_0^i \dots s_{m_i}^i$ le chemin suivi par l'agent i pendant l'intervalle de temps $[0, t_{\Pi}]$ avec la stratégie Π . Par définition, $\cup \{s_j^i\}_{i=1..r, j=0..m_i} = V$. De plus, $t_{\Pi} \leq \max_i c(s_0^i, \dots, s_{m_i}^i)$. Soit π'_i , la stratégie monoagent d'aller-retour définie sur le chemin $s_0^i \dots s_{m_i}^i$. La stratégie multiagents $\Pi' = \{\pi'_1 \dots \pi'_r\}$ effectue un aller-retour complet (c'est à dire que les agents reviennent à leur position initiale après avoir visité tous les noeuds) en un temps égal à deux fois le poids max des chemins $s_0^i \dots s_{m_i}^i$ et donc inférieur ou égal à $2t_{\Pi}$. \square

Lemme 5 *Pour toute stratégie multiagents d'aller-retour $\Pi = \{\pi_1 \dots \pi_r\}$ dans laquelle chaque stratégie π_i est associée aux chemins $s_1^i \dots s_{m_i}^i$, il existe un cycle S couvrant tous les noeuds de G tel que $\frac{c(S)}{2r} \leq \max_i c(s_1^i \dots s_{m_i}^i) + \max\{c_{ij} \mid (i, j) \in MST(G)\}$. Ici, $MST(G)$ désigne le plus petit arbre couvrant de G .*

Preuve. Considérons $E_{\Pi} = \cup_{i=1}^r \cup_{j=1}^{m_i} \{(s_j^i, s_{j+1}^i)\}$, l'ensemble des arêtes parcourues par les agents. Du fait que les chemins parcourus par les agents peuvent être disjoints, le sous-graphe (V, E_{Π}) n'est pas nécessairement connexe : il est au pire composé de r régions connexes distinctes. Or, il est connu que pour tout partitionnement d'un graphe connexe G en q régions connexes distinctes, l'ensemble des $q-1$ plus petites arêtes reliant ces régions entre elles appartient à $MST(G)$. Donc, il existe T_{Π} , un sous-ensemble de moins de r arêtes de $MST(G)$ tel que $G_{\Pi}(V, E_{\Pi} \cup T_{\Pi})$ soit connexe. Remarquons que

$$\begin{aligned} c(E_{\Pi} \cup T_{\Pi}) &= c(E_{\Pi}) + c(T_{\Pi}) \\ &\leq r \max_i c(s_1^i \dots s_{m_i}^i) + r \max\{c_{ij} \mid (i, j) \in MST(G)\} \end{aligned}$$

Il nous reste à construire un cycle à partir des arêtes de G_{Π} . Soit $T = MST(G_{\Pi})$. Soit S , le cycle défini comme le parcours de T en profondeur d'abord. On a clairement $c(S) \leq 2c(E_{\Pi} \cup T_{\Pi})$. \square

Définition 9 *Soit P , un partitionnement de G en $\{P_1, \dots, P_q\}$ régions connexes avec $q \leq r$. Soient q cycles $\{s_0^i \dots s_{m_i}^i\}$ visitant chacun tous les noeuds de G_i . Soient $r_1 \dots r_q \in \mathbb{N}^*$, tels que $\sum_{i=1}^q r_i = r$. Toute stratégie multi-agents de parcours de G telle que dans chaque G_i , il y ait r_i agents parcourant le sous-graphe en suivant une stratégie à cycle unique $s_0^i \dots s_{m_i}^i$, sera appelée stratégie mixte, et nous désignerons la classe correspondante par $\Pi_{\{(s_0^1 \dots s_{m_1}^1, r_1), \dots, (s_0^q \dots s_{m_q}^q, r_q)\}}$. La classe de toutes les stratégies mixtes sera désignée par Π_{Mixte} .*

Notons que $\Pi_{Cycle} \subset \Pi_{Mixte} \subset \Pi_{Regions}$. Par ailleurs, le lemme 2 nous indique que pour un partitionnement P donné, la stratégie régionalisée optimale est une stratégie mixte.

5 Comparaison entre les stratégies individuellement régionalisées et à cycle unique dans le cas métrique

Dans cette section, nous comparons les deux classes de stratégies présentées précédemment. Le théorème montre que selon le critère $MaxMaxIdl$ et à $3 \max_{ij}\{c_{ij}\}$ près, il existe une stratégie à cycle unique meilleure que toute stratégie régionalisée. Dans le cas où le graphe est unitaire ($c_{ij} \in \{0, 1\}$), alors la stratégie à cycle unique devient nettement plus intéressante que la stratégie régionalisée.

Dans la suite de l'exposé $opt_{\Pi}(G)$, $opt_{\Pi_{r-Regions}}(G)$, $opt_{\Pi_P}(G)$, et $opt_{\Pi_{Cycle}}(G)$ désigneront les valeur optimale du critère $MaxMaxIdl(G)$ respectivement sur les classes de toutes les stratégies multiagents, des stratégies régionalisées, des stratégies individuellement régionalisées selon P , et des stratégies à cycle unique. Lorsqu'il n'y a pas d'ambiguïté, on omettra de mentionner G .

Le lemme suivant, qui montre comment construire un cycle à partir de stratégies individuellement régionalisées, est un préliminaire au théorème principal.

