REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
Université Dr. YHIA FARES de MEDEA
Faculté Des Sciences Et De La Technologie
Département du Génie Électrique
et de l'Informatique

Mémoire de Fin d'Études

Présenté pour l'obtention du Diplôme de Master en Génie Électrique

Option : Systèmes Intelligents et Communicants

Thème

Conception et implémentation d'une méta-heuristique à base d'agent pour le problème de voyageur de commerce

Préparé et Présenté par:

⇨ *driss yamina*
⇨ *otmani yasmina*

Proposé et Dirigé par :

⇨ *Mr. KHELDOUN AHMED*

Année Universitaire 2011-2012

Remerciements

Tous d`abord nous remercions **الله عز وجل** *pour nous avoir guidé vers le bon chemin de la lumière et de savoir, pour nous avoir donné du courage et de la volonté afin de pouvoir réaliser ce mémoire.*

Nous exprimons nos sincères remerciements à nos parents pour tout ce qu`ils ont pu nous apporter.

Nous tenons à remercier les membres du jury pour avoir eu l'obligeance d'accepter et d`apprécier notre travail.

Nous tenons à remercier notre promoteur ***M. Kheldoun Ahmed*** *pour son aide, sa patience, sa disponibilité, et sa compréhensibilité.*

A tous les enseignants de la faculté des sciences et de la technologie et surtout les enseignants du département du génie électrique et de l'informatique.

Nous remercions de profond de cœur : ***M.Bouhada Mounir, M.oueld zmirli Mohamed, M.Mazari Redha.***

Enfin, nous remercions, de tout cœur, tous ceux qui ont contribué de prés ou de loin à la réalisation de ce travail

Dédicace

Je dédié ce modeste projet du master tout d'abord :

*A celle qui son image toujours dans ma pensée, à celle qui sacrifie pour me voir une femme réussie dans sa vie, dans le moment ou tous les yeux dorment, elle ne peut pas dormir. À celle qui possède un cœur tolérant, à celle qui fait couler des larmes de joie « **Ma mère** » que j'adore.*

« Hafidaha Allah w Raàha »

*A notre star qui brille et à celui qui sourit toujours, et sacrifie dans sa jeunesse et sa vieillesse, à celui qui travaille tout le temps pour me donner la force et le courage « **Mon père** » que j'adore toujours que le dieu prend soi de lui. « **Adamaho Allah Ly** »*

A tous les bougiés mes chères frères :

*A mon deuxième père, le noyau de notre famille **Mohamed** et sa femme **Nassira** et ses enfants **Mahmoud Alghazali**, **Kadour**, **Lokman**, **Hadjer.***

*A mon frère **kharchi** et sa femme **Hada** et leurs enfants **Sara**,**Younes et Ahmed Yacine**.*

*A mon frère **Abdelkader** et sa femme **Assia** et leurs enfants **Ayoub** et **Zakaria.***

*A mon frère **Slimane** et sa fiancée **Fathia.** A notre héro de famille **Ahmed**.*

*A mon fiancé **Abdelmadjid** et à toute sa famille.*

A mes chères sœurs :

*A ma sœur qui mon a quitté **Fatima Azahraa**« **Allah Yarhamha** », à son marie et leurs enfants **Hanane**, **Hamza**, **Chayma** et **Aymèn**.*

*A ma sœur **Massouda** et son marie et leurs enfants **Naima**, **Bilal, Oussama**, **Zahia**.*

*A ma chère sœur **Khadija** et son marie et leurs enfants **Sohaib**, **Yousef**.*

*A ma sœur **Nawal** et son marie et leur bébé **AbdAlhadi**.*

*A mon binôme **Yasmina** et à toute sa famille.*

*A mes préfères amies : **Aicha, Fayza, Hizia, Razika, Nadia.***

*A mes copines de chambre : **Nora, Noura, Naima.***

*A toute la famille **DRISS** et **OUABEL**.*

A tous mes amis que j'aime et qui m'aiment.

AMINA

.



Dédicace



Ce travail est dédié :

A mes très chers ***parents*** *pour leur soutien durant toute ma carrière,*

Pour leur bienveillance, leurs efforts constants dans mes études,

Pour leur encouragement,

A toute la famille,

A mes frères ***Badeldinne, Bilal et Salime***

A mes sœurs ***Hadjira, Zoulika et Wahiba***

A mon binôme ***Amina*** *et à toute sa famille,*

A tous mes amis que j'aime et qui m'aiment.



YASMINA

Table des matières

Liste des figures vi
Liste des tableaux vi
Glossaire 7
Introduction générale .....**Error! Bookmark not defined.**

Chapitre I: Généralités sur les Systèmes Multi-Agent

I. Introduction .....**Error! Bookmark not defined.**
II. Concept d'agent .....**Error! Bookmark not defined.**
II.1. Définition d'agent .....**Error! Bookmark not defined.**
II.2. Les caractéristiques d'un agent.....**Error! Bookmark not defined.**
II.2.1. Autonomie .....**Error! Bookmark not defined.**
II.2.2. Flexibilité 4
II.2.3. Situation dans un environnement .....**Error! Bookmark not defined.**
II.2.4. Perception .....**Error! Bookmark not defined.**
II.2.5. Action .....**Error! Bookmark not defined.**
II.2.6. Communication 6
II.2.7. Rationalité .....**Error! Bookmark not defined.**
II.2.8. Mobilité .....**Error! Bookmark not defined.**
II.2.9. Véracité.....**Error! Bookmark not defined.**
II.2.10. Bienveillance.....**Error! Bookmark not defined.**
II.2.11. Adaptation.....**Error! Bookmark not defined.**
II.2.12. Sécurité .....**Error! Bookmark not defined.**
II.2.13. Apprentissage.....**Error! Bookmark not defined.**
II.3. Typologie des agents.....**Error! Bookmark not defined.**
II.3.1. Selon la nature.....**Error! Bookmark not defined.**
II.3.2. Selon l'utilisation .....**Error! Bookmark not defined.**
III. Système Multi-Agents.....**Error! Bookmark not defined.**
III.1. Définition.....**Error! Bookmark not defined.**
III.2. Diversité des Systèmes Multi Agents .....**Error! Bookmark not defined.**
III.3. Les différents types des SMA.....**Error! Bookmark not defined.**
III.4. Les caractéristiques d'un SMA.....**Error! Bookmark not defined.**
III.5. Communication entre les agents dans les SMA .**Error! Bookmark not defined.**
III.5.1. Communication par partage d'information **Error! Bookmark not defined.**
III.5.2. Communication par envoi de massages 11
III.6. Langages de communication .....**Error! Bookmark not defined.**
III.7. Champs d'application des systèmes multi-agents .....**Error! Bookmark not defined.**
III.8. Les avantages de SMA.....**Error! Bookmark not defined.**
IV. Conclusion.....**Error! Bookmark not defined.**

Chapitre II: Colonie de fourmis

I. Introduction .....**Error! Bookmark not defined.**
II. Historique.....**Error! Bookmark not defined.**
III. Principe de l'algorithme de colonie de fourmis .Error! Bookmark not defined.

IV. Similarités et différences avec les fourmis réelles .....Error! Bookmark not defined.
IV.1. Points communs .....**Error! Bookmark not defined.**
IV.2. Différences .....**Error! Bookmark not defined.**
V. Applications .....Error! Bookmark not defined.
V.1. Pont binaire de Deneubourg .....**Error! Bookmark not defined.**
V.1.1. Expérience du double pont binaire .....**Error! Bookmark not defined.**
VI. Effet de la coupure d'une piste de phéromone...Error! Bookmark not defined.
VII. Colonies de fourmis et résolution de problèmes Error! Bookmark not defined.
VIII. L'algorithme selon PVC .....Error! Bookmark not defined.
VIII.1. Colonie de fourmis .....**Error! Bookmark not defined.**
IX. Conclusion .....Error! Bookmark not defined.

Chapitre III: Conception d'une méta-heuristique à base d'agent pour le PVC

I. Introduction .....Error! Bookmark not defined.
II. Représentation du problème de PVC.....Error! Bookmark not defined.
II.1. Enoncé .....**Error! Bookmark not defined.**
II.2. Enoncé formel.....**Error! Bookmark not defined.**
III. Adaptation de la Colonie de fourmis à base d'agent au problème de PVC Error! Bookmark not defined.
III.1. Architecture générique du système .....**Error! Bookmark not defined.**
III.1.1. AGENT INIT .....**Error! Bookmark not defined.**
III.1.2. Les AGENTS FOURMIS .....**Error! Bookmark not defined.**
III.1.3. Critères d'arrêt .....**Error! Bookmark not defined.**
III.2. Communication inter-agents .....**Error! Bookmark not defined.**
III.3. Les règles des agents fourmis .....**Error! Bookmark not defined.**
IV. Conclusion .....Error! Bookmark not defined.

Chapitre IV: Expérimentations

I. Environnement expérimental .....Error! Bookmark not defined.
I.1. Outils utilisées .....**Error! Bookmark not defined.**
II. Processus de simulation .....Error! Bookmark not defined.
III. Description des tests.....Error! Bookmark not defined.
III.1. Description de problème .....**Error! Bookmark not defined.**
III.2. Description et résultats des tests effectués .....**Error! Bookmark not defined.**
IV. Conclusion .....Error! Bookmark not defined.

Conclusion générale.....Error! Bookmark not defined.
Bibliographie .....Error! Bookmark not defined.
Annexe .....Error! Bookmark not defined.

Liste des figures

Figure I-1: Capacités de perception et d'action de l'agent dans un environnement. 4
Figure I-2: Structure d'un système à base de tableau noir.....10
Figure I-3 : Structure d'un système à envoi de message. 11

Figure II-1: Pont binaire de Deneubourg.....18
Figure II-2: Expérience du double pont binaire. 19
Figure II-3: Effet de la coupure d'une piste de phéromone.....20

Figure III-1 : Graphe montre la représentation du problème de PVC de 6 villes. 25
Figure III-2: Architecture globale du système. 27
Figure III-3: Architecture de l'agent INIT. 27
Figure III-4: Architecture de l'agent fourmis. 29
Figure III-5: Communication de type directe entre Agent INIT et Agent fourmis 29
Figure III-6: Communication de type indirecte entre Agent fourmis et Agent fourmis. ..30

Figure IV-1: Schéma représente le processus de simulation. 33
Figure IV-2: Variation de distance en fonction du nombre d'agent 34
Figure IV-3: Variation de distance en fonction du nombre d'itération.....36
Figure IV-4 : Variation de distance en fonction de la vitesse d'évaporation 37

Liste des tableaux

Tableau III-1: Matrice de poids représente le Problème de PVC de 6 villes.....26

Tableau IV-1 : Les valeurs des paramètres de test 1.....34
Tableau IV-2: Les résulats de test 1. 34
Tableau IV-3: Les valeurs des paramètres de test 2 .....**Error! Bookmark not defined.**5
Tableau IV-4: Les résulats de test 2.....**Error! Bookmark not defined.**5
Tableau IV-5: Les valeurs des paramètres de test 3 .....**Error! Bookmark not defined.**6
Tableau IV-6: Les résulats de test 3. .....**Error! Bookmark not defined.**6

Glossaire

SMA : Système Multi Agents.

IAD: Intelligence artificielle distribuée.

KQML: Knowledge Query and Manipulation.

FIPA: Foundation of Intelligent physical Agents.

FIPA_ACL : FIPA Agent Communication Language.

OCF ou ACO : Optimisation par Colonie de Fourmis ou Ant Colony Optimization.

PVC : Problème du voyageur de commerce.

AS: Ant System.

ACA: Ant Colony Algorithm.

Résumé

Le problème de voyageur de commerce (PVC) est l'un des problèmes d'optimisation combinatoire qui fait partie de la classe des problèmes NP-complet. Cette classe de problèmes est caractérisée par le fait qu'il n'y a pas d'algorithme polynomial pour trouver la solution optimale d'un problème. Plusieurs travaux sont initiés dans le but de trouver une solution satisfaisable par l'utilisation des heuristiques, des approximations,…

Les systèmes multi-agents forment un paradigme pour la conception des systèmes complexes et proposent des outils pour les analyser, les concevoir et les implanter. L'approche agent considère les systèmes comme des sociétés composées d'entités autonomes et indépendantes, appelées agents, qui interagissent en vue de résoudre un problème ou de réaliser collectivement une tâche. Cette approche semble intéressante dans le cadre des méta-heuristiques, tout d'abord pour aborder la distribution et l'adaptation.

L'objectif de ce projet est de proposer et de développer une nouvelle méthode combinant les systèmes multi-agents et la méta-heuristique colonie de fourmis pour résoudre le problème PVC.

