République Algérienne Démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid -El Taref
Faculté des Sciences el da le Technologie
Département D'Informatique

الجمهورية الديمقراطية الشعبية الجزائرية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد - الطارف
كلية العلوم والتكنولوجيا
قسم الاعلام الالي

MEMOIRE

Présenté par

DRICI Hacene

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Une Approche de Planification d'une Trajectoire basée sur les Systèmes Multi Agents

Soutenu le: 15/09/ 2022 Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Président	Melle. Ziani A	MCB	Chadli Bendjedid El-Tarf
Rapporteur	Mr. Benmachiche A	MCA	Chadli Bendjedid El-Tarf
Examinateur	Mme. Bougarne I	MCB	Chadli Bendjedid El-Tarf

Année Universitaires: 2021/2022

Remerciment

Nous remerciment en premier lieu ; Dieu le tout puissant, le seigneur de tous les temps, pour son aide et sa grande miséricorde, car c'est grâce à lui que nous sommes arrivés à accomplir et à présenter ce modeste travail.

Nous tenons à exprimer nos vifs remerciements à notre encadreur Abdelmadjid Benmachiche ;

Pour Les considérables qu'il a fourni afin de nous aider et nous éclairer ces précieux conseils et dont l'amabilité, le soutient morale et la patiente exemplaire nous ont aidés à mener à bien notre travail, et surtout pour sa gentillesse.

Je tenue à remercier tous les membres du Jury, pour le grand honneur qu'ils m'ont fait en acceptant d'examiner ce travail et d'y apporter leurs critiques constructives et leurs valeureuses remarques.

Nous remercions également tous les responsables, les personnels et les enseignants du département Math et Informatique pour tout le savoir qu'ils nous ont transmis durant le cycle universitaire.

Grand merci à toute personne ayant aidé de près ou de loin à l'accomplissement de ce travail.

Dedicase

A ceux qui ont la première faveur dans ma vie,

A ceux qui ont consacré toute leurs vies pour

Nous guider et surveiller, à ceux que je dois ce

Que je suis,

A la personne j'aime le plus au monde : ma mère

A mon cher père, je n'ai jamais vu autant d'amour.

Que l'amour que tu me donnes.

A mon frère bilel walid samir et mes sœurs wided mayssa nadia

A ma belle mère, mon beau père et ma belle famille drici

Ma douce sœur qui a eu la patience de me supporter durant ce mémoire,

Et qui m'a soutenu et encouragé pendant tous les moments difficiles vécus,

Je t'aime beaucoup ma chère.

Et à vous mes amies ziad et aymen et nouh et chahinez hodna mes collègues

Qui ensoleillez ma vie, et avec

Qui j'ai ce très grand plaisir de partager des instants de bonheurs continue

Et en particulier

A tous qui ont m'ont encouragé, aidé de près ou de loin pour réaliser ce

Travail et me souhaité le succès et le bonheur dans ma

Vie……

A tous ceux qui me sont chers je dédie ce modeste trav

DRICI HACENE

ABSTRACT

The design of artificial systems has undergone an epistemic revolution throughout the past fifty years. It entailed transcending the traditional dichotomy between the organic and inorganic worlds by attempting to imbue machines with the adaptability and autonomy found in living systems. Evolutionary robotics aims to design machines capable of continuously learning new skills in a continuous, uncontrolled and changing world. Based on this observation, Our work offers a non-exhaustive vision of theresearch themes associated with the field of mobile robotics, and presents the scientific obstacles that remainto be lifted to lead to the development of an autonomous robot.The latter's autonomy necessitates the accomplishment of tasks of control and perception of the environment in a coordinated manner.

Navigation is one of these, and it is critical to the robot's interaction with its evolutionary environment. It entails determining the robot's trajectories to follow a pre-determined path while avoiding mobile or immovable impediments.

Our solution is based on centralized solution Algorithm to find the shortest path. After then, the navigation problem is described as a constraint optimization problem, with a fitness function that quantifies the difference between the best robot path and the other random paths.

The impediments are implemented as limitations that penalize the robots' movement, to allow them to change positions while avoiding obstructions.

Our strategy has been put into action, and various situations have been put to the test. The collected results indicate the method's robustness as well as its performance.

Keywords: MobileRobot,, centralized solution CBS ,Path Finding,Obstacle Avoidance

RÉSUMÉ

La conception des systèmes artificiels a connu une révolution épistémique au cours des cinquante dernières années. Il s'agissait de transcender la dichotomie traditionnelle entre les mondes organique et inorganique en tentant d'imprégner les machines de l'adaptabilité et de l'autonomie que l'on retrouve dans les systèmes vivants.La robotique évolutive vise à concevoir des machines capables d'acquérir en permanence de nouvellescompétences dans un monde continu, incontrôlé et changeant. Partant de ce constat, Notre travail offre unevision non exhaustive des thèmes de recherche associés au domaine de la robotique mobile, et présente lesobstacles scientifiques qui restent à lever pour aboutir au développement d'un robot autonome.La navigation est l'une d'entre elles, et elle est essentielle à l'interaction du robot avec son environnement évolutif. Il s'agit de déterminer les trajectoires du robot pour suivre un chemin prédéterminé tout en évitant les obstacles mobiles ou immobiles.Notre solution est basée sur l'algorithmed solutions centralisée de plus court chemin. Ensuite, le problème de navigation est décrit comme un problème d'optimisation de contraintes, avec une fonction de fitness qui quantifie la différence entre la meilleure trajectoire du robot et les autres trajectoires aléatoires.Les entraves sont implémentées comme des limitations qui pénalisent le mouvement des robots, pour leur permettre de changer de position tout en évitant les obstructions.Notre stratégie a été mise en œuvre et diverses situations ont été mises à l'épreuve. Les résultats recueillis indiquent la robustesse de la méthode ainsi que ses performances.

Mots clés: Robotmobile, solution e CBS, recherchedechemin, évitement d'obstacle.

ملخص

شهد تصميم الأنظمة الاصطناعية ثورة معرفية على مدار الخمسين عامًا الماضية. لقد استلزم تجاوز الانقسام التقليدي بين العالمين العضوي وغير العضوي من خلال محاولة إضفاء القدرة على التكيف والاستقلالية الموجودة في الأنظمة الحية على الآلات.

تهدف الروبوتات المتطورة الى تصميم اليات قادرة على التعلم المستمر لمهارات جديدة في عالم مستمر و غير متحكم فيه و من غير بناء على هده الملاحظة . يقدم عملنا رؤية غير شاملة لموضوعات البحث المرتبطة بمجال الروبوتات المتنقلة و يعرض العقبات العلمية التي لا يزال يتعين رفعها لتؤدي الى تطوير روبوت مستقل

يعد التنقل أحد هذه العناصر، وهو أمر بالغ الأهمية لتفاعل الروبوت مع بيئته التطورية. يستلزم تحديد مسارات الروبوت لإتباع مسارمحدد مسبقًا مع تجنب العوائق المتحركة أو الثابتة.

يعتمد حلنا على خوارزمية الحل المركزي لأقصر مسارً. بعد ذلك، يتم وصف مشكلة التنقل على أنها مشكلة تحسين القيد، مع وظيفةاللياقة التي تحدد الفرق بين أفضل مسار للروبوت والمسارات العشوائية الأخرى.

يتم تنفيذ العوائق كقيود تعاقب حركة الروبوتات، للسماح لها بتغيير المواضع مع تجنب العوائق.

قد تم وضع استراتيجيتنا موضع التنفيذ ، وتم اختبار المواقف المختلفة. تشير النتائج التي تم جمعها إلى متانة الطريقة بالإضافة إلىأدائها.

الكلمات المفتاحية: روبوتمتحرك،حل مركزي،اكتشافالمسار،تجنبالعقوبات

Sommaire

Liste de figure	**10**
Liste de tableau	**11**
Liste d'abréviations	**12**
Introduction generale	**13**
CHAPITRE 01:L'ETAT DE L'ART	
1. Introduction	**15**
2.1.exemple d'application	**15**
3.1. robots industriels	**19**
4.1.Les moyens de perception en robotique mobile	**18**
5.1.Architecture des robots mobiles	**18**
6.1.La structure mécanique et motricité	**19**
7.1.Traitement des informations et gestion des taches	**20**
8.1.Navigation autonome	**20**
9.1.Les avantage de la robotique	**20**
10.1.Les inconvénients de la robotique	**20**
2.Stratégie de navigation	**22**
2.1.La meta heuristique	**23**
2.2.Colonie de fourmis	**23**
2.3.Optimisation de binaire bactérien	**25**
2.4.Colonnie d'abeilles	**27**
2.5.Alghorithme génetique	**30**
2.6. Alghorithme CBS	**31**
2.7.La recherche taboue	**31**
2.8.Comparaisons entre les methodes	**33**
Conclusion	**36**

CHAPITRE 02 : CONCEPTION	
3.Introduction	**36**
3.1. Presentation de travaile	**36**
3.1.1.Le problématique	**36**
3.1.2.Le solution	**36**
3.2.L'architecture d'application	**37**
4.La modélisation conceptuelle	**37**
4.1.Le diagramme de cas utilisation	**38**
4.2.Le diagramme de sequnce	**39**
4.3.Le diagramme de classe	**40**
4.4.Approche centralisée	**40**
4.5.Definition de problème	**41**
4.6.Problème d'entrée	**41**
4.7.Taches mapf	**41**
4.8. Fonction de cout	**41**
5.Enquete sur les alghorithme mapf	**42**
5.1.L'alghorithme de recherche basée sur les conflits	**42**
5.2.Traitement d'un neoud	**43**
5.3.resoudre de conflits	**43**
5.4.conflits bors	**43**
Conclusion	***46***
CHAPITRE 03 : IMPLIMENTATION	
6.Introduction	**44**
7.Envirenement logiciel	**44**
8.Interface d'application	**47**
9.Codage de boutton de reglage d'un application	**51**

10.Domaines d'application	**53**
Conclusion	**56**
Conclusion generale	**57**
References	

Liste de figures

Figure	Titre	Page
	Chapitre I	
Figure1 1	*Exemple de robots commercieaux ou de recherché*	**15**
Figure 1.2	*Architecture modulaire d'un robots mobile*	**17**
Figure 1.3	*Types de structure mecaniques*	**18**
Figure 1.4	*Classification de meta heuristiques*	**22**
Figure 1.5	*Experience de selections des branches les plus courts par colonnie fourmis*	**23**
Figure 1.6	*Representation vutuelle du fonctinnement de loptimissation de recherché BFO*	**26**
Figure 1.7	*Classification de colonnie d'abbeilles*	**27**
Figure 1.8	*Conception basée sur un alghorithme d'abbeille vurtuelle*	**28**
Figure 1.9	*Colonnie d'abbeiles artificial*	**29**
Figure 1.10	*Organigrammme de l'alghorithme génetique*	**30**
	Chapitre II	
Figure 2.1	*Architecture de système*	**36**
Figure 2.2	*Diagramme de cas d'utulisation*	**37**
Figure 2.3	*diagramme de sequence*	**38**
Figure 2.4	*diagramme de classe*	**39**
	Chapitre III	
Figure 3.1	*Logo officielle PYTHON*	**40**
Figure 3.1	*Logo oficielle NYMPY*	**40**
Figure 3.3	*Logo oficielle SPICY*	**46**
Figure 3.4	*Logo oficielle MATPLOTLIB*	**47**
Figure 3.5	*Application generale*	**47**
Figure 3.6	*Map size*	**48**
Figure 3.7	*Le nombre d'agent*	**48**
Figure 3.8	*Position d'obstacles lieu debut et lieu fin d'agents*	**48**

Figure 3.9	*Fournir des solution*	***48***
Figure 3.10	*Exemple 3 agent*	***49***
Figure 3.11	*Solution programme*	***49***
Figure 3.12	*fournir des solutions finales et afficher leur chemin ement en detaille*	***50***

Liste des abréviations

- **AI** : Artificial intelligence
- **GA** : Genetic Algorithm
- **VBA** : Virtual Bee Algorithm
- **BCO** : Bee Colony Optimization
- **BOD** : Dance Bee Optimization
- **ABC** : Artificial Bee Colony
- **ANN** : Artificial Neural network.
- **CBS:** conflit based search
- **BFO** : Bacterial forging optimization

Depuis quelques années, un intérêt croissant est porté, au sein de la communauté robotique, au développement des systèmes intelligents autonomes dans le cadre de la robotique mobile [1]. Un tel intérêt peut être perçu comme une conséquence logique à l'apparition des applications potentielles des machines intelligentes (dans l'industriel, pour des services, manutention ou encore d'aide à la mobilité des personnes âgées ou handicapées,…). L'objectif est de mettre les robots dans des situations variées telles que les interventions sur des sites accidentés, la manipulationsur sites sensibles ou nucléaires, ainsi que pour l'exploration des sites maritimes …etc. [2]

L'enjeu principal de la robotique mobile actuelle consiste à développer des systèmes denavigation intelligents, où la navigation autonome est un axe de recherchequi vise à donner à une machine la capacité de se mouvoir dans un environnement sans assistance, ni intervention humaine pour accomplir un but desire. La tâche de la navigation consiste à donner au robot la possibilité d'obtenir les informations dont il a besoin pour raisonner et le doter de capacité de locomotion adaptée à son environnement [3] .Cependant, elle implique des systèmes complexes dans la réalisation, où leur maîtrise pose d'importants problèmes non seulement technologiques mais aussi scientifiques. Un robot mobile autonome est unsystème mécanique qui doit être en mesure de prendre des décisions pour effectuer des mouvements en fonction des informations sur sa position et sur l'environnement dans lequel il évolue. [4]

De manière générale, on regroupe sous l'appellation robots mobiles l'ensembledes robots à base mobile (FIL04). Les autres robots sont distingués par le type de locomotion qu'il est marcheurs, sous-marins ou aériens [5].

La recherche en robotique mobile s'intéresse à la conception des systèmes intelligents dotés par des techniques de commande efficaces pour le déplacement d'unrobot mobile, où la sécurité est prioritaire par rapport à l'optimalité ([6] [7]. En règle générale, on ne connaît pas toujours de méthode exacte pour trouver la solution d'un problème d'optimisation en recherche opérationnelle. Dans ce cas on peut d'abord tenter de voir si le problème que l'on étudie n'a pas d'autres équivalents

ayant étés déjà résolus. Si l'on n'a toujours pas trouvé de méthode de résolution alorson utilise ce que l'on appelle une heuristique, c'est-à-dire un algorithme qui donne unesolution approchée. Ces algorithmes sont assez intuitifs ou simples. On les déduits grâce à des observations et en faisant preuve de bon sens. Leur principe consistesouvent à explorer un certain nombre de solutions et de mémoriser la meilleure. Ilspeuvent faire intervenir le hasard : cela permet de balayer un plus grand nombre demais il faut les exécuter plusieurs fois pour tendre au mieux vers la solution optimale

Le problème d'optimisation Combinatoire a longtemps privilégié les méthodes exactes au détriment des méthodes approchées. Les raisons de cette préférence sont multiples et tiennent tant à des facteurs historiques que théoriques. En particulier, le recours à des modèles exacts permet de bénéficier de résultats théoriques forts accompagnant les notions de convergence et d'optimalité globale. [9]

Cependant, depuis la fin des années 80, les méthodes approchées, et plus particulièrement les métaheuristiques suscitent un intérêt croissant de la part de la communauté. Ce succès tient pour une grande part à leur capacité à fournir des solutions d'excellente qualité au prix d'une consommation en ressources réduite. Les métaheuristiques bénéficient également d'autres avantages significatifs tels que la faculté à s'adapter rapidement à des modifications structurelles du problème (ajout ousuppression de contraintes). [10]

Ces caractéristiques les rendent parfaitement adaptées aux exigences du milieu industriel, ce qui explique l'arrivée sur le marché d'un nombre croissant d'outils d'aideà la décision intégrant des métaheuristiques. Parmi ces métaheuristiques, nous pouvons citer les algorithmes génétiques, le recuit simulé, les réseaux de neurones, les algorithmes à estimation de distribution, l'optimisation par essaim de particules et l'optimisation par CBS. [9]

Durant les dernières années, on assiste à une évolution croissante du nombre d'études menées sur les animaux vivant en groupe ou en société et plus particulièrement les insectes sociaux. Ces études éthologiques dans la théorie de l'auto-organisation ont inspiré un grand nombre de chercheurs pour développer des algorithmes de CBS et essaims particulaires.