Lemme 6 Soit $G = (V, E)$, un graphe connexe métrique sur lequel doivent patrouiller r agents. Soit $P = \{P_1, \dots, P_r\}$, un partitionnement de ce graphe en r régions chacune connexe. Soit, sur chacune de ces régions P_i , un cycle $s^i = s^i_1 \dots s^i_{m_i}$ couvrant l'ensemble des noeuds de P_i et ne passant jamais deux fois dans le même sens par la même arête⁵. On peut alors en combinant ces cycles créer un cycle unique S vérifiant :

$$c(S) \leq \sum_{i=1}^r c(s^i) + r \times \max_{ij}\{c_{ij}\}$$

Preuve. Soit $E_{intra} = \{(i, j) \in E \mid \exists x \in 1..r, i \in P_x, j \in P_x\}$ et soit $E_{inter} = \{(i, j) \in E \mid \exists x, y \in 1..r, x \neq y, i \in P_x, j \in P_y\}$. Soit \hat{E}_{inter} un plus petit sous-ensemble de E_{inter} , tel que le graphe $(V, E_{intra} \cup \hat{E}_{inter})$ soit connexe. Autrement dit, \hat{E}_{inter} représente un arbre reliant les régions entre elles (voir figure 5).

Soit $E_i^\uparrow = \cup_{j=1}^{m_i-1} (s^i_j, s^i_{j+1})$, l'ensemble des arcs (orienté) intervenant dans le cycle s^i . Du fait que chaque arc (orienté) n'est présent qu'une fois dans s^i , le digraphe (graphe orienté) $G_i^\uparrow(P_i, E_i^\uparrow)$ a la propriété d'être eulérien, c'est à dire qu'il possède un cycle visitant tous ses noeuds et passant une seule fois par chaque arête. De plus, $c(E_i^\uparrow) = c(s^i)$. Soit \hat{E}_{inter}^\uparrow , l'ensemble des arcs (orientés) obtenu en doublant chaque arête de \hat{E}_{inter} .

Rappelons qu'un digraphe est eulérien ssi il est connexe et que pour chacun de ses noeuds, le nombre d'arcs entrants est égal au nombre d'arcs sortants. Par construction,

⁵Clairement, un tel cycle existe toujours.

Procédure visiter($x, i, DejaVisitees$)

1. Soit $s_1^i \dots s_{m_i}^i$, un cycle sur G_i parcourant tous les noeuds de G_i , et ajusté tel que $s_1^i = s_{m_i}^i = x$.
 2. $DejaVisitees \leftarrow DejaVisitees \cup \{i\}$
 3. **Pour** $k = 1 \dots m_i - 1$ **faire**
 4. **Pour toute** arête $(s_k^i, y) \in \hat{E}_{inter}$ telle que la région j de y n'appartienne pas à $DejaVisitees$ **faire**
 5. **renvoyer** (s_k^i, y)
 6. visiter($y, j, DejaVisitees$)
 7. **renvoyer** (y, s_k^i)
 8. **fin pour**
 9. renvoyer (s_k^i, s_{k+1}^i)
 10. **fin pour**
-

\hat{E}_{inter}^\uparrow ne vérifie que la deuxième condition, et il lui manque la connexité pour être eulérien. Par définition, aucun des digraphes $G_1^\uparrow, \dots, G_r^\uparrow, \hat{E}_{inter}^\uparrow$ n'a d'arc en commun. Donc, le digraphe $G^\uparrow(V, \cup_{i=1}^r E_i^\uparrow \cup \hat{E}_{inter}^\uparrow)$ est connexe, et sur chacun de ses noeuds, le nombre d'arcs entrants et sortants est égal. Il est donc eulérien. Il existe donc un cycle S parcourant l'ensemble des noeuds de V et une seule fois chaque arc de $\cup_{i=1}^r E_i^\uparrow \cup \hat{E}_{inter}^\uparrow$. Celui ci vérifie :

$$c(S) = c(\hat{E}_{inter}^\uparrow) + \sum_{i=1}^r c(E_i^\uparrow) = c(\hat{E}_{inter}^\uparrow) + \sum_{i=1}^r c(s^i). \text{ Du fait que } \hat{E}_{inter}^\uparrow \text{ décrit un arbre reliant les } r \text{ régions, il n'a que } r - 1 \text{ arêtes, et donc } c(\hat{E}_{inter}^\uparrow) = 2 \times c(\hat{E}_{inter}) < 2r \times \max_{i,j} \{c_{ij} \mid (i, j) \in E\}. \quad \square$$

Remarquons que le lemme 6 nous montre l'existence d'un cycle S ayant des propriétés intéressantes, sans nous indiquer comment construire ce cycle. Une solution consisterait à extraire du digraphe G^\uparrow un cycle eulérien, par exemple avec l'algorithme de Fleury. Nous proposons un algorithme ad-hoc permettant de construire S : étant donné x , un noeud quelconque de V et P_i , la région à laquelle x appartient, l'appel à la procédure visiter($x, i, \{\}$) renverra une suite d'arcs constituant ce cycle S . La figure 5 illustre le type de cycle obtenu par cet algorithme, et la figure 6 montre un exemple de cycle obtenu par l'algorithme.

Théorème 2 Soit $G(V, E)$, un graphe connexe métrique sur lequel doivent patrouiller r agents. on a : $opt_{\Pi_{Cycle}} \leq opt_{\Pi_{r-Regions}} + 3 \times \max_{i,j} \{c_{ij}\}$.

