

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Conduite de Projets Informatiques.**

Présenté par

Sadia BEN TOUIRAD

Salima AMARA

Thème

Conception et réalisation d'une application sous android pour l'orientation.

Mémoire soutenu publiquement le 06/10/2016. devant le jury composé de :

Président : M Rachida AOUDJIT

Encadreur : M Ferroudja SEDDOUD

Examineur : M Yasmine YESLI

Examineur : M BOUSENINA

TABLES DES MATIERES

TABLES DES MATIERES1

LISTE DES FIGURES

INTRODUCTION GENERALE.....	1
CHAPITRE1ANDROID	2
1.INTRODUCTION.....	3
2. DEFINITION.....	3
3. HISTORIQUE DE VERSION	3
4. ARCHITECTURE DE LA PLATFORME ANDROID :	4
4.1. PREMIER NIVEAU : NOYAU LINUX.....	5
4.2. DEUSIEME NIVRAU : ENVIRONNEMENT D'EXCUTION	6
4.2.1. LES LIBRAIRIE.....	6
4.2.2. L'ENVIRONNEMENT D'EXECUTION:.....	6
4.2.2.1. CORE LIBRAIRIES.....	6
4.2.2.2. DALVIK.....	7
4.3. TROISIEME NIVEAU: MODULE DE DEVELOPPENT D'APPLICATION.....	7
4.3.1CORE PLATEFORME SERVICES.....	8
4.3.2. HARDWARESERVICES.....	8
4.4. QUATRIEME NIVEAU: APPLICATION.....	9
5. COMPOSITE D'UNE APPLICATION ADROID :	9
5.1. LES ACTIVITES	9
5.2. LESSERVIES.....	10
5.3. LES BROADCAST RECEIVERS.....	10
5.4. LES CONTENTS PROVIDERS	10
5.5. LES INTENTS.....	10
6. CYCLE DE VIE D'UNE APPLICATION ANDROID :....	10
7. DEVELOPPEMENT SOUS ANDROID :.....	11
7.1. ENVIRONNEMENT DEDEVELOPPEMENT.....	11
7.1.1. LE JDK.....	12
7.1.2. LE SDK ANDROID.....	12
7.1.3 .L'IDE ECLIPSE.....	13
7.1.4. LE PLUGIN ADT POUR ECLIPSE	13

7.1.5. L'EMULATEUR DE TELEPHONE	14
7.2. LA STRUCTURE D'UN PROJET	14
8. CONCLUSION..	16
CHAPITRE2 GEOLCALISATION.....	17
1. INTRODUCTION.....	18
2. GEOLOCALISATION.....	18
2.1. TECHNIQUE DE GEOLOCALISATION.....	19
2.1.1. GSM.....	19
2.1.2. GEOLOCALISATION PAR WIFI.....	20
2.1.3.GEOLOCALISATION PAR ADRESSE IP.....	21
2.1.4. GEOLOCALISATION PAR RFID.....	22
2.1.5. GEOLOCALISATION PAR GPS.....	22
2.1.5.1. PRESENTATION	23
2.1.5.2. HISTORIQUE.....	23
2.1.5.3. FONCTIONNEMENT.....	24
2.2.COMBINAISON DE DIFFERENTES TECHNIQUES.....	25
2.3. TEL RELEVÉ D'INFORMATION.....	25
2.4. LE SYSTEME DE COORDONNEES LOCALES	26
2.5. PLATFORME LOGICIEL DE GEOLOCALISATION.....	26
3.OPEN STREET MAP.....	29
3.1. PROJET ET OUTILS CONNU PAR LA CARTOGRAPHIE COLABORATIF.....	29
3.1.1. GOOGLE MARKER ET GOOGLE MY MAPS	29
3.1.2. USHAHIDI.....	30
3.1.3. WIKI MAPIA.....	33
3.2. MISE EN OUVRE D'OPENSTREET MAP.....	31
4.CONCLUSION	32
CHAPITRE3OPTIMISATION PAR COLONIE DE FOURMI.....	33
1. INTRODUCTION.....	34
2. LES FOURMI REELLES.....	34
2.1.CARACTERISTIQUES D'UNE COLONIE DE FOURMI.....	35
2.2. EXPERIENCE D'OBSTACLE	36
2.3.EXPERIENCE A DOUBLE BRANCHE	36
3.FOURMI ARTIFICIELLE	37

3.1. SIMILARITE ENTRE LES FOURMI REELES ET ARTIFICIELLES	38
2.2. DEFFIRENCE ENTRE LA FOURMI ENTRE LES FOURMI REELES ET ARTIFICIELLES.....	38
4. L'OPTIMISATION PAR COLONIE DE FOURMIS (ACO).....	39
4.1. REPRESENTATION DU PROBLEME.....	39
4.2. LES PROPRIETE DE FOURMIS	40
4.3. CARATERISTIQUE DE L' ALGORITHME ACO.....	41
4.3.1.LES PISTE DE PHEROMONE	41
4.3.2. L'UTILISATION DE L'INFORMATION HEURISTIQUE	41
4.3.3. LE NOMBRE DE FOURMI DE LA COLONIE M.	42
4.3.4. LA LISTE CANDIDATE.....	42
4.3.5. L'EXPLORATION ET L'EXPLOITATION DE L'ESPACE DE RECHERCHE.....	42
4.3.6. LA REGLE DE TRANSITION D'ETAT.....	43
4.4. LES EFFET S DE L'ALGORITHME ACO	43
4.4.1. L'EVAPORATION DE PHEROMONE..	43
4.4.2.LES ACTION DE DEMON.....	44
4.5. ACTIVITES DES FOURMIS	44
4.5.1. STRATEGIE DE CONSTRUCTION	46
4.2.2. STRATEGIE PAR AMELIORATION.....	46
4.6. LES ALGORITHME ACO.....	46
4.6.1. ALGORITHME DE BASE	46
4.6.2. ANT SYSTEM & ELITISME.....	49
4.6.3. MAX-MIN ANT SYSTEME.....	49
4.6.4. ANT COLONY SYSTEM (ACS)	49
5. LE PARALLELISME DE ALGORITHME ACO.....	53
6. CONCLUSION.....	53
CHAPITRE4CONCEPTION.....	54
1. INTRODUCTION.....	55
2. HYPOTHESES DE TRAVAIL.....	55
3. DESCRIPTION DE LA SOLUTION	56
3.1. STRUCTURE DES TABLES	56
4. DESCRIPTION DE L'ALGORITHME VANTS.....	58
5. COMPORTEMENT DE L'ALGORITHME AU NIVEAU NOUEDS	64
6. CONCEPTION DE LA BASE DE DONNEES.....	66

6.1. DICTIONNAIRE DE DONNEES	66
6.2. MODELE ENTITE ASSOCIATION	67
6.3. DESCRIPTION TEXTUELLE	68
7. FONCTIONNEMENT DE L'APPLICATION.....	70
8. CONCLUSION	70
CHAPITRE 5 REALISATION.....	71
1.INTRODUCTION.....	72
2.LES DIFFERENTES CLASSES DE L'ALGORITHME FOURNI.....	72
3. VARIATION DE LA PHEROMONE EN FONCTION DE TEMPS.....	73
4. PREREQUIS LOGICIEL.....	73
4.1. LANGUAGE DE PROGRAMMATION C.....	73
4.2 IDE ECLIPSE.....	73
4.3. ANDROID STUDIO.....	74
5. PRESENTATION DE L'APPLICATION.....	75
6.CONCLUSION.....	77
CONCLUSION ET PERSPECTIVES.....	78

LISTE DES FIGURES

FIGURE 1.1 : LOGO D'ANDROID.....	3
FIGURE1.2: LES VERSIONS D'ANDROID	4
FIGURE 1.3: ARCHITECTURE D'ANDROID	5
FIGURE 1.4 : LA MACHINE VIRTUELLE DALVIK	7
FIGURE1.5 : CYCLE DE VIE D'UNE APPLICATION ANDROID.	11
FIGURE 1.6 : LE SDK ANDROID	13
FIGURE1.7 : L'EMULATEUR	14
CHAPITRE 2	
FIGURE 2.1 : PRINCIPE DE LA GEOLOCALISATION PAR GPS	19
FIGURE 2.2: GEOLOCALISATION PAR WIFI.	21
FIGURE 2.3 : ARCHITECTURE D'UN SYSTEME DE GEOLOCALISATION PAR GPS. ..	27
FIGURE2.4 : LOGO D'OPENSTREETMAP.	31
CHAPITRE 3	
FIGURE 3.1 : EXPERIENCE DE L'OBSTACLE [DRÉ01].....	36
FIGURE 3.2 : EXPERIENCE A DOUBLE BRANCHES [DRE03].....	37
FIGURE 3.3 : LE PSEUDO-CODE DE LA META-HEURISTIQUE ACO [DOR07].....	39
FIGURE 3.4 : PSEUDO-CODE D'ACTIVITE DES FOURMIS [GUE07].....	45
FIGURE 3.5 : PVC OPTIMISE PAR L'ALGORITHME ANT SYSTEM [DRE03].....	48
FIGURE 3.6 : LA PROCEDURE ACS DE LA STRATEGIE CONSTRUCTIVE	51
FIGURE 3.7 : LA PROCEDURE ACS DE LA STRATEGIE PAR AMELIORATION	52
CHAPITRE 4	
FIGURE 4.1.REPRESENTATION D'UNE CARTE ROUTIERE SOUS FORME D'UN GRAPHE	56
FIGURE 4.2 : LA MATRICE DE PHEROMONES $T_I = [T_{IJD}]$ AU NIVEAU DU NŒUD I. 57	
FIGURE 4.3 : LA TABLE DE STATISTIQUE $M_I(M_{ID}, \Sigma_{ID}^2, W_{ID})$	58
FIGURE4.4 : DEPLACEMENT D'UNE VANT.....	59
FIGURE 4.5 : DETECTION D'UN CYCLE PAR UNE VANT.....	60
FIGURE4.6 : MISE A JOUR DES TABLES DE PHEROMONES CORRESPONDANT A CHAQUE NŒUD VISITE.	63
FIGURE 4.7 : ORGANIGRAMME DU CHEMIN OPTIMAL.....	64
FIGURE 4.8 MODELE ENTITE-ASSOCIATION DECRIVANT LA BASE DE DONNEES	67
FIGURE 4.9: FONCTIONNEMENT GENERAL DE L'APPLICATION.....	71

CHAPITRE 5

FIGURE 5.1 : LA VARIATION DE PHEROMONE EN FONCTION DE TEMPS.....	73
FIGURE 5.2 : INTERFACE GRAPHIQUE DE L'APPLICATION1.....	75
FIGURE5.3: INTERFACE GRAPHIQUE DE L'APPLICATION2.....	76
FIGURE5.4: INTERFACE GRAPHIQUE DE L'APPLICATION3.....	77

Remerciements

D'abord nous remercions le bon dieu de nous avoir donné santé, courage, volonté et foi pour réaliser ce travail.

Nous tenons à exprimer notre profonde gratitude à notre encadreur M^{me} Seddoud Farroudja, pour tout ce qu'elle nous a apporté comme aide, connaissances et conseils pour l'accomplissement de ce travail.

Nous remercions vivement les membres du jury pour avoir accepté d'évaluer notre travail.

Aux enseignants de département d'informatique de l'UMMTO pour l'effort qu'ils ont déployé afin d'assurer notre Formation, pour leurs compétences, et surtout leur modestie.

Nous aimerions aussi remercier nos familles, nos amis, ainsi que tous ceux qui ont contribué de prêt ou de loin à la réalisation de ce travail.

Dédicace

*À ma très chère mère et mon très cher père que je remercie très fort pour
tous les sacrifices qu'ils ont faits pour moi.*

À mes sœurs et mon frère qui m'ont donné leur soutien.

À ma chère grand-mère que dieu la protège.

À ces personnes avec qui j'ai vécu le bonheur du mot "amitié"...

B.Sadia

Dédicace

*À ma très chère mère et mon très cher père que je remercie très fort pour tous
les sacrifices qu'ils ont faits pour moi.*

À mes frères qui m'ont donné leur soutien.

À ces personnes avec qui j'ai vécu le bonheur du mot "amitié"...

A.Salima

Introduction général

Un des problèmes ennuyeux de notre mode de vie est le voyage pendulaire. La situation devient encore pire lorsque la circulation est mauvaise, voire franchement désagréable lorsqu'on se retrouve coincé dans un embouteillage.

L'idéal serait bien sûr de pouvoir connaître la position GPS, de chaque voiture, afin de repérer à l'avance l'itinéraire le moins encombrée possible et le plus rapide pour atteindre la destination.

En nous inspirant de la population myrmécéenne, qui a compris que l'union fait la force, nous pourrions grandement améliorer la situation :

Lorsqu'une fourmi cherche à savoir quel chemin suivre pour arriver jusqu'à sa destination, elle se fie aux informations que ses congénères ont laissées sur la route sous forme de phéromones. Plus l'empreinte « phéromonique » est forte, plus le chemin a été emprunté et meilleur il est.

En copiant le phénomène de mutualisation on peut imaginer des voitures-fourmis qui transmettent cette même information entre elles, ce qui permettrait au conducteur de faire le choix inverse : aller là où il n'y a personne.

L'objectif de notre travail est de réaliser une application sous android pour l'orientation routière.