Dans le cadre des travaux de cette thèse, on va donc :

Le chapitre 01 : Dans ce chapitre, nous réalisons une description détaillée sur la

Robotique mobile et les méthodes CBS.

Chapitre 02 : Est dédiée à la présentation du projet (problématique, objectif,fonctionnement) et la modélisation conceptuelle.

Chapitre 03 : Met l'emphase sur les étapes de réalisation, l'environnement detravail, ainsi que les résultats obtenus à travers quelques interfaces d'application.

Ce mémoire s'achève par une conclusion générale résumant les connaissances

Acquissent au travers du projet.

.

1. Introduction

Les robots aujourd'hui ont un impact considérable sur de nombreux aspects de la vie moderne, de la fabrication industrielle aux soins de santé, le transport et l'exploration de l'espace et le profond de la mer. Demain, des robots seront aussi omniprésents et personnelle comme les ordinateurs personnels.

Le rêve de créer des machines qui sont qualifiés et intelligentes a fait partie de l'humanité depuis le début du temps. Ce rêve est en train de devenir une partie dela réalité de notre monde.

La résolution des problèmes combinatoire est assez délicate puisque le nombre fini réalisable croit généralement avec la taille du problème, ainsi que sa complexité. Cela a poussé les chercheurs à développer de nombreuses méthodes de CBS et en intelligence artificielle (IA).

2.1. Robotique mobile

La robotique mobile est une discipline en plein essor avec notamment l'apparition des drones volants, des robots sous-marins détecteurs de mines, des robots voiliers ou encore des robots aspirateurs, cette ouvrage présente les différents outils et méthodes qui permettent la conception de robots mobiles. Ces systèmes généralement autonomes et supervisés par un opérateur humain sont capables de se déplacer dans un environnement plus ou moins connu. Illustré par des

Simulation et des exercices corrrigées, La robotique mobile décrit les principes essentiels de la modélisation des robots, développant les notions d'actionneur, de capteur, de régulateur et de guidage. Les outils permettant une simulation tridimensionnelle sont également traités ainsi que les bases théoriques d'une localisation fiable des robots dans leur environnement. [11]

La robotique mobile est un domaine dans lequel l'expérience pratique est Particulièrement illustratrice et importante pour la compréhension des problèmes.

Au-delà des méthodes présentées dans ce texte, les travaux dirigés ou le projet

Pratique associé que réalisent les étudiants apporteront également leur lot de connaissances irremplaçables

3.1. Exemples d'applications

Aujourd'hui, le marché commercial de la robotique mobile est toujours relativement restreint en dehors des robots aspirateurs vendus à plusieurs millions d'exemplaires. Cependant, il existe de nombreuses perspectives de développement qui en feront probablement un domaine important

dans le futur. Les applications des robots peuvent se trouver dans de nombreuses activités « ennuyeuses, salissantes ou dangereuses » mais également pour des applications ludiques ou de service, comme l'assistance aux personnes âgées ou handicapées. [15]

Figure 1.1: exemple des robots commerciaux ou de recherche. [42]

Parmi les domaines d'applications possibles de la robotique, citons :

La robotique de service (hôpital, bureaux, maison).

La robotique de loisir (jouets, robot 'compagnon').

La robotique industrielle ou agricole (entre pôts logistiques, récolte de productions agricoles, mines).

La robotique en environnement dangereux (spatial, industriel, militaire,catastrophes naturell

3.2. Robots industriels

Le marché des robots industriels est estimé à 35 milliards de dollars et devrait atteindre 90 milliards de dollars en 2022. En termes d'unité, le stock de robots industriels opérationnels dans le monde passera de 1 631 600 unitéS Fines 2015 à 2 589 000 unités fin 2019.

Ces chiffres confirment une tendance très forte dans le développement des robots. A quoi ressemblent ces robots industriels ?

On compte trois composants principaux pour caractériser un robot :

La mécanique comprenant l'intégralité des pièces du robot. Cette partie est importante car la précision, la rapidité et la charge maximale du robot seront directement liés à la conception mécanique de celui-ci.

L'électronique qui se caractérise par le système de commande afin de piloter tous les moteurs et recevoir des informations sur l'environnement du robot grâce à des capteurs.

L'informatique qui permet de rendre le robot "intelligent", en le faisant Collaborer avec l'utilisateur et son environnement. [19]

Jusqu'à présent, les robots ont été programmés pour effectuer des consignes bien spécifiques. Avec le développement de l'informatique et l'arrivée de l'intelligence artificielle, les robots vont de plus en plus évoluer. Ils seront bientôt capables de pleinement réagir par rapport à leur environnement.

Les applications liées aux robots sont quasiment infinies. En voici certaines impliquant un haut degré de performance :

- Soudage à l'arc (automobile etc…)
- Assemblage
- Peinture et Pulvérisation
- Découpe laser
- Conditionnement
- Inspection de production
- Mesure laser et Vérification de pièce
- Robots mobiles (pour l'inspection en zone sensible par exemple
- Grâce aux robots ces applications sont exécutées à haute endurance, rapidité etprécision.
- Gravure PCB

3.1. Les moyens de perception en robotique mobile

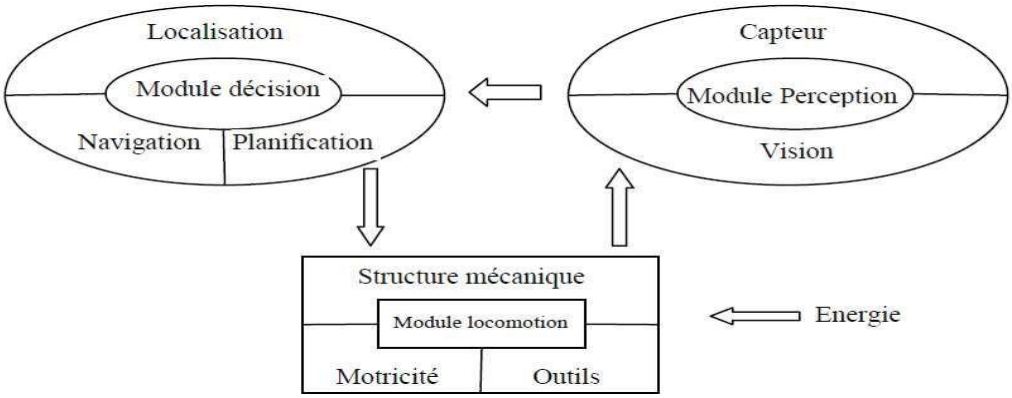
La perception est un domaine crucial de la robotique. C'est autour de ce concept qu'est bâtie la structure d'un robot apte à exécuter des tâches complexes ou à évoluer dans un univers inconnu ou mal connu. L'élément de base du système de perception est le capteur qui a pour objet de traduire en une information exploitable des données représentant des caractéristiques de l'environnement. Les moyens utilisés pour la perception de l'environnement sont nombreux. Nous citons:

- Les systèmes de vision globale.
- Les télémètres laser et ultrasonores.
- Les capteurs optiques et infrarouges.

4.1. L'architecture des robots mobiles :

L'architecture des robots mobiles se compose de quatre parties essentielles :

- La structure mécanique et la motricité.
- Le système de localisation.
- Les organes de sécurité.
- Le système de traitement des informations et gestion de tâches.



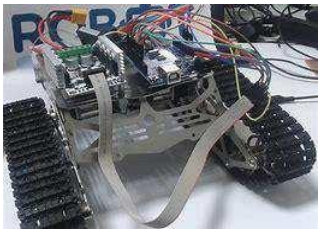
-

Figure 1.2 :Achitecture modulaire d'un robot mobile

5.1.La structure mécanique et motricité :

Il existe quatre types de structures mécaniques assurant la motricité :

Robot à rampant



Robot a chenille

Robot a roues

Robot à marcheur

Figure 1.3.: types de structures mécaniques.[42]

Les robots mobiles à roues : La mobilité par roues est la structure mécanique la plus communément appliquée .Cette technique assure selon l'agencement et les dimensions des roues un déplacement dans toutes les directions avec une accélération et une vitesse importantes .Le franchissement d'obstacles ou l'escaladede marches d'escaliers est possible.

Les robots à chenilles : L'utilisation des chenilles présente l'avantage d'une bonneadhérence au sol et d'une faculté de franchissement d'obstacles .L'utilisation est orientée vers l'emploi sur sol accidenté ou de mauvaise qualité au niveau de l'adhérence.[16]

Les robots marcheurs : Les robots marcheurs sont destinés à réaliser des tâches variées dont l'accès au site est difficile .Leur anatomie à nombreux degrés de libertépermet un rapprochement avec les robots manipulateurs .La locomotion est commandée en termes de coordonnées articulaires.

Les différentes techniques étudiées se rapprochent de la marche des animaux et notamment de celle des insectes.[20].

6.1 Traitement des informations et gestion des taches

L'ensemble de traitement des informations et gestion des tâches constitue le noyau du module d'informatique central qui établit les commandes permettant au robot mobile de réaliser un déplacement et d'activer les divers organes en accord avec l'objectif. A ce niveau le problème qui se pose est le problème de génération de plan qui consiste à établir la manière dont le robot se déplace par rapport à des connaissances aprioriques « statiques » ou obtenues en cours d'évolution « dynamiques »[16] La génération des plans repose sur trois concepts :

- La stratégie de navigation.
- La modélisation de l'espace.

7.1. La navigation autonome

La navigation autonome est un domaine de recherche très actif, c'est un thème de recherche dans des laboratoires de haut niveau. Le but pour un robot est de trouver le plus court chemin d'un point de départ jusqu'au point d'arrivé. Ces dernières années, la navigation collective attire de plus en plus l'attention des chercheurs que lanavigation d'un seul robot à cause de son efficacité, sa rapidité, sa robustesse et sa capacité à résoudre des taches complexes.

Un navigateur désigne généralement un module chargé de faire se mouvoir le

Robot mobile dans son milieu d'évolution.

Les applications de ces systèmes de navigation robustes :

- l'exploration d'environnement.
- Infrastructures urbaines.
- livraison par drones ou de robots terrestre.

8.1. Les avantage de la robotique

8.1.1. La robotique industrielle

Le développement de la robotique a permis de mettre en marche une révolution dans l'industrie puisqu'il a permis l'automatisation de la production. La robotique industrielle a aussi permis une

augmentation des rendements et une meilleure qualité au niveau du produit final grâce à la précision extrême des robots industriels. [18]

8.1.2. La robotique militaire

Les robots éclaireurs permettent aujourd'hui de limiter le danger pour les troupes humaines en précédant les soldats et en les prévenant en cas de danger. La robotique militaire est donc de plus en plus adaptée aux missions en terrains hostiles comme le montre le robot LS3 (ou Big Dog) capable de parcourir les terrains à une vitesse accrue en transportant d'importantes charges de matériel. Cela permet aux soldats de se déplacer plus rapidement et en se fatigant moins.[21]

8.1.3. La robotique de recherche et d'exploration

Il existe aussi de nombreux avantages dans la recherche et l'exploration. Les sous-marins ne pouvant pas se rendre à des profondeurs extrêmes, ils sont remplacés par des robots d'exploration sous-marine, plus résistants.

8.1.4. La robotique médicale

La robotisation dans le domaine de la santé est très importante car elle permet notamment d'obtenir de meilleurs résultats au niveau des opérations chirurgicales grâce à la précision des robots.

8.1.5. Les inconvénients de la robotique

La robotique a créé de nouveaux problèmes qui n'existaient pas auparavant. Ces

nouveaux problèmes doivent être résolus pour une utilisation optimisée des robots.[25]

8.1.7. La robotique militaire

Dans le domaine militaire, on retrouve beaucoup d'inconvénients. En effet, si un robot est très utile pour des taches simples comme le transport de marchandises ou les missions d'éclaireur, ils ne peuvent pas prendre de décisions par eux-mêmes. Ilsrestent très onéreux.

8.1.8. La robotique industrielle

Les robots peuvent être dangereux pour les ouvriers travaillant dans les usines.

8.1.9. La robotique médicale

Les robots peuvent réduire le temps d'action des médecins lors des opérations. Cependant, le prix d'achat des robots est toujours un problème car les hôpitaux ne possèdent pas tous les moyens de s'équiper de robots chirurgicaux.[25]

8.1.10. La robotique de services

Cette robotique, qui tente de s'inclure dans la société, pourrait modifier les rapports Entre les humains.

2. Stratégies de navigation

Les procédures de parcours permettant à un robot mobile de se déplacer et d'arriver à un objectif sont

exorbitants, semblables aux caractérisations qu'on peut en faire

2.1. Le méta heuristique :

Les métaheuristiques sont des algorithmes stochastiques permettent de fournir des solutions de bonne qualité à des problèmes combinatoire dont on ne connait pas de méthodes exacte pour les résoudre, ou bien qui nécessitent un temps de calcule élevé ou un grand espace de stockage.

Elles utilisent généralement des processus aléatoires et itératifs pour trouver une solution optimale.

Elles sont souvent inspirées par des systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé), en biologie de l'évolution (cas des algorithmes Génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis, des sauts de grenouilles, de l'optimisation par essaims de particules). Le principe des métaheuristiques est de minimiser ou de maximiser une fonction objective. Leur avantage est de trouver un minimum global à un problème de minimisation et de ne pas rester bloqué sur un minimum local [30].

En générale, l'utilisateur demande des méthodes efficaces et rapides permettent d'atteindre un optimum avec une précision acceptable dans un temps raisonnable, mais il a besoin aussi des méthodes simple à utiliser.

Un des enjeux des métaheuristiques est de faciliter le choix d'une méthode et de simplifier son réglage pour l'adapter au mieux à un problème posé.

2.1.1. Principe générale des métaheuristiques

Il existe une multitude de métaheuristiques. Elles tentent toutes d'améliorer itérativement une solution existante. Elles passent d'une solution à une autre dans l'espace de recherche jusqu'à ce qu'une solution optimale soit trouvée ou le nombre d'itération soit dépassé :

A partir d'une solution initiale qui peut choisie de façon aléatoire ou bien construite par un autre algorithme, ces méthodes explorent le voisinage en appliquant des transformations ou des mouvements. [24]

Le choix de la prochaine solution s'effectue dans l'ensemble de voisinage de

Manière à avoir une solution plus intéressante.

la recherche locale s'arrête généralement quand un certain nombre d'itérations est atteint ou lorsqu'une solution de qualité est obtenue.

2.1.2. Classification des métaheuristiques

Les métaheuristiques peuvent être classées de différentes façons. On peut distinguer celles qui travaillent avec une population de solution de celles qui ne manipulent qu'une seule solution à la fois.

Les méthodes qui tentent itérativement d'améliorer une solution sont appelées de recherche locale ou méthodes de trajectoire. Les exemples le plus connus de ces méthodes sont : recherche tabou et le recuit simulé.[25]

Les algorithmes génétiques, l'optimisation par essaim de particules et les algorithmes de colonie de fourmis présentent les exemples les plus connus des méthodes qui travaillent avec une population.

