
Le problème de la patrouille multiagent – Étude de convergence de l'évaluation des stratégies cycliques

Fabrice Lauri¹, Abderrafaa Koukam¹, Jean-Charles Créput¹

Institut de Recherche sur les Transports, l'Énergie et la Société (IRTES)
Laboratoire Systèmes et Transports (SeT) – Rue Thiery-Mieg, Belfort
fabrice.lauri@utbm.fr

RÉSUMÉ. Le problème de la patrouille multiagent implique une équipe d'agents qui doivent visiter les lieux stratégiques d'un environnement le plus fréquemment possible. Ce problème d'optimisation consiste généralement à déterminer une stratégie de patrouille multiagent minimisant la pire oisiveté d'un graphe, c'est-à-dire la plus grande durée pendant laquelle un nœud n'a pas été visité. Nous proposons dans cet article d'étudier de manière théorique l'évaluation des stratégies de patrouille cycliques. Une telle stratégie est constituée de n couples de pré-cycles et de cycles, n étant le nombre d'agents patrouilleurs. Chaque cycle définit la liste des nœuds qu'un agent doit visiter indéfiniment, le premier nœud étant le même que le dernier nœud. Chaque pré-cycle définit la liste des nœuds visités par un agent pour atteindre son cycle de patrouille. Nous présentons dans cet article les conditions de convergence permettant à un algorithme efficace d'évaluer en un nombre fini d'étapes la pire oisiveté de telles stratégies. Un algorithme d'évaluation basé sur ces résultats théoriques est également décrit.

ABSTRACT. Patrolling an environment involves a team of agents whose goal consists in continuously visiting its most relevant areas as frequently as possible. The current research that tackles this complex multi-agent problem usually defines the environment as a graph upon which agents move and seek to optimize the worst idleness of a graph, that is the delay during which a node remains unvisited. This paper deals with the problem of evaluating cyclic multi-agent patrolling strategies, those where every agent ultimately visits the same set of nodes indefinitely. We provide theoretical results about the computation time required for the convergence in the evaluation of any such strategy. An evaluation algorithm is also presented.

MOTS-CLÉS : patrouille multiagent, stratégies de patrouille cycliques, étude théorique.

KEYWORDS: multi-agent patrolling, cyclic patrolling strategies, theoretical results.

DOI:10.3166/RIA.28.1-23 © 2014 Lavoisier

1. Introduction

1.1. Problème de la patrouille multiagent

Une patrouille est une mission impliquant une équipe composée de plusieurs entités mobiles dont le but est de visiter indéfiniment les régions pertinentes d'un environnement, afin de le superviser, le contrôler ou le protéger. Un groupe de drones à la recherche de foyers d'incendies pour contribuer à la conservation d'une forêt, une équipe de robots aspirateurs cherchant la moindre poussière, des postiers réalisant leur tournée quotidienne ou une colonie de fourmis à la recherche de nourriture sont autant d'exemples de patrouilles. Réaliser une telle tâche nécessite une coordination efficace de la part des membres impliqués.

Les techniques de résolution du problème de la patrouille multiagent peuvent être classées en deux familles. Les techniques de la première famille ne disposent que d'une connaissance partielle de l'environnement. Dans ce cas, chaque agent de la patrouille agit de manière plutôt réactive, afin de sélectionner le prochain lieu à visiter. Parmi les travaux abordant la patrouille dans des environnements inconnus, nous pouvons citer ceux de Wagner *et al.* (1998 ; 1999), de Sempé (2004) et de Glad *et al.* (2008 ; 2010). Les techniques de la seconde famille disposent d'une connaissance parfaite de l'environnement. Dans ce cas, l'environnement à patrouiller est généralement modélisé par un graphe. Chaque nœud représente une zone connexe de l'environnement et un arc représente la distance ou le coût du déplacement entre deux zones. Dans cet article, nous nous focalisons essentiellement sur le problème de la patrouille multiagent dans un environnement connu. Les techniques de la seconde famille peuvent être utilisées dans de nombreux domaines, incluant par exemple la gestion de réseaux (Reuter, Baude, 2002), le sauvetage par des robots de personnes en danger (Poulet *et al.*, 2011 ; Kitano, 2000) ou la surveillance d'un territoire face à des menaces ennemies (Jiang *et al.*, 2013 ; Poulet *et al.*, 2011 ; Bošanský *et al.*, 2011 ; Machado, Almeida, *al.*, 2002).

Le problème de la patrouille multiagent dans des environnements connus représentés par des graphes a été formalisé très récemment (Machado, Ramalho, *al.*, 2002). Ce problème consiste à déterminer, avant ou pendant la patrouille effective, pour chaque agent patrouilleur, les nœuds du graphe à visiter afin que la population d'agents minimise un certain critère de performance globale.

1.2. Critère de performance d'une stratégie de patrouille

Ce critère de performance est en général basé sur le concept d'oisiveté d'un nœud (Machado, Ramalho, *al.*, 2002), qui représente la période de temps pendant laquelle le nœud est resté non visité. L'oisiveté d'un nœud est nulle lorsqu'un agent se trouve sur le nœud et elle s'accroît dès que l'agent quitte le nœud. Dans les travaux de (Machado, Almeida, *al.*, 2002 ; Almeida *et al.*, 2004 ; Santana *et al.*, 2004 ; Chevalyere, 2004 ; 2007 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; Marier *et al.*, 2009 ; Poulet *et al.*, 2011), plusieurs stratégies de patrouille ont été envisagées et validées

expérimentalement, en utilisant un critère basé sur l'oisiveté. Par exemple, l'un de ces critères d'évaluation, le critère de pire oisiveté, consiste à déterminer, pour une stratégie de patrouille donnée, la plus grande durée pendant laquelle un nœud est resté non visité. Ce critère est particulièrement adapté pour évaluer une patrouille destinée à collecter régulièrement et le plus vite possible des informations géographiquement distribuées.

D'autres critères de performance ont été utilisés. Par exemple, dans (Agmon *et al.*, 2011), les auteurs s'emploient à minimiser un critère basé sur la fréquence de visite de nœuds prioritaires, pour faire face aux incertitudes environnementales qui peuvent accroître le temps de trajet des agents lors de leur patrouille. Dans (Poulet *et al.*, 2011), les auteurs étudient le problème de la patrouille multiagent sous l'hypothèse qu'un agent peut s'arrêter d'effectuer la tâche de patrouille. Deux critères sont utilisés pour l'évaluation des stratégies. Le premier est basé sur le concept d'oisiveté d'un nœud. Le second, dénommé le temps de stabilisation, consiste à déterminer le temps nécessaire à la patrouille pour atteindre un état stable. Dans (Jiang *et al.*, 2013), les attaques d'adversaires statiques sont régulées par une technique issue de la théorie des jeux et minimisant un critère de performance représentant la probabilité d'apparition d'un évènement. Dans (Božanský *et al.*, 2011), les attaques d'adversaires mobiles sont anticipées par une technique également issue de la théorie des jeux et minimisant la durée d'une attaque.

Dans cet article, nous ne nous intéressons qu'aux travaux qui utilisent un critère de performance basé sur le concept d'oisiveté d'un nœud (Machado, Ramalho, *al.*, 2002 ; Machado, Almeida, *al.*, 2002 ; Almeida *et al.*, 2004 ; Santana *et al.*, 2004 ; Chevalyere, 2004 ; 2007 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014 ; Marier *et al.*, 2009 ; Poulet *et al.*, 2011). Dans ces travaux, les stratégies de patrouille sont déterminées par des approches issues de différents domaines. Des heuristiques permettent à des agents de choisir le meilleur prochain nœud à visiter dans (Machado, Ramalho, *al.*, 2002). Des mécanismes de négociation (Almeida *et al.*, 2004), des techniques d'apprentissage par renforcement (Santana *et al.*, 2004), des techniques basées sur la théorie des graphes (Chevalyere, 2004 ; 2007) ou des métaheuristiques basées sur des colonies de fourmis (Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014) furent également utilisés pour résoudre ce problème. Ces approches permettent d'obtenir des résultats expérimentaux satisfaisants sur des graphes composés d'une centaine de nœuds et d'arcs. Toutes ces techniques fournissent uniquement des solutions approchées à ce problème lorsque le graphe est complexe ou que la taille de la population d'agents est grande.