Abstract

Traveling Salesman Problem (TSP) is one of the problems of combinative optimization which belongs to the class of the problems NP-Hard. This class of problems is characterized by the fact that there is no polynomial algorithm to find the optimal solution of a problem. Several work is initiated with an aim of finding a satisfaisable solution by the use of heuristic, of the approximations…

The multi-agents systems form a paradigm for the complex systems design and propose tools to analyze, conceive and establish them. The agent approach considers the systems as companies made up of autonomous and independent entities, called agents, which interact in order to solve a problem or to carry out a task collectively. This approach seems interesting within the framework of méta-heuristic, to tackle the distribution and the adaptation.

The aim of this project is to propose and develop a new method combining the multi-agents systems and the méta-heuristic Ant Colony to solve TSP problem.

ملخص

مشكلة البائع المتجول هي واحدة من مشاكل التحسين التوافقي التي تنتمي إلى فئة المشاكل المعقدة.و الواقع أنه لا توجد خوارزمية متعددة الحدود لإيجاد الحل الامثل لهذه الفئة .علما أنها أجريت عدة دراسات لإيجاد هذا الحل عن طريق استخدام الطرق التقريبية.

تعتبر الأنظمة المتعددة الوكلاء نهجا واعدا لتحليل، تمثيل و إنجاز الأنظمة المعقدة ذلك أنها تمثل مجتمعا يتكون من كيانات مستقلة تملك القدرة على حل مشاكلها دون أي تدخل خارجي تدعى الوكيل، التي تتفاعل من أجل حل مشكلة أو تحقيق مهمة جماعية.

و الهدف من هذا العمل هو اقتراح و تطوير طريقة جديدة تجمع بين الأنظمة المتعددة الوكلاء و مستعمرات النمل لحل مشكلة البائع المتجول.

Introduction générale

La nature a souvent été une source d'inspiration inégalée pour tous les domaines scientifiques. Les chercheurs ont toujours porté un intérêt grandissant à l'étude et la compréhension des phénomènes qui s'y produisent. En effet, pour résoudre tout type de problèmes complexes, les chercheurs se sont souvent tournés vers la nature pour essayer de comprendre comment elle fonctionne et comment des problèmes très complexes sont résolus. C'est le cas notamment des systèmes collectifs tels que les colonies d'insectes, capables d'accomplir des tâches difficiles, dans des environnements dynamiques et variés sans pilotage ou contrôle externe. L'analogie avec la nature a souvent constitué les prémisses de solutions à des problèmes scientifiques de grande envergure. Le raisonnement par analogie est de toute évidence un merveilleux outil de travail.

L'exemple le plus manifeste d'analogie dans le domaine de l'optimisation est celui des colonies d'insectes, en particulier les colonies de fourmis, d'abeilles. Ces insectes, malgré une cognition individuelle très limitée, possèdent des capacités étonnantes pour faire face collectivement à des problèmes complexes, tels que ceux posés par la recherche de nourriture, la reproduction, la construction de nids. Ces capacités nous mettent face au constat d'une performance collective au niveau global inexistante au niveau individuel.
Cette problématique dépasse le cadre des sociétés d'insectes. D'autres exemples biologiques issus en général des sociétés animales illustrent la notion d'« intelligence » collective. Une partie de la réponse à cette problématique se trouve dans l'idée de l'auto-organisation et de l'émergence. L'un des domaines scientifiques qui n'a pas échappé à l'attrait des systèmes naturels et notamment biologiques est l'informatique et précisément l'Intelligence Artificielle (IA). Les informaticiens ont repris les principes d'auto-organisation et d'émergence qui orchestrent les sociétés d'insectes pour comprendre et concevoir des systèmes complexes. L'IA et particulièrement les Systèmes Multi-Agents (SMAs) ont été à l'origine de nombreuses recherches inspirées de la biologie [Bon94 , Dro93]. Les SMAs ont constitué une révolution dans la manière de penser la résolution des problèmes. Ils postulent que l'intelligence nécessaire à la résolution d'un problème n'est pas une propriété cognitive d'un seul agent, mais d'un ensemble d'agents simples en interaction. En effet, la solution émerge des interactions que les agents entretiennent entre eux et avec leur environnement.

Citons dans ce cadre les travaux

de Marvin Minsky [Min86] qui dans son ouvrage « The society of mind2 » explique comment l'intelligence humaine peut surgir à partir d'un grand nombre de petits processus en interaction qu'il appelle agents. Chaque agent ne peut, à lui seul, effectuer que quelques tâches simples ne demandant ni intelligence ni réflexion. Pourtant, le regroupement de ces agents peut aboutir à la véritable intelligence. Ce postulat a soulevé une question fondamentale : faut-il concevoir des agents comme des entités déjà intelligentes ? C'est-à-dire capables de résoudre certains problèmes par eux-mêmes, ou bien faut-il les assimiler à des êtres très simples réagissant directement aux modifications de leur environnement.

Objectifs

L'objectif de ce mémoire consiste à appliquer une méta-heuristique colonies de fourmis à base d'agent pour résoudre le problème de voyageur de commerce.

Chapitre I

Généralités sur les Systèmes Multi-Agents

I. Introduction

Les systèmes multi-agents ***(SMA)*** se situent dans un domaine ***d'intelligence artificielle distribuée(IAD).*** Ces systèmes sont particulièrement adaptés pour proposer des solutions à des problèmes complexes.

L'idée fondamentale est que plutôt de tenter de résoudre un problème par une seule entité intelligente, il serait plus intéressent de faire coopérer d'autres entités plus simples, afin de résoudre le problème par une équipe. Chaque entité a une perception et des compétences limitées qui ne leur permettent pas de résoudre le problème individuellement. Ces entités utilisent la coopération et la coordination entre elles pour résoudre le problème.

Par conséquent, les ***SMA*** constituent actuellement une approche prometteuse pour l'analyse, la conception et la réalisation des systèmes complexes. Ils connaissent un intérêt croissant, dû à l'ampleur des problèmes abordés et aux résultats encourageants obtenus (gestion de fabrication [Par87], gestion de trafic aérien [Ngu04], e-commerce [Als09], systèmes médicaux [Rot98],…etc.)

Dans ce chapitre, nous allons décrire les concepts qui découlent du domaine de recherche que constituent les SMA. L'intérêt premier est porté d'abord sur le concept de "***l'Agent***" qui constitue la brique fondamentale du domaine. Ensuite nous précisons la notion de ***" systèmes multi-agents"*** en définissant les concepts clés tels que l'interaction, l'organisation et la communication.

II. Concept d'agent

II.1. Définition d'agent

Il existe dans la littérature plusieurs définitions du concept "***Agent***" [Jen98a] [Rus95][Bri01] [Zam03]. Elles ne sont pas contradictoires les unes avec les autres, elles diffèrent selon ***le type d'application*** pour laquelle est conçu l'agent.

Nous citons, les définitions les plus courantes :

- *« Un agent est une entité autonome, physique ou virtuelle, qui est capable d'agir sur son environnement sur lequel il dispose d'une représentation partielle, et qui peut communiquer avec d'autres agents, et éventuellement se reproduire et dont son comportement vise à la satisfaction de ses besoins et de ses objectifs à partir de tous les autres éléments (Perceptions, représentations, actions, communications et ressources) dont il dispose »* Selon Ferber [Fer95].
- *« Un agent est une entité informatique située dans un environnement, qui est capable d'agir de manière autonome et flexible de façon à remplir ses objectifs, c'est à dire ceux qui lui ont été assignés lors de sa conception »* Selon Jennings & Wooldridge [Jen98a] [Jen98b].

II.2. Les caractéristiques d'un agent

Un agent doit être ***autonome***, ***flexible***, *situé dans un environnement*, il ***agit***, il ***perçoit*** et il ***communique***. Nous essayons maintenant de clarifier ces notions.

II.2.1. Autonomie

Il est difficile de définir la notion d'autonomie d'une manière précise, [Jen98a] lui accorde le sens suivant : que le système peut agir sans une intervention directe d'humains ou d'autres agents, et que le système possède le contrôle total sur ses propres actions et sur son état interne. On pourrait dire ainsi que le moteur d'un agent, c'est lui-même. Il a toujours la possibilité de répondre par l'affirmative ou le refus à des requêtes provenant des autres agents. L'agent donc n'est pas guidé par l'extérieur mais par ses tendances [Ferb95].

II.2.2. Flexibilité

L'agent doit également être flexible, la flexibilité peut être vue comme une forme de l'intelligence. Etre flexible signifie que l'agent est [jen98a]:

- ✓ ***Réactif*** : l'agent doit être capable de percevoir son l'environnement et de répondre à temps aux changements qui peuvent affecter cet environnement.

- ✓ ***Proactif*** : l'agent n'agit pas seulement d'une manière réactive (en fonction de son environnement) mais il doit avoir un comportement orienté objectifs et que l'agent peut prendre des initiatives.
- ✓ ***Social*** : l'agent est capable d'interagir avec les autres agents intelligents et humains pour qu'il puisse atteindre ses propres objectifs et aider les autres dans leurs activités.

II.2.3. Situation dans un environnement

L'agent est implanté dans l'environnement, sur lequel il agit à partir des entrées sensorielles qu'il reçoit de ce même environnement. Les actions de l'agent modifient son environnement [Jen98b]. Contrairement aux systèmes experts, ces derniers n'interagissent pas directement avec l'environnement, ils ne le perçoivent pas, mais ils reçoivent les informations via un utilisateur qui joue le rôle d'intermédiaire entre le système et l'environnement.

II.2.4. Perception

La perception est un moyen de recevoir de l'information d'un environnement qui est souvent complexe et évolutif. L'agent doit percevoir directement son environnement par ses propres capteurs.

II.2.5. Action

La définition du mot action dépend souvent du domaine de système conçu [Dri01]. L'agent est capables d'agir, et non pas seulement de raisonner comme dans les systèmes d'IA classique. L'action, repose sur le fait que les agents accomplissent des actions qui vont modifier l'environnement des agents et donc leurs prises de décision futures [Fer95].

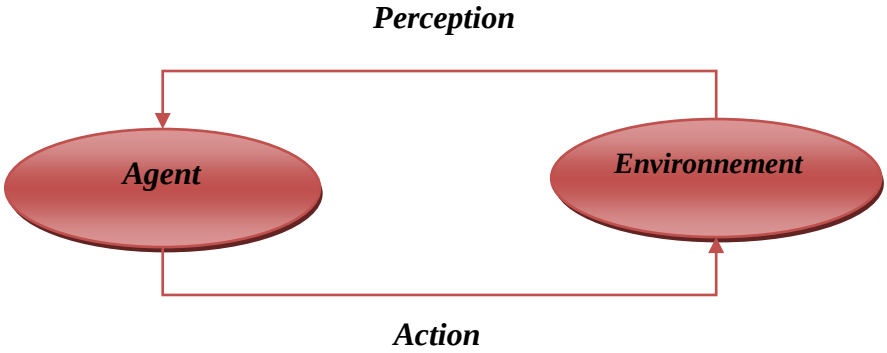


Fig. I.1. Capacités de perception et d'action de l'agent dans un environnement

II.2.6. Communication

Un agent fait un acte délibéré de transfert d'informations vers un ou plusieurs autres agents. La communication entre agents est l'un des modes principaux d'interaction existant entre les agents. L'interaction ne se résume pas à la communication, elle a d'autres formes : la négociation, la coopération, la coordination.

II.2.7. Rationalité

Un agent rationnel est un agent qui agit d'une manière lui permettant d'obtenir le plus de succès possible dans la réalisation des tâches qu'on lui a assignées. Il dispose de critères d'évaluation de ses actions. Il suit le principe suivant (dit de rationalité) : "Si un agent sait qu'une de ses actions lui permet d'atteindre un de ses buts, il la sélectionne" [New82].

II.2.8. Mobilité

Capacité de déplacement dans un environnement à un autre environnement.

II.2.9. Véracité

L'agent ne doit pas communiquer de fausses informations.

II.2.10. Bienveillance

Les agents ne doivent pas avoir des objectifs contradictoires, et chaque agent essaye toujours de faire ce qui lui a été demandé [Woo95].

II.2.11. Adaptation

Un agent adaptatif est capable d'apprendre en fonction de son expérience passée et de son évolution [Fra97].

II.2.12. Sécurité

Un agent doit se protéger des attaques extérieures par rapport aux autres agents, à ses environnements d'exécution et pendant son déplacement sur le réseau pour les agents mobiles [Esp00].

II.2.13. Apprentissage

Un agent est capable d'apprendre et d'évoluer en fonction de cet apprentissage. Un agent est aussi capable de changer de comportement (en fonction des expériences passées).

II.3. Typologie des agents

II.3.1. Selon la nature

On distingue deux catégories selon leurs capacités de résolution de problème : agent réactif et agent cognitif. D'autres types d'agents, qualifiés d'hybrides, utilisant ces

deux types de comportement et bénéficient des avantages de chaque type, sont ensuite apparus.