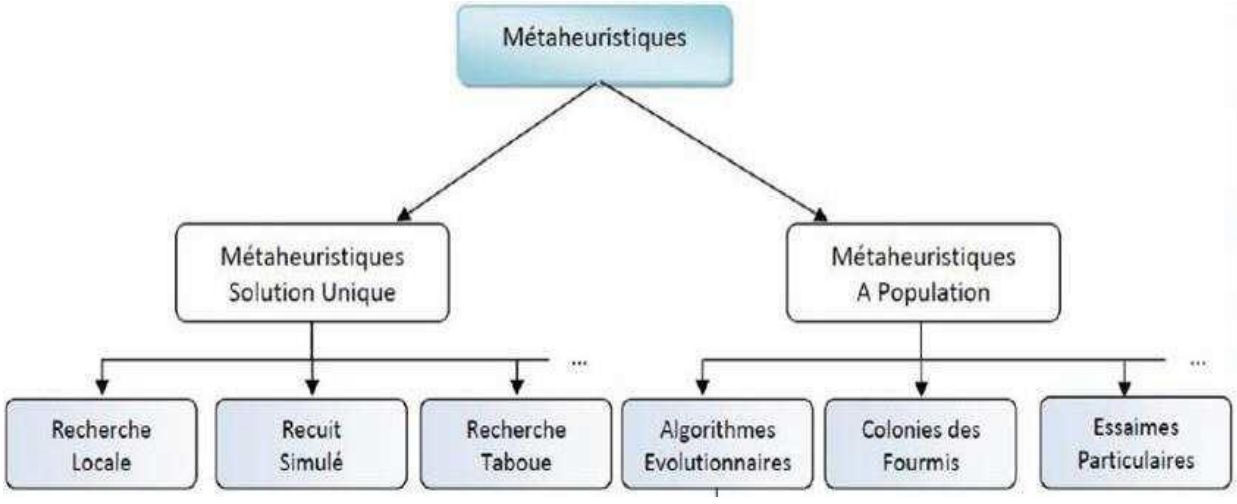


Figure 1.4 : classification des métaheuristiques.[25]

On peut également distinguer les métaheuristiques qui fonctionnent sans mémoire et celles qui mémorise des informations, et qui peuvent revenir sur des solutions qu'elles ont déjà examinée. On trouve donc la mémoire à court terme (celles qui sauvegarde les derniers mouvements effectués). Et la mémoire à long terme (qui concerne des paramètres synthétique plus généraux). [19]

2.2. Colonie de fourmis:

L'algorithme des colonies de fourmis (Ant Colony Optimisation) est un algorithme d'intelligence en essaim dont le principe est basé sur la manière dont les fourmis cherchent leurs nourritures et retrouvent leur chemin pour retourner dans la fourmilière. **[33]**. Les algorithmes de colonies de fourmis forment une classe des métaheuristiques proposée pour des problèmes d'optimisation difficiles. Ces algorithmes s'inspirent des comportements collectifs de dépôt et de suivi de piste observés dans les colonies de fourmis. Une colonie d'agents simples (les fourmis)

Communiquent indirectement via des modifications dynamiques de leur environnement (les pistes de phéromones) et construisent ainsi une solution à un problème en s'appuyant sur leur expérience collective [34]. Initialement, les fourmis explorent les environs de leur nid de manière aléatoire. Sitôt qu'une source de nourriture est repérée par une fourmi, son intérêt est évalué (quantité et qualité) et la fourmi ramène un peu de nourriture au nid. Les fourmis peuvent déposer des phéromones au sol, grâce à une glande située dans leur abdomen et former, ainsi, despistes odorantes qui pourront être suivies par leurs congénères.

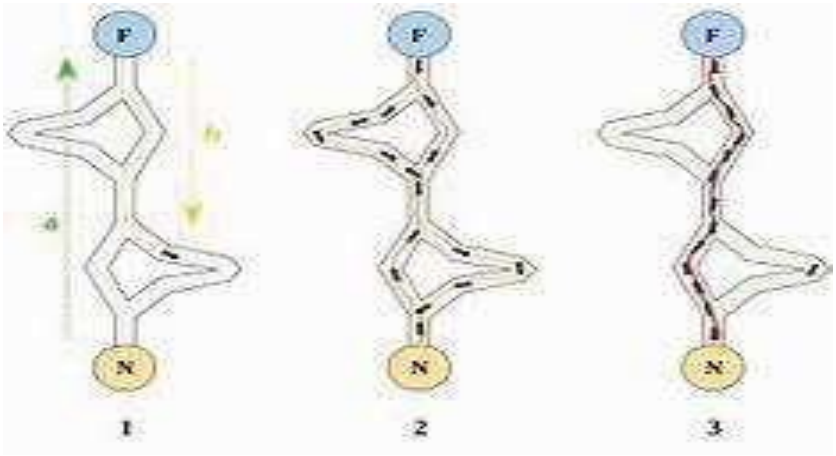


Figure 1.5 : Expérience de sélection des branches les plus courtes par une colonie de fourmis [30]

Les traces laissées s'accumulent au fur et à mesure que la piste est rejointe par plus
De congénère. Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter, sans que les individus aient une visionglobale du trajet.

L'idée est de représenter le problème à résoudre sous la forme de la recherche d'un meilleur chemin dans un graphe, appelé graphe de construction, puis d'utiliser des fourmis artificielles

pour rechercher de bons chemins dans ce graphe. Le comportement des fourmis artificielles est inspiré des fourmis réelles : elles déposentdes traces de phéromone sur les composants du graphe de construction et elles choisissent leurs chemins relativement aux traces de phéromone précédemment déposées ; ces traces sont évaporées au cours du temps. Intuitivement, cette communication indirecte via l'environnement fournit une information sur la qualité des chemins empruntés afin d'attirer les fourmis et d'intensifier la recherche dans les itérations futures vers les zones correspondantes de l'espace de recherche. Ce mécanisme d'intensification est combiné à un mécanisme de diversification, essentiellement basé sur la nature aléatoire des décisions prises par les fourmis, qui garantit une bonne exploration de l'espace de recherche. Le compromis entre intensification et diversification peut être obtenu en modifiant les valeurs des paramètres déterminant l'importance de la phéromone. Les fourmis artificielles ont aussi d'autres caractéristiques, qui ne trouvent pas leur équivalent dans la nature. En particulier, elles peuvent avoir une mémoire, qui leur permet de garder une trace de leurs actions passées. Dans la plupart des cas, les fourmis ne déposent une trace de phéromone qu'après avoir effectué un chemin complet, et non de façon incrémentale au fur et à mesure de leur progression. Enfin, la probabilité pour une fourmi artificiellede choisir un arc ne dépend généralement pas uniquement des traces de phéromone, mais aussi d'heuristiques dépendantes du problème permettant d'évaluer localement la qualité du chemin [37]

2.3. Optimisation du butinage bactérien :

L'optimisation de la recherche de nourriture bactérienne (BFO) , un nouvel algorithme proposé par Passino, simule principalement les comportements d'Escherichia coli dans le processus de recherche de nutriments.En chimiotaxie, reproduction, élimination, dispersion. De plus, comme pour l'optimisation des fonctions, Pendant le contexte de recherche de nourriture, les bactéries entreprennent généralement quatre actions importantes impliquant que la position d'une bactérie peut être considérée comme une solution réalisable dans la région de recherche. Les bactéries ajustent leurs propres positions en culbutant selon des directions aléatoires et en nageant ave[c

certaines tailles de pas pour trouver constamment l'emplacement optimal. En raison de Nombreux avantages tels que sa forte robustesse et ses bonnes performances dans la recherche locale,

L'algorithme BFO est progressivement devenu populaire et jusqu'à présent, il a été appliqué à des types de Domaines pratiques tels que l'aménagement des installations la sélection des fonctionnalités [27], la formation Machine d'apprentissage extrême du noyau et ainsi de suite. À l'heure actuelle, les recherches théoriques Pertinentes sur l'algorithme BFO en sont encore au stade initial. En d'autres termes, par rapport aux autres Algorithmes traditionnels intelligents en essaim tels que l'optimisation en essaim de particules (PSO) et l'algorithme génétique (GA) , le développement de l'algorithme BFO n'est pas assez mature. Le professeur Passino a créé la méthode Bacterial Foraging optimization (BFO) en 2002 [33], qui est une Technique révolutionnaire d'optimisation de l'intelligence des essaims basée sur le comportement de recherche de nourriture d'e. Coli. Même s'il en est encore aux premiers stades de l'étude, l'algorithme BFO surpasse les Autres algorithmes d'optimisation en termes de convergence rapide et de recherche globale en raison de sa structure et de son comportement individuels bactériens simples, de divers types et caractéristiques de groupe et d'un cycle De vie efficace. Une bactérie peut faire l'une des deux choses suivantes lorsqu'elle se nourrit : culbuter ou Nager. L'orientation de la bactérie est modifiée par le mouvement de culbutage. La bactérie va migrer dans sa Direction actuelle en nageant, ce qui est l'étape de chimiotaxi La chimiotaxie est poursuivie jusqu'à ce qu'une Bactérie se déplac dans la direction du gradient de nutriments positifs. Après un nombre donné de nages complètes,La meilleure moitié de la population se reproduit et le reste de la population est éliminé. [32] Pour éviter les optima locaux, un événement d'éliminationdispersion est effectué, dans lequel certaines bactéries Sont liquidées au hasard avec une très faible probabilité, et de nouveaux remplacements sont initialisés à des

Positions aléatoires dans tout l'espace de recherche. BFO est composé de trois composants principaux : Le processus de chimiotaxie est le premier ; l'étape de reproduction est la seconde ; et l'étape d'élimination- Dispersion est la troisième.

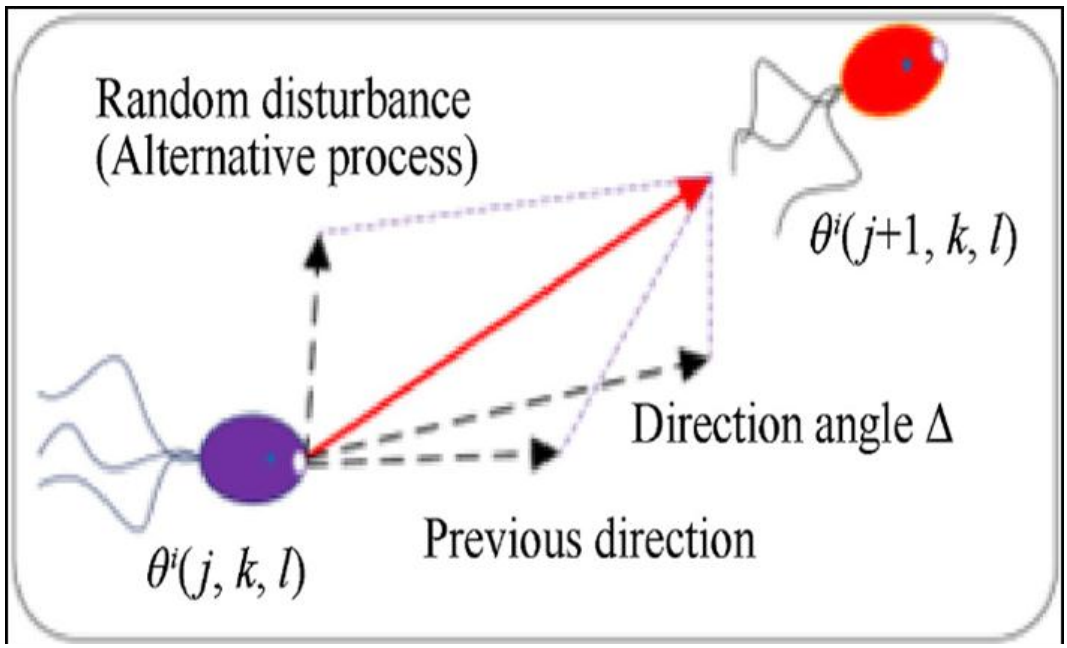


Figure1.6: representation visuellle du fonctionnement de l'optimisation de la recherche de nourriture bactérienne [31]

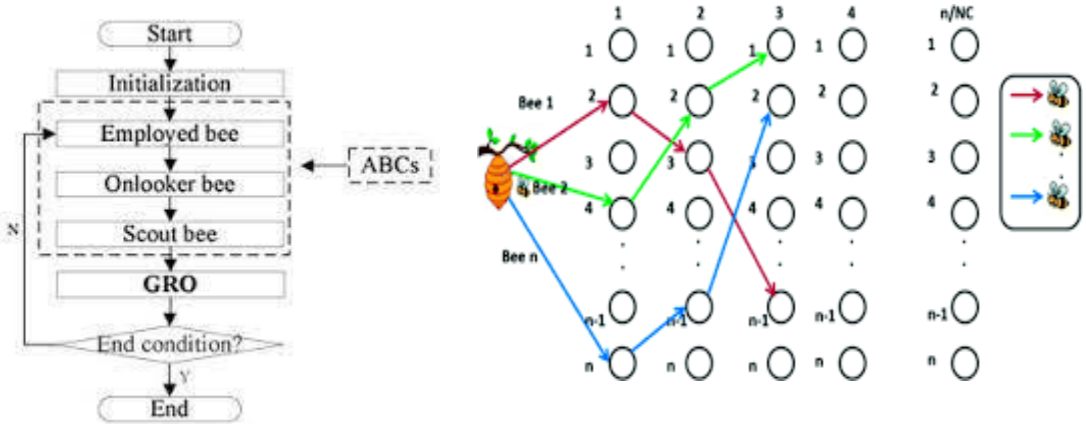
2.4. Colonie d'abeilles :

Au cas où nous n'utiliserions qu'une partie de la nature ou du comportement des abeilles et ajouter quelques nouveaux faits saillants, nous pouvons planifier une classe de nouveaux calculs. Dans ce qui suit, nous abordons une peu de calculs (les plus populaires), sans être exhaustifs, compte tenu de la conduite des les abeilles pendant la recherche [35]. L'amélioration ABC a été proposée par Dervis Karaboga pour prendre soin du monde authentique et problème mathématique dans, qui est suscité par la conduite intelligente des multitudes de bourdons pourobtenir un hotspot pour leur nourriture. Les modèles de calcul ABC comprennent trois rassemblements d'abeilles :

les abeilles éclaireuses trouver toutes les positions de source de nourriture arbitrairement en fonction des mouvements, a utilisé une abeille mellifère en abusant d'un source de nourriture provenant des abeilles éclaireuses et des abeilles spectatrices pour évaluer la nourriture qualité ABC a été largement utilisé dans quelques applications dans divers domaines comme la préparation organisations neuronales, applications de traitement du signal et groupe de personnes IA [38]. L'ensemble la construction du calcul ABC est la suivante:

- Etape 1 : mise en place des aménagements (le nombre d'habitants dans les sources alimentaires). Étape 2 : Les abeilles mellifères employées recherchent de nouvelles sources de nourriture, y compris plus de nectar à l'intérieur des quartiers de la source de nourriture.
- Étape 3 : Les abeilles passantes évaluent la qualité de la nourriture en se basant sur les données fournies par les abeilles utilisées.
- Étape 4 : Commencer à chercher de nouveaux arrangements au hasard par les abeilles éclaireuses. [14]
- Étape 5 : Répétez les étapes 2, 3 et 4 jusqu'à ce que le meilleur arrangement soit obtenu

Figure 1.7 : classification de colonie d'bbeilles [31]



2.5.1 Algorithme d'abeille virtuelle

Ce calcul a été créé par XinShe Yang en 2005 [39] pour aborder les mathématiques problèmes d'amélioration, cela peut faire progresser les capacités et les problèmes discrets, même si ce ne sont que des capacités avec deux paramètres papa ont été donnés à titre d'illustrations spécifiques. Le plan de jeu de la VBA le calcul commence par un miel virtuel êtreLa troupe, chaque abeille se déplace arbitrairement dans le espace de poursuite, et la plupart du temps, l'espace de chasse ne peut être qu'un espace 1-D ou 2-D. Les principales avancées des capacités virtuelles de calcul des abeilles pour la

rationalisation des capacités sont : • Création d'une population d'abeilles multi-spécialistes ou virtuelles. [37]

• Chaque abeille est liée à un vecteur de réponse avec quelques limites à améliorer.

• Codage des capacités d'amélioration (capacités cibles) et transformation en alimentation virtuelle (Nourriture virtuelle)

• Définition d'une règle pour transmettre le parcours et la distance d'une manière similaire à la réalité remise en forme des abeilles (la danse des abeilles). [36]

• Mettez à jour une population de personnes dans de nouveaux postes pour la recherche alimentaire virtuelle, en faisant une danse pour caractériser la distance et le cap ; "la danse virtuelle du waggle"Après un certain temps de développement, le mode le plus élevé, en nombre de virtuels les abeilles mellifères ou la puissance/récurrence des abeilles mellifères qui font la visite est élevée, est liée à la meilleure évaluation.