Dans les travaux abordés précédemment, l'évaluation des stratégies de patrouille repose sur un algorithme itératif, que nous nommerons *SEPS* (*Standard Evaluation of Patrolling Strategies*) par la suite. Cet algorithme modifie la valeur du critère de performance à chaque itération en simulant le déplacement des agents patrouilleurs. Il se termine après un nombre T d'itérations défini a priori par le concepteur. Considérons par exemple que le critère de pire oisiveté est utilisé pour évaluer une stratégie de patrouille multiagent. Notons (c_1, c_2, \dots, c_T) la suite des valeurs de pire oisiveté dé-

terminées au cours des T itérations de l'algorithme d'évaluation et c^* la pire oisiveté théorique associée à une stratégie de patrouille multiagent donnée. La pire oisiveté est initialisée à la valeur 0. À chaque itération t , l'oisiveté de chaque nœud est mise à jour sur la base des déplacements des agents. La pire oisiveté c_t est ensuite calculée comme étant l'oisiveté maximale des nœuds à l'itération t de la simulation. La pire oisiveté c_T associée à une stratégie de patrouille est déterminée ainsi de manière empirique par cet algorithme à l'issue de T itérations de simulation. Il est important de remarquer que, d'une part, pour certaines stratégies de patrouille, la suite des valeurs déterminées par l'algorithme d'évaluation *SEPS* converge vers la valeur théorique de pire oisiveté c^* lorsqu'il existe un nombre d'itérations T_{max} tel que, pour une valeur $\epsilon > 0$ choisie aussi petit que possible, quelque soit $t \geq T_{max}$, $|c_t - c^*| < \epsilon$. Par abus de langage, nous disons dans ce cas que la pire oisiveté calculée (par un algorithme itératif produisant une suite de valeurs) a convergé, ou que la pire oisiveté empirique a convergé. Pour de telles stratégies, *SEPS* peut donc fournir une valeur inexacte lorsque $T < T_{max}$. D'autre part, il peut exister des stratégies de patrouille multiagent ayant une pire oisiveté théorique non bornée. Trivialement, de telles stratégies peuvent s'envisager lorsqu'un nœud n'est pas du tout visité par les agents lors de la patrouille ou lorsque les visites de certains nœuds deviennent de plus en plus espacées au cours d'une patrouille. L'algorithme itératif d'évaluation *SEPS* donnera également une valeur erronée pour de telles stratégies. Dans ce cas, la précision de la pire oisiveté calculée dépend du nombre d'itérations T fixé a priori. Pour que les meilleures stratégies émergent lors du processus de résolution, le nombre d'itérations T doit donc être fixé avec précaution pour garantir une évaluation précise et efficace de n'importe quelle stratégie. Un nombre d'itérations T trop faible peut en effet avoir pour conséquence de supprimer (respectivement de conserver) par erreur de bonnes (respectivement de mauvaises) stratégies de patrouille lorsque le processus de résolution de la technique d'optimisation repose sur une population de solutions. Un nombre d'itérations T trop grand peut alourdir le temps de calcul de la méthode de résolution.

1.3. Contributions et organisation de l'article

La contribution de cet article est quadruple. Premièrement, une formulation plus générale du problème de la patrouille multiagent est posée. Deuxièmement, une nouvelle famille de stratégies de patrouille multiagent est définie, celle des stratégies de patrouille multiagent dites cycliques. Ces stratégies englobent les stratégies à cycle unique, les stratégies mixtes et les stratégies régionalisées étudiées par Chevalyre (2004 ; 2007). L'un des principaux avantages de ces stratégies cycliques est de pouvoir être évaluées de manière exacte en un nombre fini d'étapes. L'autre avantage d'une stratégie cyclique est de pouvoir être représentée par une structure de données de taille finie qui peut être stockée en mémoire. Une illustration d'une telle stratégie est donnée dans la section 4 les définissant. Troisièmement, les fondements théoriques permettant d'évaluer de manière exacte n'importe quelle stratégie de patrouille multiagent cyclique selon le critère de pire oisiveté sont présentés. Les résultats théoriques exposés ici peuvent s'appliquer également lorsque l'évaluation porte sur des stratégies

à cycle unique, régionalisées ou mixtes, car elles représentent des stratégies incluses dans l'ensemble des stratégies cycliques. Quatrièmement, un algorithme efficace, de complexité polynomiale, permettant d'évaluer des stratégies cycliques, est décrit.

Cet article est organisé de la manière suivante. Une formulation plus générale du cadre mathématique du problème de la patrouille multiagent est proposée à la section 2 et la section 3 aborde l'état de l'art de la patrouille dans des environnements connus. Les stratégies de patrouille dites cycliques sont définies à la section 4. Des résultats théoriques sur les conditions de convergence de l'évaluation des stratégies de patrouille cycliques sont exposés à la section 5. Sur la base de ces résultats, est présenté à la section 6 un algorithme permettant d'évaluer en un nombre fini d'étapes toute stratégie de patrouille multiagent cyclique selon le critère de pire oisiveté du graphe. Enfin, la section 7 conclut et indique les directions de recherche envisageables pour poursuivre ces travaux.

2. Formulation mathématique du problème de la patrouille multiagent

L'environnement à patrouiller est représenté par un graphe connexe orienté et pondéré $G = (V, E, c)$. V représente l'ensemble des régions stratégiques pertinentes. $E \subset V^2$ indique les moyens de déplacement entre ces régions et satisfait les propriétés suivantes :

- **Reflexivité** : $(x, x) \in E$, quel que soit $x \in V$ et
- **Connexité** : il existe un chemin (x_1, x_2, \dots, x_n) reliant le nœud $x = x_1$ au nœud $y = x_n$, où $n > 1$, quelque soient $x, y \in V$ et quelque soit $i = 1, \dots, n - 1$, $(x_i, x_{i+1}) \in E$.

La fonction $c : E \rightarrow \mathbb{R}$ est une fonction de coût qui représente une mesure de la distance (en mètre par exemple) séparant le nœud x du nœud y . La fonction de coût respecte les propriétés suivantes :

- $c(x, y) \geq 0$ pour tout $(x, y) \in E$,
- si $x = y$ alors $c(x, y) = 0$

Le graphe est considéré comme orienté pour pouvoir prendre en compte le fait que dans certains cas, pour deux nœuds différents $x, y \in V$, le coût de déplacement entre ces nœuds peut dépendre du sens de déplacement, c'est-à-dire que $c(x, y) \neq c(y, x)$.

Soit $r < |\mathcal{V}|$ le nombre d'agents patrouillant le graphe G . Chaque agent i est initialement déployé sur le nœud $sn_i \in V$ avant la tâche de patrouille et possède une vitesse de déplacement $s_i > 0$ (en m/s par exemple). L'agent i atteint un nœud y à partir d'un nœud x après $\frac{c(x,y)}{s_i}$ unités de temps (la seconde par exemple).

À chaque nœud x est associée une *oisiveté instantanée*, qui représente la période de temps pendant laquelle ce nœud est resté non visité, ainsi qu'un *facteur de pondé-*

ration $\gamma_x \in \mathbb{R}_{+*}$ ¹, qui influence l'accroissement de l'oisiveté instantanée du nœud. Lorsqu'un nœud reçoit la visite d'un agent, son oisiveté instantanée devient nulle. L'oisiveté instantanée d'un nœud x resté non visité pendant une période de temps Δt est égale à $\gamma_x \Delta t$.

On note $\mathcal{I} = (G, r, \vec{sn}, \vec{s}, \vec{\gamma})$ une instance du problème de la patrouille multiagent, avec :

- G le graphe de patrouille,
- r le nombre d'agents patrouilleurs,
- $\vec{sn} \in V^r$ les nœuds de déploiement des agents,
- $\vec{s} \in \mathbb{R}_{+*}^r$ les vitesses des agents et
- $\vec{\gamma} \in \mathbb{R}_{+*}^{|V|}$ les facteurs de pondération des nœuds.

Résoudre le problème de la patrouille multiagent consiste à élaborer une stratégie de couverture $\pi^{\mathcal{I}}$ du graphe G par les r agents, telle que chaque nœud de G est visité indéfiniment. Une telle stratégie de patrouille doit optimiser un critère de qualité donné. Par souci de clarté, les stratégies de patrouille seront maintenant notées π lorsqu'il n'y a pas d'ambiguïté sur l'instance \mathcal{I} .

Soit Π l'ensemble des stratégies de patrouille multiagent possibles.

DÉFINITION 1. — *Une stratégie de patrouille multiagent $\pi = (\pi_1, \pi_2, \dots, \pi_r)$ est constituée de r stratégies individuelles. Chaque stratégie individuelle $\pi_i : \mathbb{N}^* \rightarrow V$ associe un nœud à chaque pas de temps. $\pi_i(j)$ représente le j -ème nœud que l'agent i doit visiter, tel que :*

- $\pi_i(1) = sn_i$, c'est-à-dire que le premier nœud visité par un agent est son site de déploiement,
- $\forall j (\pi_i(j), \pi_i(j+1)) \in E$: n'importe quelle paire de nœud spécifiée dans une stratégie individuelle représente un arc du graphe.