- **Agent réactif :** Les agents réactifs sont de plus bas niveau, ils ne disposent que d'un protocole et d'un langage de communication réduit, ils n'ont pas une représentation globale de leurs environnement et ne sont pas capables de tenir compte de leurs actions passées, leurs capacités répondent uniquement à la loi stimulus/action.
- **Agent cognitif :** Contrairement aux agents réactifs, les agents cognitifs sont dotés de capacités de raisonnement et de planification ainsi que de communication, et ils possèdent une représentation explicite de leur environnement et des autres agents.
- **Agent hybride :** Les agents hybrides sont conçus pour combiner des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de l'environnement. L'agent hybride est composé de plusieurs couches, rangées selon une hiérarchie. Au plus bas niveau de l'architecture, on retrouve une couche purement réactive, la couche intermédiaire fait abstraction des données brutes et travaille avec une vision des connaissances de l'environnement et la couche supérieure se charge des aspects communication, coopération et négociation.

II.3.2. Selon l'utilisation

Certains chercheurs ont proposé une classification pour les agents en fonction de leurs caractéristiques dominantes et de leurs domaines d'application. Ils ont classé les agents selon leurs degrés d'autonomie, de coopération et d'adaptation :

- **Agents pour le E-commerce :** La montée de l'internet a bien entendu crée de nouvelles nécessités. Les agents issus de cette tendance permettent la promotion, la vente ainsi que l'achat de produits et de services par l'entremise des réseaux informatiques.
  - ✓ **Agent stationnaire :** il s'agit du cas où l'agent s'exécute toujours dans la même machine.
  - ✓ **Agent mobile :** est une entité autonome contenant du code et des données qui se déplace d'une machine à une autre sur le réseau offrant des services pour réaliser une tâche complexe. Ce code représente la suite d'instructions définissant le comportement statique de l'agent, et

les données ou ressources utilisées par l'agent dans un état d'exécutions, Cet état est composé d'une pile d'exécution et des registres.

- **Agents pour la recherche d'informations :** Ils procèdent une analyse des informations utiles sur une collection de données afin de découvrir de nouvelles connaissances.
- **Agents collaboratifs :** Ce sont des agents cognitifs non apprenants, ils sont capables de communiquer et de coopérer avec d'autres agents. Ils coordonnent entre eux par planification ou négociation pour une résolution distribuée de problèmes.

  Les agents collaboratifs sont utilisés dans les domaines qui nécessitent une décentralisation comme la maintenance de réseaux.
- **Agents d'interfaces :** Leur fonction consiste à capturer les actions de l'utilisateur, interagir avec l'utilisateur, recevoir ses spécifications et renvoyer les résultats. Les agents d'interfaces acquièrent, modélisent et utilisent les préférences de l'utilisateur.

III.Système Multi-Agents

Un système multi-agents est une société d'agents où les interactions entre les agents et leur environnement mènent à un comportement adéquat à l'atteinte d'un but global.

L'approche SMA consiste à décomposer les problèmes en agents autonomes. Ces agents sont organisés en sociétés, ils peuvent interagir à haut niveau. Cette organisation sert à décrire l'activité du système. L'identification du rôle et des capacités de chaque agent y est donc essentielle.

III.1. Définition

Une définition plus détaillée d'un SMA a été proposée par Ferber [Fer95] :

«On appelle système multi-agents un système composé des éléments suivants :

- Un environnement E, un espace disposant généralement d'une métrique.
- Un ensemble d'objets O. Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
- Un ensemble A d'agents, qui sont des objets particuliers, lesquels représentent les entités actives du système.

- Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.
- Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers ».

III.2. Diversité des Systèmes Multi Agents

Voici quelques exemples pour mieux comprendre ce qu'est un SMA et toute la diversité qui peut être engendrée par de tels systèmes :

- ***Une fourmilière***

  L'environnement : la fourmilière et l'espace alentour occupé par les fourmis.

  Les objets : les végétaux, la terre, l'eau,... toute chose inerte.

  Les agents : les fourmis, les araignées... toute chose vivante.

- ***Un écosystème***

Cet écosystème peut-être : une prairie, un bois, un groupe de proies (cervidés), un groupe de prédateurs (loups. Le principe est le même que pour la fourmilière. On peut, avec un tel système étudier l'écosystème, déterminer si des espèces vont disparaître ou de quelles façons vont évoluer les différentes populations.

- ***Un groupe de robots partant à la recherche de minerai***

Chaque robot pouvant soit chercher du minerai sur un terrain encore inconnu et/ou extraire du minerai et/ou le transporter.

III.3. Les différents types des SMA

Ouverts : les agents y entrent et en sortent librement (ex: un café).

Fermés : l'ensemble d'agents reste le même (ex: un match de football).

Homogènes : tous les agents sont construits sur le même modèle (ex: une colonie de fourmis).

Hétérogènes : des agents de modèles différents et de granularité différentes (ex: l'organisation hospitalière).

III.4. Les caractéristiques d'un SMA

D'après Sycara [Syc98], un SMA possède les caractéristiques suivantes :

- Ensemble d'agents agissant et travaillant indépendamment les uns des autres.

- Chaque agent est une partie du système et travaille dans le but d'accomplir ses tâches.
- Chaque agent est capable de communiquer et d'interagir avec d'autres agents.
- Un agent coopère avec les autres agents lorsque nécessaire.
- Un agent est capable de coordonner ses activités avec les autres agents pour accéder à des ressources partagées dont il a besoin pour réaliser ses buts.
- Chaque agent a une vue partielle du SMA.
- Chaque agent a des informations ou des capacités de résolution des problèmes limités.
- Les données sont décentralisées.
- Il n'y a aucun contrôle global du système multi-agents.

III.5. Communication entre les agents dans les SMA

Les agents communiquent entre eux à l'aide d'un langage et ils utilisent des protocoles de communication. On distingue essentiellement deux types :

III.5.1. Communication par partage d'information

La communication par partage d'information se fait lorsqu'un ensemble d'agents partage la même structure de donnée, et travaillent en mode coopératif afin de résoudre un problème global de nature centralisée selon tous les agents. Cette structure de donnée contient initialement les données du problème et est enrichie au cours de la résolution jusqu'à l'obtention de la solution.

On peut représenter ce type par la figure suivante :

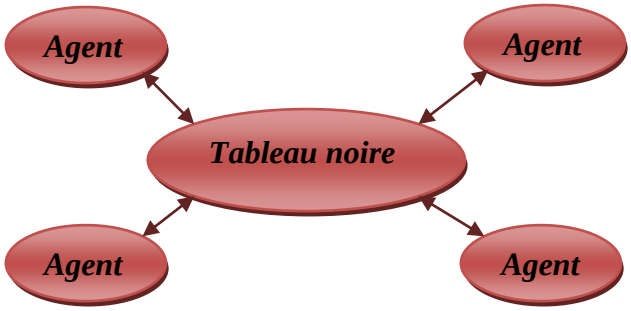


Fig. I.2 : Structure d'un système à base de tableau noir

Les agents déposent et lisent une information sur une zone de données commune.

III.5.2. Communication par envoi de massages

La communication par messages est un outil pour coordonner les actions d'agents. Elle est caractérisée par la distribution totale à la fois des connaissances, des résultats partiels et des méthodes utilisées pour aboutir à un résultat dans un SMA. Pour que l'agent communiqué, il doit pouvoir envoyer et recevoir des messages.

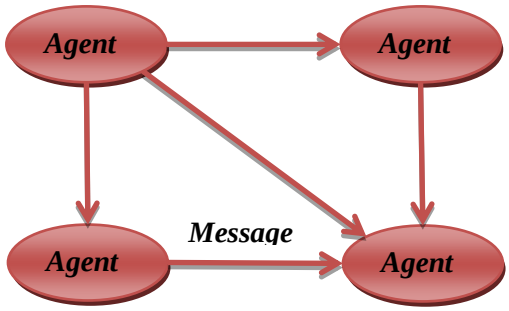


Fig. I.3 : Structure *d'un système à envoi de messages.*

La communication peut se faire point à point ou par diffusion. Dans ce contexte, l'agent peut avoir les fonctions de maître, esclave. Et les rôles passifs, actif (les deux).Un agent Pour tenir un rôle passif, doit être capable de répondre, i.e. accepter une question externe et envoyer la réponse à la source (assertion). Et Pour tenir un rôle actif, un agent doit pouvoir poser une question et faire des assertions. Il peut ainsi contrôler un autre agent par le biais de question/réponses. Dans un fonctionnement de pair, l'agent a les deux rôles.

III.6. Langages de communication

L'intérêt des langages de communication est de faciliter l'échange et l'interprétation des messages et l'interopérabilité entre les agents. Ces langages se focalisent essentiellement sur la manière de décrire exhaustivement des actes de communication d'un point de vue syntaxique et sémantique. Deux langages de communication entre agent, ***FIPA-ACL***(Foundation of Intelligent Physical Agent - Agent Communication Language) et ***KQML***(Knowledge Query and Manipulation language) [Fin94] [Lab97], ont émergé des efforts de standardisation de la communauté des SMA, portés par des besoins d'interopérabilité logicielle.

Le premier langage qui a été introduit est KQML. A l'origine, KQML a été développé pour échanger des informations et des connaissances entre des systèmes à base de connaissances. Il a été ensuite repris dans [Fin94] pour décrire les messages échangés

entre les agents. FIPA-ACL, est proposé dans le cadre d'un travail de standardisation mené au sein l'organisation FIPA12 (Foundation of Intelligent Physical Agents). Ce langage est fondé sur vingt et un actes communicatifs, exprimés par des performatifs, qui peuvent être groupés selon leurs fonctionnalités de la façon suivante :

- Passage d'information : Inform, Inform-if, Inform-ref, Confirm, Disconfirm,
- Réquisition d'information : Query-if, Query-ref, Subscribe,
- Négociation : Accept-proposal, Cfp, Propose, Reject-proposal,
- Distribution de tâches (ou exécution d'une action) : Request, Request-when,Requestwhenever, Agree, Cancel, Refuse,
- Manipulation des erreurs : Failure, Not-understood.

III.7. Champs d'application des systèmes multi-agents

Les SMA ont des applications diverses qui ont évolué avec le développement des outils technologiques, et surtout de l'Internet [Cha96]:

- **Application industrielles** : gestion de la production manufacturière, Contrôle et commande des processus (accélérateur de particules), télécommunication (gestion et contrôle des réseaux, transmission, etc..) et systèmes de transport (surveillance des véhicules automatisés (DVMT : « Distributed Vehicule Monitoring Task »), job shop flexible, Systèmes industriels flexible, etc.
- **Application commerciales *:*** gestion de l'information (Internet, filtrage, collecte), commerce électronique (agence de voyage), gestion des affaires, etc.
- **Loisirs *:*** Jeux, théâtre et cinéma interactifs.
- **Application médicales** : orientation des patients, gestion des soins.

III.8. Les avantages de SMA

L'utilisation des Systèmes Multi-Agents (SMA) présente une série d'avantages [Cha01] :

- Les SMA possèdent les avantages traditionnels de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse avec le parallélisme (Plusieurs agents qui s'exécute en même temps), et la fiabilité due à la Redondance (le même problème peut être assigné aux plusieurs agents).
- Les SMA héritent aussi des bénéfices envisageables de l'IA comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité.

- Les SMA ont l'avantage de faire intervenir des schémas d'interaction sophistiqués tels que la coopération (travailler ensemble à la résolution d'un but commun), la coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées, ou que les interactions bénéfiques soient exploitées) et négociation (parvenir à un accord acceptable pour toutes les parties concernées) .
- Les SMA peuvent répondre de l'échec individuel d'un des éléments, sans dégrader le système dans son ensemble [Mar02].

IV. Conclusion

La progression des paradigmes de programmation a conduit aux agents et systèmes multi-agents, cette nouvelle percée ait permis de répondre à certains besoins qui était jusqu'à une date récente accessible que par des humains. Les applications des agents sont multiples. On peut distinguer rapidement l'assistance à l'utilisateur, la recherche d'informations, activités commerciales, loisir et jeux …etc. Ce domaine demeure aujourd'hui un domaine rempli de défis à surmonter autrement dit un domaine très ouvert pour la recherche.

Dans ce chapitre, on a représenté les concepts des agents, et des systèmes multi-agents, défini leurs caractéristiques, et les notions fondamentales de ce domaine. Il est intéressant de noter, que l'ensemble des concepts de l'approche multi-agents font de celle-ci une approche très riche et très puissante en termes de modélisation.