• Décodage des résultats pour obtenir l'arrangement du problème. Figure 1.8. Conception basée sur l'algorithme Virtual Bees du contrôleur d'amortissement [41]

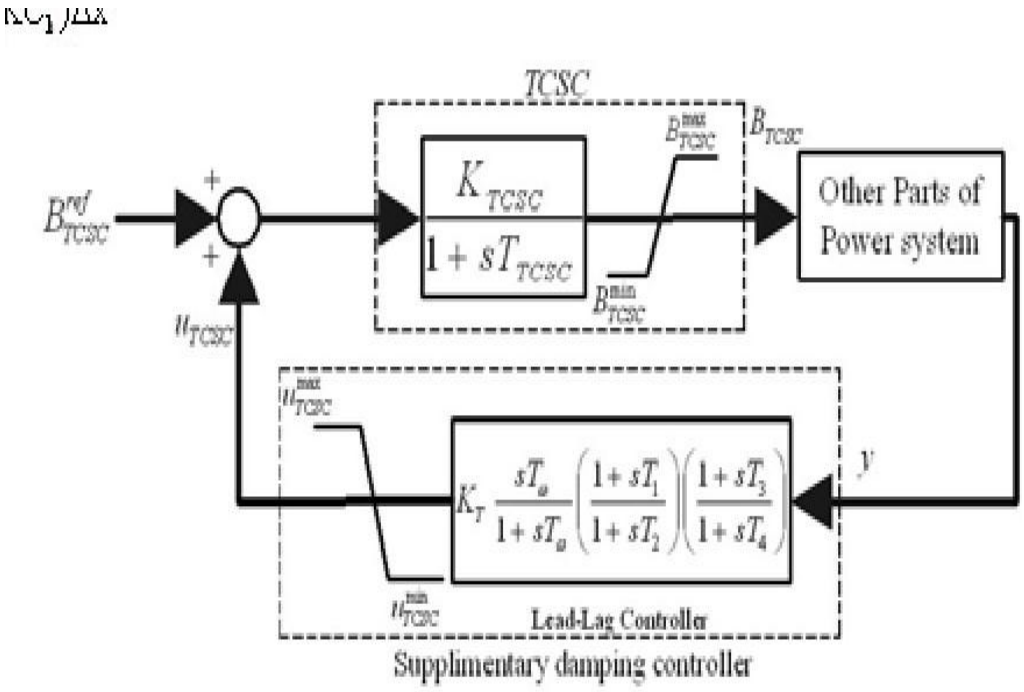


Figure1.8 :.conception basée sur un alghorithme d'abeilles virtuelles d'un controleur d'amortissement [32]

2.5.2..Colonie d'abeilles artificielles :

Le calcul ABC (Colonie d'abeilles artificielles) a été créé by Karaboga et Basturk dans 2007 , examinant le comportement des véritables abeilles mellifères pour

découvrir la source de nourriture, appelée nectar et offrir des données sur les sources de nourriture à d'autres Dans ce calcul les fausses abeilles mellifères sont caractérisées et classées en trois groupes : les abeilles mellifères (abeilles mellifères cette quête de nourriture) [34], les badauds (les abeilles de la perception) et les éclaireurs sont responsables de découvrir de nouvelles variétés alimentaires, (la nouvelle source de nectar). Pour chaque source de nourriture[38], il n'y en a qu'une à l'aide d'abeilles mellifères. [Autrement dit, la quantité de drones de travail se rapproche du nombre de nourriture sources. Parmi les avantages de la technique de la province des abeilles mellifères, l'avis peut : - Très réussi à découvrir des arrangements idéaux. - Vaincre le problème de l'idéal proche. - Facile à exécuter. - L'utilisation de quelques frontières mobiles. - Sensible aux problèmes incroyablement gênants. Pourtant, il existe de nombreuses faiblesses, comme le soulignent la plupart des calculs d'avancement, un instrument de développement et un système de valorisation, augmente un compteur d'arrangements qui ne s'améliorent pas pour arriver à une limite de bord, fixer cette limite est un problème en soi, et peu qualités peuvent tuer une réponse avant d'enquêter sur son voisinage déficient, alors qu'énorme qualités peuvent piéger le calcul dans des minima prémisses pendant quelques cycles

abeilles à la maison. [39]

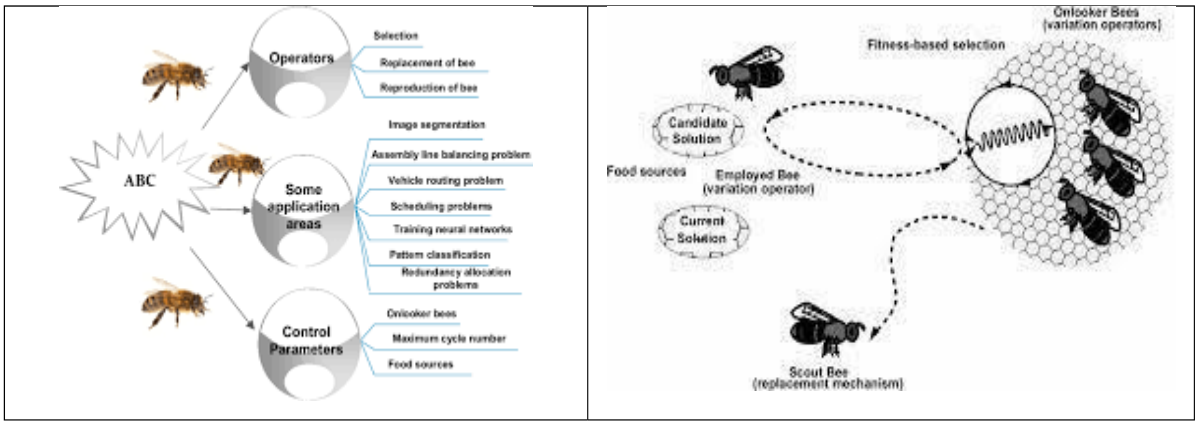


Figure1.9 : colonie d'abbeiles artificielle [42]

2.6. Algorithme génétique :

Les algorithmes génétiques ont un lieu avec un calcul familial dit Évolutif Algorithmes. Ces calculs sont des stratégies de rationalisation stochastique animées par la théorie de Darwin. Hypothèse de développement, dont l'objectif est d'obtenir un arrangement supposé, au bon moment. Les algorithmes héréditaires utilisent des stratégies issues des qualités héréditaires et du développement commun:

croisements, changements, déterminations, et ainsi de suite ..., ils adressent une amélioration stochastique technique pour la requête "0", ce qui implique que ni la congruence ni la différentiabilité ne sont fondamentales pour le bon déroulement de la stratégie. [40]

Juste l'information sur la valeur où la proximité de la capacité à renforcer est suffisante. Ainsi, la viabilité d'un calcul héréditaire repose sur la grande information sur la question à traiter. Les calculs héréditaires sont le résultat de l'examen de John Holland et de ses partenaires et doublures à l'Université du Michigan qui, aussi juste à temps qu'en 1960, ont traité 26 avec ce sujet. La curiosité suscitée par la considération de l'administrateur hybride nonobstant les changements est cet administrateur qui permet de se rapprocher de l'idéal d'un pouvoir en consolidant les qualités contenues dans les différents individus de la population. Les calculs héréditaires dépendent de la pensée du choix caractéristique et l'appliquent à un population de solutions à un problème donné [41]. Les phases d'exécution d'un calcul héréditaire :

- Population initiale : nous devrions choisir une population arbitraire of n chromosomes, chaque chromosome présente ici une zone imminente du robot, et nous pouvons également transformer c'est ainsi que chaque qualité aborde le palier suivant du robot

- Travail sur le bien-être : mesurez le bien-être de chaque chromosome dans la population.

- Choix : Créer une autre population avec des étapes de répétition jusqu'à ce que la population soit achevée. - Hybride et changement : Chaque paire crée deux enfants, dans ces activités deux chromosomes échanger au moins une section contre des informations sur les nouveaux chromosomes. À tout hasard qu'il n'y a pas de mélange, le résultat est un précis des gardiens.

- Le changement implique que la qualité d'un chromosome peut en remplacer un autre de manière arbitraire. Arrêter le test : Si cette règle n'est pas cochée alors passez à l'étape (2)

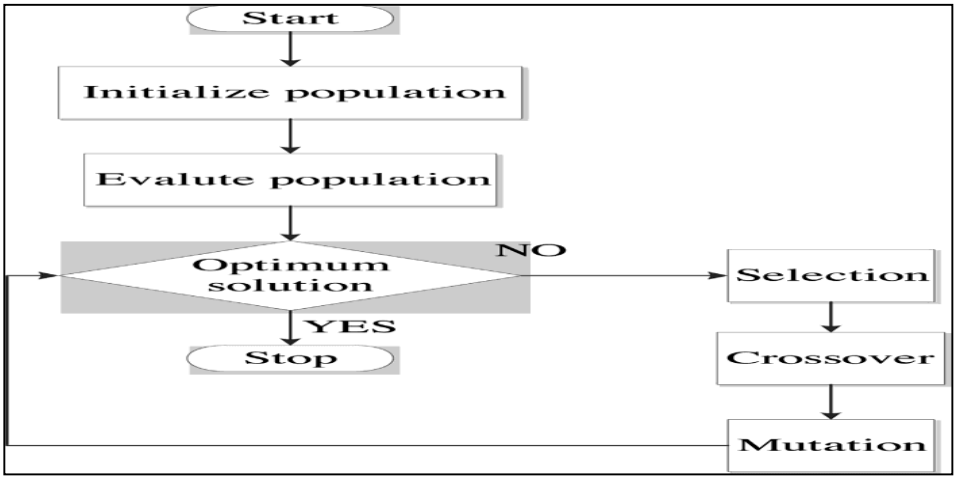


Figure1.10 : organigramme de l'alghorithme génétique [42]

2.7. Alghorithme CBS:

La position initiale et finale des agents du problème de recherche de chemin multi-agents (MAPF) est fournie. Nous trouvons un chemin pour les agents sans collisions. Nous présentons ici la Recherche Basée sur les Conflits (CBS), c'est un algorithme à deux niveaux qui ne convertit pas le problème en un seul modèle « d'agent conjoint ». Un Conflict Tree (CT) est un arbre basé sur les conflits entre agents individuels, chaque nœud représente un ensemble de contraintes sur le chemin des agents, la recherche se fait sur cet arbre à haut niveau. Au niveau bas, des recherches d'agent unique sont effectuées pour éradiquer les contraintes données par le nœud CT de haut niveau. Nous avons également l'algorithme Meta-Agent CBS (MA-CBS). MA-CBS est une généralisation de CBS. Contrairement à CBS de base, MA-CBS n'est pas limité aux recherches d'agent unique au bas niveau. Au lieu de cela, MA-CBS permet aux agents d'être fusionnés en petits groupes d'agents conjoints. Cela réduit certains des inconvénients du CBS de base et améliore encore les performances. En expérimentant, cet algorithme produit de meilleurs résultats et performances.

La centralisation fait référence au processus dans lequel les activités impliquant la planification et la prise de décision au sein d'une organisation sont concentrées sur un dirigeant ou emplacement. Dans une organisation centralisée, les pouvoirs de décision sont conservés dans le siège social et tous les autres bureaux reçoivent des commandes du bureau principal [42] .

les cadres et les spécialistes qui prennent les décisions critiques sont basés au siège social. De même, dans une structure gouvernementale centralisée, le pouvo

Concentré au sommet, et tous les autres niveaux inférieurs suivent les directions venant de le sommet de la structure organisationnelle

Les systèmes centralisés sont des systèmes qui utilisent une architecture client/serveur où un ou plusieurs nœuds clients sont directement connectés à un serveur central. Il s'agit du type de système le plus couramment utilisé dans de nombreuses organisations où un client envoie une demande à un serveur d'entreprise

2.8. La recherche taboue :

Cette méthode dont l'origine remonte à 1977, a été formalisée plus tard, 1986 par Glover, elle n'a aucun caractère stochastique et utilise la notion de mémoire pour éviter de tomber dans un optimum local [43], Le principe de cette méthode est d'examiner le voisinage de la solution courante à chaque itération. L'algorithme enregistre la meilleure solution parmi les voisins, même si elle est moins bonne que la solution courante. L'acceptation des solutions moins performantes que la solution courante permet d'éviter de tomber dans un optimum local. Pour éviter de tourner dans un cercle entre plusieurs solutions, l'algorithme interdit le passage par des solutions récemment visitées. En pratique, la méthode stocke dans une liste taboue T, les attribues des derniers solutions visitées. Dans l'itération suivante, la meilleure nouvelle solution voisine enlevé la solution la plus ancienne dans la liste. Dans d'autre cas, la méthode mémorise les mouvements réalises plutôt que les solutions. Ensuite, elle interdit les mouvements inverses. Cette technique est rapide et consomme peu de mémoire [44]

2.9.Comparaison entre les methodes :

Méthode	**Advantages**	**Disadvantages**
Meta heuristiques	**il offre souplesse d'emploi.** **il s'adapte facilement à un grand nombre de problèmes complexe**	**nombreux tests nécessaires pour trouver les bons paramètres.** **temps de calcule excessif dans certaines applications**

CBS	**-Une chaîne de commandement claire** **Une organisation centralisée bénéficie d'une chaîne de commandement claire car chaque personne au sein de l'organisation sait à qui rendre compt** **-Vision focalisée** **Lorsqu'une organisation suit une structure de gestion centralisée,** **-Coûts réduits** **Une organisation centralisée adhère à des procédures et méthodes standard qui guident l'organisation,** **- Mise en œuvre rapide des décisions** **Dans une organisation centralisée** **- Amélioration de la qualité du travail** **Les procédures standardisées et une meilleure supervision dans une organisation centralisée entraîner une amélioration de la qualité du travail**	**-Leadership bureaucratique** **La gestion centralisée ressemble à une forme dictatoriale de leadership où les employés ne sont censés fournir des résultats qu'en fonction de ce que les cadres supérieurs attribuent eux** **-Télécommande** **Les dirigeants de l'organisation subissent une énorme pression pour formuler des décisions pour l'organisation** **-Retards de travail** **La centralisation entraîne des retards dans le travail, car les enregistrements sont envoyés vers et depuis le responsable Bureau** **-Manque de loyauté des employés** **Les employés deviennent fidèles à une organisation lorsqu'ils sont autorisés initiatives dans le travail qu'ils font.**

Alghorithme de colonie de fourmis	**- Très grande flexibilité** **- Parfait pour les problèmes basés sur des diagrammes.**	**- Il a une composante d'évolution et renforcement** **- De grandes qualités peuvent piéger l'algorithme pendant de nombreux cycles et il peut tuer un répondre avant l'abus**
Les abeilles Colonie	**- Efficace pour traquer les meilleurs aménagements et simple à mettre en oeuvre** **- Libérez-vous du problème idéal à proximité** **- Sensible au complexe favorable aux imperfections..**	**- Il a une composante d'évolution et expansion** **- De grandes qualités peuvent piéger l'algorithme pour divers cycles et il peut anéantir une réponse avant l'abus**
Alghorithme genetique	**- Les AG utilisent l'encodage de papa paramètres** **- Les AG ont affaire àune population de se concentre** **- L'utilisation de la transition probabiliste règles à tenir à l'écart du voisinage ideal** **- Synthèse de formes profilées, structures et matériaux composites**	**- La question de l'aménagement moderne organizations** **- Reconnaissance des structures et apprendre en le diminuant** **- Localisation de patrons sur la bioinformatique** **- Synthèse de circuits électroniques**
Optimisation de la recherché de nourriture bactérienne	**puissante capacité de recherche parallèle, et il est simple d'échapper au minimum local.**	**- La mauvaise connexion des bactéries, qui les expose au risque d'atteindre l'optimum local plutôt que l'optimum global.**
Le recherche taboue	**- offre des économies de temps de résolution pour des programme de Gross taille.** **- algorithmes facile à mettre en œuvres**	**Paramètres peu intuitif.** **Aucune démonstration de la convergence**

Conclusion :

Dans cette partie, nous avons introduit une pointe sur la route pour les robots mobile et les différents Calculs qui nous sont accessibles pour le surveiller, pour la divulgation des moyens les plus brefs et distances Dans un climat. Nous avons également vu les différentes étapes ; nous avons également introduit le les avantages Et les limites de chaque technique et traqué qu'aucune approche d'itinéraire qui nie une réponse qui efface Tous les problèmes a été expérimentée. En conséquence, nous voyons que la route est un domaine D'exploration exceptionnellement dynamique et que de nouvelles stratégies apparaissent régulièrement. Cette vérification de l'écriture révèle l'importance d'utiliser des stratégies heuristiques pour comprendre Le mouvement question de calcul comme une question de rationalisation à la lumière d'un souci légitime De remplir quelques exigences simultanément. Dans le chapitre suivant, nous discuterons de la conception De notre proposition pour résoudre le problème de planification d'une trajectoire pour un robot mobile
Autonome utilisant alghorithme CBS.