Notre travail est structuré en cinq chapitres. Le premier chapitre décrit les généralités sur android. Le deuxième chapitre est consacré à la géolocalisation, le troisième présente d'une manière détaillée l'optimisation par colonie de fourmis ACO, La conception de l'application est illustré dans le chapitre trois. La réalisation de cette application est enfin décrite dans le cinquième chapitre.

Chapitre 1

Android



Dans ce chapitre nous allons introduire le système d'exploitation Android, son architecture puis les composantes d'une application Android ainsi la plateforme Android.

1. Introduction :

Les applications mobiles sont actuellement en pleine croissance grâce aux succès de Smartphones entre autres ceux qui tournent sous Android OS. Cette tendance offre aux entreprises de formidables opportunités pour créer de nouveaux services ou élargir ces supports de communication.

2. Définition :

Android est un système d'exploitation Open Source pour Smartphones, PDA¹ et autres terminaux mobiles, conçu par Android, une startup rachetée par Google en juillet 2005. Cet OS se différencie principalement de ses concurrents par le fait qu'il est ouvert. Le modèle économique de Google semble très pertinent, l'adoption d'Android par les fabricants sera probablement rapide du fait de la gratuité d'utilisation pour le constructeur.



Figure 1.1 : logo d'Android

3. Historique des versions :

Dans l'ensemble, les différentes versions d'Android ont toutes des noms de desserts (en anglais), qui sont sculptés et affichés devant le siège social de Google (Mountain

¹**PDA** : (*Personal Digital Assistant*) est un ordinateur de poche composé d'un processeur, de mémoire vive, d'un écran tactile et de fonctionnalités réseau dans un boîtier compact d'extrêmement petite taille.

View) et ça, depuis la sortie de la version 1.5 et suivent une logique alphabétique (de A vers Z)

Au départ, il existait deux variantes de la plateforme Android. Une de ces variantes est dédiée aux petits écrans, principalement les téléphones mobiles (toutes les versions en dessous de 3.0), et l'autre variante est dédiée pour les tablettes : Honeycomb ou Android 3.0. En octobre 2011, la fusion de ces deux variantes, pour avoir une plateforme plus versatile et uniforme, a donné naissance à la version Android 4, comme sous le nom de « Ice Cream Sandwich ». C'est la première version qui combine « Gingerbread » et « Honeycomb » pour une plateforme, à la fois, pour les tablettes et les téléphones.

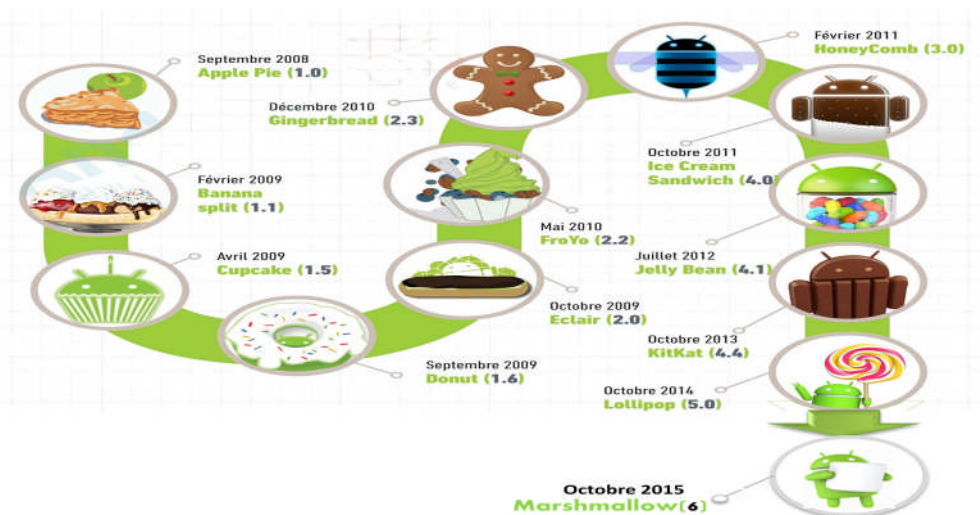


Figure1.2: les versions d'android

4. Architecture de la plateforme Android :

L'architecture de la plateforme d'android se décline, selon une démarche bottom, en quatre principaux niveaux que sont :

- ❖ Le noyau Linux
- ❖ Les bibliothèques et l'environnement d'exécution
- ❖ Le module de développement d'applications
- ❖ Les différentes applications

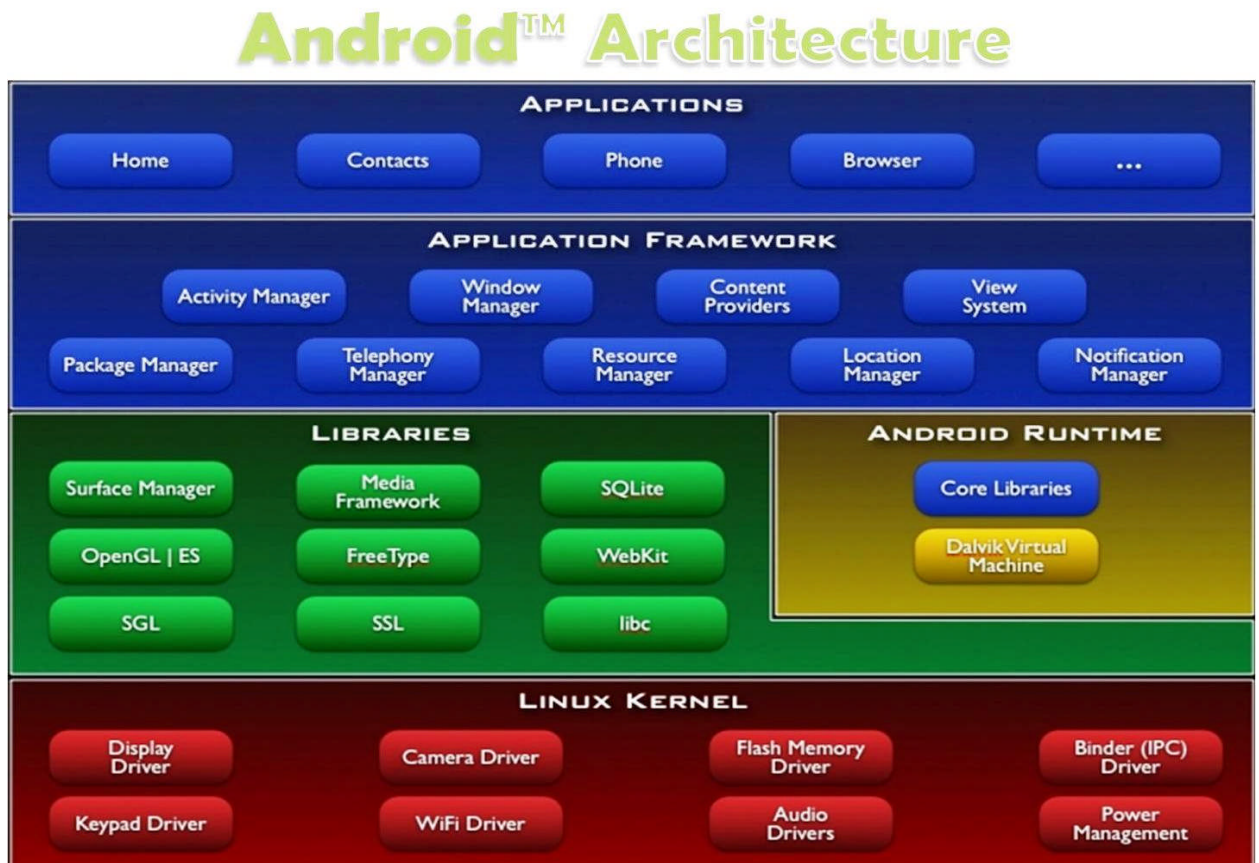


Figure 1.3: Architecture d'android

4.1. Premier niveau : noyau linux

Android s'appuie sur un noyau (kernel en anglais) linux 2.6. Pour être précis, le noyau est l'élément du système d'exploitation qui permet de faire le pont entre la partie matérielle et la partie logicielle. D'ailleurs, si vous regardez attentivement le schéma, vous remarquerez que cette couche est la seule qui gère le matériel. La version du noyau utilisée avec android est une version conçue spécialement pour l'environnement mobile, avec une gestion avancée de la batterie et une gestion particulière de mémoire.

4.2. Deuxième niveau : Librairies et environnement d'exécution

4.2.1. Les librairies :

Ces bibliothèques proviennent de beaucoup de projets open-sources, écrits en C/C++ pour la plupart, comme SQLite pour les bases de données, WebKit pour la navigation web ou encore OpenGL, afin de produire des graphismes en 2D ou en 3D.

Malgré que le développement d'application se fasse en langage JAVA, Android comprend très bien le C et le C++

4.2.2. L'environnement d'exécution :

Le runtime android est basé sur le concept de machine virtuelle, utilisée en java. Étant donné les limitations des dispositifs (peu de mémoire et la vitesse du processeur), il n'a pas été possible d'utiliser une machine virtuelle java standard. Google a pris la décision de créer une nouvelle machine virtuelle dalvik, afin de mieux répondre à ces limitations. Dans cette partie qui est l'environnement d'exécution, on retrouve essentiellement :

4.2.2.1. core libraries :

Les core libraries fournissent le langage JAVA, disponible pour les applications. Le langage JAVA fourni avec ANDROID reprend en grande partie l'API JSE 1.5 il y a des choses qui ont été mises de côté, car cela n'avait pas de sens pour Android (comme les imprimantes, swing, etc.) et d'autres APIs spécifiques requises pour Android ont été ajoutées.

4.2.2.2 Dalvik :

Les applications java, développées pour Android, doivent être compilées au format dalvik exécutable (.dex) avec l'outil (dx). Cet outil compile les (.java) en (.class) et ensuite il convertit ces (.class) en (.dex).

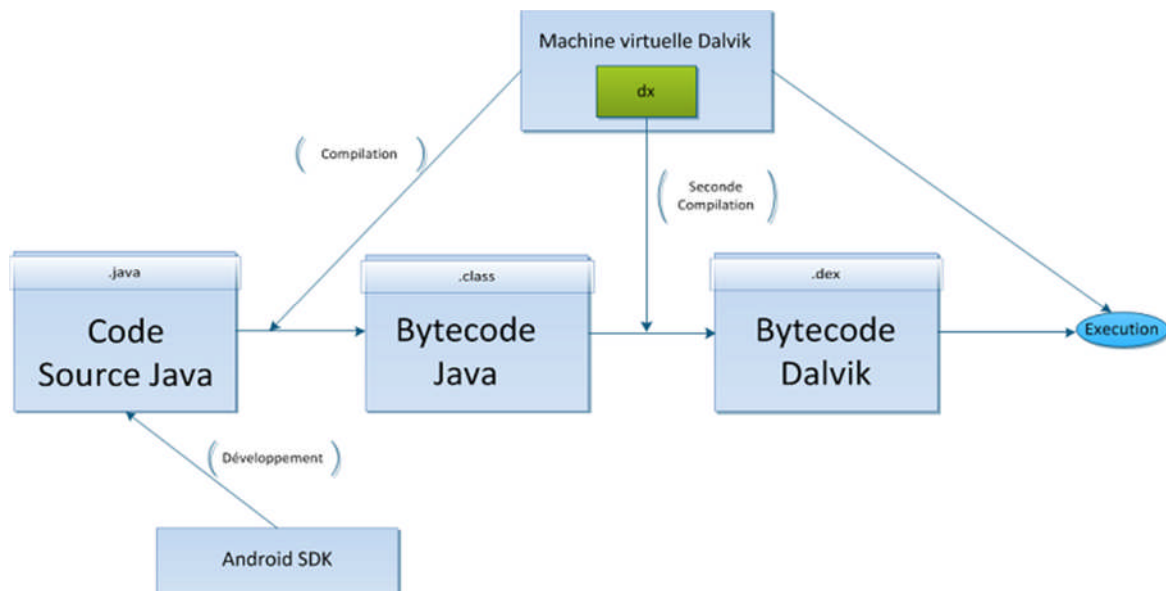


Figure 1.4 : la machine virtuelle Dalvik

4.3. Troisième niveau : module de développement d'applications

Le Framework est situé au dessus de l'android runtime et des librairies. Il fournit des API² permettant aux développeurs de créer des applications riches. A ce niveau on distingue deux types de service :

²API : Applications Programming Interface. Est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

4.3.1. core platform services :

Android introduit la notion de services. Un service est une application, qui n'a aucune interaction avec l'utilisateur et qui tourne en arrière plan. Les services essentiels au fonctionnement de la plateforme :

- ❖ Activity manager : gère le cycle de vie des applications et maintient une pile de navigation permettant d'aller d'une application à une autre ;
- ❖ Package manager : utilisé par l'Activity manager, pour charger les informations provenant des fichiers (.apk) (android package file) ;
- ❖ Window Manager : il gère les fenêtres des applications (quelle fenêtre doit être affichées devant une autre à l'écran) ;
- ❖ Resource Manager : gère tous ce qui n'est pas du code, toutes ressources (images, fichier audio, etc.) ;
- ❖ Content provider : gère le partage de données entre application.
- ❖ View système : fournit tous les composants graphiques : listes, grilles, bouton, etc.