Résoudre le problème consiste à déterminer une stratégie de patrouille multiagent qui minimise un critère de performance associé à chaque nœud $x \in V$. Plusieurs critères ont été proposés (Machado, Ramalho, *al*, 2002) afin d'évaluer la qualité d'une stratégie de patrouille multiagent sur un graphe. Notre analyse théorique se focalise sur l'utilisation du critère basé sur la pire oisiveté du graphe. Le lecteur peut consulter (Machado, Ramalho, *al*, 2002) pour avoir plus de détails sur les autres critères d'évaluation. Notons que le choix de ce critère est généralement justifié puisqu'il représente une borne supérieure des autres critères (Chevaleyre, 2007). En effet, sa minimisation garantit celle des autres.

Tous les critères d'évaluation peuvent être formulés à partir de la notion d'oisiveté instantanée d'un nœud. En supposant que les agents suivent la stratégie de patrouille π sur le graphe G , l'oisiveté instantanée $I_t^\pi(x) \in \mathbb{R}_+$ du nœud x à l'instant t représente

1. $\mathbb{R}_{+*} = \{x \in \mathbb{R} | x > 0\}$

le temps *pondéré* écoulé depuis la dernière visite d'un agent à ce nœud. Plus formellement, si le nœud x a été visité au temps t par un agent et si Δt est le temps écoulé depuis la dernière visite du nœud x , alors l'oisiveté instantanée de x au temps $t + \Delta t$ est donnée par :

$$I_{t+\Delta t}^\pi(x) = \gamma_x \Delta t \quad (1)$$

Les facteurs de pondération peuvent être utilisés pour fixer des visites prioritaires sur les nœuds. Plus le facteur de pondération d'un nœud est grand, plus vite augmente son oisiveté instantanée. Par convention, à l'instant initial, $I_0^\pi(x) = 0$, pour n'importe quelle stratégie π et pour n'importe quel nœud $x = 1, 2, \dots, |\mathcal{V}|$.

La pire oisiveté théorique du graphe associée à une stratégie de patrouille multiagent π est telle que :

$$WI^\pi = \limsup_{t \rightarrow +\infty} WI_t^\pi \quad (2)$$

où WI_t^π représente la pire oisiveté instantanée du graphe, c'est-à-dire la plus grande oisiveté instantanée sur tous les nœuds V du graphe G à l'instant t , telle que :

$$WI_t^\pi = \max_{x \in V} I_t^\pi(x) \quad (3)$$

Plus formellement, résoudre le problème de la patrouille multiagent consiste donc à déterminer une stratégie de patrouille π^* telle que pour n'importe quelle stratégie π , $WI^{\pi^*} \leq WI^\pi$, c'est-à-dire :

$$\pi^* \in \underset{\pi \in \Pi}{\operatorname{argmin}} WI^\pi \quad (4)$$

3. État de l'art du problème de la patrouille multiagent

Les travaux de Machado *et al.* (Machado, Ramalho, *al.*, 2002 ; Machado, Almeida, *al.*, 2002) sont les travaux pionniers sur le problème de la patrouille multiagent dans un environnement représenté par un graphe connu de tous les agents. Les travaux plus récents sont ceux de Almeida *et al.* (2004), Chevaleyre (2004 ; 2007), Santana *et al.* (2004), Lauri et Charpillet (2006) ; Lauri *et al.* (2006) ; Lauri et Charpillet (2008) ; Lauri et Koukam (2008), Marier *et al.* (2009) et Poulet *et al.* (2011).

3.1. Description détaillée

Dans (Machado, Ramalho, *al.*, 2002 ; Machado, Almeida, *al.*, 2002), plusieurs architectures multiagents et plusieurs critères d'évaluation sont proposés. Chaque architecture est la combinaison d'un ensemble de paramètres, comme par exemple la communication entre agents (permise vs. interdite), la portée des perceptions des agents

(locale vs. globale), l'heuristique de sélection du prochain nœud à visiter (aléatoirement, en utilisant une oisiveté individuelle ou partagée, la longueur du chemin, etc.).

Almeida *et al.* (2004) améliora les meilleures architectures proposées par (Machado, Almeida, *al.*, 2002). Les auteurs proposèrent des agents capables d'échanger des messages entre eux et de conduire des négociations au sujet des prochains nœuds à visiter. Chaque agent reçoit aléatoirement un ensemble de nœuds à visiter et utilise un système d'enchères pour vendre ses nœuds indésirables aux autres agents. Chaque agent tente de conserver les nœuds qu'il peut atteindre en un temps raisonnable. Les phases de négociation permettent aux agents d'atteindre un consensus.

Chevaleyre (2004 ; 2007) a formulé le problème de la patrouille multiagent comme un problème d'optimisation combinatoire. Il a tout d'abord prouvé qu'une stratégie de patrouille impliquant un seul agent peut être déterminée en utilisant un algorithme résolvant le problème du voyageur de commerce graphique. Dans cette variante du voyageur de commerce, les graphes ne sont pas nécessairement complets et le voyageur peut passer plusieurs fois par le même nœud. Il a ensuite étudié plusieurs classes de stratégies de patrouille et a démontré qu'elles constituaient des solutions satisfaisantes proches de solutions optimales.

Dans (Santana *et al.*, 2004), les agents sont capables d'apprendre à patrouiller en utilisant un algorithme d'apprentissage par renforcement. Chaque agent tente de résoudre un processus décisionnel de Markov (PDM) afin de déterminer, pour chaque état de l'environnement, la meilleure action à réaliser. Une action permet à un agent de se déplacer vers un nœud adjacent dans le graphe. Un état de l'environnement résume les informations essentielles dont un agent a besoin pour prendre la meilleure décision. Deux architectures d'agents apprenant par renforcement sont proposées. La première considère que les agents ne peuvent pas communiquer, la seconde considère que les agents ne peuvent communiquer que de manière indirecte en se transmettant leur intention sur leur prochaine action. La seconde architecture, nommée *Gray-Box Learner Agents (GBLA)*, donne les meilleurs résultats expérimentaux.

Toutes les approches décrites précédemment ont été évaluées dans (Almeida *et al.*, 2004) et comparées en utilisant douze configurations différentes obtenues à partir de six topologies de graphes et une population de 5 ou 15 agents. Les résultats expérimentaux montrent que la stratégie à cycle unique, l'une des classes de stratégies étudiées dans (Chevaleyre, 2004), est la meilleure dans toutes les configurations sauf une, tandis que les deux meilleures architectures proposées dans (Machado, Ramalho, *al.*, 2002 ; Machado, Almeida, *al.*, 2002) sont les moins convaincantes. Toutes les autres techniques comparées dans (Almeida *et al.*, 2004 ; Santana *et al.*, 2004) ont des performances équivalentes.

Lauri *et al.* (Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014) ont proposé plusieurs techniques basées sur l'optimisation à base de colonies de fourmis (ou *Ant Colony Optimization*), en supposant que tous les agents sont placés initialement sur le même nœud. Dans (Lauri, Charpillet, 2006 ; 2008), tous les agents sont contraints de commencer et de terminer leur cycle de patrouille sur le même nœud

de déploiement. Etant donné la nature forte de l'hypothèse émise, quelques unes des stratégies déterminées par l'algorithme proposé dans ces articles sont clairement suboptimales. Dans (Lauri, Koukam, 2008 ; 2014), les stratégies de patrouille déterminées par l'algorithme à base de colonies de fourmis sont améliorées en introduisant une phase préliminaire à la patrouille. Cette phase consiste à répartir les agents sur les nœuds les plus éloignés les uns des autres. Cette répartition est déterminée par un algorithme évolutionnaire et représente une heuristique pour constituer des pré-cycles. Les agents ne sont donc plus contraints d'initier et de terminer leur patrouille à partir de leur site de déploiement. Le nœud de départ et de fin de leur cycle de patrouille est désormais le nœud assigné pendant la phase de répartition. Les pré-cycles constituent un moyen efficace de coordonner les agents pour leur future patrouille. Les résultats expérimentaux conduits pour valider les deux techniques proposées dans (Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008) montrent qu'elles surpassent GBLA pour la majorité des graphes utilisés et pour des populations composées de 2 à 20 agents. La plupart des stratégies calculées dans (Lauri, Koukam, 2008 ; 2014) sont des stratégies régionalisées, selon la terminologie adoptée par Chevalyere (Chevalyere, 2004). Dans (Lauri *et al.*, 2006), les auteurs proposent d'étudier le problème de la patrouille dans le cadre des processus décisionnels de Markov. Ils proposent un algorithme de résolution basé sur la programmation dynamique pour résoudre des instances modestes.

Marier *et al.* (Marier *et al.*, 2009) formalise le problème de la patrouille multiagent comme un processus décisionnel semi-markovien généralisé (PDSMG). Cette modélisation leur permet de prendre en compte des patrouilles définies en temps continu et pouvant faire face à de nombreuses incertitudes (voir ci-dessus). Ils proposent un algorithme *anytime* basé sur une heuristique de recherche dans l'espace d'états du PDSMG et l'évaluent sur des graphes d'au plus 50 nœuds avec une population formée de deux agents.