3. Introduction:

Les robots sont généralement des machines qui incluent des ordinateurs ; Cependant, un contrôleur de Robot est unique à chaque robot et ne peut donc pas être transféré à un autre. Cependant, une année modèle pour Un contrôleur de robot est nécessaire à la fois dans l'industrie et dans la recherche. En termes de recherche,Le développement d'applications nécessitant qu'un robot partage son environnement avec son environnement Conduit à rechercher des architectures évolutives et fiables trèséloignées des architectures statiques des robots Traditionnels dont l'environnement immédiat est fermé au facteur humai. Les methodologies d'intelligence Artificielle (algorithme CBs) pour définir et concevoir des systèmes en temps réel devraient être utiles pour modéliser Le système complexe qu'est un contrôleur de robot dans ce contexte. Ces dernières années, de nombreuses etudes Se sont concentrées sur la planification automatique de trajectoires pour des robots sans collision dans Des environnements. De nombreuses stratégies de planification ont été proposées au cours des deux Dernières décennies, mais seules quelques-unes aident à résoudre le problème. Notre méthode de planification de chemin fournit une solution sûre et efficace au problème de planification de Base, ainsi qu'au défi d'optimisation. Le chemin est déterminé dans notre travail par la modélisation de l'algorithme

CBS.Nous découvrirons également l'environnement CBS, son mécanisme de fonctionnement et les solutions qu'il

Nous propose.

3.1. Presentation Du travaille

Notre travaille consiste a trouvé le chemin le plut court entre :

- ✓ le point de départ
- ✓ le point d’arriveé

En etilisant une technique parmi les techniques intiligent artificial pour etre bien precesu les systemès multiagents

donc dans cette partie nous detaille deferent module dans notre application.

3.1.1. La problématique :

Dans la problématique on pose la question Comment un robot trouve le plus courtChemin d’un point de départ

jusqu’au point d’arrivé pour atteindre un but désiré ?

3.1.2. La solution proposée :

Pour résoudre le problème de plus court chemin on a proposée comme une solutionle principe de l'algorithme de CBS car :

Les Agent sont des robot mobile utilisent l'environnement comme moyen de communication, l'échange d'informations indirectement en déposant les phéromones,L'existence de cette matière (phéromones) est liée à la distance, plus la distance est longue, plus la concentration de matière devient faible, si le chemin est court on remarque des trace forte de cette matière, en conséquence les fourmis suivent la tracela plus forte ce qui indique le chemin le plus court.

3.2. L'architecture de l'application :

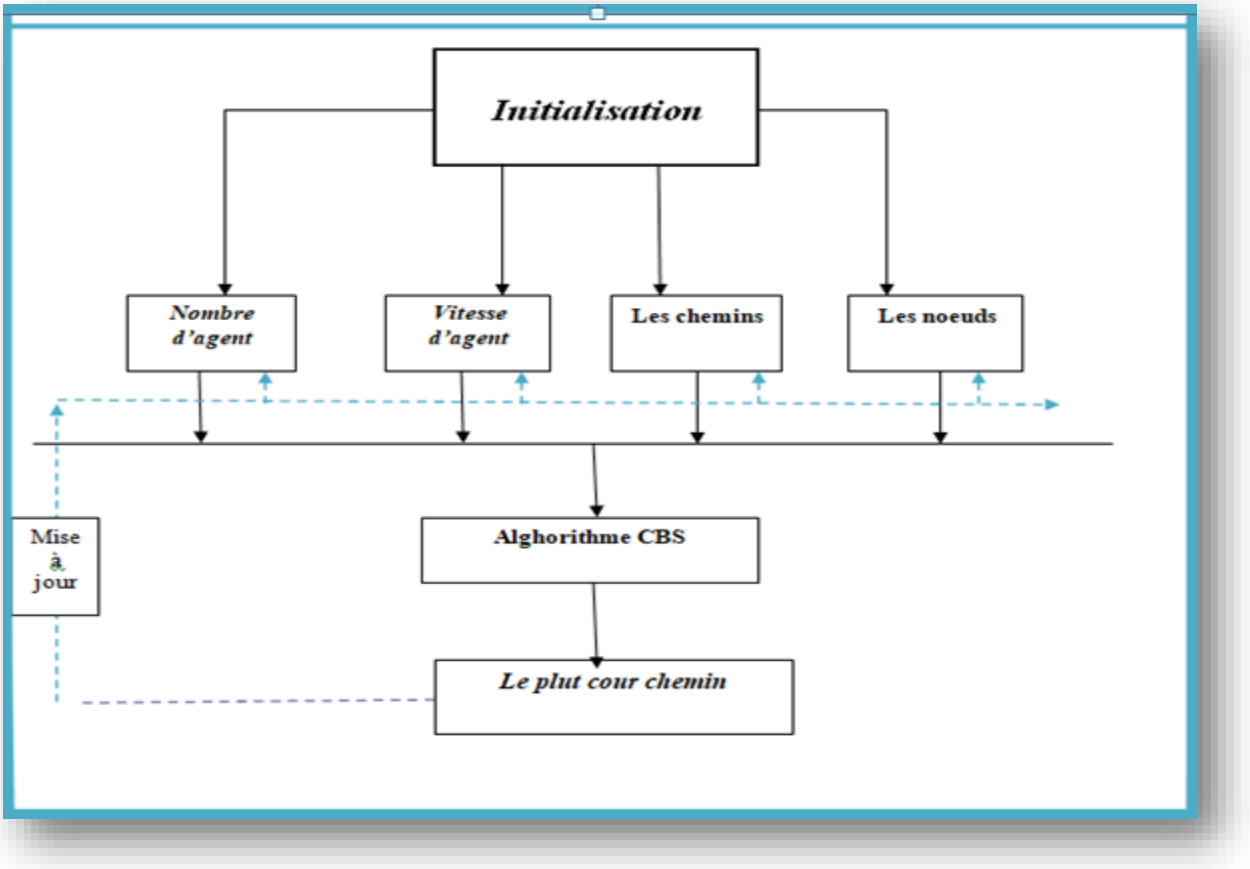


Figure 2.1: Architecture du système.

4.La modélisation conceptuelle

Le modèle conceptuel de données est une représentation statique du système d'information. Il a comme objectif de constituer une représentation claire et cohérentedes données manipulées dans le système d'information.

Dans le cadre de notre projet, nous avons opté pour le langage UML comme une approche de conception, car c'est un langage formel et normalisé en termes de modélisation objet. Son indépendance par rapport aux langages de programmation, aux domaines de l'application et aux

processus, son caractère mobile et sa souplesse ont fait de lui un langage universel. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation des solutions. L'aspect de sa notation, limite l'ambigüité et les incompréhensions.

4.1 Le diagramme de cas utilisation :

Un diagramme de cas d'utilisation pourrait résumer les spécificités des utilisateurs d'un système (également appelé acteurs) et leurs interactions avec le système. Un diagramme de cas d'utilisation efficace peut aider à consulter et à représenter les scénarios dans lesquels un système ou une application interagit avec des personnes, des organisations ou des systèmes externes, les objectifs que le système ou l'application aide ces entités (appelées acteurs) obtiennent et essentiellement la portée du système, le diagramme de cas UML Utilisation.

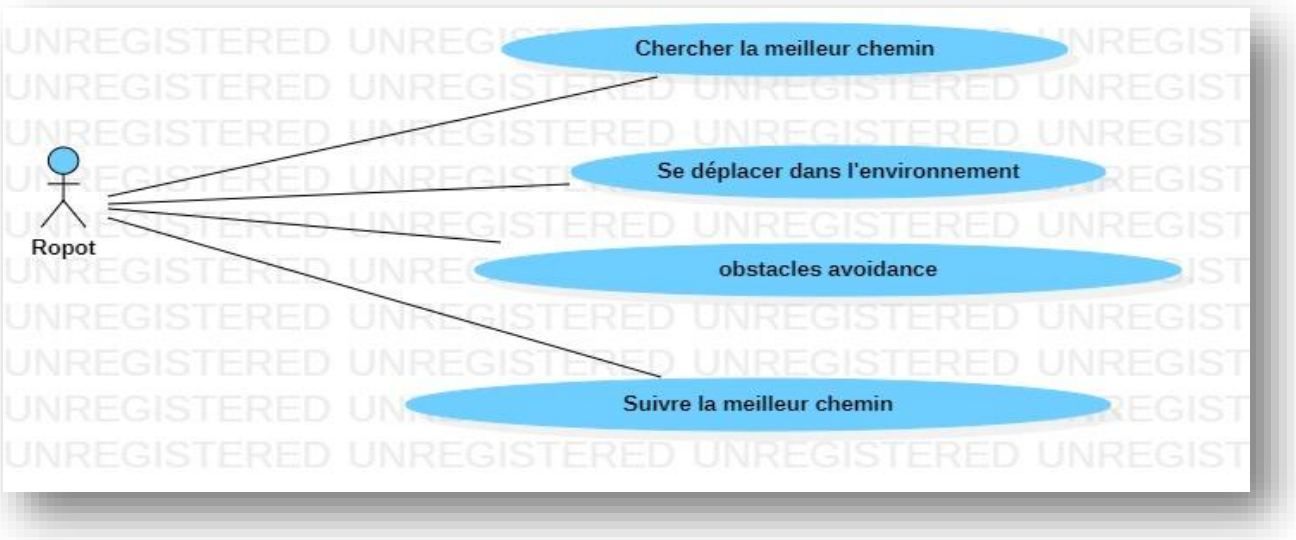


***Figure* 2.2 : diagramme de cas d'utilisation**

Est une représentation visuelle des intercommunications entre l'utilisateur et le système. Il représente la méthodologie utilisée pour analyser le système pour définir, clarifier et organiser des exigences système.

L'acteur principal de ce diagramme de cas d'utilisation est le robot, qui effectue différents types d'utilisations telles

Que la recherche et l'évitement. Avec notre robot étant l'entité unique et l'acteur principal, il comporte principalement quatre opérations fondamentales à accomplir:

Recherchez le meilleur chemin: le robot est positionné au point de départ d'un environnement avec une cible à atteindre. Il est entièrement à la hauteur de découvrir le chemin le plus court vers son objectif désigné; Suivez le meilleur Chemin trouvé: Une fois la cible acquise, le robot

commence à se déplacer vers elle avec chaque itération le récupère de plus près et avec chaque itération, elle recalcule pour trouver le point le plus proche le plus proche de la cible Jusqu'à ce qu'il atteigne; Détection et évitement des obstacles: Maintenant, notre robot fonctionne dans un environnement riche en obstacles. Les fixes sont simples et faciles à manipuler, mais les obstacles mobiles sont un peu plus difficiles et nécessitent un peu plus d'attention.

4.2.. Les diagrammes de séquence :

le diagramme de séquence sont une solution de modélisation dynamique privilégiée en UML car elles se concentrent spécifiquement sur des lignes de vie, ou les processus et objets existants simultanément, et les messages échangés entre eux pour atteindre une fonction avant la fin de la ligne de vie. Les avantages d'un diagramme de séquence doivent: illustrer les détails d'un cas d'utilisation UML. Modèle la logique d'une procédure, d'une fonction ou d'une opération sophistiquée. Affiche comment les objets et les composants interagissent les uns avec les autres pour atteindre un processus. Planifier et appréhender la fonctionnalité détaillée d'un scénario existant ou à venir

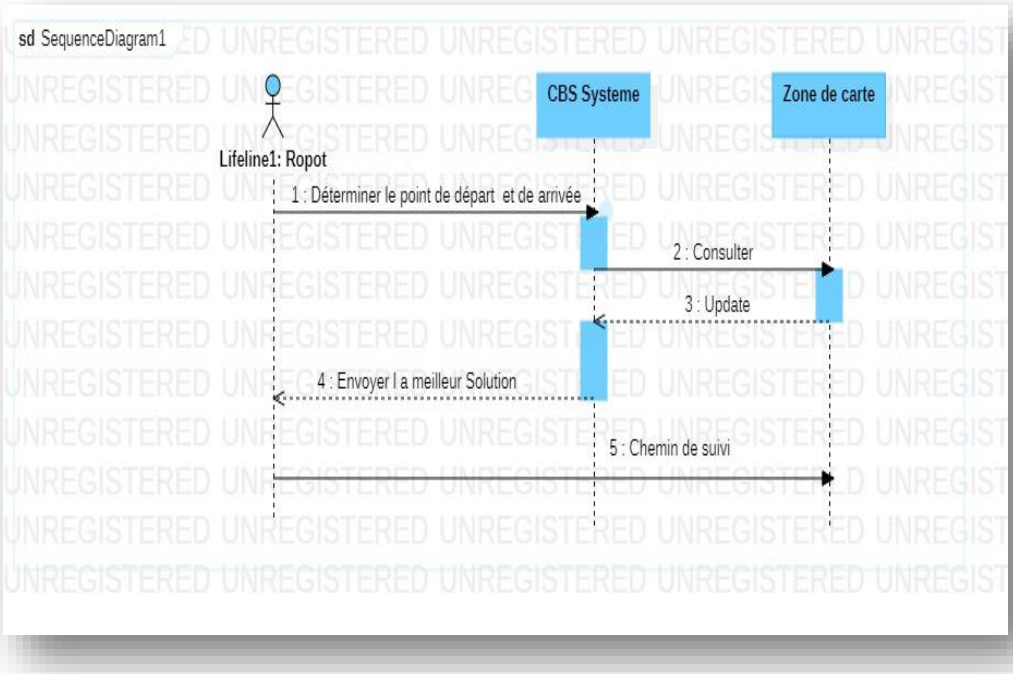


***Figure 2.3* : diagramme de sequence**

4.3. Les diagrammes de classe :

la classe sont une sorte de diagramme de structure car ils définissent ce qui doit être présent dans le système modélisé. UML a été développé comme modèle standardisé pour représenter une méthode de programmation orientée objet. Étant donné que les classes sont le bâtiment des objets, les diagrammes de classe sont les blocs de construction de UML. Les différents éléments du diagramme de classe peuvent décrire les classes qui seront programmées, les objets principaux ou les interactions entre les classes et les objets. Les avantages des diagrammes de classe UML sont les suivants: démontrer des modèles de données pour les systèmes d'information, qu'ils soient simples ou complexes. Mieux comprendre le synopsis général des schémas d'une application. Exprimer visuellement tout essentiel distinct d'un système et diffuser cette information.