4.3.2. Hardware services :

- ❖ Les services matériels (hardware services) fournissent un accès vers les API matérielles :
- ❖ Telephony service : permet d'accéder aux interfaces téléphoniques (GSM, 3G, etc.) ;
- ❖ Location Services : permet d'accéder au GPS ;
- ❖ Bluetooth Service : permet d'accéder à l'interface Bluetooth;
- ❖ Wifi Service : permet d'accéder aux interfaces WIFI;
- ❖ USB Service : permet d'accéder aux interfaces USB;
- ❖ Sensor Service : permet d'accéder aux capteurs (capteur de luminosité, de proximité, d'accélération, etc) .

4.4 Quatrième niveau : application

Il s'agit tout simplement d'un ensemble d'applications que l'on peut trouver sur Android, par exemple, les fonctionnalités de base incluent un client, pour recevoir/envoyer des e-mails, un programme pour envoyer/recevoir des SMS, un calendrier, un répertoire, etc.

5. Composantes d'une application Android :

Une application Android se compose de plusieurs éléments, dans ce qui suit, nous allons essayer de découvrir les plus importants

5.1. Les Activités :

Une activité est la composante principale pour une application Android. Elle représente l'implémentation et les interactions des interfaces

Plusieurs choix se proposent pour mettre en place l'interface visuelle :

- ❖ Utiliser un fichier XML pour décrire l'interface ;
- ❖ Créer les éléments de l'interface à l'intérieur du code java.

5.2. Les services :

Un service est, en fait, un programme tournant en tâche de fond et n'ayant pas d'interface graphique. L'exemple commun illustrant cette notion, est celui du lecteur mp3. Un lecteur mp3 ne nécessite pas, pour la plus part du temps, d'interface graphique et soit tourner en tâche de fond, laissant la possibilité aux autres applications de s'exécuter librement. Un service peut être lancé à différents moments :

- ❖ Au démarrage du téléphone ;
- ❖ Au moment d'un événement (arrivée d'un appel, SMS, mail, etc) ;
- ❖ Lancement de l'application ;
- ❖ Action particulière dans application.

5.3. Les broadcastReceivers

Un broadcastReceivers, comme son nom l'indique, permet d'écouter ce qui se passe sur le système ou sur votre application et de déclencher une action que vous aurez prédéfinie.

C'est souvent par ce mécanisme que les services sont lancés.

5.4. Les Content Providers :

Les ContentProvider sont, comme l'exprime leur nom, des gestionnaires de données .ils permettent de partager l'information entre applications.Vous pouvez accéder :

- ❖ Aux contacts stockés dans le téléphone ;
- ❖ A l'agenda ;
- ❖ Aux photos ;
- ❖ Ainsi que d'autres données depuis votre application grâce aux content providers.

5.5. Les intents :

Les intents sont des objets permettant de faire passer des messages contenant de l'information entre composants principaux. La notion d'intent peut être vue comme une demande de démarrage d'un autre composant, d'une action à effectuer

6. Cycle de vie d'une application Android :

Le diagramme d'état suivant présente les principaux états du cycle de vie d'une activité Android, il est suivi d'une description des principales méthodes événementiel du cycle de vie d'une activité.



figure1.5 : Cycle de vie d'une application Android.

7. développent sous Android

7.1. Environnement de développement sous Android :

Afin de développer des applications sous Android, un ensemble d'outils est nécessaire. Vue que les procédures d'installation de ces outils sont assez longue et fastidieuse, alors les décrire pas à pas, risquerai de prendre énormément de place ainsi beaucoup de pages. Alors, on se contentera juste d'évoquer les outils et leur intérêt.

7.1.1. Le JDK (java developpent kit)

Les applications développés pour Android étant essentiellement écrites en langage java ; un langage de programmation orienté objet qui a la particularité d'être très portable. Cela signifie qu'un programme java, fonctionnant sur Windows (par exemple), pour facilement tourner sur Mac ou GNU/Linux.

Cette petite prouesse vient du fait que java s'appuie une machine virtuelle pour s'exécuter (appelée la JVM). Pour avoir une JVM sur votre ordinateur, il vous faut télécharger le JRE. Ce dernier contient, en plus de la JVM, des bibliothèques java standards.

La JVM ne lit pas directement le code java. Elle lit un code compilé (le bytecode).

Pour passer du code java, que le développeur écrit, au code compilé, lu par la JVM, des outils spéciaux sont nécessaires. Ces outils sont inclus dans le JDK De plus, le JDK contient le JRE (et donc la machine virtuelle), ce qui est bien pratique

Pour résumer, on dira que :

- ❖ Pour un simple utilisateur de java : il doit avoir la JRE.
- ❖ Pour un développeur : il aura besoin des outils du JDK

7.1.2. Le SDK (Software Développment Kit) Android :

Un SDK, c'est-a-dire un Kit de développement logiciel, est un ensemble d'outils que met à disposition un éditeur afin d permettre de développer des applications pour un environnement précis. Le SDK Android permet, donc, de développer des applications pour android et uniquement pour Android.

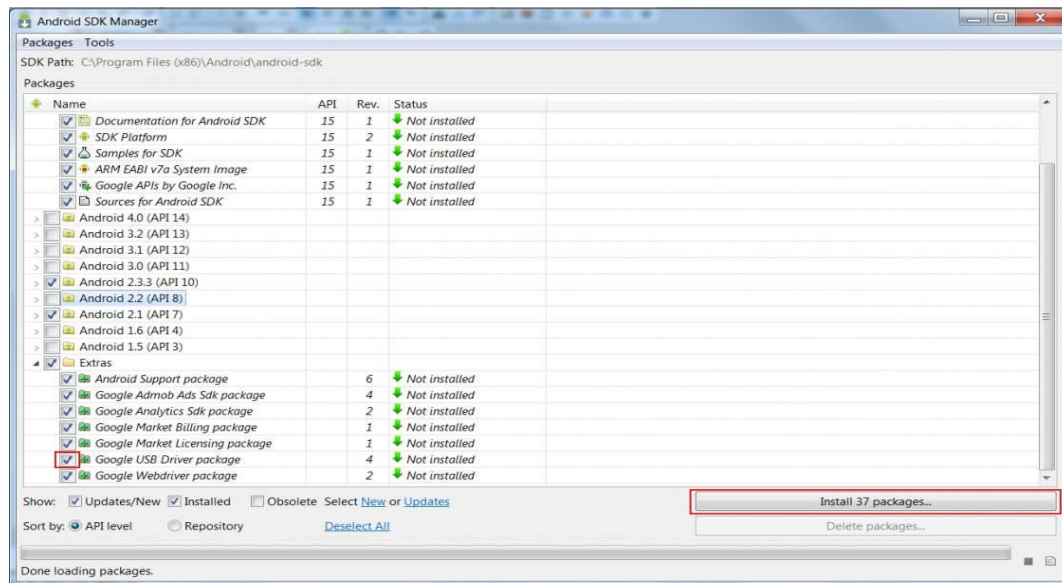


Figure 1.6 : Le SDK Android

7.1.3. L'IDE Eclipse :

Eclipse est un environnement de développement intégré. C'est un logiciel qui permet d'écrire un programme beaucoup plus facilement qu'avec le simple Bloc-notes. Outre la coloration du code, il permet d'apporter des outils très pratiques pour compiler vos programmes, les déboguer, etc. Il peut être utilisé pour programmer avec n'importe quel type de langage, mais nous l'utiliserons pour faire de la java.

De plus, Eclipse est conçu pour pouvoir être complété avec des plugins (extension).

Ainsi, il existe un plugin pour développer des applications Android que nous verrons dans partie suivante.

7.1.4. Le plugin ADT pour Eclipse :

Google fournit un plugin pour Eclipse, nommé ADT (Android Développement Tools). La fonction principale de ce plugin est créer un pont entre Eclipse et le SDK Android.

7.1.5. L'émulateur de téléphone : Android Virtuel Device

L'Android virtuel Device, aussi appelé AVD, est un émulateur de terminal sous Android, c'est-à-dire que c'est un logiciel qui fait croire à votre ordinateur qu'il est un appareil sous Android. C'est la raison pour laquelle vous n'avez pas besoin d'un périphérique sous Android pour développer et tester la plupart de vos applications. En effet, une application qui affiche un calendrier par exemple peut être bien se tester dans un émulateur, mais une application qui exploite le GPS doit être éprouvée sur le terrain pour que l'on soit certain de son comportement.

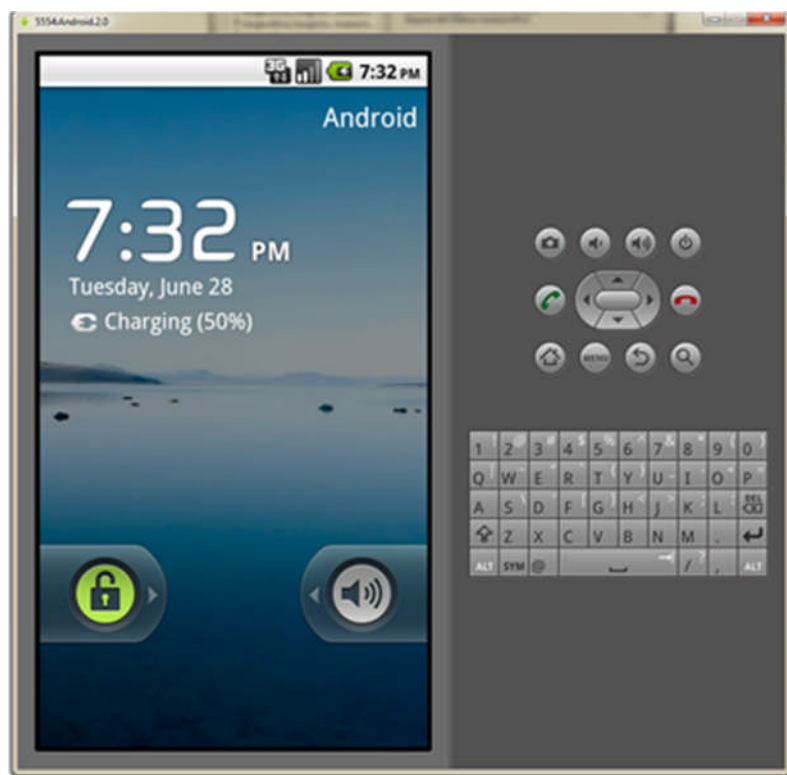


Figure1.7 : L'émulateur

7.2. La structure d'un projet :

Le système de construction d'un programme Android est organisé sous la forme d'une arborescence de répertoires spécifiques à un projet, exactement comme n'importe quel projet java. Les détails, cependant, sont spécifiques à un projet, exactement comme n'importe quel projet Java. Les détails, cependant, sont spécifiques à Android.

Tout comme bon nombre de technologies actuelles, les sources d'une application Android possèdent une structure bien définie qui doit être respectée. Ces arborescences permettent non seulement de rendre les projets plus lisibles et organisés, mais aussi de simplifier le développement. Lors de la création d'un nouveau projet, voici l'arborescence qui est automatiquement générée :

- ❖ Src : répertoire contenant l'ensemble des sources du projet. Il contient les classes, de type activités, qui gèrent entre autre le cycle de vie, as aussi les classes permettant de piloter les différentes fonctions de l'application ;
- ❖ Gen : répertoire contenant l'ensemble des fichiers générés par le plugin correspondant à l'environnement de développement. Aucune modification ne doit être faites dans ces fichiers ;
- ❖ androidManifest.xml : fichier XML décrivant l'application et ses composants, tels que les activités, les services, etc. Lors de la création d'une activité, une erreur courante pour un premier projet Android est d'oublier de la déclarer dans le fichier Manifest ;
- ❖ C'est une étape indispensable pour le fonctionnement de l'application. Le Manifest est, en quelque sorte, la carte d'identité de l'application.
- ❖ Res : répertoire contenant toutes les ressources telles que les images, les vues de l'interface graphique, etc., nécessaire à l'application. Ce répertoire est structuré par défaut de la manière suivante :
 - ❖ Res/drawable : contient les ressources de type image ;
 - ❖ Res/layout : contient les descriptions des interfaces graphiques au format XML (les vues)
 - ❖ Res/xml : contient les fichiers XML supplémentaires (non présent par défaut) ;
 - ❖ Res/menu : contient la description des menus, composant très courants d'une vue ;
 - ❖ Res/values contient diverses ressources, telles que les textes, qui sont empaquetées sans aucun traitement.
- ❖ Au moment de la compilation du projet, l'application finale est générée au format APK, dans le répertoire bin de l'arborescence. C'est ce fichier qu'il faut ensuite déployer sur les équipements, afin de pouvoir faire tourner l'application.

8. Conclusion

Dans ce premier chapitre nous avons essayé de lever le voile sur cette nouvelle race de terminaux en nous basant sur les fonctionnalités que nous pouvons trouvées sur ces système d'exploitation et l'architecture d'Android, puis la plate forme Android .

Chapitre 2

Géolocalisation



Ce chapitre a pour but d'introduire la Géolocalisation. Une attention particulière est portée aux différentes techniques de Géolocalisation, à sa plateforme logicielle ainsi que la carte collaboratif OpenStreetMap.