Poulet *et al.* (Poulet *et al.*, 2011) formule une variante du problème de la patrouille multiagent, en introduisant des priorités sur les nœuds à visiter, des critères métriques à minimiser et une population d'agents patrouilleurs de taille variable. Des expériences ont été menées pour évaluer les performances des techniques proposées dans (Almeida *et al.*, 2004) et (Chevalyere, 2004) sur des instances de cette variante du problème. Les résultats empiriques montrent que les stratégies à cycle unique de (Chevalyere, 2004) et celles formées à partir d'une des heuristiques proposées dans (Almeida *et al.*, 2004) sont clairement meilleures que les autres.

3.2. Comparaison des hypothèses de travail

Dans ces travaux, les auteurs émettent les hypothèses suivantes.

- L'environnement à patrouiller est soit un graphe non pondéré (Lauri *et al.*, 2006 ; Machado, Ramalho, *al.*, 2002 ; Machado, Almeida, *al.*, 2002), soit un graphe métrique (Machado, Ramalho, *al.*, 2002 ; Machado, Almeida, *al.*, 2002 ; Almeida *et al.*, 2004 ; Chevalyere, 2004 ; 2007 ; Santana *et al.*, 2004 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014 ; Marier *et al.*, 2009 ; Poulet *et al.*, 2011).

– Les instances traitées peuvent être de grande taille (des graphes d’une centaine de nœuds, plusieurs dizaines d’agents patrouilleurs) (Machado, Ramalho, *al*, 2002 ; Machado, Almeida, *al*, 2002 ; Almeida *et al.*, 2004 ; Santana *et al.*, 2004 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014 ; Poulet *et al.*, 2011) ou limitées en taille (une cinquantaine de nœuds au maximum, moins de 5 agents) (Marier *et al.*, 2009 ; Lauri *et al.*, 2006).

– Tous les agents ont la même vitesse de déplacement, c’est-à-dire que $s_i = 1$ pour tout $1 \leq i \leq r$, sauf dans (Lauri, Koukam, 2014).

– Tous les nœuds ont la même priorité de visite, c’est-à-dire que $\gamma_x = 1$ pour tout $x \in V$ (Machado, Ramalho, *al*, 2002 ; Machado, Almeida, *al*, 2002 ; Almeida *et al.*, 2004 ; Chevaleyre, 2004 ; 2007 ; Santana *et al.*, 2004 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; Poulet *et al.*, 2011) ou certains nœuds sont prioritaires (Lauri, Koukam, 2014 ; Marier *et al.*, 2009).

– Les sites de déploiement des agents représentent une partie des données du problème (Machado, Ramalho, *al*, 2002 ; Machado, Almeida, *al*, 2002 ; Almeida *et al.*, 2004 ; Santana *et al.*, 2004 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014 ; Marier *et al.*, 2009 ; Poulet *et al.*, 2011) ou ils sont déterminés par l’approche de résolution (Chevaleyre, 2004 ; 2007).

– Les incertitudes liées aux capteurs ou à la collecte d’informations disponibles sur les nœuds du graphe sont prises en compte (Marier *et al.*, 2009), ou non (Machado, Ramalho, *al*, 2002 ; Machado, Almeida, *al*, 2002 ; Almeida *et al.*, 2004 ; Chevaleyre, 2004 ; 2007 ; Santana *et al.*, 2004 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014 ; Poulet *et al.*, 2011).

– La taille de la population d’agents est statique (Machado, Ramalho, *al*, 2002 ; Machado, Almeida, *al*, 2002 ; Almeida *et al.*, 2004 ; Chevaleyre, 2004 ; 2007 ; Santana *et al.*, 2004 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014 ; Marier *et al.*, 2009) ou dynamique (Poulet *et al.*, 2011).

– Les stratégies de patrouille sont dynamiques et déterminées en temps-réel (Machado, Ramalho, *al*, 2002 ; Machado, Almeida, *al*, 2002 ; Almeida *et al.*, 2004 ; Marier *et al.*, 2009 ; Poulet *et al.*, 2011) ou elles sont statiques et calculées hors ligne (Chevaleyre, 2004 ; 2007 ; Santana *et al.*, 2004 ; Lauri, Charpillet, 2006 ; 2008 ; Lauri, Koukam, 2008 ; 2014).

– Aucun cycle de patrouille n’est déterminé explicitement (Machado, Ramalho, *al*, 2002 ; Machado, Almeida, *al*, 2002 ; Almeida *et al.*, 2004 ; Marier *et al.*, 2009 ; Poulet *et al.*, 2011), un cycle de patrouille est déterminé pour chaque agent à partir de son site de déploiement (Santana *et al.*, 2004 ; Chevaleyre, 2004 ; 2007 ; Lauri, Charpillet, 2006 ; 2008), ou une phase transitoire précède la patrouille proprement dite, appelé pré-cycle, où chaque agent se déplace éventuellement de son site de déploiement jusqu’au premier nœud de son cycle de patrouille, est déterminée conjointement avec un cycle de patrouille (Lauri, Koukam, 2008 ; 2014).

3.3. Limitations des approches existantes et apports de nos travaux

Dans (Portugal, Rocha, 2011), les auteurs indiquent un certain nombre de limitations inhérentes aux techniques existantes de recherche de stratégies de patrouille multiagent. Le manque d'études sur la flexibilité des approches proposées ou sur l'utilisation effective des ressources, ainsi que la nature déterministe de beaucoup d'approches centralisées existantes font partie des limitations ainsi soulevées.

D'un point de vue théorique, nous pensons qu'une des autres limitations des travaux existants est l'utilisation de critères d'évaluation mal définis (voir page 3, dernier paragraphe) pour certaines classes de stratégies de patrouille ou l'utilisation de critères qui ne traduisent pas toujours la qualité d'une stratégie en fonction de l'application visée. Le présent article émet les hypothèses de travail suivantes :

- les stratégies de patrouille multiagent considérées sont des stratégies cycliques consistantes,
- les graphes de patrouille sont pondérés,
- les nœuds peuvent avoir des priorités de visite différentes,
- les agents peuvent avoir des vitesses de déplacement différentes,
- les agents peuvent être déployés sur des nœuds différents.

Les quatre dernières hypothèses généralisent celles émises dans les précédents travaux.

Les stratégies de patrouille cycliques consistantes seront définies dans la section suivante. Le critère de pire oisiveté associé à de telles stratégies est bornée, pour les raisons qui seront abordées également dans la section suivante. Ces stratégies de patrouille cycliques peuvent être employées plus particulièrement pour des tâches de patrouille destinées à collecter régulièrement et le plus vite possible des informations géographiquement distribuées.

4. Stratégies de patrouille cycliques

4.1. Formalisation

DÉFINITION 2 (Stratégie de patrouille cyclique). — *Une stratégie de patrouille π est cyclique si et seulement si chaque stratégie individuelle π_i est définie par un couple (μ_i, l_i) tel que :*

- $\mu_i = (\mu_i(1), \mu_i(2), \dots, \mu_i(N_i))$ est une séquence finie de N_i nœuds visités par l'agent i ,
- $l_i \in \{1, 2, \dots, N_i\}$ est l'indice du premier nœud commençant le cycle,
- $\mu_i(1) = sn_i$, c'est-à-dire que le premier nœud visité par l'agent i est son nœud de déploiement,
- $\mu_i(l_i) = \mu_i(N_i)$ représente le nœud commençant et terminant le cycle de l'agent i ,

– le j -ème nœud $\pi_i(j)$ effectivement visité par un agent i est défini comme :

$$\pi_i(j) = \begin{cases} \mu_i(j) & \text{pour } j < N_i \\ \mu_i(l_i + (j - l_i) \bmod (N_i - l_i)) & \text{pour } j \geq N_i \end{cases} \quad (5)$$

Les stratégies individuelles d'une stratégie de patrouille cyclique sont caractérisées par l'existence d'un cycle et éventuellement d'un pré-cycle, définis respectivement comme suit.

DÉFINITION 3 (Cycle de patrouille). — *Le cycle de patrouille $\text{cyc}(\pi, i)$ d'un agent i dans une stratégie de patrouille cyclique π est la séquence finie des nœuds de π_i visités indéfiniment par l'agent i , tel que $\text{cyc}(\pi, i) = (\pi_i(l_i), \pi_i(l_i + 1), \dots, \pi_i(N_i))$ avec $\pi_i(l_i) = \pi_i(N_i)$.*

DÉFINITION 4 (Pré-cycle de patrouille). — *Le pré-cycle de patrouille de l'agent i dans une stratégie cyclique multiagent π est la séquence de nœuds de π_i visités par l'agent i à partir de son nœud de déploiement sn_i jusqu'au nœud $\pi_i(l_i)$ commençant son cycle de patrouille. Lorsque $l_i = 1$, π_i ne contient pas de pré-cycle.*

DÉFINITION 5 (Stratégie cyclique consistante). — *Une stratégie de patrouille cyclique est consistante si et seulement si chaque nœud de G est visité indéfiniment par au moins un agent dans son cycle de patrouille.*

Cette propriété de consistance garantit que le critère de pire oisiveté associé à une stratégie de patrouille cyclique est borné, étant donné que chacun des nœuds du graphe est assuré d'être visité régulièrement par au moins un agent.