Preuve. (Théorème 2). La première étape de cette preuve consiste à montrer que le cycle

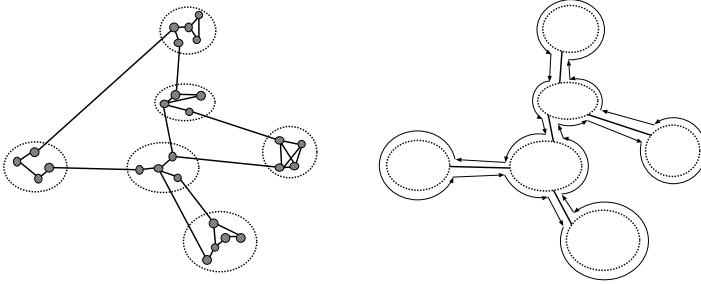


FIG. 5 – Création par l’algorithme **visiter** du cycle destiné à parcourir le graphe. A gauche, un graphe partitionné en régions (cercles en pointillés). A droite, le même graphe (sans le détail des noeuds) dont des arêtes de E_{inter} ont été retirées pour obtenir un arbre reliant les régions entre elles. Les flèches symbolisent le cycle de parcours du graphe, obtenu en combinant les cycles sur chaque région.

induit par le graph-TSP ne passe pas deux fois par la même arête dans le même sens, ceci afin de pouvoir utiliser le lemme 6.

Considérons le cycle $S = x_1, x_2, \dots, x_i, a, b, x_{i+1}, \dots, x_j, a, b, x_{j+1}, \dots, x_m$ qui passe deux fois par (a, b) et dans le même sens. Ce cycle ne peut pas être la solution d’un graph-TSP puisque l’on peut construire un cycle $S' = x_1, x_2, \dots, x_i, a, x_j, \dots, x_{i+1}, b, x_{j+1}, \dots, x_m$ qui visite les mêmes noeuds et dont le poids est inférieur : $c(S') = c(S) - 2 \times c_{a,b}$. Donc, le cycle induit par le graph-TSP ne passe pas deux fois dans le même sens sur une même arête.

Soit $\{P_1 \dots P_r\}$, une partition sur G , et $G_1 \dots G_r$, les sous-graphes associés. Nommons $gTSP(G_i)$ le cycle correspondant au *graph-TSP* sur G_i . Le lemme 2 nous indique que pour tout partitionnement P , on a $opt_{\Pi_P} = \max_{i \in 1..r} \{c(gTSP(G_i))\}$. Grâce au lemme 6 on sait qu’il existe un cycle visitant tous les noeuds de V et vérifiant :

$$\begin{aligned}
 c(S) &\leq \sum_{i=1}^r c(gTSP(G_i)) + 2r \times \max_{ij} \{c_{ij}\} \\
 &\leq r \times \max_{i \in 1..r} \{c(gTSP(G_i))\} + 2r \times \max_{ij} \{c_{ij}\} \\
 &\leq r \times opt_{\Pi_P} + 2r \times \max_{ij} \{c_{ij}\}
 \end{aligned} \tag{1}$$

Utilisons le lemme 3 pour montrer qu’une stratégie à *cycle unique* Π_s construit à partir du cycle S permet d’obtenir les propriétés suivantes :

$$\begin{aligned} \text{MaxMaxIdl}_{\Pi_s}(G) &\leq \frac{c(S)}{r} + \max_{ij}\{c_{ij}\} \\ &\leq \text{opt}_{\Pi_P} + 3 \times \max_{ij}\{c_{ij}\} \end{aligned}$$

Donc, pour toute partition $\{P_1 \dots P_r\}$, il existe toujours une stratégie à cycle unique meilleure que la meilleure stratégie régionalisée sur $P_1 \dots P_r$, à $3 \cdot \max\{c_{ij}\}$ près. \square

Corollaire 1 Soit $G = (V, E)$, un graphe connexe métrique sur lequel doivent patrouiller r agents. Soit $P = \{P_1, \dots, P_r\}$, un partitionnement de ce graphe en r régions connexes. On peut calculer en temps $O(n^3)$ une stratégie multi-agents à cycle unique $\Pi_{Chr} \in \Pi_{Cycle}$ telle qu'on ait $\text{MaxMaxIdl}_{\Pi_s} \leq \frac{3}{2} \text{opt}_{\Pi_P} + 4 \times \max_{ij}\{c_{ij}\}$.

Preuve. (Corrolaire 1) Soit S_{Chr} , le cycle obtenu en utilisant l'algorithme de Christofides [7] sur G . On sait que $c(S_{Chr}) \leq \frac{3}{2}c(S_{gTSP})$, avec S_{gTSP} désignant le cycle induit par le graph-TSP sur G . En utilisant l'équation 1, on obtient :

$$c(S_{Chr}) \leq \frac{3}{2}c(S_{gTSP}) \leq \frac{3r}{2} \times \text{opt}_{\Pi_P} + 3r \times \max_{ij}\{c_{ij}\}$$

Avec le lemme 3, on peut donc conclure que la stratégie multiagents à cycle unique $\Pi_{S_{Chr}}$ selon S_{Chr} vérifie :

$$\begin{aligned} \text{MaxMaxIdl}_{\Pi_{S_{Chr}}}(G) &\leq \frac{c(S_{Chr})}{r} + \max_{ij}\{c_{ij}\} \\ &\leq \frac{3}{2} \text{opt}_{\Pi_P} + 4 \times \max_{ij}\{c_{ij}\} \end{aligned}$$

\square

6 Approximations à partir de stratégies à cycle unique

Dans cette section, nous allons d'abord présenter un premier résultat d'approximation utilisant une stratégie à cycle unique obtenue par un algorithme de *graph-TSP*. Ce résultat comporte le terme $4 \cdot \max_{ij}\{c_{ij}\}$. Dès lors, si le graphe comporte quelques arêtes de très grand poids, la *MaxMaxIdl* obtenue par la stratégie à cycle unique sera d'autant plus grande. Pour finir, nous obtiendrons un résultat d'approximation indépendant de $\max_{ij}\{c_{ij}\}$, grâce à l'utilisation de la classe des stratégies mixtes.