Chapitre II

Colonie de fourmis

I. Introduction

L'inspiration biologique a amené de nombreux chercheurs à emprunter les principes mis en exergue chez les insectes sociaux tels que les fourmis, les abeilles, pour en faire des algorithmes et des paradigmes d'ingénierie très efficaces pour la résolution de certains problèmes.

En effet, vers la fin des années 80, des informaticiens travaillant conjointement avec des éthologues se penchent sur le fonctionnement de ces sociétés d'insectes et sur la manière dont les individus d'une même colonie arrivent à régler leurs activités et favoriser la survie de l'espèce. Ils ont abouti à la conclusion que le comportement global de ces colonies résulte d'une multitude d'interactions locales résolument simples. Par analogie, une nouvelle forme de résolution collective est née à partir de cette conclusion.

L'exemple frappant des fourmis reste le plus utilisé dans la littérature pour illustrer l'utilisation des SMAs pour la résolution de problèmes d'optimisation combinatoires [Dor96].

Alors que toutes les fourmis sont sur un même pied d'égalité, les actions des agents (fourmis) se coordonnent de telle manière que la colonie se développe et survive. L'entité collective est en mesure de résoudre des problèmes complexes tels que la recherche de nourriture, la construction de nid, la reproduction et les soins donnés aux larves et aux œufs [Dro91].

Les méta-heuristiques sont des méthodes approchées qui traitent les problèmes d'optimisation difficile.

L'optimisation par colonie de fourmis (OCF) est une méta-heuristique relativement récente qui s'inspire de l'intelligence collective des fourmis. Initialement, elle a été conçue pour résoudre un problème discret classé NP-difficile: Le problème du voyageur de commerce (PVC). Plusieurs variantes de l'OCF ont été appliquées au PVC et ont abouti à des résultats plus au moins satisfaisants. Toujours est-il que le degré de difficulté est lié à la dimension de l'espace de recherche. La résolution du problème est donc limitée par les capacités des ressources matérielles disponibles (puissance de calcul, mémoire). Le parallélisme constitue alors une arme efficace pour lutter contre cette explosion Combinatoire.

II. Historique

Dans les sections qui viendront plus tard, nous présenterons la méta-heuristique OCF (optimisation par colonie de fourmis). Toutes ces idées abstraites sont inspirées des travaux de Deneubourg sur les fourmis.

Cette méta-heuristique est relativement récente. Elle a été introduite en 1991 par Colorni, Dorigo et Maniezzo pour résoudre le problème du Voyageur de commerce. Elle s'est popularisée, puis a été l'objet d'améliorations dès 1995 et a été appliquée avec succès à d'autres problèmes d'optimisation combinatoire dès 1994.

Nous allons tout d'abord exposer les différences et les points communs entre les fourmis virtuelles et les fourmis réelles [Dor03], avant d'exposer en termes plus abstraits la métaheuristique proprement dite. Ceci expliquera comment les fourmis virtuelles peuvent être exploitées pour résoudre un problème d'optimisation combinatoire.

III. Principe de l'algorithme de colonie de fourmis

Le premier algorithme s'inspirant du comportement des fourmis calcule un chemin entre leur nid et une source de nourriture. Initialement, les fourmis quittent le nid et explorent leur voisinage de manière aléatoire.

Lorsqu'une fourmi découvre par hasard une source de nourriture, elle informe ses congénères de sa découverte en déposant sur le sol une marque volatile, nommée phéromone, lors de son retour vers le nid. Cette phéromone va guider les autres fourmis vers la même source de nourriture.

À leur retour, ces dernières déposent également de la phéromone sur le sol, et ainsi de suite. Ceci renforcera le marquage de la piste qui mène du nid à la source de nourriture découverte. Le renforcement du marquage par les phéromones de la piste la plus fréquentée optimise la collecte de nourriture. À long terme, les fourmis exploiteront uniquement la source la plus proche, car la trace conduisant aux sources éloignées s'évaporera et deviendra moins prépondérante. Cet exemple montre que la colonie de fourmis converge vers une solution efficace alors que chaque fourmi n'a accès qu'à une information locale et qu'elle est incapable de résoudre seule le problème dans un délai raisonnable.

Les fourmis utilisent l'environnement comme support de communication elles échangent indirectement de l'information en déposant des phéromones décrivant l'état de leur « travail ».

L'information échangée a une portée locale, une fourmi à accès seulement à la phéromone située à sa position et son voisinage. Ce principe d'interaction porte le nom de « stigmergie », et se retrouve chez plusieurs animaux sociaux.

Cette idée originelle s'est depuis développée pour résoudre une classe plus large de problèmes et plusieurs algorithmes ont vu le jour, s'inspirant de divers aspects du comportement des fourmis.

IV. Similarités et différences entre les fourmis virtuelles et les fourmis réelles

Les fourmis virtuelles ont une double nature. D'une part, elles modélisent les comportements abstraits de fourmis réelles, et d'autre part, elles peuvent être enrichies par des capacités que ne possèdent pas les fourmis réelles, afin de les rendre plus efficaces que ces dernières. Nous allons maintenant synthétiser ces ressemblances et différences.

IV.1. Points communs

- **Colonie d'individus coopérants :** Comme pour les fourmis réelles, une colonie virtuelle est un ensemble d'entités non-synchronisés, qui se rassemblent ensemble pour trouver une "bonne" solution au problème considéré. Chaque groupe d'individus doit pouvoir trouver une solution même si elle est mauvaise.
- **Pistes de phéromones :** Ces entités communiquent par le mécanisme des pistes de phéromone. Cette forme de communication joue un grand rôle dans le comportement des fourmis : son rôle principal est de changer la manière dont l'environnement est perçu par les fourmis, en fonction de l'historique laissé par ces phéromones.
- **Évaporation des phéromones *:*** La méta-heuristique ACO comprend aussi la possibilité d'évaporation des phéromones. Ce mécanisme permet d'oublier lentement ce qui s'est passé avant. C'est ainsi qu'elle peut diriger sa recherche vers de nouvelles directions, sans être trop contrainte par ses anciennes décisions.

- **Recherche du plus petit chemin :** Les fourmis réelles et virtuelles partagent un but commun recherche du plus court chemin reliant un point de départ (le nid) à des sites de destination (la nourriture).
- **Déplacement locaux :** Les vraies fourmis ne sautent pas des cases, tout comme les fourmis virtuelles. Elles se contentent de se déplacer entre sites adjacents du terrain. Choix aléatoire lors des transitions. Lorsqu'elles sont sur un site, les fourmis réelles et virtuelles doivent décider sur quel site adjacent se déplacer. Cette prise de décision se fait au hasard et dépend de l'information locale déposée sur le site courant. Elle doit tenir compte des pistes de phéromones, mais aussi du contexte de départ (ce qui revient à prendre en considération les données du problème d'optimisation combinatoire pour une fourmi virtuelle).

IV.2. Différences

Les fourmis virtuelles possèdent certaines caractéristiques que ne possèdent pas les fourmis réelles :

- **Elles vivent dans un monde non-continu :** Leurs déplacements consistent en des transitions d'état.
- **Mémoire (état interne) de la fourmi :** Les fourmis réelles ont une mémoire très limitée. Tandis que nos fourmis virtuelles mémorisent l'historique de leurs actions. Elles peuvent aussi retenir des données supplémentaires sur leurs performances.
- **Nature des phéromones déposées :** Les fourmis réelles déposent une information physique sur la piste qu'elles parcourent, là où les fourmis virtuelles modifient des informations dans les variables d'états associées au problème. Ainsi, l'évaporation des phéromones est une simple décrémentation de la valeur des variables d'états à chaque itération.
- **Qualité de la solution :** Les fourmis virtuelles déposent une quantité de phéromone proportionnelle à la qualité de la solution qu'elles ont découverte.
- **Retard dans le dépôt de phéromone :** Les fourmis virtuelles peuvent mettre à jour les pistes de phéromones de façon non immédiate : souvent elles attendent d'avoir terminé la construction de leur solution. Ce choix dépend du problème considéré bien évidemment.

⌦ **Capacités supplémentaires :** Les fourmis virtuelles peuvent être pourvues de capacités artificielles afin d'améliorer les performances du système. Ces possibilités sont liées au problème et peuvent être :

- ***L'anticipation :*** la fourmi étudie les états suivants pour faire son choix et non seulement l'état local.
- ***le retour en arrière :*** une fourmi peut revenir à un état déjà parcouru car la décision qu'elle avait prise à cet état a été mauvaise.

V. Applications

V.1. Pont binaire de Deneubourg [Dré04]

L'expérience montre un nid d'une colonie de fourmis, qui est séparé d'une source de nourriture par un pont à deux voies de même longueur. On laisse évoluer les fourmis sur le pont, on trace ainsi en fonction du temps, le graphe du nombre de fourmis empruntant chaque branche. Le résultat de l'expérience est exposé à la Fig. II.2.

L'illustration (a) représente la configuration physique de l'expérience. Le graphique (b) indique l'évolution de ce système en fonction du temps : on constate que les fourmis ont tendance à emprunter le même chemin (par exemple celui du haut) après une dizaine de minutes.

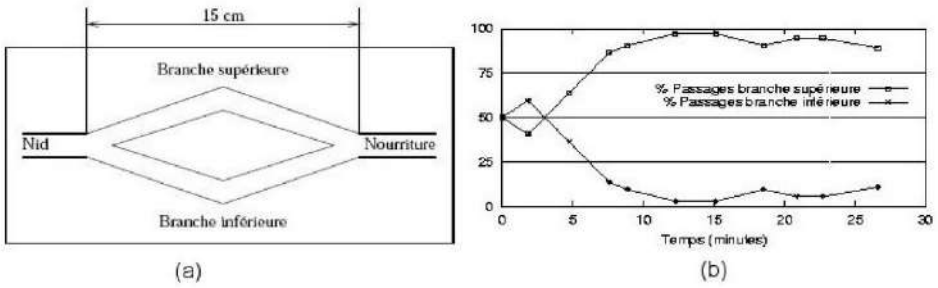


Fig. II.1 : Pont binaire de Deneubourg

Explication :

Au départ, il n'y a pas de phéromone sur le pont. Donc, chaque branche peut être choisie par une fourmi avec la même probabilité. Néanmoins, dans notre exemple, après une certaine période, des variations aléatoires ont fait qu'un peu plus de fourmis ont choisi le chemin du haut plutôt que celui du bas.

Puisque les fourmis déposent des phéromones en avançant et puisqu'elles sont plus nombreuses en haut qu'en bas, le chemin du haut comportera plus de phéromones. Cette quantité supérieure de phéromone incite plus de fourmis à choisir la branche du haut, donc la quantité de phéromone déposée augmentera encore plus.

On en déduit que plus les fourmis suivent un chemin, plus ce chemin devient intéressant à suivre. Ainsi la probabilité avec laquelle une fourmi choisit un chemin, augmente avec le nombre de fourmis qui ont pris ce chemin précédemment.

V.1.1. Expérience du double pont binaire [Dor03]

On peut se demander à présent quel serait l'effet de l'augmentation de la longueur d'une des deux branches du pont. L'effet produit sera que la branche la plus courte sera sélectionnée.

En effet, les premières fourmis qui reviennent au nid avec de la nourriture sont celles qui ont emprunté le chemin le plus court dans les deux sens. Ce chemin, marqué deux fois par les phéromones, attire plus les autres fourmis que le long chemin, qui lui est marqué une seule fois dans le sens de l'aller. Cet effet se renforce au fur du temps, jusqu'à ce que toutes les fourmis choisissent le chemin le plus court.

C'est ainsi que dans cette expérience, on voit que les variations aléatoires sont réduites, puisque les deux chemins n'ont plus la même longueur. Contrairement à la première expérience, le comportement des fourmis qui consistait à suivre les pistes de phéromones n'est plus le seul mécanisme.

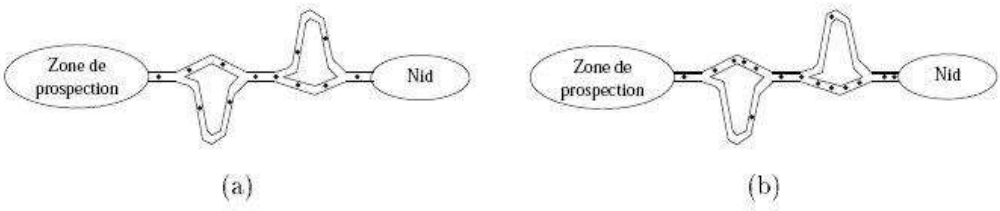


Fig. II.2 : Expérience du double pont binaire

VI. Effet de la coupure d'une piste de phéromone [Dréo04]

Cette fois, les fourmis sont en train de suivre une piste de phéromones, comme présenté à la Fig. II.4. (a). À un moment donné, on a un obstacle qui barre la route des fourmis. Les fourmis qui arrivent à côté de l'obstacle doivent choisir soit d'aller à gauche soit d'aller à droite (b). Puisqu'aucune phéromone n'est déposée le long de l'obstacle, il y a autant de fourmis qui partent à gauche qu'à droite.