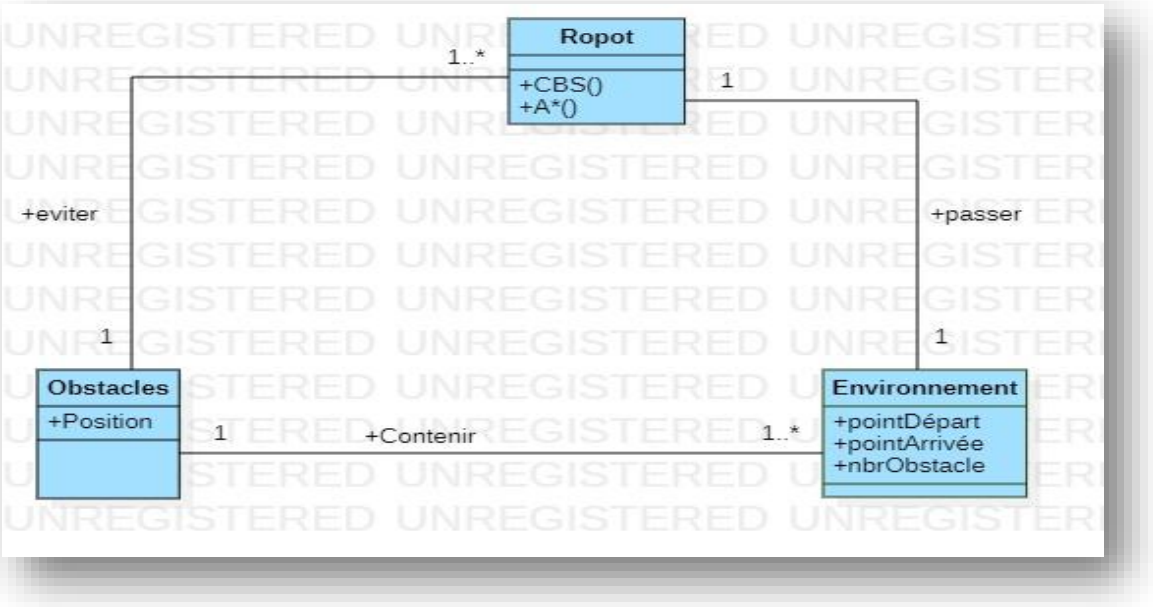


***Figure2.4*: diagramme de classe**

4.4. Approche centralisée :

Il s'agit en effet de représenter les éléments d'un ensemble en regroupant leurs caractéristiques descriptives dans

une entité centrale.Ces entités sont alors utilisées pour traiter des requêtes et trouver un ensemble de Documents correspondant aux caractéristiques définies par la requête considérée .L'adoption d'un index central Gagne aussi les grandes entreprises.Pour ces grandes entreprises, la centralisation des données est une solution fréquemment employée via la création d'un meta-annuaire fédérant les different Annuaires existants.Cependant, on lui préferre parfois un agent

particulier, le broker, proposant un Ensemble de services pouvant s'étendre au-delà de la gestion d'un index.

4.4.1. Contenu :

Le contenu qui complète le code source de CBS est organisé dans les dossiers suivants :

Démos : contient des animations des solutions obtenues par CBS.

Exemple : contient les exemples des fichiers XML d'entrée_sortie requis _fournis par l'alghorithme.

Instance*s* : contient les archives avec les cartes et les instances (dans le format pris en charge par le CBS)

Qui ont été utilisées pour l'évaluation expérimentale décrite dans les articles susmentionnés sur le CBS

ExpResults : contient les résultats tabulaires bruts obtenus lors de l'evaluation experimentale de l'alghorithme CBS.

4.6. Problème d'entrée :

L'entrée du problème de recherche de chemin multi-agents (MAPF) est :

(1)Un graphe orienté g(V,E). Les sommets du graphe sont des emplacements possibles pour les agents, et les arêtes sont les transitions possibles entre les emplacements.

(2)k agents étiquetésun1,un2…unk. Chaque agentunjea un sommet de départ,débutje∈Vet un sommet de but,objectifje∈V.Le temps est discrétisé en points temporels. Au point de tempst0agentunjeest situé à l'emplacementdébutje.

4.7. Action :

Entre des instants successifs, chaque agent peut effectuer une action de déplacement vers un sommet voisin ou une action d' attente pour rester inactif à son sommet actuel.Il existe plusieurs façons de gérer la possibilité qu'une chaîne d'agents se succède à un pas de temps donné.Cela peut ne pas être autorisé, ne peut être autorisé que si le premier

agent de la chaîne se déplace vers un emplacement inoccupé ou cela peut être autorisé même dans une chaîne

cyclique qui ne comprend aucun emplacement vide.Notre algorithme est applicable à toutes ces variations.

4.8. Contraintes MAPF :

La contrainte principale dans MAPF est que chaque sommet peut être occupé par au plus un agent à un instant

donné. Il peut également y avoir une contrainte interdisant à plus d'un agent de traverser le même bord entre des

pas de temps successifs. Un conflit est un cas où une contrainte est violée.

4.9. Tâche MAPF :

Une solution au problème MAPF est un ensemble de chemins non conflictuels, un pour chaque agent, où un chemin pour l'agentunjeest une suite de{mouvement,Attendez}actions telles que siunjeeffectue cette séquence d'actions

à partir dedébutje, cela finira dansobjectif

4.10. Fonction de coût :

Nous décrivons les algorithmes dans cet article dans le contexte d'une fonction de coût commune que nous appelon

s la somme des coûts .Makespan , par exemple, est une autre fonction de coût MAPF courante qui minimise le temps

total jusqu'à ce que le dernier agent atteigne sa destination La fonction de coût de carburant est en fait la somme des coûts

de tous les agents où seules les actions de déplacement entraînent des coûts mais les actions d'attente sont gratuites.Certaines variantes de MAPF n'ont pas de fonction de coût globale à optimiser, mais un ensemble de fonctions de coût individuelles, une pour chaque agent .Yu et Lavelle ont étudié une variante MAPF dans laquelle

au lieu d'attribuer à chaque agent une position d'objectif, un ensemble de positions d'objectif est donné et la tâche consiste à trouver une solution qui amène chacun des agents à n'importe quelle position d'objectif.

5. Enquête sur les algorithmes MAPF centralisés :

Approche centralisée peuvent être divisés en trois classes.La première classe de solveurs, utilisée dans des travaux récents, réduit MAPF à d'autres problèmes bien étudiés en informatique.La deuxième classe de solveurs est constituée de solveurs sous-optimaux spécifiques à MAPF.

La troisième classe de solveurs est la classe des solveurs optimaux.Cet article se concentre sur les solveurs optimaux, mais nous incluons un bref aperçu des autres classes ci-dessous :

5.1.. L'algorithme de recherche basé sur les conflits (CBS) :

Nous passons maintenant à la description de notre nouvel algorithme, l’algorithme de recherche basé sur les conflits .Plus tard, , nous présentons une généralisation à CBS appelée recherche basée sur les conflits de méta-agents .En revanche, dans un problème de recherche de chemin à agent unique,k=1, et l'espace d'état n'est linéaire que dans la taille du graphe.

CBS résout MAPF en le décomposant en un grand nombre de problèmes de recherche de chemin à agent unique contraints.Chacun de ces problèmes peut être résolu en temps proportionnel à la taille de la carte et à la longueur de la solution, mais il peut y avoir un nombre exponentiel de tels problèmes à agent unique.

Haut niveau : Dans la section suivante, nous décrivons le processus de haut niveau de CBS et l'arbre de recherche

Qu'il recherche.

L'arbre des contraintes :

La racine du CT contient un ensemble vide de contraintes.L'enfant d'un nœud dans le CT hérite des contraintes du parent et ajoute une nouvelle contrainte pour un agent.Un ensemble de k chemins, un chemin pour chaque agent.Le nœud N dans le CT est un nœud de but lorsqueN.la solutionest valide, c'est-à-dire que l'ensemble des chemins pour tous les agents n'a pas de conflits.Le niveau supérieur effectue une recherche du meilleur premier sur le CT où les nœuds sont classés par leurs coûts.

5.2.3. Traitement d'un nœud dans le CT :

La recherche de bas niveau renvoie un chemin le plus court pour chaque agent, qui est cohérent avec toutes les contraintes associées àunjeau nœud N .Une fois qu'un chemin cohérent a été trouvé pour chaque agent ces chemins

Sont ensuite validés par rapport aux autres agents.La validation est effectuée en parcourant toutes les étapes de

Temps et en faisant correspondre les emplacements réservés par tous les agents.Si deux agents ne prévoient pas

D'être au même endroit en même temps, ce nœud CT N est déclaré comme nœud de but, et la solution actuelle qui contient cet ensemble de chemins est renvoyé.Si toutefois, lors de la validation, un conflitC= est trouvé pour

Deux agents, la validation s'arrête et le nœud est déclaré nœud non-but.

5.2.4. Résoudre UN conflit:

Par conséquent, au moins une des contraintes ou doit être ajouté à l'ensemble des contraintes dansN.contraintes.L'enfant de gauche résout le conflit en ajoutant la contrainte et le bon enfant ajoute la contrainte

.Au lieu de cela, il ne peut enregistrer que sa dernière contrainte et extraire les autres contraintes en parcourant le chemin De N à la racine via ses ancêtres.De même, à l'exception du nœud racine, la recherche de bas niveau ne doit être effectuée que pour l'agentunjequi est associé à la contrainte nouvellement ajoutée.Les chemins des autres agents

Les mêmes car aucune nouvelle contrainte n'est ajoutée pour eux**.**

5.2.5 Conflits de bord :

Pour plus de simplicité, nous avons décrit uniquement les conflits qui se produisent dans les sommets.Mais si, selon

La définition du problème, les agents ne sont pas autorisés à traverser le même bord dans une direction opposée, des conflits de bord peuvent également se produire.Nous définissons un conflit d'arête comme étant le tuple où deux

Agents "échangent" leurs emplacements entre le pas de temps t et le pas de tempst 1. Une contrainte de bord est

Définie comme, où l'agentunjeest interdit de commencer à se déplacer le long du bord dev1àv2au pas de temps t .Le cas échéant, les conflits d'arêtes sont traités par le niveau supérieur de la même manière que les conflits de sommets.

Conclusion

Des environnements inconnus avec des obstacles dynamiques sur la trajectoire du robot mobile forcent l'algorithme

de planification de trajectoire environnement dans ce travail. Pour trouver le chemin le plus court, cette méthode utilise un algorithme CBS. Dans une dynamique créer un critère de robustesse. Trouver une solution à une bonne performance sensible à l'incertitude est l'objectif de Le robot est capable d'atteindre la destination en évitant les obstacles sur son chemin, selon les données de simulation. Environnement d'obstacles mouvants, le robot peut passe

r de sa position initiale à sa position finale sans entrer en collision. De nouveaux trajets en ligne puis calculer la trajectoire tout en répondant aux exigences des petites heures locales ce critère. L'objectif est d'identifier le chemin le plus rapide pour se rendre à destination. Contrôlabilité. Nous avons présenté notre méthode de navigation autonome pour un robot mobile dans un environnement rempli d'obstacles Nous discutons de notre méthode bactérienne pour guider un robot mobile Et autonome dans ce chapitre. Il a été décidé de pour résoudre des problèmes de calcul extrêmement difficiles qui nécessitent un calcul haut performance pour calculer En plus de naviguer dans des robots mobiles dans des situations dynamiques et compliquées.

6. Introduction :

Dans la section précédente, nous avons mieux compris notre problème. Nous avons vu où se situe notre problème et avons proposé une solution, le CBS. Ensuite, nous avons fait un examen approfondi de notre méthode choisie où nous avons compris exactement comment elle fonctionne et sur quelles bases. Nous avons utilisé le modèle conceptuel

pour une explication plus approfondie des techniques et de son fonctionnement. Des diagrammes comme l'activité

et la séquence ont été utilisés pour aider à expliquer plus efficacement. Dans ce chapitre, cependant, nous obtenons une vue plus pratique. Nous commençons par la présentation des outils et équipements utilisés dans la conception et la

en œuvre du robot de recherche de trajectoire basé sur CBS. Nous avons choisi PYTHON comme langage de programmation principal. Nous avons choisi PYTHON comme langage de programmation principal. Python est l'un des langages de programmation les plus dignes de mention dans le monde moderne. Il figure parmi les langages de programmation à la croissance la plus rapide dans le monde. Il est adaptable, flexible, extrêmement efficace et simple

à utiliser et à développer. Il a une société extrêmement active ainsi. Il est utilisé dans d'innombrables organisations en raison de ses nombreux atouts de paradigme de programmation et de sa mise en œuvre de la gestion automatique de la mémoire. En raison de sa vaste bibliothèque standard, Python est également souvent qualifié de langage inclus dans la batterie.

Outils:

7. Environnement logiciel :

Python est de loin le langage de programmation open source le plus connu et le plus utilisé par les professionnels de l'informatique. Python est un langage de programmation de haut niveau, orienté objet et diagnostiqué, avec une sémantique dynamique. Ses structures de données intégrées de haut niveau, associées à un typage dynamique et à une liaison dynamique, le rendent extrêmement attrayant pour le développement rapide d'applications et pour une utilisation en tant que langage de script ou de collage pour connecter des composants existants.

Python est une syntaxe simple et facile à apprendre qui met en évidence la lisibilité et, par conséquent, réduit les dépenses de maintenance du programme. Python prend en charge les modules et les packages, qui renforcent la modularité du programme et la réutilisation du code.

Python est un langage de programmation renommé. Il a été développé en 1991 par Guido van Rossum. Il est utilisé pour le développement Web (côté serveur), le développement de logiciels, les mathématiques et les scripts système.

Python peut être utilisé : • Sur un serveur pour construire des applications web. • Aux côtés d'un logiciel pour créer des flux de travail. • Pour gérer des mégadonnées et effectuer des calculs complexes. • Pour un prototypage rapide ou pour le développement de logiciels prêts pour la production.

Python peut être lié à des systèmes de bases de données. Il peut également lire et modifier des fichiers. Syntaxe Python comparée à d'autres langages de programmation :

Python a été développé pour la lisibilité et les ressemblances avec la langue anglaise avec l'influence Des mathématiques. • Python exploite de nouvelles lignes pour réaliser une commande, par rapport à d'autres langages de programmation qui utilisent généralement des points-virgules ou des parentheses

. • Python s'appuie sur l'indentation, en appliquant des espaces blancs, pour spécifier une portée ; comme la portée des boucles, des fonctions et des classes. Alors que d'autres langages de programmation appliquent des accolades à cette fin.

Bibliothèques utilisées La bibliothèque standard de Python est très complète et présente un large éventail de structures. La bibliothèque comprend des modules intégrés (écrits en C) qui fournit l'accès aux fonctionnalités du système telles que les E/S de fichiers qui seraient autrement inaccessibles aux programmeurs Python, ainsi que des modules composés en Python qui fournissent des solutions standardisées pour de nombreuses difficultés qui surviennent dans programmation quotidienne. Peu de ces modules sont explicitement développés pour encourager et améliorer la portabilité des programmes Python.

Figure 3.1 : logo officiele Python [41]

- ***NumPy***

NumPy est le package essentiel pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux masqués et des matrices) et une variété de modèles pour des procédures rapides sur des tableaux, y compris mathématiques, logiques, manipulation de forme, tri, sélection, E/S , transformées de Fourier discrètes, algèbre linéaire de base, opérations statistiques de base, simulation aléatoire et bien plus encore.

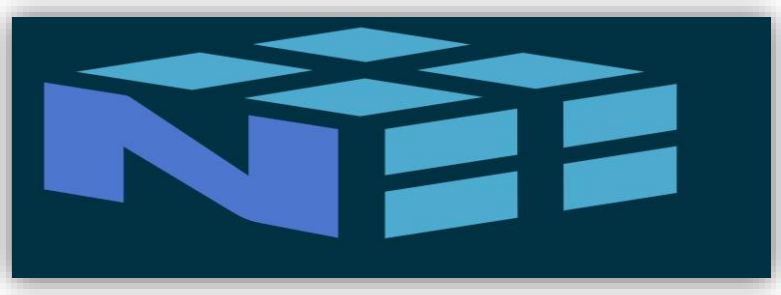


Figure 3.2 : logo officiele numpy[41]

- ***SciPy***

SciPy, une bibliothèque scientifique pour Python, est une bibliothèque open source sous licence BSD pour les mathématiques, les sciences et l'ingénierie. La bibliothèque SciPy s'appuie sur NumPy, qui fournit une manipulation de tableau N-dimensionnel appropriée et rapide. La principale motivation pour créer la bibliothèque SciPy est les fonctions avec les tableaux NumPy. Il fournit de nombreuses pratiques numériques conviviales et efficaces telles que des modèles d'intégration et d'optimisation numériques. L'interpolation est la méthodologie de localisation d'une valeur entre deux points sur une ligne ou une courbe.