Théorème 3 Soit $G = (V, E)$, un graphe connexe métrique et r , un nombre quelconque d'agents. En $O(n^3)$, on peut trouver une stratégie $\Pi_{Chr} \in \Pi_{Cycle}$ telle que $MaxMaxIdl_{\Pi_{Chr}} \leq 3 \times opt_{\Pi} + 4 \times \max_{ij} \{c_{ij}\}$. De plus, en $O(n \log n)$, on trouve une stratégie $\Pi_{MST} \in \Pi_{Cycle}$ telle que $MaxMaxIdl_{\Pi_{MST}} \leq 4 \times opt_{\Pi} + 4 \times \max_{ij} \{c_{ij} \mid (i, j) \in MST(G)\}$ et que les agents n'empruntent que les arêtes du plus petit arbre couvrant de G .

Preuve. A l'aide du lemme 4, on sait qu'il existe une stratégie $\Pi = \{\pi_1 \dots \pi_r\} \in \Pi_{AR}$ dans laquelle chaque stratégie monoagent π_i est associée à un chemin $s_0^i \dots s_{m_i}^i$, tel que $\max_i \{c(s_0^i \dots s_{m_i}^i)\} \leq opt_{\Pi}$. De plus, pour toute stratégie $\Pi \in \Pi_{AR}$, le lemme 5 nous informe qu'il existe un cycle S parcourant tous les noeuds de G et tel que $\frac{c(S)}{2r} \leq \max_i \{c(s_0^i \dots s_{m_i}^i)\} + \max_{ij} \{c_{ij} \mid (i, j) \in MST(G)\}$. Avec l'algorithme de Christofides [7], on peut obtenir en $O(n^3)$ un cycle S_{Chr} passant par tous les noeuds et de poids inférieur à $\frac{3}{2}$ du poids minimal. De même, avec l'algorithme du plus petit arbre couvrant, on obtient en $O(n \log n)$ un cycle S_{MST} correspondant à l'exploration en profondeur d'abord de cet arbre, et de poids inférieur à 2 du poids minimal. Dès lors :

$$\begin{aligned} c(S_{Chr}) &\leq \frac{3}{2}c(S) \\ &\leq 3r \cdot \max_i \{c(s_0^i \dots s_{m_i}^i)\} + 3r \max_{ij} \{c_{ij} \mid (i, j) \in MST(G)\} \\ &\leq 3r \cdot opt_{\Pi} + 3r \max_{ij} \{c_{ij} \mid (i, j) \in MST(G)\} \\ c(S_{MST}) &\leq 2c(S) \leq 4r \cdot opt_{\Pi} + 4r \max_{ij} \{c_{ij} \mid (i, j) \in MST(G)\} \end{aligned}$$

L'application du lemme 3 nous permet de créer à partir du cycle obtenu par l'algorithme de Christofides une stratégie à cycle unique Π_{Chr} telle que $MaxMaxIdl_{\Pi_{Chr}}(G) \leq \frac{c(S_{Chr})}{r} + \max_{ij} \{c_{ij}\} \leq 3 \cdot opt_{\Pi} + 4 \cdot \max_{ij} \{c_{ij}\}$. De même, $MaxMaxIdl_{\Pi_{MST}}(G) \leq \frac{c(S_{MST})}{r} + \max_{ij} \{c_{ij} \mid (i, j) \in MST(G)\} \leq 4 \cdot opt_{\Pi} + 4 \cdot \max_{ij} \{c_{ij} \mid (i, j) \in MST(G)\}$.

□

7 Approximations à partir de stratégies mixtes

Dans cette section, nous utiliserons les stratégies mixtes pour obtenir un résultat d'approximation qui ne dépende pas de $\max_{ij} \{c_{ij}\}$. Rappelons qu'une stratégie mixte se compose de partitions dans lesquels un ou plusieurs agents se déplacent en parcourant des cycles. Notons que la partition triviale formée d'une seule région ne permet pas d'obtenir un résultat d'approximation ainsi que le suggère le résultat de la section précédente, et que déterminer une partition en r régions "équilibrées" est un problème difficile. Notre

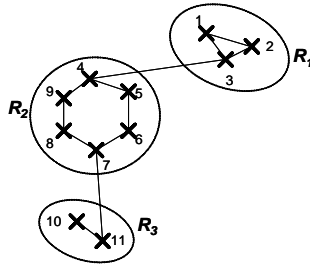


FIG. 6 – Graphe partitionné en 3 régions. Seuls les arêtes issues du graph-TSP de chaque région ainsi que les arêtes de \hat{E}_{inter} sont visibles. Si l’on fournit à l’algorithme **visiter** les cycles issus du graph-TSP et les arêtes de \hat{E}_{inter} , alors l’appel à $visiter(1, 1, \{ \})$ renverra : (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 11), (11, 10), (10, 11), (11, 7), (7, 8), (8, 9), (9, 4), (4, 3), (3, 1).

approche consistera à définir un critère sur les partitions, dont nous montrerons qu’il suffit pour établir un résultat d’approximation.

Tout d’abord, nous définirons la notion de partition $\alpha - propre$, et montrerons comment on peut construire de telles partitions. Par la suite, nous montrerons que pour toute partition $opt_{\Pi} - propre$, il existe une stratégie mixte qui permet d’obtenir le résultat d’approximation voulu. Bien entendu, nous ignorons la valeur de opt_{Π} . Néanmoins, les lemmes 7 et 8 nous montrent comment construire un ensemble d’au plus r partitions parmi lesquelles se trouve nécessairement une partition $opt_{\Pi} - propre$. Le théorème principal de cette section montre que le problème est 15-approximable. Avant d’en débiter la démonstration qui se trouve à la fin de cette section, il nous faut introduire la notion de partition $\alpha - propre$ et en obtenir quelques propriétés.