Il est important de noter que c'est en particulier la prise en compte explicite des pré-cycles, d'une part dans la structure d'une stratégie de patrouille et d'autre part dans l'évaluation d'une stratégie de patrouille, qui nous a permis de formaliser le concept de stratégies cycliques puis de découvrir des résultats avantageux sur les conditions de convergence de l'évaluation exacte de telles stratégies lorsque le critère de pire oisiveté est utilisé.

Par la suite, Π^{cyclic} désigne l'ensemble des stratégies de patrouille multiagent cycliques consistantes pour une instance donnée du problème.

4.2. Exemple

Pour illustrer les notions de cycle et de pré-cycle, considérons le graphe représenté à la figure 1, patrouillé par deux agents déployés sur le nœud 1.

Soit la stratégie de patrouille cyclique $\pi = (\pi_1, \pi_2)$, tel que :

- $\pi_1 = ((1, \mathbf{4,5,6,8,7,4}), 2)$ et
- $\pi_2 = ((\mathbf{1,2,3,2,1}), 1)$.

où les cycles de patrouille sont indiqués en gras.

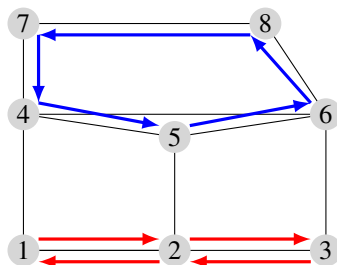


Figure 1. Graphe composé de 8 nœuds et de 11 arcs, patrouillé par deux agents. Les flèches représentent la séquence des nœuds visités par les agents lors de leur cycle de patrouille

Dans cette stratégie de patrouille, l'agent 1 visite indéfiniment les nœuds 4, 5, 6, 8 et 7 : ces nœuds forment son cycle de patrouille. Le pré-cycle de l'agent 1 est représenté par le chemin (1, 4). L'agent 2 effectue directement son cycle de patrouille, formé des nœuds 1, 2 et 3, sans avoir préalablement parcouru un pré-cycle.

4.3. Motivation

L'étude des stratégies de patrouille multiagent cycliques est grandement motivé par l'intuition que nous avons quant à la structure des stratégies de patrouille multiagent optimales lorsque le critère de pire oisiveté est utilisé. Cette intuition peut être formulée par la conjecture suivante :

PROPOSITION 6. — *Pour n'importe quelle instance du problème de la patrouille multiagent, parmi l'ensemble des stratégies de patrouille multiagent possibles, il existe au moins une stratégie de patrouille multiagent cyclique qui est optimale au sens du critère de la pire oisiveté du graphe.*

À ce jour, nous n'avons pas trouvé de contre-exemples permettant de réfuter cette affirmation.

5. Évaluation efficace des stratégies de patrouille multiagent cycliques

Déterminer une stratégie optimale $\pi^* \in \Pi^{cyclic}$ qui minimise l'équation 2 (section 2) implique de construire puis d'évaluer plusieurs stratégies $\pi \in \Pi^{cyclic}$ lors de la recherche. L'efficacité d'un algorithme capable de résoudre une instance large du problème de la patrouille multiagent dépend donc fortement du temps de calcul requis pour notamment évaluer une stratégie cyclique. En effet, plus le nombre de stratégies évaluées par l'approche de résolution est grand, plus la probabilité de découvrir une stratégie optimale est grande. Cette section fournit les preuves théoriques sur les conditions de convergence de l'évaluation de n'importe quelle stratégie de patrouille multiagent cyclique. Un algorithme efficace d'évaluation exacte de telles stratégies est présenté à la section suivante.

Nous utiliserons les notations suivantes dans la suite de cette section :

- $\mathbb{I}_{\{p\}}$ est la fonction qui retourne 1 lorsque le prédicat p est satisfait et 0 sinon.
- $c(\pi_i, x) = \sum_{k=1}^{j-1} c(\pi_i(k), \pi_i(k+1))$ est le coût du chemin à partir du nœud de déploiement $\pi_i(1)$ jusqu'au nœud $x = \pi_i(j)$.
- $\mu = (\mu(1), \mu(2), \dots, \mu(N_\mu))$ est un chemin composé de N_μ nœuds à partir de $\mu(1)$ jusqu'à $\mu(N_\mu)$ et tel que $\forall i (\mu(i), \mu(i+1)) \in E$.
- $c(\mu) = \sum_{k=1}^{N_\mu-1} c(\mu(k), \mu(k+1))$ est le coût de μ .
- $E(\mu) = \{\mu(k) | 1 \leq k \leq N_\mu\}$ est l'ensemble des nœuds apparaissant dans la séquence de nœuds μ .
- $n^\pi(x) = \sum_{k=1}^r \mathbb{I}_{\{x \in E(\text{cyc}(\pi, k))\}}$ est le nombre d'agents visitant le nœud x dans leur cycle de patrouille.
- $n^{\pi_i}(x) = \sum_{j=l_i}^{N_i} \mathbb{I}_{\{x=\pi_i(j)\}}$ est le nombre de fois que le nœud x apparaît dans le cycle de patrouille de π_i .
- $S(\pi, i) = \{x^* | x^* \in \text{argmin}_{x \in E(\text{cyc}(\pi, i))} n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}\}$ est l'ensemble des nœuds du cycle de patrouille π_i qui sont le moins souvent visités par l'agent i .
- $p(\pi, i) \in S(\pi, i)$ est la *nœud pivot* du cycle de patrouille de π_i de l'agent i . Le nœud pivot est le nœud le moins souvent visité dans le cycle de patrouille et tel qu'il apparaît à la position $k < N_i$ dans π_i , avec k aussi proche que possible de N_i .
- $WI_T^\pi(x) = \limsup_{t \rightarrow T} I_t^\pi(x)$ est la pire oisiveté du nœud x après une période de temps T lorsque les agents suivent la stratégie de patrouille π .
- $WI^\pi(x)$ est la pire oisiveté du nœud x lorsque les agents suivent π pendant une période de temps assurant sa convergence. En d'autres termes, $WI^\pi(x) = WI_\infty^\pi(x)$.
- $WI_T^\pi = \limsup_{t \rightarrow T} WI_t^\pi$ est la pire oisiveté du graphe G après une période de temps T lorsque les agents suivent la stratégie de patrouille π .

Nous pouvons énoncer le problème de convergence de l'évaluation d'une stratégie de patrouille cyclique de la manière suivante :

"Identifier les conditions nécessaires et suffisantes déterminant, pour n'importe quelle stratégie cyclique $\pi \in \Pi^{\text{cyclic}}$, la période de temps T^π garantissant que $WI^\pi = \limsup_{t \rightarrow T^\pi} WI_t^\pi$, c'est-à-dire tel que pour tout $T > T^\pi$, $\limsup_{t \rightarrow T} WI_t^\pi = \limsup_{t \rightarrow T^\pi} WI_t^\pi$."

Soit $P_x(n)$ la propriété suivante définie pour tout nœud x :

$P_x(n)$: "La pire oisiveté $WI^\pi(x)$ de n'importe quel nœud $x \in V$ visité par n agents converge après une période de temps $T_x(n) = \min \{T_{x,i}\}_{1 \leq i \leq n}$ où $T_{x,i}$ correspond à la période de temps nécessaire à l'agent i pour qu'il visite le nœud x exactement $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ fois².

2. Par souci de clarté, les n agents d'indices $i = 1, 2, \dots, n$ sont supposés visiter le nœud x .

Nous allons tout d'abord démontrer dans le théorème 1 qui suit que la propriété $P_x(n)$ associée à un nœud x donné est vraie pour tout n . La démonstration est basée sur un raisonnement par induction. Le lemme 1 définit une borne supérieure sur la pire oisiveté $WI^\pi(x)$ associée à un nœud x visité uniquement par un seul agent. Cette borne supérieure dépend du temps écoulé pour que cet agent visite $1 + \mathbb{I}_{\{x=\pi_i(l_i)\}}$ fois le nœud x . Le lemme 2 stipule que la propriété $P_x(n)$ est vraie pour $n = 1$, en s'appuyant sur le lemme 1.