Interpolation : la classe interp1d dans scipy.interpolate est une technique pratique pour créer une fonction fondée sur des points de données fixes, qui peuvent être évalués n'importe où dans l'environnement représenté par les données fournies à l'aide d'une interpolation linéaire.

***Figure 3.3 :* logo officiele scipy [41]**

- ***Matplotlib*** :

Matplotlib est une biblithèque complète pour la costruction de visualizations statiques animées et interactives en python

Matplotlib fournit des chiffres de qualité publication dans un mélange de formats papier et d'environnements

interactifs sur toutes les plateformes.

Figure 3.4 : **logo officiele matplotlib [41]**

8. Interfaces de l'application :

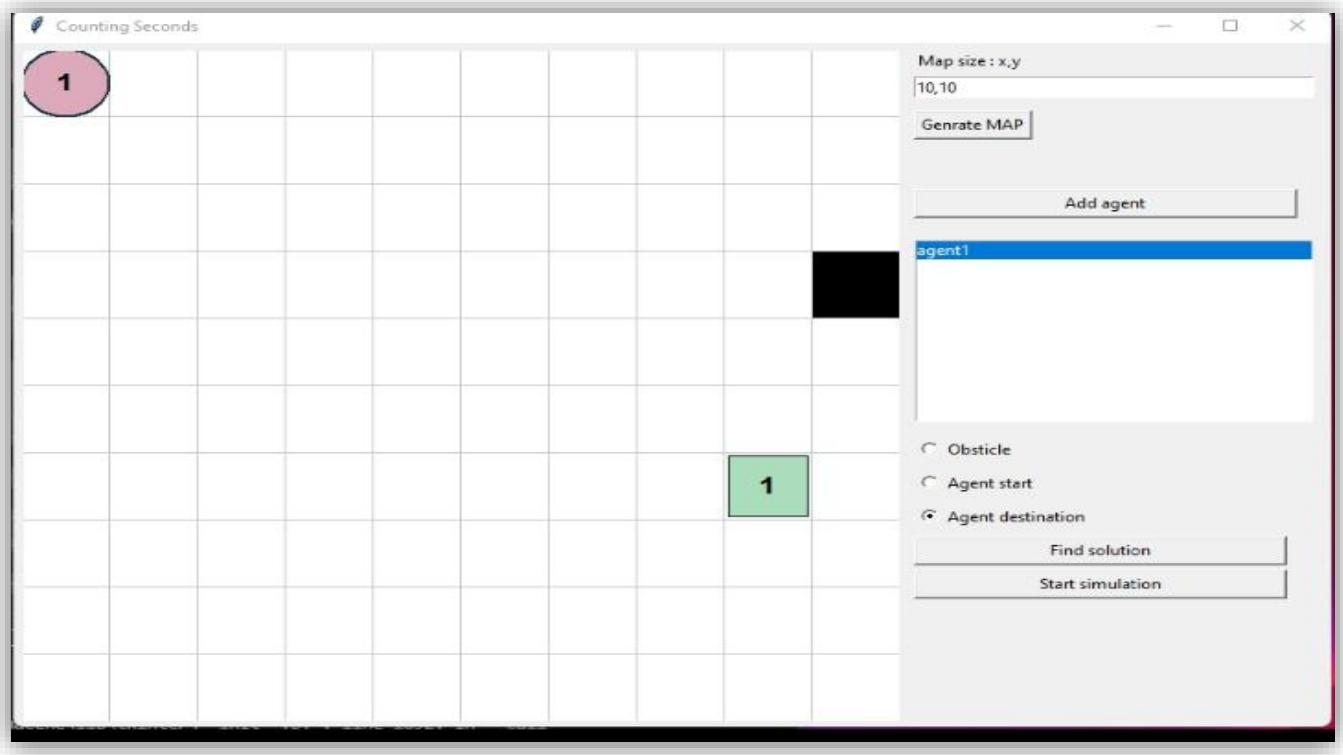


Figure 3.5 : application generale

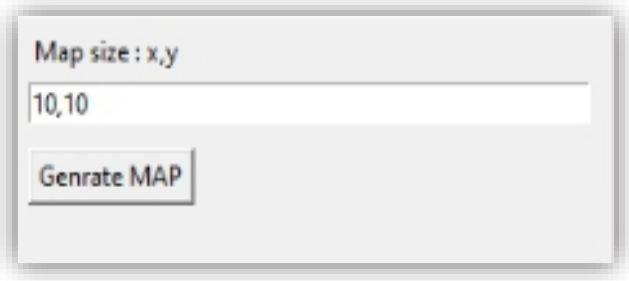


Figure 3.6 : maps size

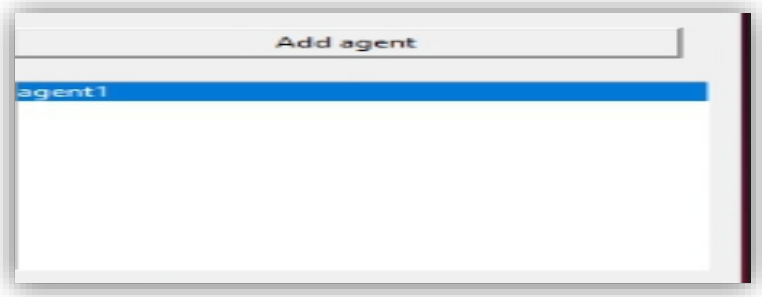


Figure 3.7 : le nombre d'agent

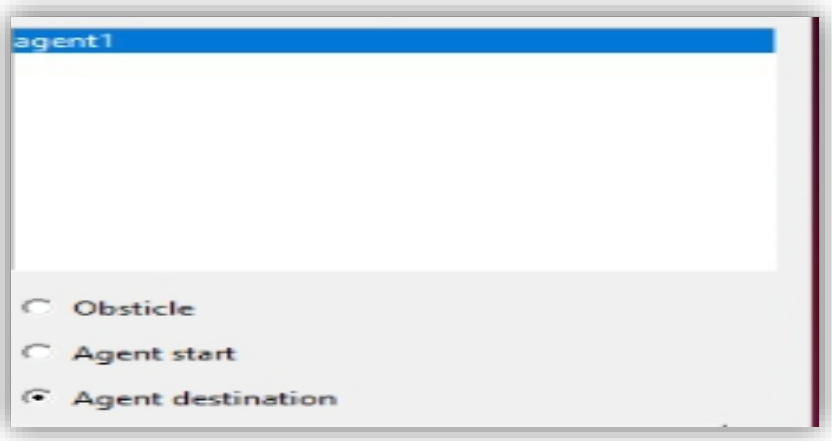


Figure 3.8 : détermination de la position des obstacles, lieu début agent
En plus de situer la fin d'agent

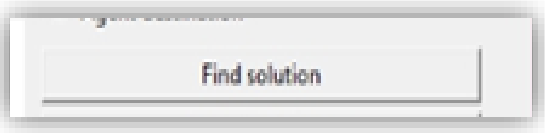


Figure 3.9 : fournir des solutions

8.1. Exemple de 3 Agent :

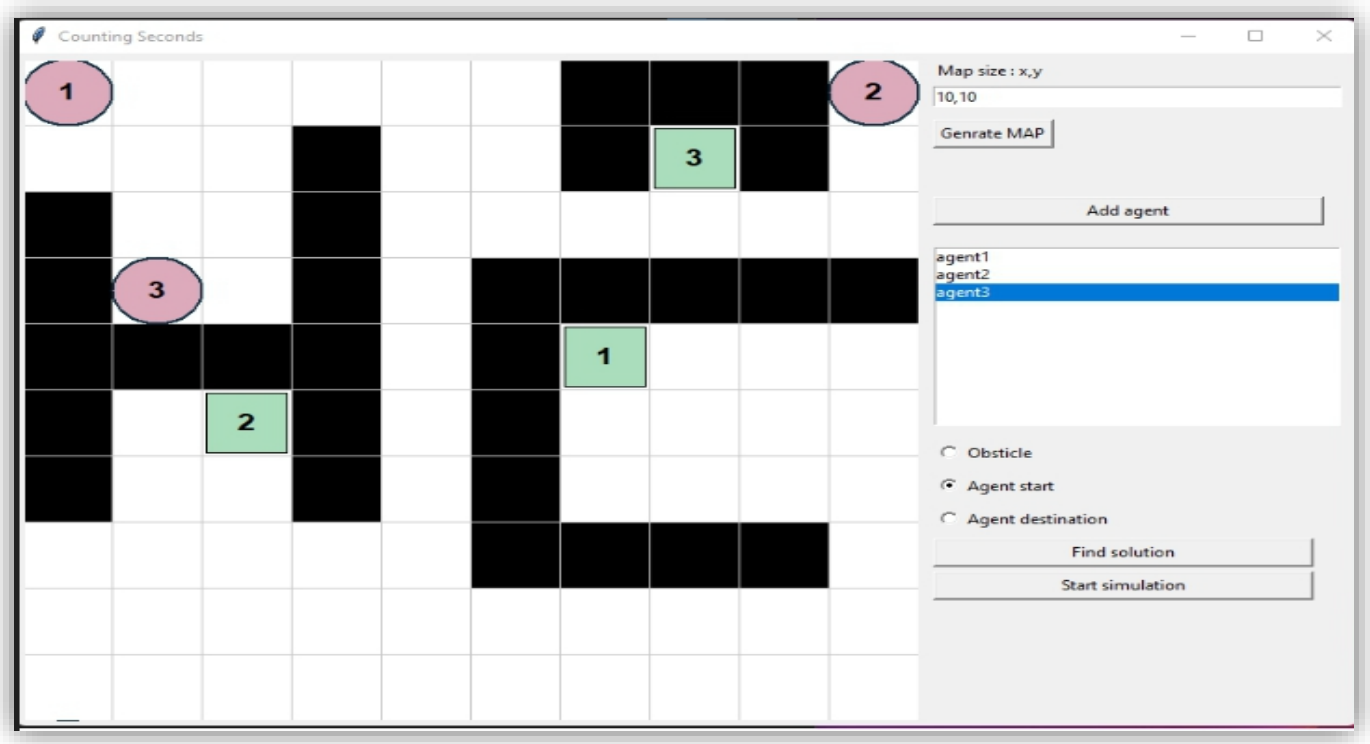


Figure3.10 : un exemple de 3 agent dans lequel les obstacles et les positions de départ et d'arrivée ont identifiés

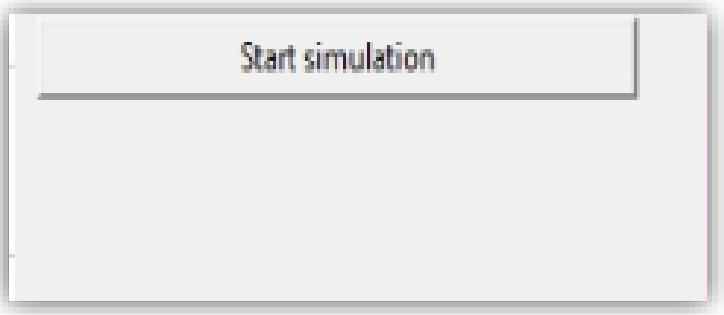


Figure 3.11 : solution du programme

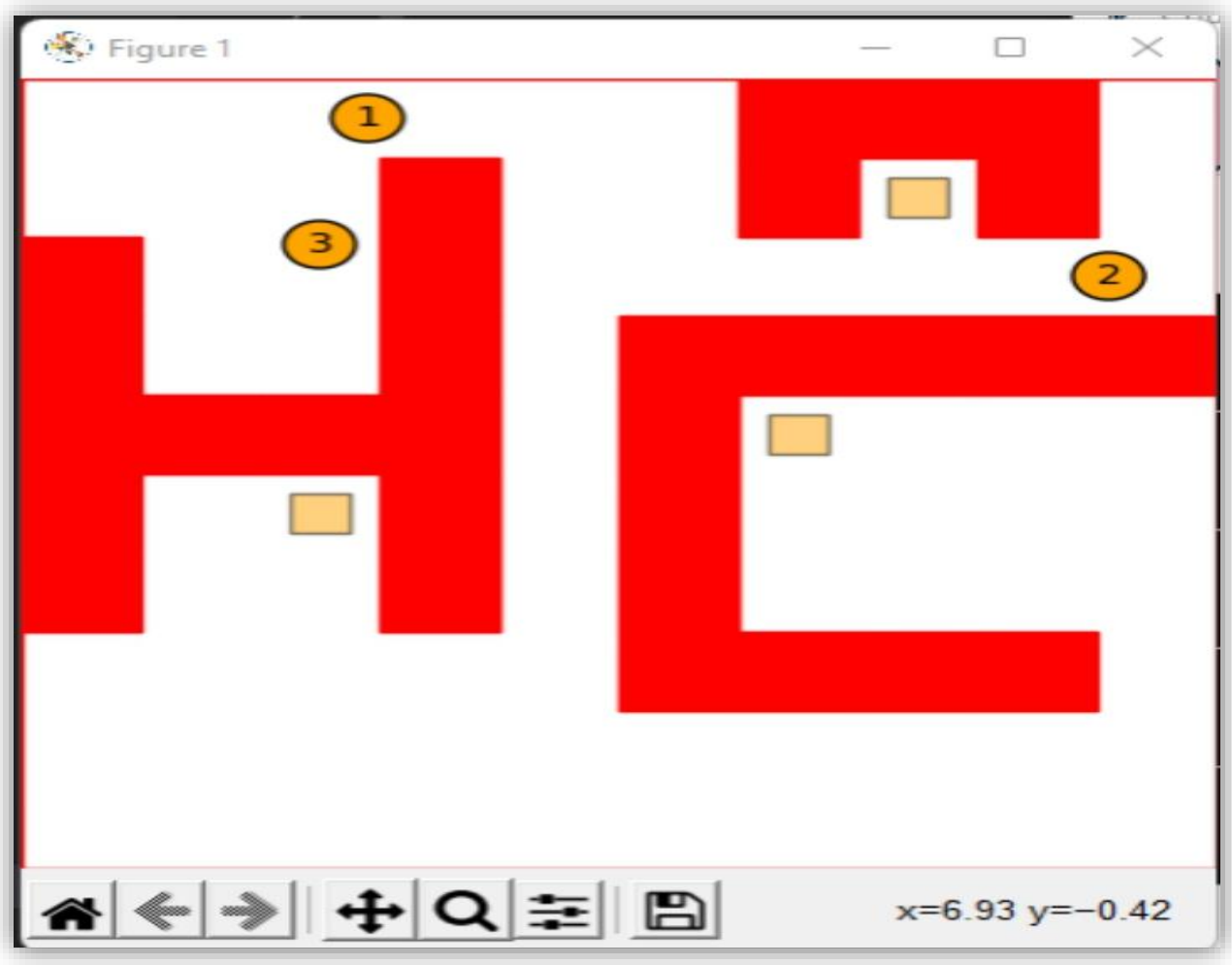


Figure 3.12 : fournir des solutions finales et afficher leur chemin ement en detaille

8.2.Pseudo-code et exemple :

L'algorithme : présente le pseudo-code pour CBS, ainsi que le MA-CBS plus avancé qui sera expliqué plus tard.

Les lignes 11 à 18 ne sont pertinentes que pour MA-CBS. Pour l'instant, supposons que ledevrait fusionner () function renvoie toujours false. Le niveau supérieur a la structure d'une recherche du meilleur premier.