Définition 10 Soit $G(V, E)$, un graphe connexe. Une partition $\{P_1 \dots P_q\}$ de G est dite $\alpha - propre$ ssi :

- Pour chaque sous-graphe G_i , l’arbre couvrant minimal $MST(G_i)$ ne possède pas d’arête de poids supérieur à α .
- Toute arête de G reliant deux régions est toujours de poids strictement supérieur à α .

Remarquons que si l’on choisit $\alpha < \min_{ij}\{c_{ij}\}$, la seule partition $\alpha - propre$ est $\{\{1\}, \{2\}, \dots, \{n\}\}$. D’autre part, si $\alpha \geq \max_{ij}\{c_{ij}\}$, seul $\{V\}$ est $\alpha - propre$. Le lemme suivant généralise la construction de partitions pour α quelconque.

Lemme 7 Soit $G(V, E)$ un graphe connexe et tel que $\forall (i, j) \in E, c_{ij} > 0$. Il existe une classe $\wp = \{\wp^0 \dots \wp^{n-1}\}$ de partitions de G , telle que chaque partition \wp^k comporte k régions et peut être calculée en $O(n \log n)$, et que $\forall \alpha \geq 0, \exists \wp^k \in \wp, \wp^k$ est $\alpha - propre$.

Preuve. On désigne par $MST(G)$ l'ensemble des arêtes du plus petit arbre couvrant de G , et on peut le calculer en $O(n \log n)$. Soit $a_1 \dots a_{n-1}$, les arêtes de $MST(G)$ triées par ordre de poids décroissant. Soit $T_G^k = MST(G) \setminus \{a_1, \dots, a_k\}$. Clairement, le graphe $G^k(V, T_G^k)$ possède k composantes connexes. On définit donc \wp^k comme étant la partition dont chaque région correspond à une composante connexe de G^k . Montrons que pour $\alpha \in [c_{a_{k+1}}, c_{a_k}[$, \wp^k est $\alpha - propre$.

D'abord, T_G^k a été obtenu en éliminant de l'arbre couvrant minimal de G les arêtes de poids supérieur ou égal à c_{a_k} , ce qui a pour effet de créer des composantes connexes distantes entre elles d'au moins c_{a_k} . Autrement dit, toute arête de G reliant deux régions de \wp^k est de poids supérieur ou égal à c_{a_k} . Ensuite il est clair que le MST de chaque composante connexe de T_G^k est inclus dans $MST(G)$, et donc dans T_G^k . Cela implique qu'aucune arête de poids supérieur à $c_{a_{k+1}}$ n'appartient au MST des partitions de \wp^k . Cela montre bien que pour $\alpha \in [c_{a_{k+1}}, c_{a_k}[$, \wp^k est $\alpha - propre$. Par ailleurs, \wp^0 est la partition comportant une seule région, et est donc $\alpha - propre$ pour $\alpha \in [0, c_{a_{n-1}}[$, et \wp^{n-1} qui est la partition formée de n régions (1 noeud par région) est $\alpha - propre$ pour $\alpha \in [c_{a_1}, \infty[$. Finalement, pour toute valeur de $\alpha \geq 0$, il existe une partition de \wp qui soit $\alpha - propre$. \square

Lemme 8 Soit $G(V, E)$, un graphe et r agents patrouillant ce graphe. Toute partition $\{P_1 \dots P_q\}$ qui soit $(opt_{\Pi}) - propre$ vérifie $q \leq r$.

Preuve. Preuve par l'absurde : supposons qu'on ait une partition $\{P_1 \dots P_q\}$ formée de $q > r$ régions et qui soit $(opt_{\Pi}) - propre$. Du fait que le nombre de régions est supérieur au nombre d'agents, à l'instant zéro, il y aura au moins une région inoccupée ("pigeon hole principle"). Quelque soit la stratégie multiagents Π suivie, pour atteindre cette région, tout agent mettra un temps supérieur à opt_{Π} (puisque toute arête reliant deux régions a un poids supérieur à opt_{Π}), ce qui implique que $\forall \Pi, MaxMaxIddl_{\Pi}(G) > opt_{\Pi}$, ce qui est une impossibilité. \square

Lemme 9 Soit $G(V, E)$, un graphe et r agents patrouillant ce graphe. Pour toute partition $\{P_1 \dots P_q\}$ de G qui soit $opt_{\Pi} - propre$, il existe une distribution $r_1 \dots r_q$ des agents sur les différentes régions et une stratégie $\Pi' \in \Pi_{\langle P_1, r_1 \rangle \dots \langle P_q, r_q \rangle}$ telle que $MaxMaxIddl_{\Pi'}(G) \leq 2 \times opt_{\Pi}$. Il s'ensuit que $opt_{\Pi_{\langle P_1, r_1 \rangle \dots \langle P_q, r_q \rangle}} \leq 2 \times opt_{\Pi}$.