LEMME 7. — *La pire oisiveté d'un nœud x visité indéfiniment par un seul agent i , c'est-à-dire tel que $x \in E(\text{cyc}(\pi, i))$ et $n^\pi(x) = 1$, satisfait l'équation :*

$$WI^\pi(x) \leq \frac{\gamma_x}{s_i} \max\{c(\text{cyc}(\pi, i)), c(\pi_i, x)\} \quad (6)$$

PREUVE 8. — Soit t_x le temps écoulé jusqu'à la première visite au nœud $x \in E(\text{cyc}(\pi, i))$. Alors, avant la patrouille proprement dite, la pire oisiveté de x après une période de temps t_x est égale à :

$$WI_{t_x}^\pi(x) = \limsup_{t \rightarrow t_x} I_t^\pi(x) = \frac{\gamma_x}{s_i} c(\pi_i, x)$$

La pire oisiveté de x converge après une période de temps $T > t_x$ qui correspond à la durée nécessaire à l'agent i pour compléter son cycle de patrouille une fois et revenir au nœud x . Dans le cas où le nœud x apparaît une seule fois dans $\text{cyc}(\pi, i)$ ou que x est le nœud de départ du cycle de patrouille et qu'il apparaît exactement deux fois dans π_i , c'est-à-dire que $n^{\pi_i}(x) = 1 + \mathbb{I}_{\{x=\pi_i(l_i)\}}$, alors sa pire oisiveté satisfait l'équation :

$$WI^\pi(x) = \begin{cases} WI_{t_x}^\pi(x) & \text{si } \frac{\gamma_x}{s_i} c(\text{cyc}(\pi, i)) \leq WI_{t_x}^\pi(x). \\ \frac{\gamma_x}{s_i} c(\text{cyc}(\pi, i)) & \text{sinon} \end{cases}$$

Si $n^{\pi_i}(x) > 1 + \mathbb{I}_{\{x=\pi_i(l_i)\}}$, alors l'équation ci-dessus devient une inégalité inférieure. Donc dans le cas général, $WI^\pi(x) \leq \frac{\gamma_x}{s_i} \max\{c(\text{cyc}(\pi, i)), c(\pi_i, x)\}$. ■

LEMME 9. — *La pire oisiveté d'un nœud x visité indéfiniment par un seul agent i converge après une période de temps $T_{x,i}$ correspondant à la durée nécessaire à l'agent i pour visiter le nœud x $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ fois exactement. En d'autres termes, $P_x(1)$ est vérifié.*

PREUVE 10. — Le lemme 7 stipule que la pire oisiveté d'un nœud x visité indéfiniment par un seul agent i converge dès que l'agent i a complété son cycle de patrouille une fois et est revenu au nœud x . Si $x = \pi_i(l_i)$, alors x apparaît au moins deux fois dans $\text{cyc}(\pi, i)$. Dans ce cas, la pire oisiveté de x converge une fois que x a été visité exactement $n^{\pi_i}(x)$ fois. Si $x \neq \pi_i(l_i)$, alors x apparaît au moins une fois dans

$\text{cyc}(\pi, i)$. Dans ce cas, la pire oisiveté de x converge dès que x a été visité exactement $n^{\pi_i}(x) + 1$ fois. ■

THÉORÈME 11. — *La pire oisiveté $WI^\pi(x)$ de n'importe quel nœud $x \in V$ visité par n agents converge après une période de temps $T_x(n) = \min \{T_{x,i}\}_{1 \leq i \leq n}$, où $T_{x,i}$ correspond à la durée nécessaire à l'agent i pour qu'il visite le nœud x exactement $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ fois.*

PREUVE 12. — Supposons que $P_x(n)$ est vraie et prouvons par induction que $P_x(n+1)$ est également vraie. $P_x(n+1)$ signifie que : "La pire oisiveté $WI^\pi(x)$ de tout nœud $x \in V$ visité par $n+1$ agents converge après une période de temps $T_x(n+1) = \min \{T_x(n), T_{x,n+1}\}$ ". Si $T_x(n) \leq T_{x,n+1}$, puisque $P_x(n)$ est vraie, alors $WI^\pi(x) = WI_{T_x(n)}^\pi(x)$. Si $T_{x,n+1} < T_x(n)$, c'est-à-dire si l'agent $n+1$ visite le nœud x plus rapidement que les autres, alors $WI^\pi(x) = WI_{T_{x,n+1}}^\pi(x)$. Donc $WI^\pi(x) = WI_{T_x(n+1)}^\pi(x)$, ce qui stipule que $P_x(n+1)$ est vraie sachant que $P_x(n)$ est vraie. Puisque $P_x(1)$ est vraie (lemme 9) et que pour tout n , $(P_x(n) \Rightarrow P_x(n+1))$, alors $P_x(n)$ est vraie pour tout n . ■

De ce théorème nous pouvons en déduire le corollaire suivant :

COROLLAIRE 13. — *La pire oisiveté de chaque nœud x converge après une période de temps $T_x(n^\pi(x))$, c'est-à-dire :*

$$WI^\pi(x) = WI_{T_x(n^\pi(x))}^\pi(x) = \limsup_{t \rightarrow T_x(n^\pi(x))} I_t^\pi(x) \quad (7)$$

PREUVE 14. — Chaque nœud x est visité indéfiniment par $n^\pi(x)$ agents. En utilisant le théorème 11, $WI^\pi(x) = WI_{T_x(n^\pi(x))}^\pi(x)$. ■

Nous démontrons dans le théorème 2 que le critère de pire oisiveté converge une fois que les pires oisivetés de tous les nœuds ont convergé.

THÉORÈME 15. — *La pire oisiveté du graphe G lorsque les agents suivent la stratégie de patrouille cyclique π converge après une période de temps T^π , c'est-à-dire :*

$$WI^\pi = \limsup_{t \rightarrow T^\pi} WI_t^\pi \quad (8)$$

où :

$$T^\pi = \max_{x \in V} T_x(n^\pi(x)) \quad (9)$$

T^π représente la période de temps nécessaire à la convergence de la pire oisiveté de chacun des nœuds. T^π correspond également à la durée nécessaire pour que chaque agent i visite chaque nœud x de son cycle de patrouille $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ fois.

PREUVE 16. — L'équation 2 peut être reformulée comme suit :

$$\begin{aligned}
WI^\pi &= \max_{x \in V} \limsup_{t \rightarrow +\infty} I_t^\pi(x) \\
&= \max_{x \in V} WI^\pi(x) \\
&= \max_{x \in V} \limsup_{t \rightarrow T_x(n^\pi(x))} I_t^\pi(x) \tag{10}
\end{aligned}$$

$$\begin{aligned}
&= \limsup_{t \rightarrow T^\pi} \max_{x \in V} I_t^\pi(x) \tag{11} \\
&= \limsup_{t \rightarrow T^\pi} WI_t^\pi
\end{aligned}$$

L'équation 10 mène à l'équation 11 en utilisant l'équation 9. ■

Le théorème 3 ci-dessous introduit le critère d'arrêt de l'algorithme *AECPS* (*Accurate Evaluation of Cyclic Patrolling Strategies*) qui permet une évaluation exacte et efficace des stratégies cycliques de patrouille. Ce critère correspond à la durée nécessaire T^π pour que chaque agent i visite au total $\sum_{x \in E(\text{cyc}(\pi, i))} (n^{\pi_i}(x) + 1)$ nœuds de son cycle.

THÉORÈME 17. — *Les propositions suivantes sont équivalentes :*

- T^π correspond à la durée nécessaire pour que chaque agent i visite chacun des nœuds x de son cycle de patrouille $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ fois.
- T^π correspond à la durée nécessaire pour que chaque agent i visite son nœud pivot $x = p(\pi, i)$ exactement $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ fois.
- T^π correspond à la durée nécessaire pour que chaque agent i visite au total $\sum_{x \in E(\text{cyc}(\pi, i))} (n^{\pi_i}(x) + 1)$ nœuds de son cycle.

PREUVE 18. — Pour n'importe quel agent i , lorsque le nœud pivot $x = p(\pi, i)$ a été visité $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ fois par l'agent i , tous les autres nœuds $y \neq x$ dans π_i ont été visités au moins $n^{\pi_i}(y) + \mathbb{I}_{\{y \neq \pi_i(l_i)\}}$ fois. En effet, lorsque l'agent i a complété son cycle de patrouille la première fois, chaque nœud y a été visité $n^{\pi_i}(y)$ fois. Autrement dit, à l'issue de son premier cycle, l'agent a rendu visite à $\sum_{x \in E(\text{cyc}(\pi, i))} n^{\pi_i}(x)$ nœuds au total. Le second cycle de patrouille permet à l'agent i de visiter chaque nœud une fois de plus. Donc chaque nœud est assuré d'être visité un nombre de fois suffisant à la convergence de sa pire oisiveté dès que :

- le nœud pivot a été visité au second cycle de patrouille ou
- le nombre total de nœuds visités par l'agent i lors de son cycle de patrouille est égal à $\sum_{x \in E(\text{cyc}(\pi, i))} (n^{\pi_i}(x) + 1)$. ■

6. Évaluation des stratégies cycliques de patrouille multiagent

Nous présentons maintenant l'algorithme *AECPS* (*Accurate Evaluation of Cyclic Patrolling Strategies*). Sur la base des résultats théoriques précédents, cet algorithme

permet d'évaluer en temps polynomial et de manière exacte n'importe quelle stratégie cyclique de patrouille multiagent. Une analyse de sa complexité exprimée en termes de temps de calcul est détaillée à la section 6.2. Une comparaison empirique entre *AECPS* et l'algorithme itératif standard *SEPS* est enfin présentée à la section 6.3.