Néanmoins, puisque le chemin de droite est plus court que celui de gauche, les fourmis qui l'empruntent, vont retrouver plus vite la piste de phéromone de départ. Pour chaque fourmi allant du nid à la nourriture, on associe également une fourmi qui va de la nourriture au nid (en fait elles ont été séparées par l'apparition brutale de l'obstacle). Les phéromones de ces fourmis vont se superposer à droite. Donc quand elles vont rejoindre le chemin initial, le chemin de droite sera deux fois plus imprégnée de phéromone que la piste de gauche, où les deux fourmis n'ont pas encore pu rejoindre la piste initiale (ce chemin étant plus long).

Les fourmis qui arrivent à l'obstacle à partir de ce moment, préféreront suivre la piste de droite. Le nombre de fourmis qui passent par la droite va augmenter, ce qui augmentera encore la concentration de phéromones. De plus, l'évaporation des phéromones sera plus forte sur la piste de gauche du fait que sa longueur est supérieure. La piste de gauche sera dont rapidement abandonnée, parce qu'elle en est beaucoup moins imprégnée : les fourmis passeront toutes très rapidement par la piste la plus courte.

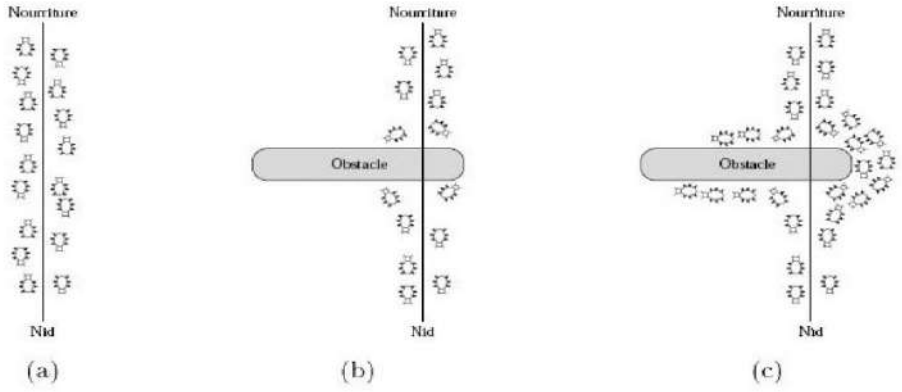


Fig. II.3 : Effet de la coupure d'une piste de phéromone

VII. Colonies de fourmis et résolution de problèmes

À l'origine, l'optimisation par colonie de fourmis (en anglais Ant Colony Optimization - ACO) a été conçue pour résoudre le problème du voyageur de commerce(PVC). L'objectif de ce problème étant de trouver la tournée la plus courte pour servir un ensemble de villes. Le premier algorithme proposé dans ce contexte fut appelé « Ant System » (AS) par son auteur Marco Dorigo. Chaque fourmi représente un agent qui en se déplaçant d'une ville à une autre, laisse des traces sur le chemin parcouru [Dor92]. Le choix des villes à visiter se fait à l'aide d'une probabilité basée sur un

compromis entre l'intensité de la trace laissée et la visibilité qui prend en considération la distance entre les villes. Finalement, une fourmi possède une sorte de mémoire tabou lui rappelant la liste des villes déjà visitées afin de lui permettre de former une solution admissible.

D'autres variantes d'amélioration de cet algorithme ont été proposées.

Nous pouvons citer Ant-Q [Dor96], Ant Colony System (ACS) et Max-Min Ant System [Stu00].

VIII. L'algorithme selon PVC

Soit un graphe $G(X,U)$ tel que :

- X est un ensemble de sites représentant des villes.
- U est un ensemble d'arcs tel que : $U = \{(i,j) | \ i,j \in X\}$
- La distance entre chaque paire de villes est notée par d_{ij}.

Supposons qu'un voyageur de commerce doit passer par l'ensemble des villes une seule fois. Il s'agit donc de trouver un circuit hamiltonien de distance minimale (la somme des distances parcourues) sur le graphe U.

La distance parcourue entre deux villes est représentée par :

$$d = \sqrt{\left[\left(x_1^i - x_1^j\right)^2 + \left(x_2^i - x_2^j\right)^2\right]}$$

Si pour tout i et j ***appartient à X***, $d_{ij} = d_{ji}$ alors nous sommes dans le cas d'un PVC symétrique.

Autrement, si (il existe) i et j ***n'appartient pas à X,*** $d_{ij} \neq d_{ji}$ alors le PVC devient asymétrique.

VIII.1. Colonie de fourmis

Nous définissons dans ce qui suit les variables utilisées dans l'algorithme [COL 96]. Soit :

- $b_i(t)$ le nombre de fourmis se trouvant à la ville i à l'instant t.
- $m = \sum_1^n b_i(t)$ le nombre total de fourmis.
- $Path_{ij}$ le plus court chemin entre la ville i et la ville j.
- d la distance Euclidienne utilisée :

$$d = \sqrt{\left[\left(x_1^i - x_1^j\right)^2 + \left(x_2^i - x_2^j\right)^2\right]} \quad \longrightarrow (1)$$

$t_{ij}(t+1)$ est l'intensité de la trace de fourmis sur le chemin $Path_{ij}$ à l'instant $t+1$ tel que :

$$t_{ij}(t+1) = rt_{ij}(t) + Dt_{ij}(t, t+1) \quad \longrightarrow (2)$$

Où

- ✓ r est un coefficient d'évaporation
- ✓ $Dt_{ij}^k(t, t+1)$ est la quantité de phéromones déposées par la $k^{ième}$ fourmi sur le chemin $Path_{ij}$ entre l'instant t et l'instant $t+1$.

- $t_{ij}(0)$ est initialisée arbitrairement avec un nombre positif constant.
- La visibilité est la quantité $h_{ij} = 1/d_{ij}$
- La probabilité de transition de la ville i à la ville j est définie par :

$$P_{ij}(t) = \frac{[t_{ij}(t)]^{\alpha}[h_{ij}]^{\beta}}{\sum_{j=1}^{n}[t_{ij}(t)]^{\alpha}[h_{ij}]^{\beta}} \quad \longrightarrow (3)$$

Où :

α et β sont des paramètres de contrôle. Ils apportent un équilibre entre la visibilité et l'intensité. A l'instant t, les fourmis font un compromis entre la visibilité et l'intensité de la trace des phéromones présentes sur tous les chemins pour choisir la prochaine destination j.

VIII.2. Description de l'algorithme

- ➢ L'étape initialisation consiste à affecter les fourmis à chaque nœuds et à initialiser l'intensité de la trace de phéromones à chaque chemin.
- ➢ Les fourmis se déplaceront ensuite en choisissant leur destination suivant la probabilité P_{ij} définie précédemment. Cette probabilité est définie en fonction de 2 paramètres : t_{ij} le nombre de fourmis ayant emprunté le même chemin $Path_{ij}$ qu'auparavant et h_{ij} qui permet de favoriser les villes se trouvant dans un environnement assez fermé (villes rapprochées).
- ➢ A chaque fois qu'une fourmi se déplace, l'intensité qu'elle dépose à son passage est utilisée pour le calcul de la valeur de la nouvelle trace sur le chemin $Path_{ij}$. D'autre part, la ville visitée est ajoutée à la liste taboue.
- ➢ Lorsque l'ensemble des fourmis s'est déplacé, toutes les traces sont utilisées pour le calcul des nouvelles probabilités de transition P_{ij}.

- Après n déplacements, la liste taboue de chaque fourmi est supposée pleine, et donc le plus court chemin est mémorisé et la liste est vidée.

On peut résumer tout ça par un pseudo code de l'algorithme colonie de fourmis :

1. Initialisation

T :=0
Initialiser $t_{ij}(t)$ *pour chaque chemin* $Path_{ij}$ *arbitrairement*
Affecter $b_i(t)$ *fourmis à chaque noeud* ***i***.
$Dt_{ij}(t,t+1)=0$ *Pour chaque* ***i*** *et* ***j***.

2. Répéter jusqu'à saturation de la liste taboue (n-1 fois)

Pour i=1 à n faire *(pour chaque ville)*
Pour k=1 à $b_i(t)$ ***faire*** *(pour chaque fourmi se trouvant à la ville* ***i*** *à l'instant* ***t****)*
Choisir la ville destination en considérant la probabilité P_{ij} *selon (3).*
Déplacer la $k^{ième}$ *fourmi vers cette ville.*
Insérer le libellé de cette ville dans la liste taboue de la fourmi ***k****.*
Calculer l'intensité selon (1)
En calculant $Dt_{ij}^{k}(t,t+1)$ *selon (2)*
Calculer $t_{ij}(t+1)$ *et* $P_{ij}(t+1)$
Fin pour
Fin pour

Retenir le plus court chemin à partir de la liste taboue qu'on remet à zéro.
Si *la condition d'arrêt n'est pas satisfaite (le nombre de cycles atteint)*
alors
$t=t+1$,
$Dt_{ij}(t,t+1)=0$,
pour *chaque* ***i*** *et* ***j*** *aller à l'étape 2*
sinon *Afficher le plus court chemin* ***fsi***
Fin.

IX. Conclusion

Les fourmis sont capables de résoudre collectivement des problèmes complexes, comme trouver le plus court chemin entre deux points dans un environnement accidenté. Pour cela, elles communiquent entre elles de façon locale et indirecte, grâce à une hormone volatile, appelée phéromone : au cours de sa progression, une fourmi laisse derrière elle une trace de phéromone qui augmente la probabilité que d'autres fourmis passant à proximité choisissent le même chemin.

Ce mécanisme de résolution collective de problèmes est à l'origine des algorithmes à base de fourmis artificielles. Ces algorithmes ont été initialement proposés dans [Dor92], comme une approche multi-agents pour résoudre des problèmes d'optimisation combinatoire. 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 déposent des 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.

Chapitre III

Conception d'une méta-heuristique à base d'agent pour le PVC

I. Introduction

À travers ce chapitre, nous montrons comment adapter la méta-heuristique optimisation par colonie de fourmis à base d'agent afin de résoudre le problème de PVC.

II. Représentation du problème de PVC

II.1. Enoncé

Le problème de voyageur de commerce consiste à visiter une et une seule fois un nombre fini de villes et revenir à son point d'origine. Trouvez l'ordre de visite des villes qui minimise la distance totale parcourue par le voyageur ». Ce problème d'optimisation combinatoire appartient à la classe des problèmes NP-Complets.

II.2. Enoncé formel

Le problème du voyageur de commerce peut être modélisé à l'aide d'un graphe non orienté constitué d'un ensemble de sommets et d'un ensemble d'arrêtes. Chaque sommet représente une ville, une arrête symbolise le passage d'une ville à une autre, et on lui associe un poids pouvant représenter une distance, un temps de parcours ou encore un coût. Ci-dessous, un exemple de graphe à 6 sommets.

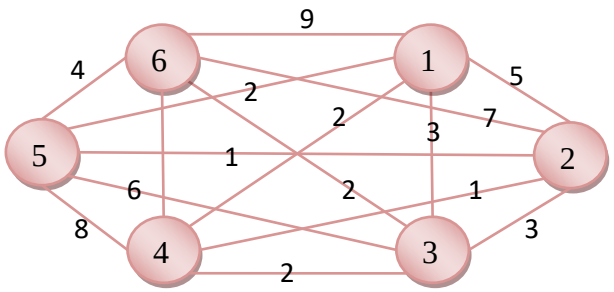


Fig. III.1 : Graphe montre la représentation du problème de PVC de 6 villes

Comme il existe une arrête entre chaque paire de sommets, on dit que ce graphe est « complet ». Pour tout graphe, une matrice de poids peut être établie. En lignes figurent les sommets d'origine des arêtes et en colonnes les sommets de destination ; le poids sur chaque arrête apparaît à l'intersection de la ligne et de la colonne correspondante. Pour notre exemple, cette matrice est la suivante :

	1	2	3	4	5	6
1	**0**	5	3	2	2	9
2	5	**0**	3	1	1	7
3	3	3	**0**	2	6	2
4	2	1	2	**0**	8	1
5	2	1	6	8	**0**	4
6	9	7	2	1	4	**0**

Tableau. III.1 : Matrice de poids représente le problème de PVC de 6 villes

Propriétés de la matrice de poids

1. Elle est carrée: il y a autant de lignes que de colonnes.
2. Il n'y a que des zéros sur la diagonale.
3. Elle est symétrique: $m_{ij} = m_{ji} \qquad \forall i, j = \overline{1,n}$

III. Adaptation de la Colonie de fourmis à base d'agent au problème de PVC

Les points suivants montrent l'adaptation de la colonie de fourmis à base d'agent au problème de PVC.