1. Introduction :

De nos jours on entend par localisation géographique, le positionnement sur un système de référence une personne ou un objet, par exemple un véhicule. Ce système de référence n'est ni plus ni moins qu'un système cartographique : une carte routière, ou un plan de ville.

Actuellement, toute automobile, tout bateau, tout avion, etc... , qui voudrait se repérer géographiquement, utilisera, entre autres le G.P.S¹. qui est le système le plus répandu à l'heure actuelle.

Les appareils utilisés pour la localisation doivent être muni d'une puce GPS qui permet de récupérer les données satellites et de les traiter afin de positionner l'individu ou l'objet sur une carte donnée.

2. Géolocalisation : [1]

La géolocalisation (géoréférencement) est un procédé permettant de positionner un objet ou une personne sur un plan ou une carte à l'aide de ses coordonnées géographiques.

Cette opération est réalisée à l'aide d'un terminal capable d'être localisé (grâce à un système de positionnement par satellites ou d'autres techniques) et de publier ses coordonnées géographiques. Les positions enregistrées peuvent être stockées au sein du terminal ou être transmises en temps réel vers une plateforme logicielle de géolocalisation. La transmission temps réel nécessite que le terminal soit équipé d'un moyen de télécommunication de type GSM², GPRS³, UMTS⁴ radio ou satellite lui permettant d'envoyer les positions à des intervalles réguliers. Ceci permet de visualiser la position du terminal au sein d'une carte à travers une plateforme de géolocalisation le plus souvent accessible depuis internet.

¹ GPS: Système Global de Positionnement.

² GSM: Global System for Mobile.

³ GPRS: General Packet Radio service.

⁴ UMTS: Universal Mobile Telecommunication System.



Figure 2.1 : principe de la géolocalisation par GPS.

2.1. Techniques de géolocalisation :

Différents techniques de géolocalisation existent, selon le niveau de précision, le coût, la couverture réseau ou le droit d'accès. Dans ce qui suit nous présenterons quelques unes de ces techniques.

2.1.1. Géolocalisation par GSM (Global System for Mobile communication) [Neviere, 2001]

Cette technique permet le positionnement d'un terminal GSM en se basant sur certaines informations relatives aux antennes GSM auxquelles le terminal est connecté. Positionnement par GSM peut aller de 200 mètres à plusieurs kilomètres, selon que le terminal se trouve en milieu urbain (où la densité d'antennes est supérieure), ou en milieu rural.

Plusieurs techniques existent :

- ❖ Différence de temps observée ou EOTD (enhanced-observed time difference) : le terminal calcule le temps écoulé entre l'émission et la réception de la requête envoyée à l'antenne, il peut alors calculer sa distance par rapport à celle-ci.
- ❖ Temps d'arrivée (time of arrival)
- ❖ Angle d'arrivée (angle of arrival)
- ❖ Cell ID (identifiant de cellule)

Aujourd'hui, la méthode GSM la plus utilisée est celle du Cell ID (identification de la cellule radio¹⁰), c'est la technologie la moins coûteuse car il n'y a pas de matériel à mettre en place. Du moment que le portable est dans une zone couverte par le réseau, il se connecte à une antenne relais GSM. Puis le portable est localisé grâce à une base de données faisant le lien entre les identifiants des cellules et les positions géographiques des antennes. Cette localisation est très rapide (moins de 5 secondes) mais elle est peu précise car elle dépend du nombre d'antennes relais et de leur rapprochement (plus l'antenne est isolée, plus la zone de couverture est vaste et moins la localisation est précise, la précision est de l'ordre de 50 à 500 mètres dans un milieu urbain alors qu'elle peut varier de 5 à 15Km dans les régions isolées).

Les bases de données peuvent être mises à disposition par les opérateurs pour leurs abonnés, ou par des sociétés privées qui recensent les antennes GSM ou ayant des partenariats avec les opérateurs.

2.1.2. Géolocalisation par WIFI:[2]

Les points d'accès WiFi, qui se trouvent dans la plupart des "box Internet", émettent en permanence des signaux permettant à des ordinateurs ou à des téléphones mobiles de les reconnaître et de s'y connecter. Ces signaux contiennent notamment un numéro d'identification unique propre à chaque point d'accès (appelé "BSSID⁵"), ce numéro est utilisé par les Smartphones pour géolocaliser une personne.

Fonctionnement de la géolocalisation à partir des points d'accès wifi :

Lorsqu'une personne lance une application de géolocalisation sur son Smartphone, celui-ci peut lister les points d'accès WiFi à sa portée, et interroger une base de données qui permet d'associer ce point d'accès à une position géographique. Le Smartphone va donc être capable de géolocaliser précisément le propriétaire du Smartphone.

Plusieurs sociétés telles que Google, Skyhook Wireless, Microsoft et Apple ont donc constitué des bases cartographiques recensant ces points d'accès WiFi pour fournir leurs services de géolocalisation.

Ces bases peuvent être créées de deux manières :

⁵ BSSID: Basic Set IDentifier

- ❖ à partir des données transmises par les téléphones mobiles eux-mêmes lorsqu'ils demandent à être géolocalisés.
- ❖ à partir des données WiFi collectées par des véhicules se déplaçant dans les rues des villes et sur les principaux axes routiers.

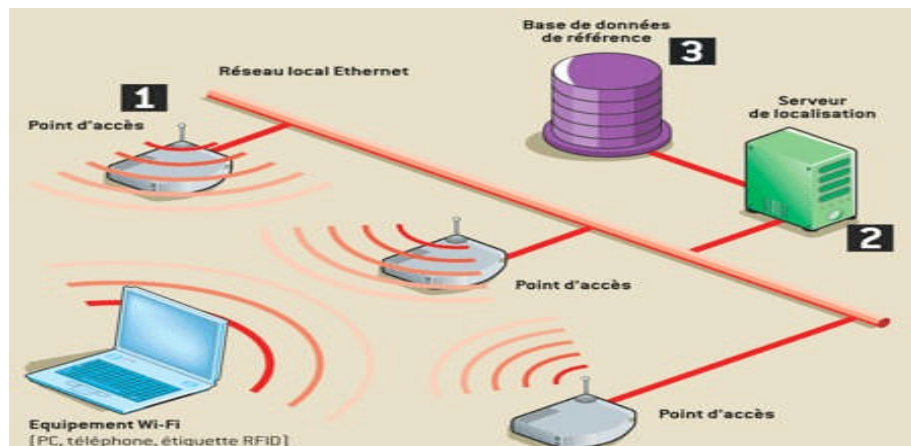


Figure 2.2: Géolocalisation par Wifi.

2.1.3. Géolocalisation par adresse IP⁶ (sur internet) : [3]

Cette méthode permet de déterminer la position géographique d'un ordinateur ou de n'importe quel terminal connecté à internet en se basant sur son adresse IP. Les adresses IP sont gérées par l'IANA⁷, une organisation qui s'occupe de découper les blocs d'adresses IP disponibles et de les distribuer de façon très contrôlée aux pays qui en demandent. Toutes ces attributions étant très bien documentées, il est possible de savoir dans quel pays se trouve un terminal connecté à internet grâce à son adresse IP. On peut même obtenir un niveau de précision de l'ordre de la ville en se basant sur la distribution des adresses IP faite par les fournisseurs d'accès à internet.

La précision de la géolocalisation par identification de l'adresse IP utilisée est cependant sans commune mesure avec celle permise par un GPS ou par triangulation des

⁶ IP: internet protocol.

⁷ IANA: Internet Assigned Numbers Authority.

téléphones mobiles. On considère généralement que les meilleurs référentiels d'adresses IP sont globalement fiables à l'échelle des villes.

2.1.4. Géolocalisation par RFID (Radio Frequency Identification)

Les technologies de RFID ont pour objectif d'identifier quelque chose à l'aide d'une étiquette passive réagissant à un signal de radio fréquence. Lors de cette sollicitation l'étiquette RFID va fournir de l'information sur l'objet. Elle peut aussi être capable d'enregistrer de l'information. Le RFID va ainsi permettre de déterminer quel objet ou produit traverse un point de passage, où se trouvent les capteurs RFID, mais aussi d'indiquer dans l'étiquette de l'objet qu'il est passé par tel point de passage. L'intérêt est évident, en particulier pour toutes les chaînes logistiques. On parle parfois de RFID passif, pour illustrer le fait que l'étiquette RFID ne fait que réagir, mais n'est pas capable par elle-même d'émettre.

2.1.5. Géolocalisation par GPS :

Le Global Positioning System (GPS) est un système de navigation par satellite qui a été développé par l'US Department of Defense (DoD) au début des années 1970. Initialement, le GPS a été développé comme un système militaire pour répondre aux besoins des militaires américains. Toutefois, il a fini par être autorisé aux civils, et maintenant c'est un système à double usage qui peut être consulté par les utilisateurs civils et militaires [Kleusberg, 1990].

2.1.5.1. Présentation :

GPS fournit un positionnement continu et des informations de synchronisation, n'importe où dans le monde dans toutes les conditions météorologiques. Parce qu'il sert un nombre illimité d'utilisateurs ainsi que d'être utilisé pour des raisons de sécurité, le GPS est un système à sens unique (passif) [Phillips, 1996]. L'utilisateur peut seulement recevoir les signaux.

Mis en place par le Département de la Défense des États-Unis. Il est très rapidement apparu que des signaux transmis par les satellites pouvaient être librement reçus et exploités, et qu'ainsi un récepteur pouvait connaître sa position sur la surface de la Terre,

avec une précision sans précédent, dès l'instant qu'il était équipé des circuits électroniques et du logiciel nécessaires au traitement des informations reçues. Une personne munie de ce récepteur peut ainsi se localiser et s'orienter sur terre, sur mer, dans l'air ou dans l'espace au voisinage proche de la Terre.

2.1.5.2. Historique :

En 1960 l'armée américaine (à la demande du président Richard Nixon) a lancée un projet de recherche sur le GPS dont la réalisation a été confiée à Ivan A. Getting qui a conçu le principe d'un groupe de satellites gravitant en orbite et émettant des ondes radio UHF captées par des récepteurs GPS.

En 1978, un premier satellite est lancé. En 1983, suite au crash d'un avion de la compagnie Korean Airlines et la mort de ses 269 passagers, le président Ronald Reagan, propose que la technologie GPS soit disponible gratuitement aux civils. En 1995, le déploiement des 24 satellites opérationnels (plus 4 en réserve) est achevé. Le système devient alors opérationnel en permanence sur l'ensemble de la planète, avec une précision limitée à une centaine de mètres pour un usage civil. En 2000, le président Bill Clinton confirme l'intérêt de la technologie à des fins civiles et autorise une diffusion non restreinte des signaux GPS, permettant une précision d'une dizaine de mètres.

2.1.5.3. Fonctionnement:[1]

Le GPS comprend au moins 24 satellites tournant à 20 200 km d'altitude. Ces satellites émettent en permanence un ou plusieurs codes pseudo-aléatoires, datés précisément grâce à leur horloge atomique¹², et par un message de navigation. Ce message, transmis à 50 bit/s, inclut en particulier les éphémérides¹³ permettant le calcul de la position des satellites, ainsi que des informations sur leur horloge interne.

Les codes sont un code C/A (acronyme de « coarse acquisition », en français : « acquisition brute ») de débit 1,023 Mbit/s et de période 1 ms, et un code P (pour « Précis ») de débit 10,23 Mbit/s et de période 1 semaine. Le premier est librement accessible, le second est réservé aux utilisateurs autorisés car il est le plus souvent chiffré : on parle alors de code Y.

Ainsi, un récepteur GPS qui capte les signaux d'au moins quatre satellites équipés de plusieurs horloges atomiques peut, en calculant les temps de propagation de ces signaux entre les satellites et lui, connaître sa distance par rapport à ceux-ci et, par trilatération, situer précisément en trois dimensions n'importe quel point placé en visibilité des satellites GPS, avec une précision de 3 à 50 mètres pour le système standard. Dans certains cas, seuls trois satellites peuvent suffire. La localisation en altitude (axe des Z) n'est pas d'emblée correcte alors que la longitude et la latitude (axe des X et des Y) sont encore bonnes. On peut donc se contenter de trois satellites lorsque l'on évolue au-dessus d'une surface « plane » (océan, mer).

Étant un système développé pour les militaires américains, une disponibilité sélective a été prévue certaines informations, en particulier celles concernant l'horloge des satellites, peuvent être volontairement dégradées et priver les récepteurs qui ne disposent pas des codes correspondants de la précision maximale. Pendant quelques années, les civils n'avaient ainsi accès qu'à une faible précision (environ 100 m). Le 1er mai 2000, le président Bill Clinton a annoncé qu'il mettait fin à cette dégradation volontaire du service.

2.2. Combinaison de différentes techniques :

Il existe plusieurs inconvénients à l'utilisation d'une seule technique de géolocalisation :

- ❖ **La dépendance au réseau GPS** : l'incapacité de l'utiliser en intérieur et le temps de réponse à l'allumage.
- ❖ **La dépendance au réseau GSM** : sa couverture géographique, l'accès au réseau GPRS pour exploiter l'information.
- ❖ **La dépendance à la présence de bornes d'accès WiFi** : en zone rurale par exemple.