Preuve. Soit Π_{opt} , une stratégie multiagents optimale, et soit $s_1^i \dots s_{m_i}^i$, les noeuds parcourus par l'agent i jusqu'au temps $t = opt_{\Pi}$ selon la stratégie Π_{opt} . Par définition,

$V = \cup_{i=1..r} \cup_{j=1..m_i} \{s_j^i\}$. D'après le lemme 4, la stratégie multiagents Π' obtenue en parcourant ces chemins d'avant en arrière vérifie $MaxMaxIdl_{\Pi'}(G) \leq 2.opt_{\Pi}$. De plus, du fait que $\forall i \in 1..r, c(s_1^i \dots s_{m_i}^i) \leq opt_{\Pi}$, toutes les arêtes figurant dans les chemins $s_1^i \dots s_{m_i}^i$ sont de poids inférieur ou égal à opt_{Π} . Donc, en suivant la stratégie Π' , les agents n'emprunteront que des arêtes de poids inférieur ou égal à opt_{Π} . Or, toute arête reliant deux régions P_i et P_j a un poids nécessairement supérieur à opt_{Π} . Par conséquent, lorsqu'ils suivent la stratégie Π' , les agents ne changent jamais de région. Π' est donc une stratégie régionalisée en plus d'être une stratégie d'allers-retours. \square

Lemme 10 Soit $G(V, E)$, un graphe et r agents patrouillant ce graphe. Soit $\{P_1 \dots P_q\}$, une partition de G qui soit opt_{Π} - propre. Soit S_{MST}^i , le cycle obtenu en parcourant en profondeur d'abord l'arbre couvrant minimal du i^{ieme} sous-graphe $G_i(V, E \cap P_i^2)$. Alors, il existe une répartition $r_1^* \dots r_q^* \in 1..r$ des r agents telle que la stratégie mixte $\Pi_{MST}^{r_1^* \dots r_q^*}$ consistant à faire parcourir chaque cycle S_{MST}^i par r_i^* agents vérifie $MaxMaxIdl_{\Pi_{MST}^{r_1^* \dots r_q^*}}(G) \leq 12.opt_{\Pi}(G)$.

Preuve. Soit une stratégie à cycle unique $\Pi_{MST(G_i)}^{r_i}$ sur la région P_i faisant suivre le cycle S_{MST}^i par r_i agents. D'après le théorème 3, on a alors, pour tout répartition $r_1 \dots r_q$ des agents sur les différentes régions :

$$\begin{aligned} MaxMaxIdl_{\Pi_{MST(G_i)}^{r_i}}(G_i) &\leq 4opt_{\Pi^{r_i}}(G_i) + 4. \max\{c_{ij} \mid (i, j) \in MST(G_i)\} \\ &\leq 4opt_{\Pi^{r_i}}(G_i) + 4.opt_{\Pi}(G) \end{aligned}$$

Notons que le passage de $\max\{c_{ij} \mid (i, j) \in MST(G_i)\}$ à opt_{Π} provient du fait que la partition est opt_{Π} - propre, et donc qu'aucune arête du $MST(G_i)$ n'excède en poids la valeur de opt_{Π} .

Selon le lemme 9, il existe une répartition $r_1^* \dots r_q^*$ des agents sur les différentes régions telle que $opt_{\Pi_{\langle P_1, r_1^* \rangle \dots \langle P_q, r_q^* \rangle}} \leq 2 \times opt_{\Pi}$. On a donc

$$\forall i, \quad opt_{\Pi^{r_i^*}}(G_i) \leq opt_{\Pi_{\langle P_1, r_1^* \rangle \dots \langle P_q, r_q^* \rangle}}(G) \leq 2 \times opt_{\Pi}(G)$$

Par conséquent, pour tout i , on a

$$MaxMaxIdl_{\Pi_{MST(G_i)}^{r_i^*}}(G_i) \leq 8 \times opt_{\Pi}(G) + 4 \times opt_{\Pi}(G)$$

ce qui implique que $MaxMaxIdl_{\Pi_{MST}^{r_1^* \dots r_q^*}}(G) \leq 12 \times opt_{\Pi}$. \square

Remarquons que le lemme ci-dessus utilise les cycles induits par l'arbre couvrant minimal, au lieu d'utiliser les cycles obtenus par l'algorithme de Christofides. Si nous avions opté pour ce dernier choix, nous n'aurions pas été certain que les cycles obtenus ne passaient pas par des arêtes de poids supérieur à opt_{Π} , ce qui aurait rendu la démonstration impossible. Notons que si nous avions autorisé aux agents d'être positionnés à l'instant zéro entre deux noeuds, alors nous aurions pu utiliser l'algorithme de Christofides, en obtenant les mêmes bornes que ci-dessus.

Le lemme ci-dessus ne nous indique pas comment calculer la répartition des agents dans chaque région : il nous dit juste qu'il existe une répartition telle qu'en parcourant les MST des différentes régions, on obtiendra une valeur du critère $MaxMaxI dl$ inférieure à 12 fois l'optimale. Le lemme suivant nous montre qu'il est possible de calculer la répartition $\hat{r}_1 \dots \hat{r}_q$ proche de la répartition optimale.

Lemme 11 *Soit G , un graphe parcouru par r agents, et $\{P_1 \dots P_q\}$, une partition de G qui soit opt_{Π} - propre. Soit S_{MST}^i , le cycle obtenu en parcourant en profondeur d'abord l'arbre couvrant minimal du i^{ieme} sous-graphe $G_i = (V \cap P_i, E \cap P_i^2)$. On peut calculer une répartition $\hat{r}_1 \dots \hat{r}_q$ des agents dans les régions telle que la stratégie mixte $\Pi_{MST}^{\hat{r}_1 \dots \hat{r}_q}$ consistant à faire parcourir chaque cycle S_{MST}^i par \hat{r}_i agents vérifie $MaxMaxI dl(G)_{\Pi_{MST}^{\hat{r}_1 \dots \hat{r}_q}} \leq 14 \times opt_{\Pi}$.*