III.1. Architecture générique du système

Le système est constitué d'un ensemble d'agents, ces agents travaillent en collaboration pour atteindre un objectif commun. On peut représenter ce système par le schéma suivant :

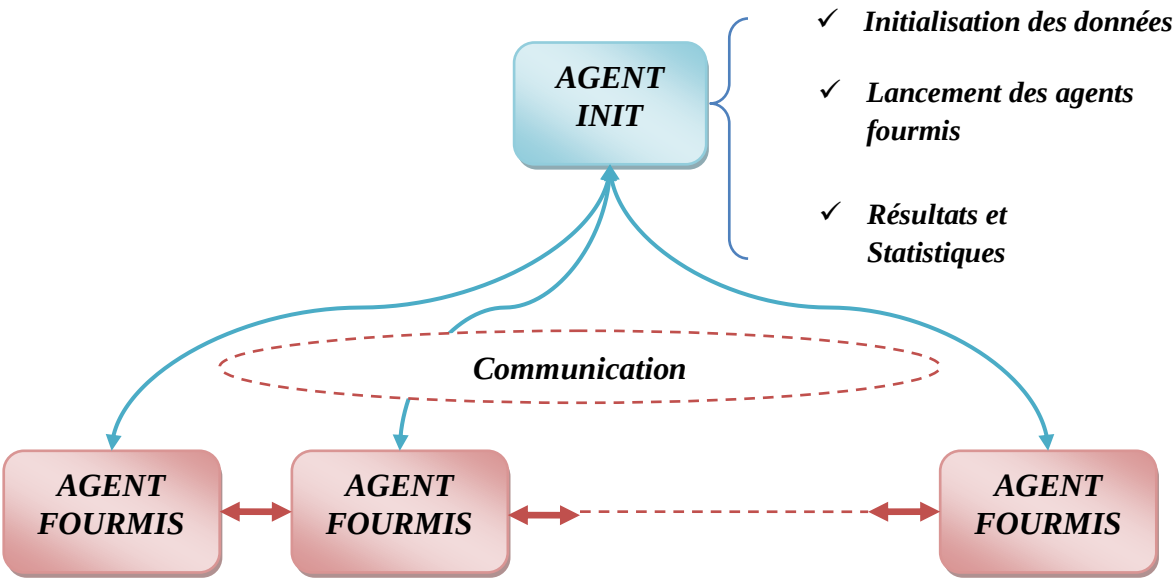


Fig. III.2. Architecture globale du système

Nous détaillons dans ce qui suit, le rôle de chaque composant de ce système :

III.1.1. AGENT INIT

Est un agent principale dans le système, travaille avec les autres agents pour réalisé le but globale. Et les points suivants résument les tâches de l'agent INIT:

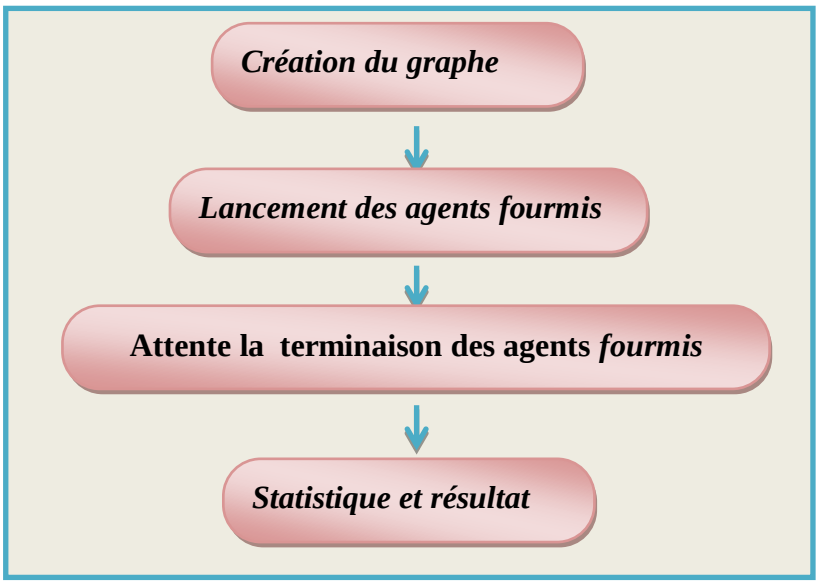


Fig. III.3. Architecture de l'agent INIT

- ***Génération du graphe et initialisation des données :*** Cet agent est le responsable de la création de la matrice représentant le problème.

- ***Lancement des agents fourmis : i.e.*** L'AGENT INIT qui lance un ensemble des agents fourmis.
- ***Résultat et Statistique : i.e.*** L'AGENT INIT qui fait une vision globale sur les modifications du graphe selon les mises à jour de la quantité des phéromones, et après la satisfaire d'un critère d'arrêt L'AGENT INIT choisit la bonne solution (le plus court cycle).

III.1.2. Les AGENTS FOURMIS

Dans notre travail, les agents fourmis déposent des 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.

Il possède les propriétés suivantes :

- Chaque agent fourmis possède un comportement de fourmis.
- Lorsqu'un agent fourmis passe par un arc, il dépose une quantité de la phéromone.
- Tous les agents fourmis se rassemblent ensemble pour trouver une "bonne" solution au problème considéré i.e. le plus court cycle.
- Les agents fourmis coopèrent d'une manière organisée et leur comportement résulte d'interactions simples entre eux (p*ar exemple une fourmi suit une autre fourmi*). Ces interactions permettent à l'ensemble de résoudre des problèmes difficiles comme la recherche de chemins optimaux (le plus court cycle) dans un graphe.
- Chaque agent fourmis possède trois états :
  - ✓ **Cherche premier nœud :** parmi les villes existant l'agent fourmis va choisir la ville de départ.
  - ✓ **Parcours de graphe :** après la recherche de la ville de départ l'agent fourmis parcours la liste des chemins possibles depuis la ville courant, et choisit leur prochaine ville selon la plus grande quantité de phéromone.

    Si la prochaine ville n'est pas déjà visitée alors la ville courante reçoit cette ville. On applique le même principe pour choisir les autres villes jusqu'à la terminaison de cycle.

L'agent fourmis peut avoir une mémoire, qui leur permet de garder les villes visitées.

- ✓ **Rentre :** après la visite du toutes les villes et l'agent fourmis n'a pas trouvé d'autre villes donc l'agent fourmis doit rentrer.

On peut représenter les états précédents par le schéma suivant :

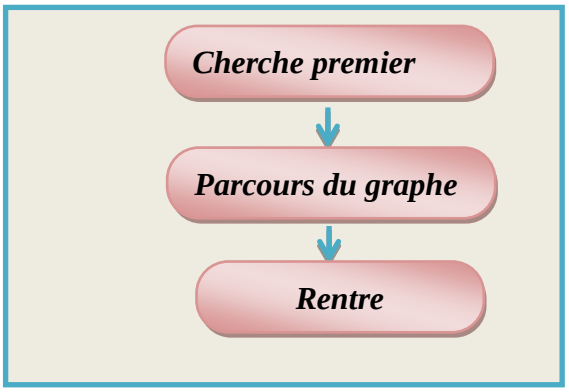


Fig. III.4. Architecture de l'agent fourmis

III.1.3. Critères d'arrêt

Le critère d'arrêt peut être défini comme étant :

- Un nombre maximum d'itération.
- Le temps : on peut fixer un temps limite après lequel la recherche doit s'arrêter.

III.2. Communication inter-agents

Dans un système multi-agent ou la communication joue un rôle central, en tant que processus permettant l'émergence d'un comportement global cohérent du système.

⌦ **Agent INIT vs Agent fourmis :** lorsqu'un agent fourmis termine son cycle il envoie un message à l'agent INIT pour lui confirmer sa terminaison. Cette communication se fait à travers de deux primitives send et receive.

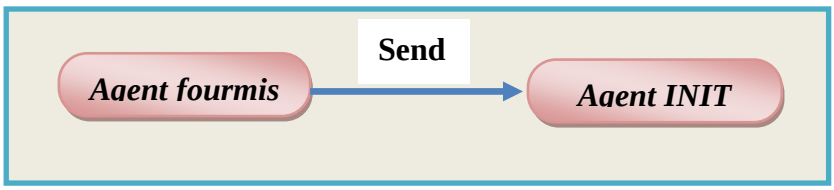


Fig. III.5. Communication de type directe entre Agent INIT et Agent fourmis

⌦ **Agent fourmis vs Agent fourmis :** ce type de communication est appelée stigmergie. En effet, c'est une communication asynchrone qui se fait à travers le changement dynamique de l'environnement (i.e. les pistes de phéromone) et construisant ainsi une solution à un problème, en s'appuyant sur leur expérience collective.

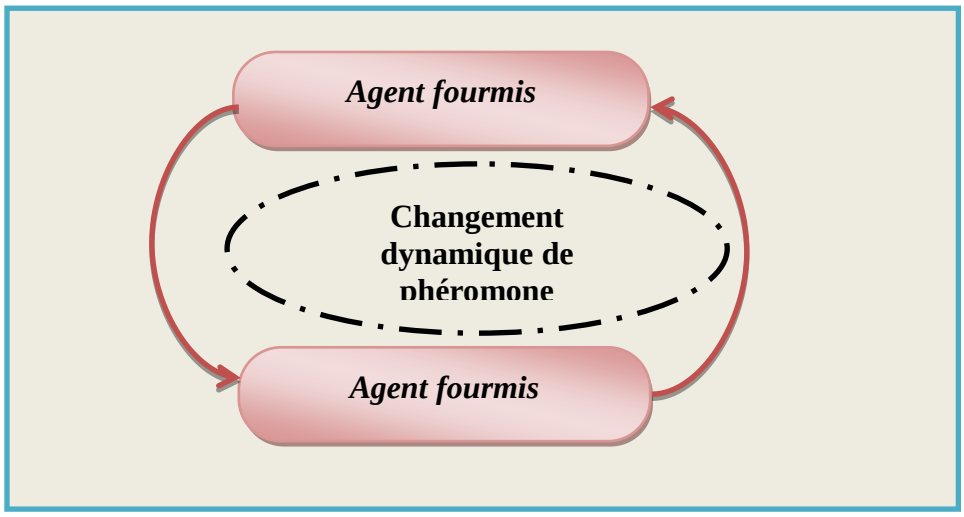


Fig. III.6. Communication de type indirecte entre Agent fourmis et Agent fourmis

III.3. Les règles des agents fourmis

Nous appliquons les idées de l'ACA (ant colony algorithm) au système multi-agents. Ici toutes les fourmis sont des agents indépendants. Les agents souhaitent accomplir une certaine tâche obéissant à certaines règles locales. Maintenant, nous définissons les règles de comportement pour les agents fourmis.

Règle 1 : l'agent fourmis doit trouver le plus court cycle grâce à un certain nombre de ville visitée. Lorsque l'agent fourmi choisit une ville, il se base sur la quantité de phéromone et la visibilité de la ville. Les deux derniers paramètres se calculent comme suivant :

- La visibilité est la quantité $h_{ij} = 1/d_{ij}$ (d_{ij} est la distance entre les villes i et j)
- La probabilité de transition de la ville ***i*** à la ville ***j*** est définie par :

$$P_{ij}(t) = \frac{[t_{ij}(t)]^{\alpha}[h_{ij}]^{\beta}}{\sum_{j=1}^{n}[t_{ij}(t)]^{\alpha}[h_{ij}]^{\beta}}$$

Sachant que :

α est un paramètre pour l'intensité de la piste de phéromone (t_{ij}).

β est un paramètre pour la visibilité des nœuds (h_{ij}).

Règle 2 : lorsqu'un agent fourmi trouve un obstacle dans une route, il rentre à la ville précédente et choisit une autre ville de la même manière.

IV. Conclusion

À travers ce chapitre, nous avons montré comment adapter la méta-heuristique « Colonie de fourmis à base d'agent » au problème de voyageur de commerce.

Dans le prochain chapitre, et afin d'évaluer notre étude, nous allons procéder à tester notre application développée.

Chapitre IV

Expérimentations

I. Environnement expérimental

Dans ce chapitre, nous allons présenter les différentes expérimentations effectuées sur notre application. En effet, nous donnerons une description générale sur les tests effectués ainsi que les résultats obtenus.

I.1. Outils utilisées

Cette partie est consacrée à la présentation des différents outils utilisés pour l'implémentation du notre travail.

⌦ JAVA

Nous avons choisi le langage JAVA, ce choix se justifie par :

- JAVA est un langage multi-plateforme qui permet aux concepteurs, selon le principe : « write once, run every where », d'écrire un code capable de fonctionner dans tous les environnements (quelque soit le système d'exploitation).
- JAVA est un langage orienté objet, simple qui réduit le risque d'erreurs et d'incohérence.
- JAVA est dotée d'une riche bibliothèque de classe comprenant la gestion des interfaces graphiques.