Des dispositifs qui combinent ces trois techniques et qui sont capables de géolocaliser le terminal dans n'importe quelle situation existent. La précision de ce positionnement va varier en fonction des technologies disponibles, mais le temps de réponse à l'allumage et l'adaptabilité s'en verront améliorées. Ceci permet par exemple de géolocaliser une personne à l'extérieur en utilisant le GPS, et de garder sa trace à l'intérieur des bâtiments ou des tunnels en utilisant la technologie GSM couplée au WiFi pour plus de précision.

2.3. Télérèlevé d'information :

Elle consiste à récupérer à distance une série d'informations issues de capteurs ou de systèmes informatiques, électroniques ou électriques. La géolocalisation est très souvent couplée à des systèmes de télérèlevé via des boîtiers télématiques, ce qui permet de combiner la position géographique d'un terminal ou d'un véhicule à une série d'informations annexes relatives à l'objet géolocalisé.

Dans un véhicule par exemple, ces boîtiers peuvent se connecter au chronotachygraphe⁸ (pour le transport routier) ou à divers capteurs ou voyants, ce qui permet de relever des informations telles que :

- ❖ la vitesse du véhicule
- ❖ les kilomètres parcourus
- ❖ l'état d'une porte (ouverte/fermée)
- ❖ l'état d'une remorque (accrochée/décrochée)
- ❖ la température (pour les camions frigorifiques)

2.4. Le système de coordonnées locales

Le système de coordonnées utilisé par le système GPS (ECEF⁹) est sans réel intérêt pour l'utilisateur. Une conversion dans un système plus pratique est nécessaire.

Le système le plus approprié est le système E, N, U (East, North, Up) qui s'exprime en général par :

- ❖ La latitude Φ
- ❖ La longitude λ
- ❖ L'altitude h

N est la distance mesurée le long de la droite localement perpendiculaire à l'ellipsoïde entre l'observateur et l'intersection avec l'axe polaire

⁸ **Le chronotachygraphe:** est un appareil électronique enregistreur de vitesse, de temps de conduite et d'activités (conduite, travail, disponibilité et repos) installé dans un véhicule de transport routier.

⁹ ECEF : earth-centered, earth-fixed.

Il faut également tenir compte que la terre n'est pas une sphère mais un ellipsoïde dont les demi grands axes s'expriment par les valeurs a et b.

Le système ENU le plus couramment utilisé est le système WGS84 (World Geodetic Survey 1984).

2.5. Plateformes logicielles de géolocalisation :

❖ Composant :

Les principaux composants d'une plateforme de géolocalisation sont les suivant :

- **Terminal communicant** : C'est le terminal qui reçoit ses coordonnées géographiques (via GPS ou tout autre moyen) et qui les envoie via un réseau de télécommunications à la plateforme ou les utilise localement si la plateforme est hébergée au niveau local (c'est le terminal lui-même qui s'occupe du traitement de l'information).
- **System informatique capable de recevoir, stocker et traiter l'information** : il s'agit des serveurs qui hébergent l'infrastructure et qui s'occupent de la réception des données (envoyées par les terminaux dans le cas où la plateforme logiciel est hébergée sur un site distant), leur traitement ainsi que leur mise à disposition des utilisateurs.
- **Module cartographique** : c'est le module qui va permettre d'afficher la position des terminaux sur un fond cartographique adapté. Il prend en charge les calculs de distances, d'itinéraires, détecte l'interaction avec les zones et permet d'avoir accès à des informations terrain (sens interdits, restrictions pour les poids lourds, vitesses autorisées...).

❖ Architecture temps réel :

La donnée générée par un terminal sur le terrain est une information brute qui peut être exploitée et couplée à d'autres données, pour pouvoir l'exploiter elle doit être transmise à une plateforme logicielle qui va la traiter, la présenter graphiquement à l'utilisateur et l'associer à d'autres données afin d'enrichir les informations relatives à l'état du terminal ou de la flotte de terminaux.

Voici quelques étapes de la chaîne de traitement :

- Le terminal détermine sa position géographique grâce à une des technique de géolocalisation (de préférence GPS);
- Il envoie ces données vers une plateforme logicielle (soit par le réseau GSM/GPRS soit par un réseau satellitaire) ou les stocke au niveau de la plateforme locale (dans le cas où elle est installée au niveau local) ;
- La plateforme logicielle de géolocalisation traite la donnée et positionne le terminal géographiquement sur une carte moyennant la précision offerte par la technique de géolocalisation utilisée. De plus, en combinant plusieurs informations, notamment récupérées via un système de télérelève (trafic routier, autonomie du véhicule, points à visiter etc...), des calculs d'itinéraires ou de tournées peuvent par exemple être générés.
- Cette carte ainsi que tous les traitements effectués sont mis à disposition l'utilisateur à travers un portail web hébergé sur un serveur accessible depuis internet, ou à travers une application métier installée sur le terminal.



Figure 2.3 : architecture d'un système de géolocalisation par GPS.

❖ Fonctionnalités :

Voici une liste de fonctions typiquement offertes par les plateformes de géolocalisation professionnelles :

Suivi en temps réel de terminaux.

- Affichage d'un historique de déplacements.
- Création de points d'intérêt.
- Configuration d'alertes automatiques par courriel ou SMS sur des événements.
- Localisation des itinéraires ainsi que le chemin le plus court entre deux points distants.
- Génération de rapports périodiques (temps de conduite, arrêts, vitesses moyennes, zones couvertes...).
- Fonds cartographiques variables (cartes classiques, cartographie photographique, cartes de fonds marins, cartes provenant d'un SIG etc).

3. OSM (Open Street Map)[4]:

Open Street Map (OSM) est une initiative privée à but non lucratif visant à produire et partager des cartes collaboratives vectorielles en libre-diffusion du monde et en particulier du réseau routier. Partie d'Angleterre, OSM a désormais largement franchi les frontières de ce pays et la qualité des données s'est améliorée de manière spectaculaire à partir de 2008. Grâce à des outils de saisie très évolués (et open source) et à l'effort des bénévoles, les données disponibles sont désormais assez complètes.

3.1. Projet et Outils connus pour la cartographie collaborative:

Il y a différents moyens de faire une carte collaborative. Le plus simple est de s'asseoir avec les autres et d'utiliser un stylo et du papier. Bien qu'étant collaborative dans son processus de création, la technique papier-stylo ne permet pas facilement la collaboration au delà du groupe réduit qui peut se réunir autour de la feuille de papier. Même si la feuille est très grande, il y a une limite physique au nombre de personnes pouvant travailler en même temps sur celle-ci. Open Street Map permet à des groupes de collaborer sur des cartes électroniques, ce qui augmente considérablement ce nombre. Mais ce ne sont pas les seules options. Outre le couple papier-stylo et les outils entourant Open Street Map, il existe d'autres façons de créer et partager des cartes collaboratives.

3.1.1. Google Map Maker et Google My Maps:[5]

Google Map Maker est un moyen de corriger et d'ajouter des informations à Google Maps et Google Earth. Google permet par exemple d'ajouter les routes manquantes ou incorrectes, ou des points d'intérêt. Les modifications apportées par les nouveaux utilisateurs passent par un processus de vérification avant d'être publiées.

Google My Maps est quand à lui un onglet associé à Google Maps qui permet d'enregistrer des informations dans la carte personnelle d'un utilisateur. Celui-ci peut alors décider rendre sa carte publique, privée, ou seulement partagée avec certaines personnes. Il lui est ainsi possible d'inviter des personnes précises à collaborer sur sa carte ou de rendre sa carte disponible au monde entier pour que chacun puisse y contribuer.

Si l'utilisation de Google Maps est gratuite, les données ne sont pas libres pour autant. Contrairement à Open Street Map, Google interdit des usages commerciaux. Il restreint aussi l'usage de ses cartes pour des services qui entreraient en compétition avec Google. Selon les conditions d'utilisation du projet, votre contribution devient ainsi un produit dérivé de Google. La compagnie se réserve en effet les usages commerciaux des données. Google s'appuie sur l'aide de visiteurs pour améliorer sa cartographie, sans pour autant en faire un bien public

3.1.2. Ushahidi: [6]

Ushahidi est un projet à but non commercial et un logiciel open-source permettant la collecte d'informations collaboratives et l'affichage de ces informations sur une carte et un calendrier. Ushahidi est un mot Swahili qui veut dire témoin ou témoignage. Son but initial était de créer des cartes des cas de violence déclarés au cours des troubles postélectorales au Kenya en 2007 -2008. Le site permettait notamment de collecter les témoignages de violence envoyés par email et SMS, en les plaçant sur une carte.

Aujourd'hui, on pourrait le présenter comme une initiative de cartographie sociale, combinaison d'activisme social, de journalisme citoyen et d'information géographique. D'un point de vue logiciel, Ushahidi fournit un mécanisme à un observateur local pour soumettre un rapport via son téléphone mobile ou internet, avec le lieu et l'heure de l'évènement. L'idée d'Ushahidi est de permettre aux utilisateurs de soumettre des « rapports » directement via l'application Ushahidi, à travers les réseaux sociaux, ou même par SMS. Il est possible de rapidement mettre en place une version d'Ushahidi en utilisant

CrowdMap ou en téléchargeant le code source et en l'installant sur un serveur. Le logiciel est ainsi utilisé par plusieurs projets de cartographie collaborative.

3.1.3.WikiMapia[7]:

WikiMapia est un projet commercial qui offre « une carte interactive modifiable dont le but est de créer et de maintenir à jour une carte gratuite, complète et multilingue du monde entier.» WikiMapia est surtout utilisée pour référencer des points d'intérêts rassemblés collectivement. Ce projet utilise les vues satellitaires de Google Maps et permet de les annoter avec un système de style "Wiki". Ainsi, WikiMapia fournit un support de carte modifiable par tout le monde, sans qu'il soit nécessaire d'avoir un compte pour consulter ou contribuer au projet.

À la différence d'OpenStreetMap qui a expressément obtenu de Microsoft le droit d'utiliser son imagerie aérienne, WikiMapia propose un travail dérivé de sources propriétaires (Google Earth). Par ailleurs, le travail contributif n'est pas placé sous licence libre, mais sous la licence Creative Commons BY-NC-SA1 (paternité, non commercial et partage des conditions à l'identique). On doit comprendre que le projet se réserve les usages commerciaux des données. WikiMapia génère des revenus avec des publicités Google et la plateforme reste la propriété privée de ses fondateurs.

3.2. Mise en œuvre d'Open Street Map :

Traditionnellement, les données cartographiques sont collectées et tenues à jour par des organisations à but lucratif, qu'il s'agisse d'éditeurs privés ou d'entreprises publiques. Bien souvent, l'utilisation de ces cartes est restreinte, et dans certains cas, quand il n'est pas rentable de créer ou de maintenir des cartes, aucune de ces organisations n'édite de cartes.

OpenStreetMap a été créé pour combler ces manques, c'est une carte collaborative à l'échelle mondiale, ouverte, librement modifiable (disponible sous licence ODbL). À la manière de Wikipédia, tous les internautes naviguant dans le web peuvent contribuer à la création et à la numérisation de cartes. Des éditeurs permettent de réaliser en ligne des cartes en se basant sur un fond d'image satellitaire. Cependant, ces images satellitaires ne couvrent pas toujours en haute résolution l'ensemble du globe. C'est pourquoi il est

possible d'introduire des données provenant de récepteurs GPS. Il suffit de réaliser un itinéraire et de positionner le récepteur GPS en mode enregistrement, puis de le restituer sur le serveur de données d'OpenStreetMap situé au Royaume-Uni et géré par la fondation OpenStreetMap. Cela a pour avantage de les rendre visibles dans les outils d'édition des cartes et facilite la couverture internationale : une personne séjournant dans une autre région ou un autre pays que le sien peut publier les tracés de ses parcours, à charge pour les habitants permanents de les compléter.

Les points d'intérêts (POI, en anglais « point of interest »), c'est-à-dire, toutes les mentions utiles (noms, largeur, nature du revêtement, sens uniques, parcs, zones résidentielles et d'activités, barrières, pistes cyclables, boîtes aux lettres, cabines téléphoniques, commerces, fontaines, etc.) sont notés, soit en les écrivant, soit en les photographiant, soit en les décrivant sur un appareil d'enregistrement audio. Tous les modes de locomotion terrestre possibles sont utilisés : à pied, à deux-roues, à rollers, à skis, en véhicule automobile particulier, en bus, en train...

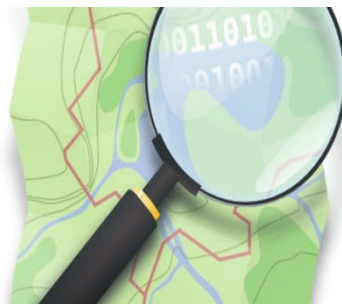


Figure2.4 : logo d'OpenStreetMap.

4. Conclusion :

Dans le présent chapitre nous avons illustré quelques principes sur les différentes techniques de géolocalisation et les technologies associées, puis nous avons vu une brève introduction à la cartographie en exposant le modèle de données d'OpenStreetMap.

Chapitre 3

Optimisation par Colonie de Fourmis



Ce chapitre introduit la méthode d'Optimisation par Colonie de Fourmis (ACO). Nous évoquons le comportement des fourmis réelles, celui des fourmis artificielles et les principaux algorithmes utilisés dans la littérature.