6.1. Algorithme *AECPS*

L'algorithme présenté ci-dessous calcule la pire oisiveté *WI* associée à une stratégie cyclique π de patrouille de r agents évoluant sur un graphe G , en prenant en compte les vitesses des agents et les facteurs de pondérations des nœuds.

Entrées: Graphe de patrouille G , nombre d'agents $r \in \mathbb{N}$, vitesses des agents $\vec{s} \in \mathbb{R}_{+*}^r$, facteurs de pondération des nœuds $\vec{\gamma} \in \mathbb{R}_{+*}^{|V|}$, stratégie de patrouille cyclique $\pi = (\pi_1, \pi_2, \dots, \pi_r) \in \Pi^{cyclic}$.
Sortie: Pire oisiveté du graphe $WI \in \mathbb{R}_+$.

```

1:  $I(x) \leftarrow 0$  pour chaque nœud  $x \in V$ 
2:  $WI \leftarrow 0$ 
3: Pour chaque agent  $i \in [1; r]$  Faire
4:    $cn(i) \leftarrow 1$ 
5:    $pn(i) \leftarrow 2$ 
6:    $d(i) \leftarrow c(\pi_i(cn(i)), \pi_i(pn(i)))$ 
7:    $n(i) \leftarrow \sum_{x \in E(cyc(\pi, i))} (n^{\pi_i}(x) + 1)$ 
8: Fin Pour
9: Répéter
10:   $\Delta_t \leftarrow \min_{i \in [1; r]} \frac{d(i)}{s_i}$ 
11:  Pour chaque nœud  $x \in V$  Faire
12:     $I(x) \leftarrow I(x) + \gamma_x \times \Delta_t$ 
13:  Fin Pour
14:   $WI \leftarrow \max(WI, \max_{x \in V} \{I(x)\})$ 
15:  Pour chaque agent  $i \in [1; r]$  Faire
16:     $d(i) \leftarrow d(i) - \Delta_t \times s_i$ 
17:    Si  $d(i) = 0$  Alors
18:      Si  $cn(i) \geq l_i$  et  $n(i) > 0$  Alors
19:         $n(i) \leftarrow n(i) - 1$ 
20:      Fin Si
21:    Mettre à jour les indices  $cn(i)$  et  $pn(i)$ , respectivement du nœud courant
    et du prochain nœud à atteindre.
22:     $d(i) \leftarrow c(\pi_i(cn(i)), \pi_i(pn(i)))$ 
23:     $I(\pi_i(cn(i))) \leftarrow 0$ 
24:  Fin Si
25: Fin Pour
26: Jusqu'à  $n(i) = 0$  pour tout agent  $i \in [1; r]$ 

```

Les différentes structures de données utilisées pour calculer la pire oisiveté WI du graphe G sont initialisées de la ligne 1 à la ligne 8, c'est-à-dire :

- l'oisiveté instantanée $I(x)$ de chaque nœud x ,
- la pire oisiveté du graphe WI ,
- l'indice $cn(i)$ du nœud courant de chaque agent i , en sachant que le nœud courant de chaque agent i est donné par $\pi_i(cn(i))$,
- l'indice $pn(i)$ du prochain nœud à atteindre pour chaque agent i ,
- le nombre total $n(i)$ de nœuds que chaque agent i doit visiter une fois entré dans son cycle de patrouille,
- la distance $d(i)$ séparant chaque agent i du prochain nœud.

La ligne 10 permet de déterminer le délai minimal nécessaire à un agent pour atteindre son prochain nœud. Les lignes 11 à 13 permettent de mettre à jour les oisivetés instantanées des nœuds du graphe. La mise à jour de la pire oisiveté WI du graphe est réalisée à la ligne 14. De la ligne 16 à la ligne 24, chaque agent i se déplace pendant un délai Δ_t sur l'arc reliant son nœud courant au prochain nœud indiqué par sa stratégie individuelle de patrouille π_i . Si la distance parcourue est suffisante (lignes 16 et 17) pour atteindre son prochain nœud, les nœuds courant et prochain (ligne 21) ainsi que la distance les séparant (ligne 22) sont mis à jour pour assurer le prochain déplacement. Les lignes 18 à 20 décomptent le nombre de nœuds restant à visiter lorsque l'agent entre dans son cycle de patrouille. À la ligne 23, l'oisiveté du nœud courant de l'agent i est annulée.

Les déplacements des agents s'arrêtent lorsque la convergence du critère de pire oisiveté est atteinte, ce qui est le cas lorsque chaque agent i a visité au total le nombre $n(i)$ (initialisé en ligne 5) de nœuds dans son cycle de patrouille (test à la ligne 26).

6.2. Analyse de complexité

Soient :

- n le nombre de nœuds du graphe G ,
- r le nombre d'agents patrouilleurs,
- $m = \max_{i \in [1, r]} (N_i - l_i + 1)$ la longueur du plus long des r cycles de patrouille,
- k le nombre d'itérations permettant d'atteindre la convergence de la pire oisiveté.

La complexité en terme de temps de calcul de l'algorithme *AECPS* est en $\mathcal{O}(rmn + kn + kr)$. En effet :

- la ligne 1 est en $\mathcal{O}(n)$,
- les lignes 4 à 6 sont en $\mathcal{O}(r)$,
- la ligne 7 est en $\mathcal{O}(rmn)$, car pour chacun des r agents, le calcul du nombre de fois $n^{\pi_i}(x)$ où chaque nœud x est dans le cycle est en $\mathcal{O}(m)$ et la somme est en $\mathcal{O}(n)$,

- la ligne 10 et les lignes 15 à 25 sont en $\mathcal{O}(kr)$ et
- les lignes 11 à 14 sont en $\mathcal{O}(kn)$.

Notons que *SEPS* (l’algorithme itératif standard se terminant à l’issue de T itérations) a une complexité en temps de calcul de $\mathcal{O}(Tn + Tr)$. Dans l’algorithme *SEPS*, la ligne 7 et les lignes 18 à 20 sont absentes. Le test de la ligne 26 est remplacé par celui sur le nombre restant d’itérations à exécuter. Cependant *SEPS* ne garantit pas la convergence de la pire oisiveté, comme l’illustre la comparaison empirique ci-dessous.

6.3. Comparaison empirique de *AECPS* et *SEPS*

Pour souligner l’importance de disposer d’un algorithme d’évaluation exacte et efficace, nous avons conduit une série d’expériences d’évaluation de stratégies de patrouille en utilisant le graphe *MapA* (figure 2). Ce graphe est communément utilisé pour ce problème dans les travaux précités. Six stratégies de patrouille ont été évaluées successivement par les algorithmes *AECPS* et *SEPS*. Chacune de ces stratégies correspond à l’implication de 2, 5, 10, 15, 20 ou 25 agents, respectivement. Ces stratégies de patrouille ont été obtenues par l’algorithme présenté dans (Lauri *et al.*, 2008b).

Les résultats de ces expériences ont été reportés dans le tableau 1, où :

- r désigne le nombre d’agents patrouilleurs,
- k représente le nombre d’itérations nécessaire à la convergence du critère de pire oisiveté pour l’algorithme *AECPS*,
- T est le nombre d’itérations utilisé par l’algorithme *SEPS*,
- WI indique la valeur de la pire oisiveté.

Les temps d’évaluation indiqués sont exprimés en secondes et représentent le temps de calcul de 1000 évaluations successives d’une stratégie de patrouille. Les pires oisivetés ayant convergé sont indiquées en gras.