⌦ JADE

Concernant le développement des agents nous avons opté pour la plateforme JADE (Java Agent Développement Framework), qui est une plateforme gratuite et open source qui facilitent le développement des systèmes à base d'agents répondant aux spécifications FIPA, les agents développés sous la plateforme Jade, sont entièrement écrits en Java. Ses principales caractéristiques sont :

- Jade n'est couplée à aucune méthodologie de développement.
- Jade simplifie l'implémentation d'un SMA à travers un MiddleWare en répondant aux spécifications de la FIPA, une librairie de classes que les

utilisateurs peuvent utiliser et étendre ainsi qu'un ensemble d'outils graphiques qui permettent le débogage et l'administration du SMA.

- Jade assure une communication transparente par échange de messages dans le langage normalisé FIPA-ACL.

II. Processus de simulation

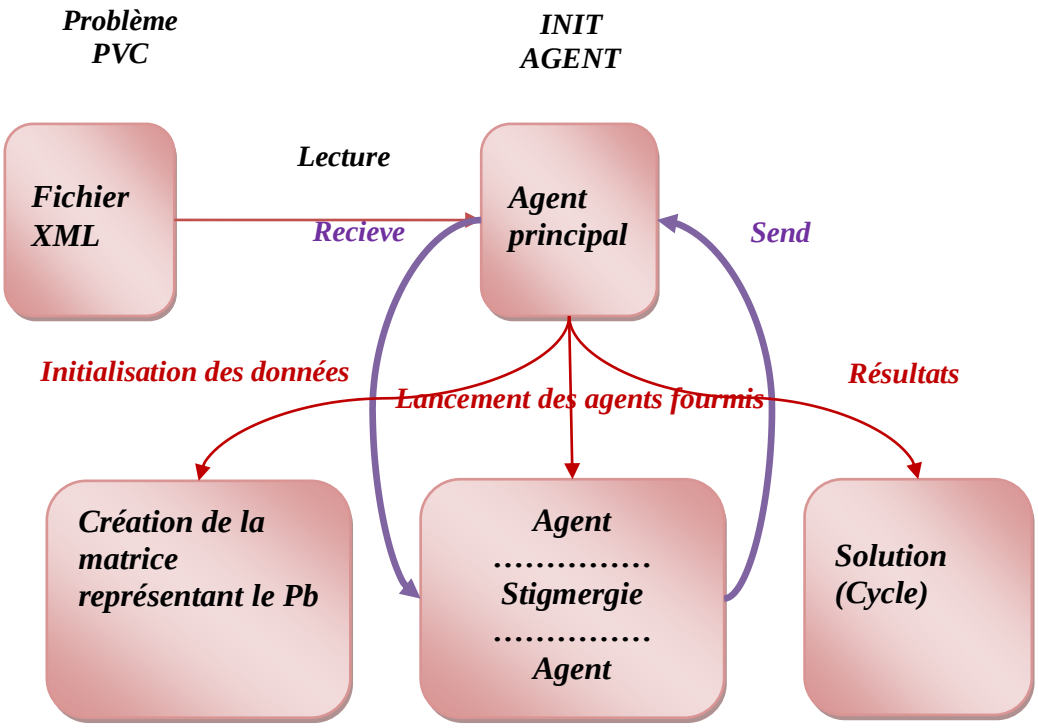


Fig. IV.1: Schéma représente le processus de simulation

III. Description des tests

III.1. Description de problème

Il existe plusieurs problèmes sur PVC, nous avons choisit le problème «Att48» pour faire des tests. Ce problème contient 48 villes d'US (Padberg/Rinaldi) [*www.dcs.gla.ac.uk/~pat/af2009/tspDemo/**att48**.tsp*].

Sachant que la solution optimale de ce problème est : 10628 Km.

III.2. Description et résultats des tests effectués

Les tests sont organisés comme suit :

(1) Variation de distance en fonction du nombre d'agents

Durant ce test, nous allons tester la variation de la distance de la solution trouvée (cycle) en fonction du nombre d'agents. Durant ce test, les différents paramètres ont été fixés comme montre le tableau suivant :

Paramètre	Valeur
Nombre d'itération	100
Vitesse d'évaporation de la phéromone	50
Nombre de phéromone à évaporer	0.5

Tableau. IV.1 : Les valeurs des paramètres de test 1.

Le tableau suivant montre les résultats obtenus durant ce premier test :

Nombre d'agents	**1**	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**
Distance(Km)	46704	47765	48414	45754	47561	46307	45667	45331	45293	43154	42372

Tableau. IV.2 : Les résultats de test 1.

Les valeurs des distances (D_f) sont des parties entières obtenues par le calcul de la moyenne entre dix valeurs de tests par la formule suivante :

$$D_f = \frac{\sum_{j=1}^{10} D_j}{10}$$

Où D_j représente la valeur de distance obtenue à la $j^{ème}$ test.

Le graphe suivant montre la variation de la distance trouvée en fonction du nombre d'agents :

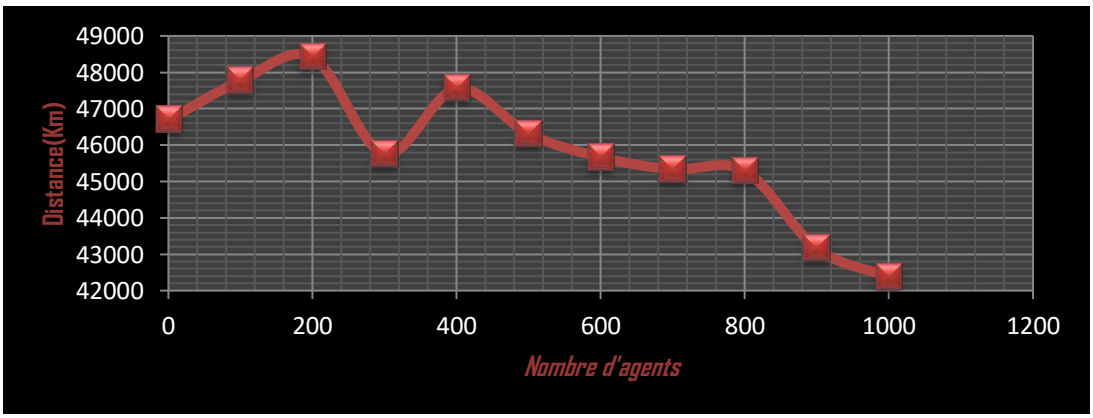


Fig. IV. 2: Variation de distance en fonction du nombre d'agents

❖ Interprétation du résultat

Cette expérience montre que la méta-heuristique colonie de fourmis à base d'agent peut améliorer la solution par le fait de travail collaboratif. En effet, à chaque fois qu'on augmente dans le nombre d'agents, alors on augmente dans la quantité de phéromone qui influence sur le choix des chemins. Par conséquent, la probabilité de trouver des solutions avec moins de coût s'augmente ce qui explique l'amélioration de la solution à chaque fois qu'on augmente dans le nombre d'agents.

(2) Variation de distance en fonction du nombre d'itérations

Dans ce deuxième test, nous allons tester la variation de la distance en fonction du nombre d'itérations. Durant ce test, les différents paramètres ont été fixés comme montre le tableau suivant :

Paramètre	**Valeur**
Nombre d'agents	10
Vitesse d'évaporation de la phéromone	50
Nombre de phéromone à évaporer	0.5

Tableau. IV.3 : Les valeurs des paramètres de test 2.

Le tableau suivant montre les résultats obtenus durant cette expérience :

Nombre d'itérations	**100**	**500**	**1000**	**1500**	**2000**	**2500**	**3000**	**3500**	**4000**	**4500**	**5000**
Distance(Km)	48323	45407	43177	45917	45475	44694	43142	42923	44238	43181	43032

Tableau. IV.4 : Les résultats obtenus de test 2.

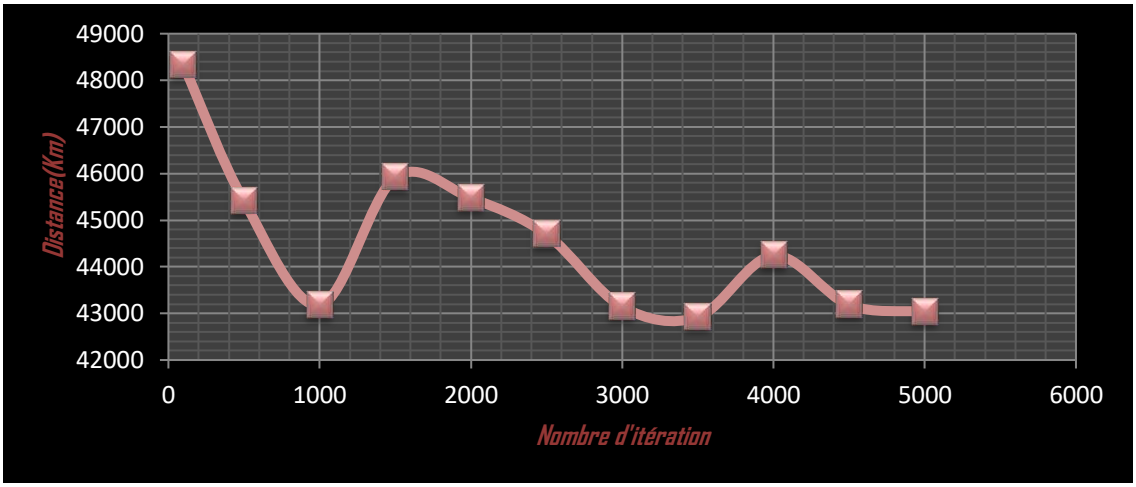


Fig. IV.3 : Variation de distance en fonction du nombre d'itérations

❖ Interprétation du résultat

La méta-heuristique colonie de fourmis à base d'agent Dans cette expérience donne une amélioration de la solution. En effet, à chaque fois qu'on augmente dans le nombre d'itération, alors on augmente dans l'espace de recherche et la quantité de phéromone pour choisir la meilleure solution.

(3) Variation de distance en fonction de la vitesse d'évaporation

Ce test consiste à tester la variation de la distance en fonction de vitesse d'évaporation. Pour cela, les différents paramètres ont été fixés comme montre le tableau suivant :

Paramètre	**Valeur**
Nombre d'agents	10
Nombre d'itération	100
Nombre de phéromone à évaporer	0.5

Tableau. IV.5 : Les valeurs des paramètres de test 3.

Le tableau suivant montre les résultats obtenus durant ce test :

vitesse d'évaporation	**1**	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**	**180**	**200**
Distance(Km)	58149	47349	47868	43956	43811	47449	46368	45791	47121	46863	44024

Tableau. IV.6 : Les résultats obtenus de test 3.

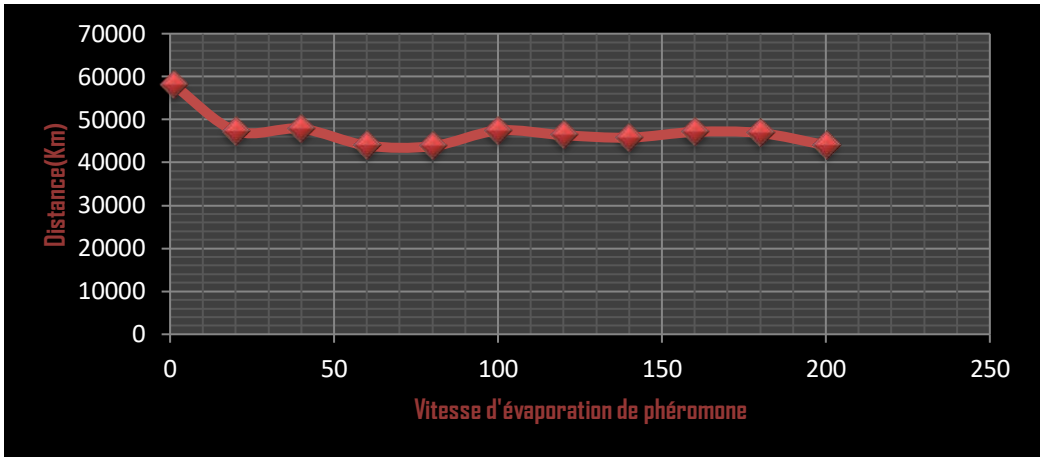


Fig. IV.4 : Variation de distance en fonction de la vitesse d'évaporation

❖ Interprétation du résultat

Dans cette expérience, à chaque fois qu'on augmente dans la vitesse d'évaporation de phéromone, alors les agents fourmis éliminent les chemins dans lesquels leurs quantités de phéromones sont négligeables.