1. Introduction

Les recherches sur le comportement collectif des insectes sociaux fournissent aux informaticiens des méthodes puissantes pour la conception d'algorithmes d'optimisation combinatoire et de contrôle distribué. Ainsi, proposé par Marco Dorigo en 1991, le premier algorithme à base de colonies de fourmis est utilisé comme une approche multi-agents pour la résolution du problème du voyageur de commerce (PVC)[DOR96(a)].

Cet algorithme est fortement inspiré du comportement collectif des fourmis réelles. Ces dernières sont capables de trouver le plus court chemin entre leur fourmilière et une source de nourriture. Les fourmis partagent donc un savoir commun qui permet d'indiquer aux autres fourmis des directions à suivre ou à éviter. En effet, à chaque fois qu'une fourmi explore un chemin, bon ou mauvais, elle enrichit la connaissance collective de la colonie via le dépôt de phéromone. Cependant, une seule fourmi n'est, souvent, pas capable de trouver ce court chemin : il faut toute une colonie. Il s'agit d'un exemple vivant de système auto-organisé.

Aujourd'hui, l'idée originale s'est diversifiée pour résoudre une classe plus large de problèmes. Ainsi, plusieurs algorithmes ont vu le jour, s'inspirant du comportement des fourmis réelles.

2. Les fourmis réelles

Les fourmis réelles sont capables de trouver le plus court chemin reliant leur fourmilière et une source de nourriture, malgré le fait qu'elles soient aveugles [IBR03]. En effet, elles sont en mesure d'établir cette route écourtée grâce, exclusivement, à une communication indirecte appelée « *stigmergie* ». Ainsi, sur tout le long de leurs chemins, allant de leur nid jusqu'à la source de nourriture, les fourmis déposent une substance chimique, sécrétée par une glande située sous leurs abdomens, appelée « *phéromone* ». Dès lors, leurs congénères, étant très sensibles à cette dernière, s'y appuient pour déterminer lequel, parmi les chemins déjà visités par les autres fourmis, est le plus intéressant. Ainsi, si un chemin se trouve fortement phéromonné, cela signifie

que plusieurs fourmis l'ont jugé avantageux et que leurs congénères devront le considérer avec intérêt. Toutefois, la phéromone s'évapore avec le temps et permet donc l'exploration d'autres chemins probablement meilleurs.

Cependant, il est important de faire remarquer que, bien qu'une fourmi soit capable de trouver la plus courte voie reliant son nid à la source de nourriture, c'est exclusivement le comportement de l'ensemble de la colonie qui détermine, à coup quasiment sûr, le chemin le plus court.

2.1. Les caractéristiques d'une colonie de fourmis

Une colonie de fourmis est caractérisée par :

- ❖ **Le parallélisme** : dans une colonie de fourmis, plusieurs fourmis travaillent en même
- ❖ Temps pour trouver le plus court chemin reliant la fourmilière et la source de nourriture.
- ❖ **La flexibilité** : une colonie de fourmis est capable de s'adapter aux modifications de son environnement (elle capable de détourner des obstacles par exemple).
- ❖ **La robustesse** : une colonie de fourmis continue toujours à travailler même si certains de ses individus sont défaillants.
- ❖ **La décentralisation** : une colonie de fourmis n'obéit pas à une autorité centralisée (c'est-à-dire qu'il n'existe pas de hiérarchie entre les fourmis).
- ❖ **L'auto-organisation** : une colonie de fourmis est capable de s'auto-organiser (s'organiser seule) pour trouver d'elle-même le plus court chemin qui n'est préalablement, pas connu.

2.2. Expérience de l'obstacle [DOR04]

La figure suivante montre un cas simple pour illustrer le comportement des fourmis réelles.

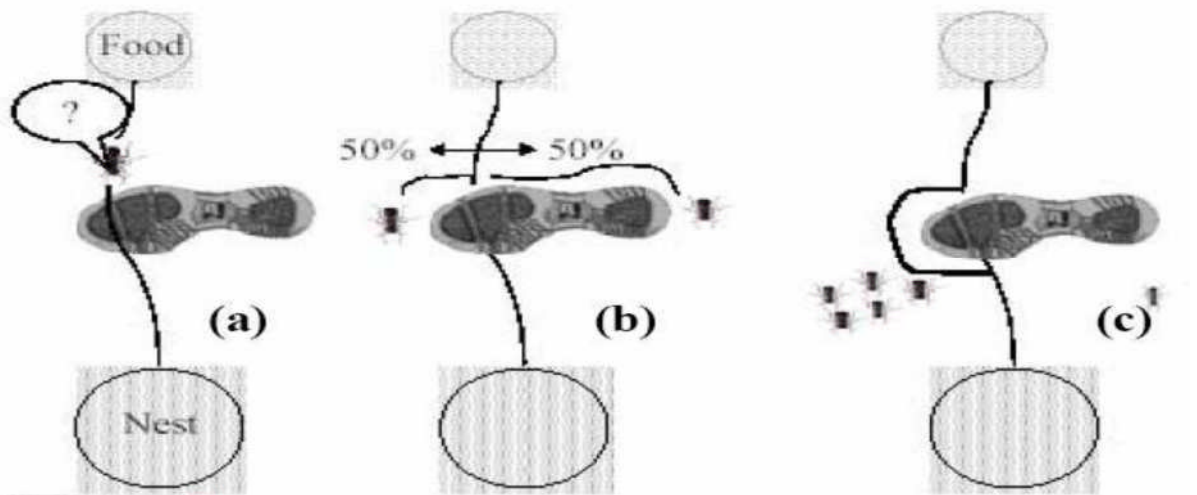


Figure 3.1 : Expérience de l'obstacle [DRÉ01].

Avant de mettre l'obstacle, les fourmis suivaient le plus court chemin entre le nid et la source de nourriture, (a) un obstacle est mis au début, les fourmis se partagent en deux chemins aléatoirement, (b) il est évident que les fourmis qui ont pris le plus court chemin vont atteindre le nid plus rapidement, et dans leurs retour déposent encore de la phéromone, (c) après une durée transitoire toutes les fourmis suivent ce chemin [LAA05].

2.3. Expérience à double branches

Pour plus de confirmation des résultats de la première expérience, GOSS en 1989 [DRE03] a fait une expérience similaire à la précédente avec un double pont de longueurs différentes.

Cet obstacle à deux branches est mis entre la fourmilière et la source de nourriture pour avoir plusieurs chemins de longueur différente donc mieux observer le comportement des fourmis (figure 3.2).

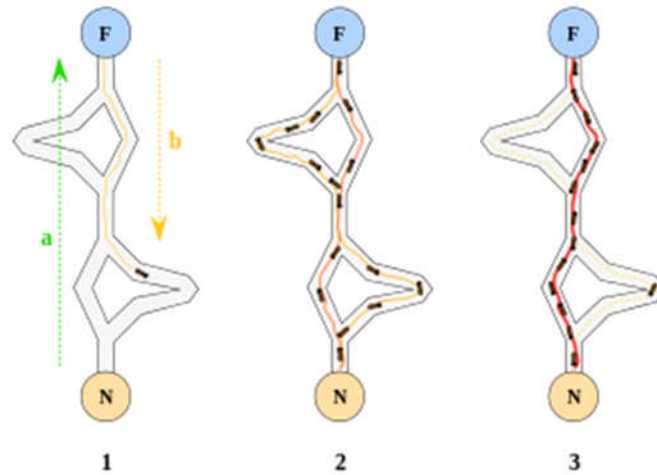


Figure 3.2 : Expérience à double branches [DRE03].

Au début, les fourmis s'intéressent à arriver à la nourriture à travers les deux branches de façon aléatoire, les fourmis qui prenaient le plus court chemin arrivent rapidement au nid et au retour elles trouvent leur chemin grâce à la phéromone donc la concentration de ce produit chimique va augmenter ce qui va mener les autres fourmis de suivre ce plus court chemin.

3. Les fourmis artificielles

Etant donné que les fourmis réelles résolvent efficacement le problème du plus court chemin, les chercheurs ont pensé à transposer ces mécanismes pour la résolution de problèmes relevant de la recherche opérationnelle. Ainsi, sont nées les fourmis artificielles.

Cependant, les fourmis artificielles ont une double nature. D'une part, elles modélisent le comportement abstrait des fourmis réelles. D'une autre part elles peuvent être enrichies par des capacités supplémentaires pour plus de performance. Ci-dessous, sont présentées les similarités et différences [DOR04] entre ces deux types de fourmis.

3.1 Similarités entre les fourmis réelles et artificielles

Les fourmis artificielles ont les caractéristiques et capacités des fourmis réelles dont :

- ❖ **La coopérativité et la concurrence** : les fourmis réelles coopèrent de façon asynchrone et concurrente pour trouver le chemin le plus court entre la fourmilière et la source de nourriture. Alors que, les fourmis artificielles collaborent dans le but de trouver une solution optimale au problème d'optimisation considéré.
- ❖ **La phéromone comme moyen de communication** : quand les fourmis (réelles et artificielles) explorent un chemin, elles laissent tout au long de ce chemin une quantité de phéromone. Dès lors, les autres fourmis pourront s'y appuyer pour choisir le bon chemin à suivre.
- ❖ **L'évaporation de la phéromone** : après un certain temps, la phéromone s'évapore favorisant ainsi l'exploration et la recherche d'autres chemins, peut être, meilleurs.
- ❖ **L'utilisation d'une décision probabiliste** : lorsqu'il s'agit de choisir le chemin à suivre et qu'aucun des chemins candidats n'est préalablement exploré, les fourmis (réelles et artificielles) choisissent aléatoirement une route candidate.

3.2. Différences entre les fourmis réelles et artificielles

Les caractéristiques, énumérées ci-dessous, sont exclusivement propres aux fourmis artificielles.

- ❖ Les fourmis artificielles ne sont pas totalement aveugles puisqu'elles sont dotées d'une mémoire leur permettant de garder trace des actions préalablement accomplies.
- ❖ Les fourmis artificielles déposent une quantité de phéromone proportionnelle à la qualité de la solution trouvée.
- ❖ Les fourmis artificielles ont la faculté de ne mettre à jour les pistes de phéromones qu'une fois la solution trouvée.
- ❖ Les pistes de phéromone s'évaporent plus rapidement que chez les fourmis réelles.

- ❖ La probabilité pour qu'une fourmi choisisse une destination ne dépend pas exclusivement des traces de phéromones mais aussi d'une heuristique qui évaluent localement la qualité du chemin.
- ❖ Les fourmis artificielles sont dotées de capacités supplémentaires comme l'anticipation, la recherche locale et le retour arrière.

4. L'optimisation par colonies de fourmis (ACO)

La méta-heuristique ACO est constituée d'une étape d'initialisation et trois procédures, ces trois procédures nécessitent une certaine synchronisation qui est accomplie par le Scheduler-des-activités. Selon DORIGO, le pseudo code de l'ACO est :

```
Set parameters, initialize pheromone trails
Schedule_Activities
ConstructAntSolutions
DaemonActions {optional}
UpdatePheromones
End_Schedule_Activities
```

Figure 3.3 : Le pseudo-code de la méta-heuristique ACO [DOR07].

La procédure `Schedule_Activities` est répétée jusqu'à ce que le critère d'arrêt est satisfait[DOR07]. Ces trois composantes sont plus détaillées dans ce qui suit.

4.1. Représentation du problème

Les ACO sont souvent représentés par un graphe $G=(C, L)$, où les nœuds (sommets) sont les composants de C et l'ensemble fini de connexions L qui relie ces composants. Les contraintes du problème considéré sont implémentées par un ensemble fini Ω qui définit les règles de déplacement des fourmis artificielles [DRE03], ainsi nous avons :

- $C = \{c_1, c_2, \dots, c_n\}$.
- $L = \{l_{ij} / (c_i, c_j) \in (C \times C)\}$ où l_{ij} représente le passage du nœud i vers le nœud j .
- J_{ij} est le coût associé à la connexion l_{ij} .
- $\Omega \equiv \Omega\{C, L, t\}$ est défini sur les éléments de C et L à l'instant t .
- Etat du problème $S = \langle c_i, c_j, \dots, c_k, \dots \rangle$ est une séquence d'éléments de C , si S_1 et S_2
- sont deux états S_2 est voisin de S_1 si on peut atteindre S_2 à partir de S_1 par un seul
- pas logique ($S_2 = \langle S_1, c_i \rangle$), NS dénote l'ensemble de voisins de S .
- Si S est l'ensemble de toutes les séquences possibles, on pose S' le sous ensemble de S qui contient toutes les séquences réalisables.
- $\Psi \in S'$ est une solution au problème si elle satisfait toutes les contraintes Ω .
- $J\Psi(t)$ est le coût associé à la solution Ψ qui est en fonction des coûts $J_{ij}(t)$.
- Un sommet i possède une « liste des candidats » contenant l'ensemble de ses sommets favoris (les sommets les plus proches).

Une solution au problème consiste à trouver dans ce graphe, un chemin réalisable avec un coût minimal et qui respecte les contraintes du problème, les fourmis sont conçues pour effectuer cette recherche.

Durant le processus de recherche les fourmis charge des informations en phéromone artificielle, qui peut être associée soit au sommet soit à la connexion l_{ij} et que l'on stocke sa valeur dans la variable τ_{ij} , les informations phéromones peuvent être enrichies par des valeurs heuristiques η_{ij} spécifiques au problème traité.