Tableau 1. Comparaison empirique de l’algorithme d’évaluation proposé *AECPS* et de l’algorithme d’évaluation standard *SEPS*.

r	<i>AECPS</i>			<i>SEPS</i> $T = 50$		<i>SEPS</i> $T = 100$		<i>SEPS</i> $T = 500$	
	k	WI	Temps	WI	Temps	WI	Temps	WI	Temps
2	131	6 733	0,045	5 175	0,018	6 733	0,03	6 733	0,16
5	190	4 026	0,077	2 768	0,021	4 026	0,042	4 026	0,20
10	322	2 520	0,16	1 854	0,026	2 469	0,051	2 520	0,24
15	434	2 477	0,25	1 725	0,035	2 432	0,064	2 477	0,29
20	543	2 348	0,36	1 569	0,041	2 090	0,075	2 348	0,35
25	523	2 283	0,39	1 475	0,045	1 890	0,084	2 283	0,37

Nous pouvons constater que seul l’algorithme *AECPS* permet de déterminer la pire oisiveté théorique en un minimum de temps pour l’ensemble des stratégies de

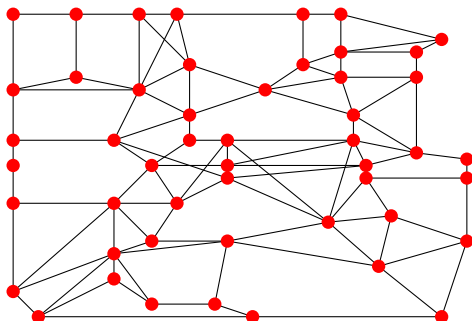


Figure 2. Graphe MapA: 50 nœuds et 104 arcs

patrouille impliquant 2, 5, 10, 15, 20 ou 25 agents. En effet, lorsque 50 itérations sont utilisées dans *SEPS*, la valeur de la pire oisiveté qui est déterminée n'a pas convergée et est loin de la valeur théorique, même si le temps de calcul est très faible. Lorsque 100 itérations sont employées, seules les stratégies impliquant 2 et 5 agents sont évaluées précisément par *SEPS* tout en nécessitant moins de temps de calcul que *AECPS*. Il est important de noter en effet qu'*AECPS* garantit la convergence de la pire oisiveté, mais pas une exécution en un temps minimum. Ceci provient du lemme 1 définissant une borne supérieure sur la pire oisiveté de tout nœud, à la base des démonstrations portant sur la convergence de l'évaluation de toute stratégie de patrouille cyclique et donc directement lié au critère d'arrêt de l'algorithme itératif *AECPS*. Lorsque *SEPS* s'exécute en 500 itérations, les pires oisivetés théoriques sont déterminées mais le temps de calcul devient conséquent par rapport à celui utilisé par *AECPS*, particulièrement lorsque le nombre d'agents impliqués est faible.

Ces résultats illustrent parfaitement la difficulté de maîtriser le compromis existant dans l'algorithme *SEPS* entre la précision de calcul du critère de performance et le temps d'exécution. Ce compromis est absent de l'algorithme *AECPS*.

7. Conclusion et perspectives

Les techniques permettant de résoudre le problème de la patrouille multiagent peuvent être utilisées dans de nombreuses applications, incluant par exemple la gestion de réseaux, la robotique mobile et les jeux vidéos. Etant donné que ce problème peut être considéré comme NP-difficile, seules des solutions approchées peuvent être déterminées lorsque la complexité du graphe et la taille de la population d'agents patrouilleurs sont grandes.

Un travail pionnier a été conduit dans cet article dans le but d'étudier les conditions de convergence de l'évaluation de la pire oisiveté des stratégies de patrouille multiagent appartenant à la classe des stratégies dites cycliques. De cette étude résulte l'énoncé d'un théorème prouvant que l'évaluation exacte de ces stratégies peut être réalisée en un nombre fini d'étapes. Un algorithme d'évaluation efficace, de complexité polynomiale, fondé sur ce théorème, est enfin proposé.

Plusieurs directions de recherche mériteraient d'être empruntées pour justifier et enrichir ces travaux. Premièrement, une étude plus approfondie au sujet de la comparaison entre la procédure standard *SEPS* et *AECPS* serait à envisager. En effet, les résultats montrent clairement le gain de *AECPS* pour un petit nombre d'agents mais ce n'est plus le cas lorsque le nombre d'agents augmente. Avoir une idée du nombre d'itérations T nécessaire à *SEPS* en fonction de la taille du graphe et du nombre d'agents permettrait de mieux apprécier la difficulté de paramétrage de cet algorithme et donc de privilégier *AECPS*. Deuxièmement, l'apport d'une preuve confirmant la conjecture 1 permettrait d'étendre significativement la portée des présents travaux. Troisièmement, il serait intéressant d'établir les conditions de convergence de l'évaluation de n'importe quelle stratégie de patrouille multiagent, selon différents critères de performance (pire oisiveté du graphe ou moyenne oisiveté du graphe par exemple). Enfin, concevoir et valider expérimentalement une technique permettant de générer des stratégies de patrouille multiagent cycliques permettrait de compléter l'arsenal des algorithmes existants déterminant des stratégies de patrouille multiagent. L'évaluation efficace de ces stratégies permet dorénavant d'accroître le nombre de solutions envisagées par un algorithme et donc la probabilité de déterminer une solution optimale.

Bibliographie

- Agmon N., Urieli D., Stone P. (2011). Ship Patrol: Multiagent Patrol under Complex Environmental Conditions. In *Autonomous Agents and Multi-Agent Systems*.
- Almeida A., Ramalho G., al. (2004). Recent Advances on Multi-Agent Patrolling. In *Brazilian Symposium on Artificial Intelligence*, p. 474-483.
- Bošanský B., Lisý V., Jakob M., Pechoucek M. (2011). Computing Time-Dependent Policies for Patrolling Games with Mobile Targets. In *Autonomous Agents and Multi-Agent Systems*.
- Chevalere Y. (2004). Theoretical Analysis of the Multi-Agent Patrolling Problem. In *International Joint Conference on Intelligent Agent Technology*, p. 302-308.
- Chevalere Y. (2007). Combinatorial Optimization and Theoretical Computer Science. In V. T. Paschos (Ed.), chap. The Patrolling Problem: theoretical and experimental results. Wiley.
- Glad A., Simonin O., Buffet O., Charpillet F. (2008). Theoretical Study of Ant-Based Algorithms for Multi-Agent Patrolling. In .
- Glad A., Simonin O., Buffet O., Charpillet F. (2010). Influence of Different Execution Models on Patrolling Ant Behaviors: from Agents to Robots. In *Autonomous Agents and Multi-Agent Systems*.
- Jiang A., Yin Z., Zhang C., Tambe M., Kraus S. (2013). Game-theoretic Randomization for Security Patrolling with Dynamic Execution Uncertainty. In *Autonomous Agents and Multi-Agent Systems*.
- Kitano H. (2000). RoboCup Rescue : A Grand Challenge for Multi-Agent Systems. In *International Conference on Multi Agent Systems*, p. 5-12.
- Lauri F., Charpillet F. (2006). Ant Colony Optimization applied to the Multi-Agent Patrolling Problem. In *IEEE Swarm Intelligence Symposium*.

- Lauri F., Charpillet F. (2008). Résolution du problème de la patrouille multi-agent en utilisant des colonies compétitives de fourmis. In *Revue d'Intelligence Artificielle*, vol. 22, p. 63–85.
- Lauri F., Charpillet F., Szer D. (2006). Analyse théorique du problème de la patrouille multi-agent en utilisant le cadre des processus décisionnels de Markov. In *Journées Francophones sur la Planification, la Décision et l'Apprentissage*.
- Lauri F., Koukam A. (2008). A Two-Step Evolutionary and ACO Approach for Solving the Multi-Agent Patrolling Problem. In *IEEE World Congress on Computational Intelligence*.
- Lauri F., Koukam A. (2014). Hybrid ACO/EA Algorithms applied to the Multi-Agent Patrolling Problem. In *IEEE World Congress on Computational Intelligence*.
- Machado A., Almeida A., *al.* (2002). Multi-Agent Movement Coordination in Patrolling. In *International Conference on Computer and Game*.
- Machado A., Ramalho G., *al.* (2002). Multi-Agent Patrolling : an Empirical Analysis of Alternatives Architectures. In *International Workshop on Multi-Agent Based Simulation*, p. 155–170.
- Marier J.-S., Besse C., Chaib-draa B. (2009). A Markov Model for Multiagent Patrolling in Continuous Time. In *International conference on neural information processing: Part ii*, p. 648–656.
- Portugal D., Rocha R. (2011). A Survey on Multi-robot Patrolling Algorithms. In *Technological innovation for sustainability*, vol. 349, p. 139–146.
- Poulet C., Corruble V., Seghrouchni A., Ramalho G. (2011). The Open System Setting in Timed MultiAgent Patrolling. In *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*.
- Reuter E., Baude F. (2002). System and Network Management Itineraries for Mobile Agents. In *International Workshop on Mobile Agents for Telecommunications Applications*, p. 227–238.
- Santana H., Ramalho G., *al.* (2004). Multi-Agent Patrolling with Reinforcement Learning. In *Autonomous Agents and Multi-Agent Systems*, p. 1122–1129. Consulté sur <http://jmvidal.cse.sc.edu/library/santana04a.pdf>
- Sempé F. (2004). *Auto-organisation d'une collectivité de robots – Application à l'activité de patrouille en présence de perturbation*. Thèse de doctorat non publiée, Université Pierre et Marie Curie (Paris, France).
- Wagner I., Lindenbaum M., Bruckstein A. (1998). Efficiently Searching a Graph by a Smell-Oriented Vertex Process. In *Annals of Mathematics and Artificial Intelligence*, vol. 24, p. 211–223.
- Wagner I., Lindenbaum M., Bruckstein A. (1999). Distributed Covering by Ant-Robots Using Evaporating Traces. In *IEEE Transactions on Robotics and Automation*, vol. 15, p. 918–933.