IV. Conclusion

Dans ce chapitre, nous avons présenté les différents tests effectués à travers de notre prototype développé. En effet, nous avons étudiés l'effet de certains paramètres de la colonie de fourmis sur le résultat de la simulation.

Conclusion générale

L'objectif primordial du présent travail consistait à contribuer à l'intégration du paradigme agents et Système multi-agent lie par une méta-heuristique colonie de fourmis, pour résoudre un problème très connu (le problème du voyageur de commerce (PVC) qui est l'un des problèmes combinatoires les plus étudiés en RO et théorie des graphes et en optimisation combinatoire. Il consiste à trouver un cycle hamiltonien qu'est un cycle qui passe par chaque sommet une fois et une seule.

Au cours de notre projet nous avons représenté :

Au premier chapitre les concepts agent et le système multi-agents, défini leurs caractéristiques, et les notions fondamentales de ce domaine. Il est intéressant de noter, que l'ensemble des concepts de l'approche multi-agents font de celle-ci une approche très riche et très puissante en termes de modélisation.

Au deuxième chapitre une méta-heuristique colonie de fourmis, et on remarque que bien qu'une seule fourmi soit capable de construire une solution (i.e. de trouver un chemin du nid à la nourriture), c'est seulement le comportement de l'ensemble de la colonie qui crée le chemin le plus court [Dréo04].

Au troisième chapitre une combinaison entre les deux SMA et Colonie de fourmis pour faire la résolution de problème du voyageur de commerce (PVC).

Nous avons implémenté une méta-heuristique colonie de fourmis à base d'agent pour résoudre le problème de PVC.

Enfin, nous avons terminé par effectuer certaines expérimentations sur ce prototype réalisé.

Perspectives :

Comme perspectives, nous proposons d'un coté d'étendre le travail courant dans le but de prendre le problème des obstacles en considération et d'autre coté d'adapter une autre méta-heuristique, à titre d'exemple les algorithmes génétiques, dans le but de faire une comparaison avec notre prototype réalisé.

Bibliographie

[Als09]	Ghalib Alshammri, "Investigation of Agent or Multi-agent Technologies in Ecommerce Systems", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.10, October, 2009.
[Bon94]	Bonabeau .E et Theraulaz .G. éditeurs (1994). Intelligence collective. Hermès.
[Ben96]	S. Benkrid and F. Merzougui. Développement d'un système d'aide à la décision par négociation en aménagement du territoire à base d'un système multi-agents.Mémoire d'ingénieur, Institut National d'Informatique, Alger, 2006.
[Bri01]	Jean-Pierre Briot, Yves Demazeau, "Introduction aux agents : Principes et architecture des systèmes multi-agents", collection IC2, Hermès, 2001.
[Cha96]	B. Chaib-draa, and D. Vanderveken. Agent communication language : towards a semantics based on success, satisfaction and recursion, in Intelligent Agents V, Agent Theories, Architectures and Language, Müller, J.P., Singh M.P., Rao A.S. (Eds), Lecture Notes on AI, Springer Verlag, Berlin, 1999.
[Cha01]	B. Chaib-draa, I. Jarras et B. Moulin,"Systèmes multiagents : Principes généraux et applications", Article à paraître dans : J. P. Briot et Y. Demazeau « Agent et systèmes multiagents » chez Hermès en 2001.
[Dro91]	Drogoul, A., Ferber, J., Corbara, B., and Fresneau, D. (1991). __ A Behavioral Simulation Model for the study of Emergent Social Structures. In (Varela and Bourgine, 1991), pages 161–170.
[Dor92]	Dorigo, M. 1992. Optimization, Learning and Natural Algorithms (in Italian). Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

[Dro93]	Drogoul, A. (1993). De la simulation multi-agents à la résolution collective de problèmes. Thèse de doctorat, University Paris 6
[Dor96]	Dorigo, M. and Gambardella, L. (1996). __ A Study of some properties of Ant-Q __. In (Voigt et al. 1996), pages 656–665.
[Dri01]	Benjamin Drieu, " L'intelligence artificielle distribuée appliquée aux jeux d'équipe situés dans un milieu dynamique: l'exemple de la RoboCup", Université Paris 8, 2001.
[Dor03]	Un chapitre dans son livre qui parle des algorithmes élaborés des colonies de fourmis notamment adaptés pour le TSP et divers problèmes d'optimisation dynamiques.
[Dréo04]	Un chapitre dans son livre qui parle de la méta-heuristique ACO, d'une manière simplifiée.
[Fin94]	T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML as an agent communication language", In Third international conference on information and knowledge management. ACM Press, November 1994.
[Fer95]	Jacques Ferber, "Les Systèmes Multi Agents: vers une intelligence collective", InterEdidion ,1995.
[Fra97]	Franklin,Stan and Graesser, "is it an Agent ,or just a program?:A Taxonomy For Autonomous Agents", Proccedings of the Third International Workshop on Agent Theories,Architecture, And Languages Springer-Verlag 1997.
[Jen98a]	N. R. Jennings and M. Wooldridge, "Applications of Intelligent Agents", Queen Mary & Westfield College, University of London, 1998.
[Jen98b]	Jennings N, Sycara K., Wooldridge M, "A Roadmap of Agent Research and evelopment" in Autonomous Agents and Multi-Agent Systems,, Kluwer Academic Publishers,Boston,1998.

[Lab97]	Y. Labrou and T. Finin, "A Proposal for a new KQML Specification", rapport technique TR CS-97-03, C.S. and E.E. dep., university of Maryland Baltimore County, février 1997.
[Min86]	Minsky, M. (1986). The Society of Mind. New York: Simon and Schuster.
[Mar02]	Marjorie le bars," Un Simulateur Multi-Agent pour l'Aide à la Décision d'un Collectif: Application à la Gestion d'une Ressource Limitée Agro-environnementale",Thèse de Doctorat, université paris ix-dauphine, France, 2002.
[New82]	A. Newell, "The Knowledge Level.Artificial Intelligence", page 87-127, 1982.
[Ngu04]	Duc-Minh Nguyen, Alexis Drogoul, Vu Duong, "Conception d'un simulateur multi-agents pour la Gestion du Trafic Aérien", Intl. Conf. RIVF'04, 2004.
[Par87]	Parunak, H. V. D," Manufacturing experience with the contract net", In: M. N. Huhns (Ed.) Distributed AI. Morgan Kaufmann, 1987.
[Rot89]	B. H. Roth, M. Hewett, R. Washington, R. Hewett, and A. Seiver, " Distributing intelligence within an individual", In Les Gasser and Michael N. Huhns, editors, Distributed Artificial Intelligence, volume 2 of Research Notes in Artificial Intelligence, pages 385-412. Pitman, 1989.
[Rus95]	Stuart Russell & Peter Norvig, P, "Artificial intelligence: A modern approach", second edition, Englewood Cliffs, NJ: Prentice Hall, 1995.
[syc98]	K. P. Sycara. Multi-agents Systems, AI Magazine, Summer 1998.
[Stu00]	T. Stützle and H.H. Hoos. MAX-MIN Ant System. Journal of Future Generation Computer Systems, special isuue on Ant Algorithms, 16:889–914, 2000.
[Woo95]	Michael Wooldridge, Nicholas R. Jennings, " Intelligent Agents:

	Theory and Practice",Manchester Metropolitan University, United Kingdom, Submitted to Knowledge Engineering Review, October 1994.Revised January 1995.
[Zam03]	Franco Zambonelli, Nicholas R. Jennings, Michael Wooldridgez, "Developing Multiagent Systems: The Gaia Methodology", Università di Modena e Reggio Emilia, 2003.
	[*www.dcs.gla.ac.uk/~pat/af2009/tspDemo/**att48**.tsp*]. Un exemple pour les teste de TSP

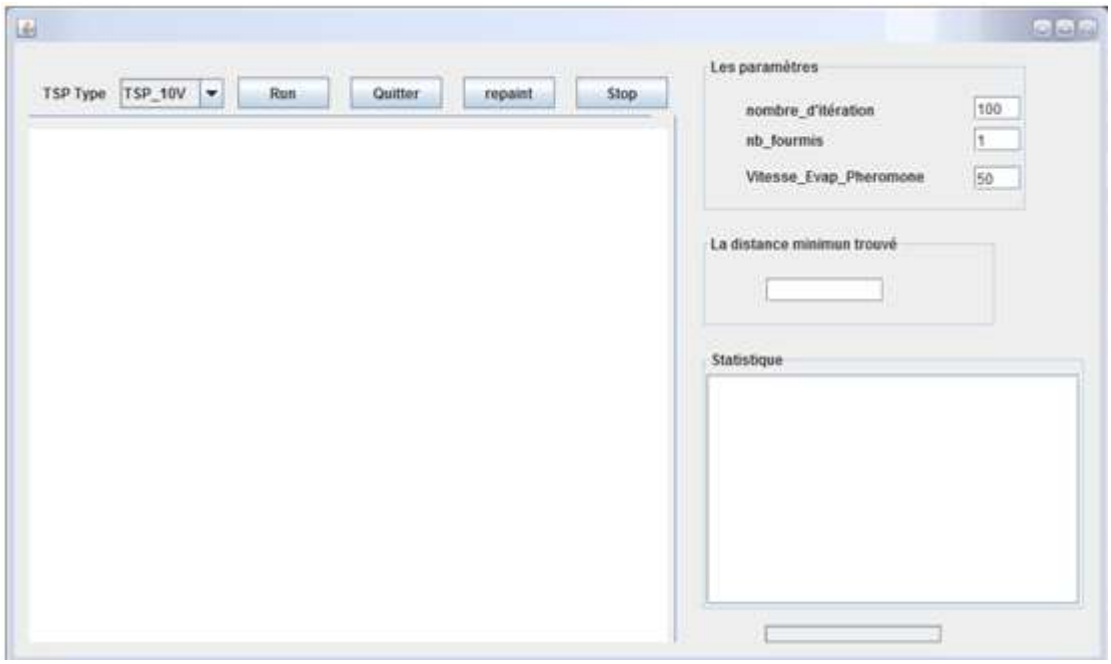
Annexe

Cette annexe donne un aperçu sur le prototype réalisée. Premièrement, nous devons représenter le graphe représentant notre problème. Pour cela, nous avons choisi le format XML.

Voici un exemple de fichier XML représente un graphe de 10 villes générées aléatoirement.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Graph xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <les_Sommets>
    - <Sommet y="390" x="69" nom="1">
      - <liste_Liaisons>
          <Liaison cout_liaison="384" next_Sommet="2"/>
          <Liaison cout_liaison="288" next_Sommet="3"/>
          <Liaison cout_liaison="432" next_Sommet="4"/>
          <Liaison cout_liaison="519" next_Sommet="5"/>
          <Liaison cout_liaison="534" next_Sommet="6"/>
          <Liaison cout_liaison="358" next_Sommet="7"/>
          <Liaison cout_liaison="317" next_Sommet="8"/>
          <Liaison cout_liaison="465" next_Sommet="9"/>
          <Liaison cout_liaison="152" next_Sommet="10"/>
        </liste_Liaisons>
      </Sommet>
    - <Sommet y="371" x="453" nom="2">
      - <liste_Liaisons>
          <Liaison cout_liaison="384" next_Sommet="1"/>
          <Liaison cout_liaison="438" next_Sommet="3"/>
          <Liaison cout_liaison="293" next_Sommet="4"/>
          <Liaison cout_liaison="289" next_Sommet="5"/>
          <Liaison cout_liaison="161" next_Sommet="6"/>
          <Liaison cout_liaison="334" next_Sommet="7"/>
          <Liaison cout_liaison="184" next_Sommet="8"/>
          <Liaison cout_liaison="137" next_Sommet="9"/>
          <Liaison cout_liaison="289" next_Sommet="10"/>
        </liste_Liaisons>
      </Sommet>
    - <Sommet y="104" x="105" nom="3">
      - <liste_Liaisons>
          <Liaison cout_liaison="288" next_Sommet="1"/>
          <Liaison cout_liaison="438" next_Sommet="2"/>
          <Liaison cout_liaison="272" next_Sommet="4"/>
          <Liaison cout_liaison="384" next_Sommet="5"/>
          <Liaison cout_liaison="523" next_Sommet="6"/>
          <Liaison cout_liaison="163" next_Sommet="7"/>
          <Liaison cout_liaison="264" next_Sommet="8"/>
          <Liaison cout_liaison="431" next_Sommet="9"/>
          <Liaison cout_liaison="192" next_Sommet="10"/>
```

Ensuite, et dans le but de lancer la simulation, nous avanos l'interface graphique présentée dans la figure suivante :



En fin, et après le lancement de la simulation, l'interface graphique montre le cycle trouvé ainsi que sa distance le temps écoulé.