4.2. Les propriétés des fourmis

Chaque fourmi K de la colonie possède les propriétés suivantes [DOR04]:

- ❖ Elle exploite le graphe G afin de trouver une solution optimale avec $J\Psi = \min J\Psi(L, t)$.

- ❖ Elle a une mémoire M^k (une liste **Tabou** des sommets déjà visités), elle peut être utilisée pour construire une solution réalisable, évaluer une solution trouvée et retracer le chemin parcouru.
- ❖ Une fourmi k dans un état $S_r = \langle S_{r-1}, i \rangle$ se déplace vers le nœud j de son voisinage N_i^k en utilisant **une règle de construction probabiliste (transition)** qui s'élabore en fonction de:
 - ❖ l'information phéromone τ_{ij} et de la valeur heuristique η_{ij} sauvegardée dans la structure de données des nœuds dénommée : **table de routage**.
 - ❖ les contraintes du problème Ω .
 - ❖ la mémoire M^k .
- ❖ Elle commence le processus de recherche de la solution par l'état initial et s'arrête lorsque l'un des états finaux est atteint.
- ❖ **La mise à jour On-line step by step** : lorsqu'une fourmi k se déplace d'un nœud i à nœud j où $j \in N_i^k$, elle met à jour la quantité de phéromone $\Delta\tau_{ij}^k$, c'est ce qu'on appelle « *la mise à jour étape par étape* ».
- ❖ **La mise à jour On-line delayed** : une fois la solution construite, la fourmi peut retracer le même chemin pour mettre à jour la phéromone avec une quantité relative à la qualité de cette solution, c'est qu'on appelle « *la mise à jour retardée* ».
- ❖ Après cela, la fourmi meurt et sera supprimée du système.

4.3. Les caractéristiques de l'algorithme ACO

Les algorithmes de colonie de fourmis doivent prendre en considération plusieurs paramètres. Parmi les plus utiles, on peut citer :

4.3.1. Les pistes de phéromone

C'est le paramètre le plus important, il représente la définition de la signification projetée par la phéromone τ_{ij} , elle dépend fortement du problème à résoudre, le choix

intuitif donne de bon résultat pour beaucoup de problèmes comme PVC dont la phéromone fait référence à la désirabilité de visiter la ville j après i [DOR01].

4.3.2. L'utilisation de l'information heuristique

L'utilisation de l'information heuristique est importante du fait qu'elle donne la possibilité d'exploiter les connaissances spécifiques du problème à résoudre, ces informations peuvent être statiques (calculées à l'initialisation et inchangeable) tel est le cas de PVC ou dynamiques (calculées au cours de traitement) tel est le cas de MAX-SAT [DOR01].

4.3.3. Le nombre de fourmis de la colonie m

Des résultats ont montré que les algorithmes d'ACO sont plus performants lorsque $m > 1$, étant donné qu'une seule fourmi peut ne pas bien explorer l'espace de recherche [DRÉ03].

4.3.4. La liste candidate

Lorsqu'on traite un problème où le voisinage est trop grand, une liste candidate peut être utilisée par les fourmis pour limiter le nombre de voisins à considérer d'un état donné en se basant sur des connaissances retenues du problème, cette liste contient l'ensemble des nœuds préférés à visiter afin d'améliorer la performance de l'algorithme. Il a été montré par expérience que dans le cas de PVC où le nombre total de villes était 2392, une solution optimale peut être atteinte avec une liste de candidate de 8 villes [REI94].

4.3.5. L'exploration et l'exploitation de l'espace de recherche

L'utilisation de méta-heuristique est relative au balancement entre le processus de diversification et d'intensification de région de recherche, par intensification on perçoit l'exploitation de l'information reliée au problème à résoudre, par contre la diversification est une injection de solution aléatoire.

Dans les algorithmes de type ACO, la façon de gérer ces deux facteurs des régions de recherche se base sur l'influence de piste de phéromone τ_{ij} et de l'information heuristique η_{ij} via les paramètres α et β respectivement dans la règle de transition, la mise à jour de phéromone selon la qualité de problème fournit l'exploitation d'espace de recherche, cependant l'absence de l'information heuristique rend l'exploration plus haute initialement et diminuera peu à peu, ce qui produira une **stagnation** de la colonie (la stagnation est une situation où les fourmis suivent la même trajectoire et construisent la même solution) [DRÉ03].

4.3.6. La règle de transition d'état

Il faut définir la règle de déplacement des fourmis artificielles d'un nœud vers le nœud voisin, qui peut être :

- a) **Règle proportionnelle Aléatoire**: l'état suivant est choisi aléatoirement selon une distribution de probabilité dépendant de la quantité de phéromone τ_{ij} et de l'information heuristique η_{ij} [DRÉ03].
- b) **Règle proportionnelle pseudo aléatoire**: dépend d'un paramètre q_0 ($0 < q_0 < 1$), qui définit une balance exploration / exploitation (cf. §5.3) [DRÉ03, DOR07].

4.4. Les effets de l'algorithme ACO[HAM03, DOR07]

Pour réaliser une modélisation plus réelle et une implémentation plus performante, des effets supplémentaires à la théorie de base ont été introduits à l'algorithme ACO :

4.4.1. L'évaporation de Phéromone

L'intensité de la phéromone est décrétementée avec le temps, cela sert à rendre les chemins les plus visités de moins en moins attirant ce qui favorise l'exploration des nouvelles régions par les fourmis sinon l'utilisation de m fourmis sera inutile, donc il y

aura une plus grande probabilité d'amélioration de la solution, éviter la stagnation et ralentir la convergence.

Selon la règle :

$$\tau_{ij}(t+1) = (1-P)\tau_{ij}(t) + P \sum F(S) \text{ [DOR07]}$$

Où F est une fonction de fitness et $p \in [0,1]$, varie selon l'algorithme ACO.

4.4.2. Les actions de Démon (Daemon Actions)

Pour améliorer les performances de l'ACO, un composant appelé le Démon qui a une vue globale de l'état d'avancement de la recherche, par conséquent, il peut intensifier la recherche en rajoutant de l'extra phéromone sur le chemin emprunté par la fourmi ayant trouvé la meilleure solution afin de favoriser les nœuds de celui ci, ou diversifier la recherche

en la dirigeant vers des nouvelles régions [DOR07]. Cette mise à jour est appelée « **offline** ».

4.5. Activité des fourmis (ConstructAntSolutions) :

Chaque fourmi construit une solution en parcourant l'espace de recherche représenté sous forme d'un graphe, le comportement des fourmis est déterminé en définissant l'état de départ, les règles de transition, les règles de construction et les règles de mise à jour de phéromone.

PROCEDURE Activité_des_fourmis ()

Tant Que (ressources disponibles) **Faire**

Création_nouvelle_fourmi();

Nouvelle_fourmi_active();

Fin Tant Que

Fin.

PROCEDURE Nouvelle_fourmi_active() {cycle de vie d'une fourmi}

Début

Initialiser la fourmi ;

M = Mise à jour de la mémoire de la fourmi ;

Tant Que (état courant \neq but) **Faire**

{but signifie une solution complète construite par la fourmi}

A = Lire la table de routage locale de la fourmi;

P = Calculer la probabilité de transition (A, M, Ω);

Etat suivant = Appliquer la règle de décision (P, Ω);

Se déplacer vers le nouvel état (état suivant);

Si (Mise à jour online step by step de la phéromone) **Alors**

Déposer de la phéromone sur les arcs visités ;

Mettre à jour la table de routage ;

Fin Si

M = Mettre à jour l'état interne de la fourmi;

Fin Tant Que

Si (Mise à jour online delayed de la phéromone) **Alors**

Evaluer la solution ;

Déposer de la phéromone sur tous les arcs visités ;

Mettre à jour la table de routage ;

Fin Si

Tuer la fourmi;

Fin.

Figure 3.4 : Pseudo-Code d'Activité des Fourmis [GUE07].

Les fourmis artificielles adoptent deux stratégies pour concevoir une solution, la stratégie par construction et la stratégie par amélioration [HAM 05] :

4.5.1. Stratégie de construction

En démarrant d'une solution vide, chaque fourmi est positionnée initialement sur un nœud du graphe choisi aléatoirement. Une fourmi se déplace d'un sommet vers un autre en choisissant successivement les sommets à rajouter à la solution en cours de construction, jusqu'à l'arrivée à une solution complète. Le choix d'un sommet à prendre se fait selon la règle proportionnelle pseudo-aléatoire.

4.5.2. Stratégie par amélioration (voisinage)

Cette stratégie consiste à générer une solution initiale aléatoire, donc on démarre à partir d'une solution complète, puis on effectue des modifications sur celle-ci afin d'améliorer sa qualité.

4.6. Les algorithmes ACO

L'ACO est le paradigme des algorithmes à base de colonie de fourmis comme AS qui a été le premier algorithme ACO, et par la suite d'autres variantes sont apparues comme ACS.

4.6.1. Algorithme de base (Ant System)

Le système des fourmis AS est le premier algorithme ACO introduit par Marco Dorigo en 1992 pour résoudre le PVC.

AS était originaire des trois algorithmes Ant-Cycl, Ant-Density, Ant-Quantity [DOR92],

il est efficace pour les problèmes de petites tailles. On peut citer les caractéristiques suivantes d'AS :

a) La règle de transition

La règle de déplacement des fourmis du nœud i vers le nœud j, appelée la règle aléatoire de transition proportionnelle, est la suivante :

$$P_{ij}^K = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in N_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta} & \text{si } j \in N_i^k \\ 0 & \text{si } j \notin N_i^k \end{cases} \dots (6) \text{ [DRE03]}$$

Où :

N_i^k : l'ensemble des nœuds voisins faisables de la fourmi k.

τ_{ij} : La quantité de le phéromone déposée sur l'arc (i,j).

η_{ij} : La valeur de l'information heuristique τ .

α, β sont deux paramètres de contrôles de balance exploration/ exploitation.

b) La mise à jour globale de la phéromone

La mise à jour locale n'est pas faite en AS, locale veut dire à chaque déplacement. La mise à jour globale est donnée par la formule suivante :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{L^k(t)} & \text{si } (i,j) \in T^K(t) \\ 0 & \text{sinon} \end{cases} \dots (7) \text{ [DRE03]}$$

Où :

$T^k(t)$ est la tournée faite par la fourmi k à l'itération t, $L_k(t)$ la longueur du trajet.

La quantité initiale de la phéromone sur les arcs est une distribution uniforme d'une petite quantité.

c) L'évaporation de la phéromone :

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \dots \dots (8) \text{ [DRE03]}$$

Où :

m : Le nombre de fourmis ayant emprunté l'arc (i, j) à l'itération t.

p : Le coefficient d'altération de la phéromone entre 0 et 1.

4.6.2. Ant system & élitisme

Une première variante a été proposée par Dorigo et al.96 par « l'introduction de fourmis élitistes », la fourmi qui a effectué le trajet le plus court dépose une quantité de phéromone plus grande, dans le but d'accroître la probabilité des autres fourmis d'explorer la solution la plus prometteuse [DRE03].

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i,j) \in T^k(t) \\ 0 & \text{sinon} \end{cases} \dots (9) \text{ [DRE03]}$$

Où Q un paramètre fixé.

On peut citer l'exemple suivant :

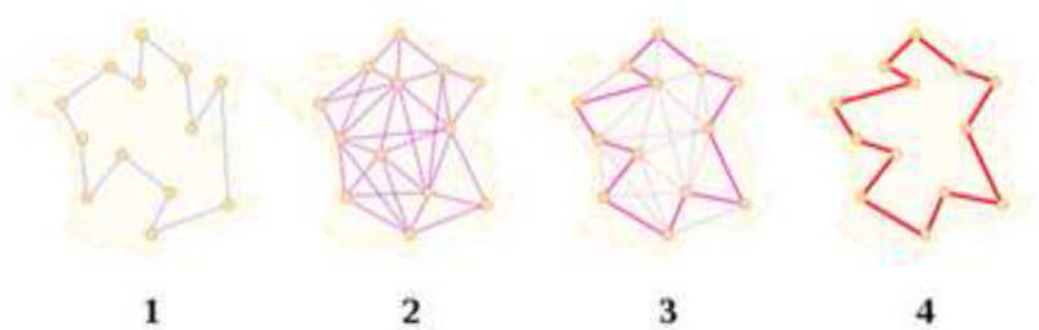


Figure 3.5 : PVC optimisé par l'algorithme Ant System [DRE03].

Les points représentent les villes et l'épaisseur des arcs la quantité de phéromone déposée,

- (1) ➔ Exemple de trajet construit par une fourmi.
- (2) ➔ Au début du calcul, tous les chemins sont explorés.
- (3) ➔ Le chemin le plus court est plus renforcé que les autres.
- (4) ➔ L'évaporation permet d'éliminer les moins bonnes solutions.

4.6.3. Max-Min Ant System [LAA05]

Cet algorithme est proposé par Stutzle et Hoos en 1997 comme une amélioration d'AS.

La différence se résume dans les points suivants :

- ✓ Seule la meilleure fourmi est autorisée à mettre à jour la phéromone.
- ✓ Afin d'éviter la stagnation la quantité de la phéromone est bornée entre T_{min}, et T_{max}.
- ✓ Les pistes des phéromones sont initialisées à leurs valeurs maximales.