Lors de la validation de la solution à deux agents donnée par les deux chemins individuels, un conflit est détecté lorsque les deux agents arrivent au sommet D au pas de temps 2. Cela crée un conflit .La recherche de bas niveau est maintenant invoquée pour que l'enfant gauche trouve un chemin optimal qui satisfait également la nouvelle contrainte.Il effectue une recherche dans le graphe sous-jacent pour trouver un chemin optimal pour l'agentunjequi satisfait toutes ses contraintes tout en ignorant complètement les autres aagents.Tout algorithme de recherche de chemin d'agent unique peut être utilisé pour trouver le chemin d'un agentunje, tout en vérifiant que les contraintes sont satisfaites.Contrairement à la recherche de chemin à agent unique,

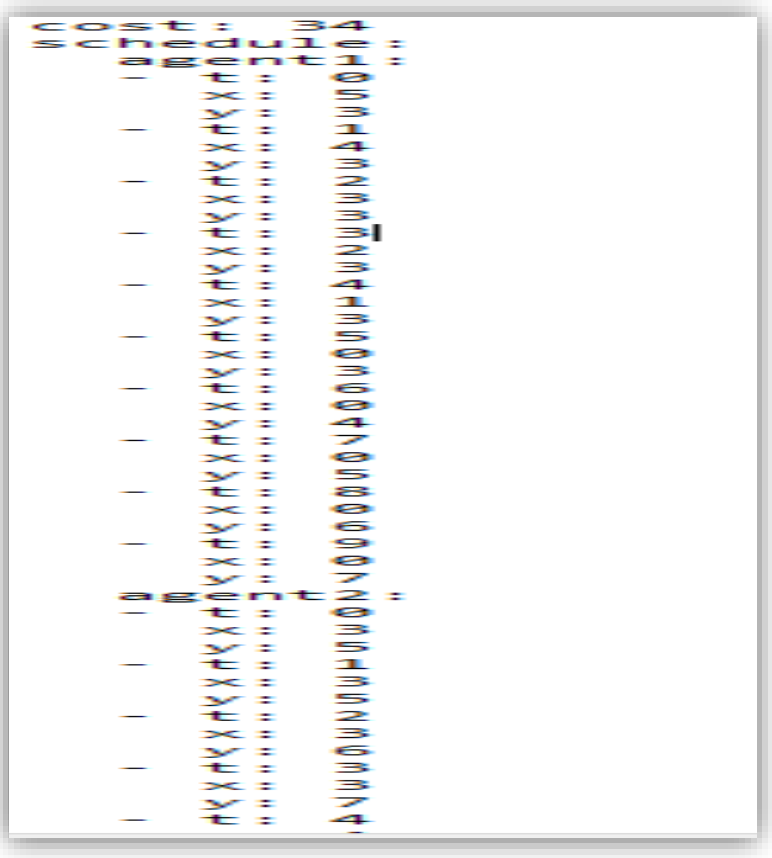
l'espace d'états de bas niveau inclut également la dimension temporelle et les « obstacles » dynamiques causés par les contraintes.

```
Input: MAPF instance
1  Root.constraints = ∅
2  Root.solution = find individual paths by the low level()
3  Root.cost = SIC(Root.solution)
4  insert Root to OPEN
5  while OPEN not empty do
6      P ← best node from OPEN // lowest solution cost
7      Validate the paths in P until a conflict occurs.
8      if P has no conflict then
9          return P.solution // P is goal
10     C ← first conflict (a_i, a_j, v, t) in P
11     if shouldMerge(a_i, a_j) // Optional, MA-CBS only then
12         a_{i,j} = merge(a_i, a_j, v, t)
13         Update P.constraints(external constraints).
14         Update P.solution by invoking low level(a_{i,j})
15         Update P.cost
16         if P.cost < ∞ // A solution was found then
17             Insert P to OPEN
18         continue // go back to the while statement
19     foreach agent a_i in C do
20         A ← new node
21         A.constraints ← P.constraints + (a_i, v, t)
22         A.solution ← P.solution
23         Update A.solution by invoking low level(a_i)
24         A.cost = SIC(A.solution)
25         if A.cost < ∞ // A solution was found then
26             Insert A to OPEN
```

9. Codage de bouton de réglage dans l'application :

```
tk.Label(r,text='Map size : x,y').place(x=615,y=5)
self.map_size_text = tk.Entry(r,width=45)
self.map_size_text.place(x=615,y=30)
self.map_size_text.insert(0,'10,10')

tk.Button(r,text='Genrate MAP',command=self.generateMap).place(x=615,y=60)

button = tk.Button(r, text='Stop', command=r.destroy)
button.place(x=100,y=100+pd)

self.pl = self.PlotView(r=r,parent=self)
self.pl.get_canvas()

add_agent_btn = tk.Button(r,text='Add agent',command=self.addAgent,width=36)
add_agent_btn.place(x=615,y=30+pd)

self.lb = tk.Listbox(r,width=45,height=10)
self.lb.insert(1,'agent1')

self.lb.place(x=615,y=75+pd)
padding = 250+pd
tk.Radiobutton(r,text='Obsticle',value=0,variable=self.dtype).place(x=615,y=padding)
tk.Radiobutton(r,text='Agent start',value=1,variable=self.dtype).place(x=615,y=padding+30)
tk.Radiobutton(r,text='Agent destination',value=2,variable=self.dtype).place(x=615,y=padding+60)

find_solution_btn = tk.Button(r,text="Find solution",command=self.findSolution,width=35).place(x=615,y=padding+90)
start_simulation_btn = tk.Button(r,text="Start simulation",command=self.startSimulation,width=35).place(x=615,y=padding+120)
r.mainloop()
```

9.1. source Out put yaml :



Cost : Cette caractéristique signifie combien de temps nous avons besoin pour terminer le travaille (les mouvments nécessaires effectués par agent).

Schedule : le tableau

A l'instant 0 : t=0.x=5.y=3

Quand agent fait un mouvement de t=1.devient x=4.y=3

L'orsqu'un agent fait un mouvement de t=2 .devient x=3.y=3

Ainsi de suite jusqu'à la fin du nombre de coups et l'arrivée de chaque agent aux positions finale, et de ces opérations

Le nombre final nombre de mouvement nécessaires pour terminer le travaile de l'application est calculé.

X et y changent d'un agent à l'autre en changeant le chemain suivi par chaque agent et en changeant les positions de debut et de fin.

9.2.Code source Imput yaml :

```yaml
agents:
- goal: !!python/tuple
  - 0
  - 7
  name: agent1
  start: !!python/tuple
  - 5
  - 3
- goal: !!python/tuple
  - 6
  - 0
  name: agent2
  start: !!python/tuple
  - 3
  - 5
- goal: !!python/tuple
  - 6
  - 7
  name: agent3
  start: !!python/tuple
  - 1
  - 7
- goal: !!python/tuple
  - 7
  - 0
  name: agent4
  start: !!python/tuple
  - 7
  - 0
- goal: !!python/tuple
  - 7
  - 7
  name: agent5
  start: !!python/tuple
  - 7
  - 7
```

```yaml
map:
  dimensions:
  - 8
  - 8
  obstacles:
  - !!python/tuple
    - 0
    - 2
  - !!python/tuple
    - 1
    - 4
  - !!python/tuple
    - 1
    - 2
  - !!python/tuple
    - 2
    - 4
  - !!python/tuple
    - 2
    - 2
  - !!python/tuple
    - 3
    - 4
  - !!python/tuple
    - 3
    - 2
  - !!python/tuple
    - 3
    - 1
  - !!python/tuple
    - 4
    - 6
  - !!python/tuple
    - 4
    - 5
```

- ***Name*** : chaque agent à un nom.
- **Start** : debut agent.
- ***Goal*** : fin agent.
- ***Map***
- ***Size*** : combien de point dimensions.
- ***Les obstacles*** : l'emplacement des obstacles .chaque obstacle a une virgule et un ordre.

10. Domaines d'application :

- Contrôle de robot.
- -L'ordonnancement de tâches.
- Clustering de données.
- Conception électroniques et mécanique.
- -Optimisation de filtre digital.
- -L'optimisation de fonction

Conclusion :

Ce chapitre présenté la phase de développement de notre projet (matériel et logiciel). À travers la présentation des interfaces de l'application nous avons donné une idée claire sur les taches qui sont réalisées dans notre application par la présentation.

Dans cette partie, nous avons mis au point un processus de décision collectiveà partir d'un modèle CBS, peuvent conduire un groupe de robots, aux capacités sensorielles et cognitives limitées, à réaliser un choix collectif pour une navigation optimale dans un environnement avec des obstacles.

Conclusion générale

Notre projet de fin d'étude traite la problématique de la navigation autonome des robots mobiles dans des environnements dynamiques pour résoudre le problème de plus court chemin par l'algorithme de colonie de fourmis.

Notre contribution porte sur le développement de nouvelles méthodes inspirées de l'intelligence artificielle pour assurer une navigation optimisée quel que soit l'environnement où doit évoluer les robots.

L'objet de notre mémoire porte sur les potentiels donnés par les technologies d'automatisation pour développer des procédés de construction durables pour les

Bâtiments. Les aspects économiques, environnementaux et sociaux de tels processus sont appréciés. Les principaux facteurs qui poussent à un changement dans le

processus de construction sont la réduction du nombre de travailleurs de la

construction, l'augmentation du coût et de la durée limitée des projets, la pression sur

L'industrie de la construction et de la consommation de matériaux de haute dans le

Secteur de la construction

En robotique mobile, la poursuite de la trajectoire qui est une tache élémentaire, représente la base de toute autre tâche du robot. Dans la stratégie que nous avons utilisée, la trajectoire à suivre est représentée par un certain nombre de points de passage entre le point de départ et le point d'arrivée. Le robot exécute, point par point, sa trajectoire, et corrige son orientation en fonction de sa position par rapport au pointdésiré.

Dans nos travaux futurs, nous essayerons d'améliorer davantage notre algorithme CBS , afin de rendre notre algorithme plus souple, et s'approcher au maximum de la longueur réelle du chemin.

Enfin, nous pensons que le travail présenté dans cette mémoire ouvre de nouvelles perspectives selon les principaux axes suivants :

L'application des architectures présentées pour des environnements, laversion de base de ces systèmes déplace le robot entre positions sans collision. L'opération est régie par un processus perception-action répété à une fréquence élevée. Le résultat est une séquence de mouvements en ligne qui conduit le robotjusqu'au but sans collisions.

References

1_ Sebastian ,T .Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots Robot. Auton. Syst., 41 (2) (2002_

- **2**_Subhrajit, B .Distributed optimization with pairwise constraints and its application to multi-robot path planning Robotics: Science and Systems (2010)
- **3**_Subhrajit, B. Topological constraints in search-based robot path planning Auton. Robots, 33 (3) (2012)
- **4**_Zahy ,B, Ariel Felner Conflict-oriented windowed hierarchical cooperative A* International Conference on Robotics and Automation (ICRA) (2014)
- **5**_Zahy ,B Roni Stern, Ariel Felner, Roie Zivan, Steven Okamoto Multi-agent path finding for self interested agents Symposium on Combinatorial Search (SOCS) (2013)
- **6**_Blai ,B. Planning as heuristic search Artif. Intell., 129 (1) (2001)
- **7**_Jorjeta Jetcheva, A .performance comparison of multi-hop wireless ad hoc network routing protocols Proceedings of the International Conference on Mobile Computing and Networking, ACM (1998).
- **8**_Benjamin ,C .Single- and dual-arm motion planning with heuristic search Int. J. Robot. Res. (2013).
- **9**_Paul , S. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications Symposium on Foundations of Computer Science, IEEE (1984).
- **10**_ Adriaan ,W. ter Mors, Cees Witteveen Push and rotate: cooperative multi-agent path planning AAMAS (2013).
- **11**_Rina Dechter, Judea Pearl Generalized best-first search strategies and the optimality of A* J. ACM, 32 (3) (1985).
- **12**_Kurt ,M. Peter Stone A multiagent approach to autonomous intersection management J. Artif. Intell. Res., 31 (2008).
- **13**_ Doga G. , A general formal framework for pathfinding problems with multiple agents AAAI (2013) .
- **14**_Michael ,E. On multiple moving objects Algorithmica, 2 (1–4) (1987).
- **15**_ Nathan ,R. Partial-expansion A* with selective node generation AAAI (2012) .
- **16**_Nathan, S. PHA*: finding the shortest path with A* in an unknown physical environment J. Artif. Intell. Res., 21 (2004).
- **17**_Cornelia, F, Glenn Wagner, Howie Choset ODrM* optimal multirobot path planning in low dimensional search spaces International Conference on Robotics and Automation (ICRA) (2013).
- **18**_Arnon, G, Distributed navigation in an unknown physical environment AAMAS, ACM (2006).

References

- **19**_Meir ,D. A* variants for optimal multi-agent pathfinding Symposium on Combinatorial Search (SOCS) (2012)
- **20**_Nathan?R. Sturtevant, Robert C. Holte, Jonathan Schaeffer Enhanced partial expansion A* J. Artif. Intell. Res., 50 (2014).
- **21**_ Lydia,M E. Kavraki Asynchronous distributed motion planning with safety guarantees under second-order dynamics Algorithmic Foundations of Robotics IX, Springer (2011).
- **22**_ Bertram Raphael ,A .formal basis for the heuristic determination of minimum cost paths Syst. Sci. Cybern., 4 (2) (1968).
- **23**_Malte, H. Understanding Planning Tasks: Domain Complexity and Heuristic Decomposition vol. 4929, Springer (2008) .
- **24**_ Nathan R. Sturtevant A new approach to cooperative pathfinding AAMAS (2008).
- **25**_Mokhtar M.. Sturtevant A polynomial-time algorithm for non-optimal multi-agent pathfinding Symposium on Combinatorial Search (SOCS) (2011) .
- **26**_Richard E. Korf Depth-first iterative-deepening: an optimal admissible tree search Artif. Intell., 27 (1) (1985).
- **27**_Richard, E. Korf Finding optimal solutions to Rubik's cube using pattern databases AAAI/IAAI (1997).
- **28**_Richard, E. Finding optimal solutions to the twenty-four puzzle AAAI (1996).
- **29**_Steven, M. Optimal motion planning for multiple robots having independent goals Robot. Autom, 14 (6) (1998).
- **30**_Ryan L, Efficient and complete centralized multi-robot path planning Intelligent Robots and Systems (IROS) (2011).
- **31**_Lucia ,P, Decentralized cooperative policy for conflict resolution in multivehicle systems Robotics, 23 (6) (2007).
- **32**_ Christopher M, Complete and scalable multi-robot planning in tunnel environments Comput. Sci. Softw. Eng. (2006)
- **33**_Jussi, R, Planning with SAT , admissible heuristics and A* IJCAI (2011), pp. 2015-2020 View Record in Scopus .
- **34**_ John M, Non-optimal multi-agent pathfinding is solved (since 1984) Symposium on Combinatorial Search (SOCS) (2012) .
- **35**_Malcolm R. Exploiting subgraph structure in multi-robot path planning J. Artif. Intell. Res., 31 (2008).
- **36**_Malcolm R. Constraint-based multi-robot path planning International Conference on Robotics and Automation (ICRA) (2010), pp. 922-928 View Record in Scopus .

References

- **37**_ Kostas E. Multi-agent pathfinding with simultaneous execution of single-agent primitives Symposium on Combinatorial Search (SOCS) (2012)
- **38**_ Nathan R. Sturtevant Conflict-based search for optimal multi-agent path finding AAAI (2012)
- **39**_ Nathan R. Meta-agent conflict-based search for optimal multiagent path finding Symposium on Combinatorial Search (SOCS) (2012)
- **40**_Meir, G, The increasing cost tree search for optimal multi-agent pathfinding IJCAI (2011).
- **41**_Meir, G, Pruning techniques for the increasing cost tree search for optimal multi-agent pathfinding Symposium on Combinatorial Search (SOCS) (2011) .
- **42**. Aissani. D .Modélisation, Note de cours.Université A/Mira, Bejaia, 2008
- **43.** Benyamina, A. Faculté des Sciences Département d'Informatique Spécialité Informatique thèse soutenus le 08 avril 2013