Preuve. Soit $\hat{r}_1 \dots \hat{r}_q$, la répartition d'agents qui minimise $\max_i (\frac{c(S_{MST}^i)}{\hat{r}_i})$. En appliquant le lemme 3, et en substituant opt_{Π} à $\max_{ij} \{c_{ij} \mid (i, j) \in MST(G_i)\}$, on obtient :

$$\max_i \left(\frac{c(S_{MST}^i)}{\hat{r}_i} \right) - opt_{\Pi} \leq MaxMaxI dl_{\Pi_{MST}^{\hat{r}_1 \dots \hat{r}_q}}(G) \leq \max_i \left(\frac{c(S_{MST}^i)}{\hat{r}_i} \right) + opt_{\Pi} \quad (2)$$

On peut appliquer le même lemme concernant une répartition $r_1 \dots r_q$ quelconque d'agents pour obtenir $MaxMaxI dl_{\Pi_{MST}^{r_1 \dots r_q}}(G) \geq \max_i (\frac{c(S_{MST}^i)}{r_i}) - opt_{\Pi}$. Or, comme $\hat{r}_1 \dots \hat{r}_q$ minimise $\max_i (\frac{c(S_{MST}^i)}{\hat{r}_i})$, on a

$$\forall r_1 \dots r_q \quad MaxMaxI dl_{\Pi_{MST}^{r_1 \dots r_q}}(G) \geq \max_i \left(\frac{c(S_{MST}^i)}{\hat{r}_i} \right) - opt_{\Pi} \quad (3)$$

En, fusionnant les équations 2 et 3, on obtient :

$$\forall r_1 \dots r_q \quad MaxMaxI dl_{\Pi_{MST}^{\hat{r}_1 \dots \hat{r}_q}}(G) \leq MaxMaxI dl_{\Pi_{MST}^{r_1 \dots r_q}}(G) + 2opt_{\Pi} \quad (4)$$

En utilisant le lemme 10, qui indique qu'il existe une répartition d'agents $r_1^* \dots r_q^*$ telle que $MaxMaxI dl_{\Pi_{MST}^{r_1^* \dots r_q^*}}(G) \leq 12opt_{\Pi}$, en déduit :

$$MaxMaxIdl_{\Pi}^{\hat{r}_1 \dots \hat{r}_q}(G) \leq 14.opt_{\Pi} \quad (5)$$

□

Le lemme ci-dessus suppose que l'on peut déterminer les entiers $\hat{r}_1 \dots \hat{r}_q$ qui minimisent $\max_i(\frac{c(S_{MST}^i)}{\hat{r}_i})$ et tels que $\sum_{i=1}^q \hat{r}_i = r$. Montrons comment effectuer ce calcul sous la condition (provisoire) que $\forall i, c(S_{MST}^i) > 0$. Soit $\gamma_i = r \times \frac{c(S_{MST}^i)}{\sum_{k=1}^q c(S_{MST}^k)}$. Le problème revient à maximiser $\min_i(\frac{\hat{r}_i}{\gamma_i})$. Il est clair que les solutions \hat{r}_i au problème vérifient $\hat{r}_i \in \{\lfloor \gamma_i \rfloor, \lceil \gamma_i \rceil\}$. L'algorithme suivant permet de trouver la solution au problème simplement :

1. On affecte $\lceil \gamma_i \rceil$ à chaque variable \hat{r}_i
 - (a) On calcule le nombre d'agents affectés en trop $r^+ = \sum_{i=1}^q \lceil \gamma_i \rceil - r$
 - (b) On détermine parmi toutes les variables $\hat{r}_1 \dots \hat{r}_q$ les r^+ variables qui ont la plus grande valeur de $\frac{\hat{r}_i - 1}{\gamma_i}$ et qui vérifient $\hat{r}_i \geq 2$
 - (c) On décrémente chacune de ces r^+ variables de 1.

Au début de cette note, on a supposé que $\forall i, c(S_{MST}^i) > 0$. Pour traiter le cas où l'un des cycles S_{MST}^i est le poids nul, il suffit d'affecter à ce cycle un agent $\hat{r}_i = 1$, et de résoudre le problème d'affectation de $r - 1$ agents sur tous les autres cycles.

Théorème 4 Soit $G(V, E)$ un graphe connexe. Il existe un algorithme capable de déterminer en $O(r \times n \log n)$ une stratégie mixte Π telle que $MaxMaxIdl_{\Pi}(G) \leq 15.opt_{\Pi}$.

Preuve. (Théorème 4). A l'aide des lemmes 7 et 8, nous pouvons obtenir en $O(r \times n \log n)$ un ensemble de r partitions $P_1 \dots P_r$, parmi lesquelles il existe au moins une partition ($opt_{\Pi} - propre$). Pour chacune de ces partitions P^k , il faudra

1. calculer les cycles $S_{MST}^{i,k}$ pour chaque région i
2. calculer la meilleure répartition $\hat{r}_1^k \dots \hat{r}_q^k$ d'agents sur ces partitions.
3. estimer $M^k = \max_i(\frac{c(S_{MST}^{i,k})}{\hat{r}_i^k}) + \max\{c_{xy} \mid (x, y) \in S_{MST}^{i,k}\}$ qui est une borne supérieure sur le critère $MaxMaxIdl$, selon le lemme 3.

On sait qu'il existe parmi ces r partitions une partition P^{k^*} qui soit $opt_{\Pi} - propre$. Des lors, la borne-sup de cette partition vérifiera : $M^{k^*} \leq MaxMaxIdl_{\Pi^{k^*}}(G) + \max\{c_{xy} \mid (x, y) \in S_{MST}^{i,k^*}\}$ et donc en appliquant le lemme 11, $M^{k^*} \leq MaxMaxIdl_{\Pi^{k^*}}(G) + opt_{\Pi} \leq 15.opt_{\Pi}$. Il est donc clair qu'en sélectionnant partition P^k et la stratégie mixte associée Π^k qui correspondent au M^k minimal sans savoir si cette partition est $opt_{\Pi} - propre$ ou non, nous saurons néanmoins que :

$$MaxMaxIdl_{\Pi^k}(G) \leq M^k \leq M^{k^*} \leq 15.opt_{\Pi}$$

□

8 Conclusion

Une étude théorique du problème de la patrouille a été entreprise. En particulier, il a été montré que des biais forts (stratégies à cycle unique) permettaient de rapprocher notre problème à un simple TSP, grâce auquel on pouvait avoir des solutions très efficaces. L'un des problèmes concernant les stratégies à cycle unique vient de la constante $\max_{ij} \{c_{ij}\}$ dans le résultat d'approximation ; cela rend en particulier cette approche moins intéressante sur des graphes comportant d'importantes variations entre la taille des arêtes. L'utilisation de stratégies mixtes (section 7) a permis de montrer que le problème est 15-approximable sans restriction quant au type de graphe.

Il est probable qu'il existe des liens entre le problème de la patrouille en le "Vehicle Routing Problem", que nous n'avons pas su mettre en évidence. Ce problème est d'ailleurs souvent traité par des approches dites "Route first, Cluster second", ou "Cluster first, Route second"[11], cette dernière approche ressemblant beaucoup à celle que nous avons utilisée dans la dernière section.

Récemment, une étude expérimentale a comparé différents algorithmes sur le problème de la patrouille [1]. Des techniques d'apprentissage par renforcement, de négociation, de méthodes multiagents ont été comparées aux stratégies à cycle unique. Il en ressort que sur l'ensemble des graphes générés, les stratégies à cycle unique donnent de meilleurs résultats que tous les autres algorithmes, quel que soit le nombre d'agents utilisés.

Références

- [1] Almeida, A., Ramalho, G. L., Santana, H. P., Tedesco, P., Menezes, T. R., Corruble, V., Chevalere, Y. (2004). Recent Advances on Multi-Agent Patrolling. In *Advances in Artificial Intelligence ? SBIA 2004 : 17th Brazilian Symposium on Artificial Intelligence*, Sao Luis, Maranhao, Brazil. Lecture notes on Artificial Intelligence 3171, Springer-Verlag.
- [2] Aydano Machado, Geber Ramalho, Jean-Daniel Zucker, Alexis Drogoul (2002). Multi-Agent Patrolling : an Empirical Analysis of Alternative Architectures. Third International Workshop on Multi-Agent Based Simulation.
- [3] Aydano Machado, Almeida Alessandro, Ramalho Geber, Zucker Jean-Daniel, Drogoul Alexis (2002). Multi-Agent Movement Coordination in Patrolling. First Workshop on Agents in Computer Games, at the 3rd International Conference on Computers and Games (CG'02), Edmonton, Canada.
- [4] Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan (1994). The minimum latency problem. In *Proceedings*

- of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing, pages 163-171, Montréal, Québec, Canada.
- [5] Johnston, M. R. (1999). *Dynamic Routing with Competition : Foundations and Strategies*. PhD Thesis, Massey University, Palmerston North, New Zealand.
 - [6] Bellmore, A. and Hong, S. (1974). Transportation of multi-salesman problem to the standard TSP, *Journal ACM* 21, p.500-504.
 - [7] Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. Tech. Rep. CS-93-13, Carnegie Mellon University, Graduate School of Industrial Administration.
 - [8] Crites and Barto (1996). Improving elevator performance using reinforcement learning. *Advances in Neural Information Processing Systems*. pp1017-1023. MIT-Press.
 - [9] R.M. Karp and J.M. Steele (1985). Probabilistic analysis of heuristics, in *The traveling sales- man problem : a guided tour of combinatorial optimization*, E.L.Lawler, J.K. Lenstra, A.H.G.Rinnooy Kan and D.B.Shmoys Eds., Wiley.
 - [10] M. A. Al-Fawzan and K. S. Al-Sultan (1996). The vehicle routing problem : a survey, in *Proceedings of the Fifth Industrial Engineering Research Conference*, pp. 269 - 274.
 - [11] Laporte G., Gendreau M., Potvin J-Y., Semet F. (2000). Classical and Modern Heuristics for the Vehicle Routing Problem, *International Transaction in Operational Research*, vol. 7, pp. 285-300.
 - [12] D. Naddef and G. Rinaldi (1993), *The Graphical Relaxation : A New Framework for the Symmetric Traveling Salesman Polytope*, *Mathematical Programming* 58, 5388.
 - [13] Reinelt G., (1994), "The Traveling Salesman : Computational Solutions for TSP Applications," *Lectures Notes in Computer Science* 840, Springer-Verlag.
 - [14] Donald Sofge, Alan Schultz, and Kenneth De Jong (2002). Evolutionary Computational Approaches to Solving the Multiple Traveling Salesman Problem Using a Neighborhood Attractor Schema. *EvoCOP, EvoWorkshops 2002, Applications of Evolutionary Computing*. LNCS 2279, p. 153.
 - [15] Tangamchit P., Dolan J., Khosla P. (2000). Learning-Based Task Allocation in Decentralized Multirobot Systems. *5th International Symposium on Distributed Autonomous Robotic Systems*, October 4-6, 2000, Knoxville, Tennessee, USA.
 - [16] Andersson, M. and Sandholm, T. (2000). Contract Type Sequencing for Reallocation Negotiation. *International Conference on Distributed Computing Systems (ICDCS)*, Taipei, Taiwan, April.
 - [17] Andersson, M. and Sandholm, T. (1999). Time-Quality Tradeoffs in Reallocation Negotiation with Combinatorial Contract Types. *National Conference on Artificial Intelligence (AAAI)*, pp. 3-10, Orlando, FL.