4.6.4. Ant colony system (ACS)

Il a été introduit par Dorigo et Gambardella en 1997, pour améliorer les performances de l'algorithme AS auxquels ont ajoutés des fonctions qui permettent d'obtenir des bons résultats avec des jeux de données plus complexes, ce que ne permettait pas l'AS. L'idée principale de l'ACS est de modéliser le problème comme la recherche du meilleur chemin dans un graphe particulier [DOR07] Nous présentons, ci-dessous, ses particularités

a) La règle de transition

La première différence importante entre l'AS et l'ACS est la règle utilisée par les fourmis durant le processus de construction, appelée la règle pseudo aléatoire proportionnelle, dépendant d'un paramètre réglable q_0 ($0 \leq q_0 \leq 1$), qui définit une balance exploration/exploitation [BOU09, DRE03] :

$$j = \begin{cases} \underset{u \in N_i^k}{argmax} [\tau_{ui}(t) \cdot (\eta_{ij})^\beta] & \text{si } q \leq q_0 \\ S & \text{si } q > q_0 \end{cases} \dots (10) \text{ [DRE03]}$$

Où q est une variable aléatoire uniformément distribuée sur $[0,1]$, N_i^k l'ensemble des nœuds de voisin faisable de la fourmi k et $S \in N_i^k$ une ville sélectionnée aléatoirement selon la probabilité :

$$p_{is}^k(t) = \frac{(\tau_{is}(t)) \cdot (\eta_{is})^\beta}{\sum_{l \in N_i^k} (\tau_{il}(t)) \cdot (\eta_{il})^\beta} \dots (11) \text{ [DRE03]}$$

Il y a deux comportements possibles :

- Pour $q > q_0$: le système tend à effectuer une exploration (diversification), le choix se fait de la même façon que pour l'algorithme AS.
- Pour $q \leq q_0$: le système tend au contraire vers une exploitation, l'algorithme exploite plus d'informations récoltées par le système.

b) La mise à jour de phéromone :

ACS adapte une nouvelle stratégie de gestion de phéromone :

• **La mise à jour locale de phéromone** en ACS, la phase d'évaporation (on-line step by step) est remplacée par la mise à jour locale pour favoriser l'émergence d'autres trajets, la fourmi k se déplace d'un nœud i vers nœud $j \in$ et la mise à jour de la phéromone s'effectue selon la formule :

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t) + p \cdot \tau_0 \quad \dots (12) \text{ [DRE03]}$$

Où τ_0 est la valeur initiale de la piste du phéromone et $p \in [0,1]$.

• **La MAJ global de phéromone** : ici, seule la meilleure piste est mise à jour, ce qui participe à une exploitation par sélection de la meilleure solution. La mise à jour globale est faite par le démon d'action « offline » selon la formule suivante :

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t) + P \Delta \tau_{ij}(t) \quad \dots (13) \text{ [DRE03]}$$

Où les arcs (i, j) appartiennent au meilleur tour T^+ de longueur L^+ et $\Delta \tau_{ij} = 1/L^+$.

c) Liste de candidats :

l'ACS incorpore une structure de données avancée appelée liste candidat, la fourmi qui désire se déplacer vers un autre nœud choisit l'un des nœuds existant dans la liste.

d) Les stratégies de l'ACS : l'algorithme ACS possède deux stratégies :

- La stratégie constructive, illustrée dans la figure ci-dessous :

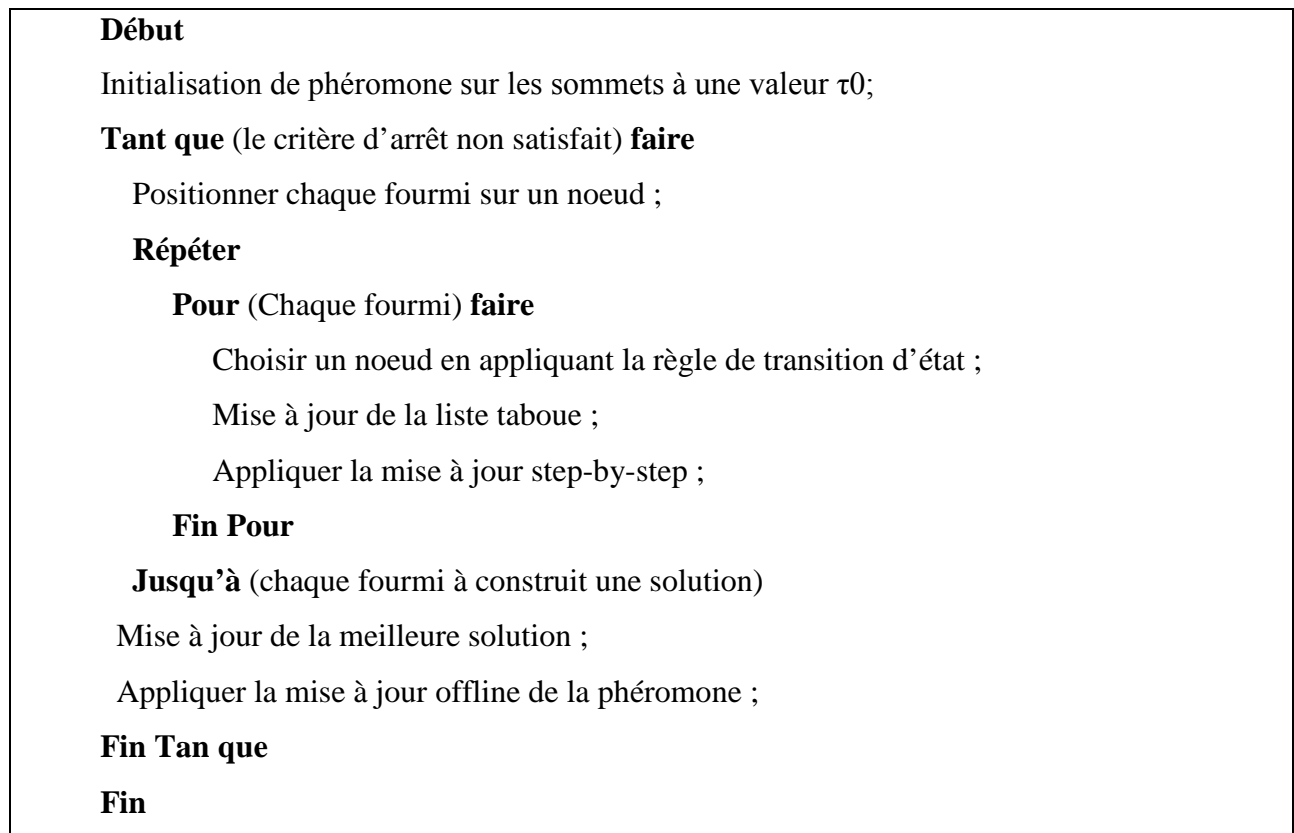


Figure 3.6 : La procédure ACS de la stratégie constructive [HAM05].

La stratégie améliorée (voisinage), montré en la figure suivante :

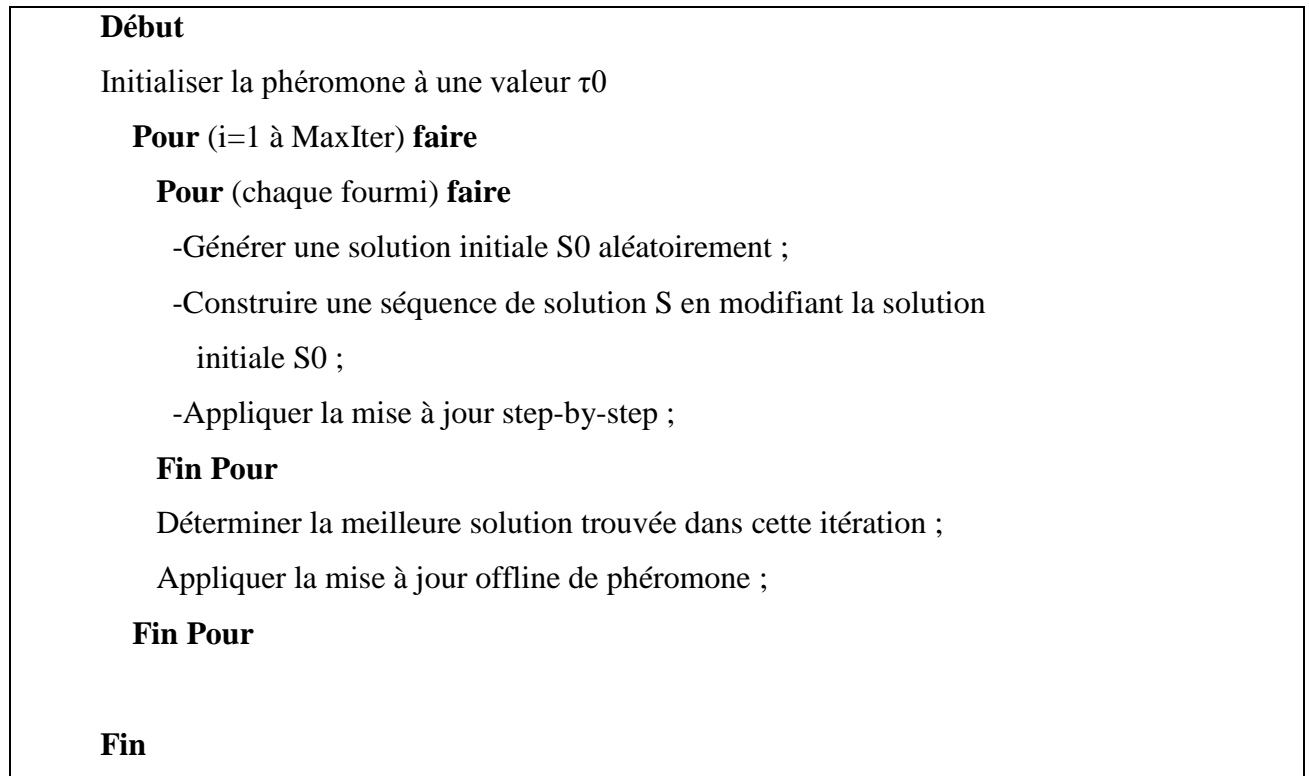


Figure 3.7 : La procédure ACS de la stratégie par amélioration [HAM05].

5. Le parallélisme des algorithmes ACO

Les fourmis d'une colonie ne prennent pas en compte les informations locales propres à son environnement, en effet le parallélisme est réalisable [DRE03]. On peut distinguer trois types de parallélisme [DOR01]:

- ❖ **Parallélisme fonctionnel** : les trois procédures de l'algorithme principal ACO (le pseudo code de méta-heuristique ACO) évaluent en même temps grâce à la synchronisation.
- ❖ **Parallélisme au niveau des fourmis** : où la colonie est subdivisée en sous colonies et l'algorithme est appliqué pour chacune en respectant la même structure du problème principal.
- ❖ **Parallélisme au niveau des données** : il divise le problème à résoudre en sous problèmes, chacun résolu par une colonie.

6. Conclusion

Dans ce chapitre, nous avons établi une analyse sur le comportement des fourmis réelles, et comment elles peuvent trouver le plus court chemin entre le nid et la source de nourriture.

Les ACO sont inspirés des fourmis réelles avec l'ajout de quelques paramètres comme la valeur de la méta-heuristique, le nombre de fourmis et les pistes de phéromone.

Nous constatons que l'algorithme des fourmis peut être appliqué à notre problème, car les route peut être représenté sous forme d'un graphe, et notre but est de trouver un chemin optimal.

Chapitre4

Conception

Dans ce chapitre, nous abordons rigoureusement la problématique, et proposons une solution d'optimisation par colonie de fourmis (ACO). Ainsi, les objectifs et le fonctionnement de notre application seront peaufinés dans leurs moindres détails.

1. Introduction :

L'objectif de notre travail est de réaliser une application sous android pour indiquer a conducteur le chemin ayant moins d'embouteillage par la méthode des colonies de fourmis, en nous inspirant de la population myrmécéenne¹, nous pourrions grandement améliorer la situation.



2. Hypothèses de travail :

Nous pouvons résumer les hypothèses du travail comme suit :

- Notre objectif est d'indiquer à un conducteur le chemin ayant moins d'embouteillage pour atteindre une certaine destination en tirant profit des informations transmises par d'autres conducteurs qui ont déjà parcouru ce chemin, les uns vont servir d'autre, en supposant que tous les conducteurs utilisent l'application android. Le système s'exécutera dans un environnement Web qui permettra de calculer ce chemin.
- Le calcul sera garanti par l'application de l'algorithme des colonies des fourmis.
- Ce dernier sera exécuté sur le graphe qui représente la carte routière pour trouver le meilleure Chemin qui sera présenté aux conducteurs comme étant le chemin optimal.
- Pour bénéficier de l'effort de la communauté, les temps de parcours précédents seront pris en compte dans le calcul de la phéromone, ce qui permettra d'obtenir de meilleurs résultats.

3. Description de la solution

La figure 4.1 illustre un exemple de cette représentation. Une route R est définie par une séquence de nœuds N allant d'un nœud source n_s et arrivant à un nœud de destination n_d . Nous avons représenté la carte routière par un graphe $G(V,A)$ où V est l'ensemble des nœuds (villes) et A l'ensemble des arcs (chemins), nous avons associé à chaque arc deux paramètres : le temps T et la distance D .

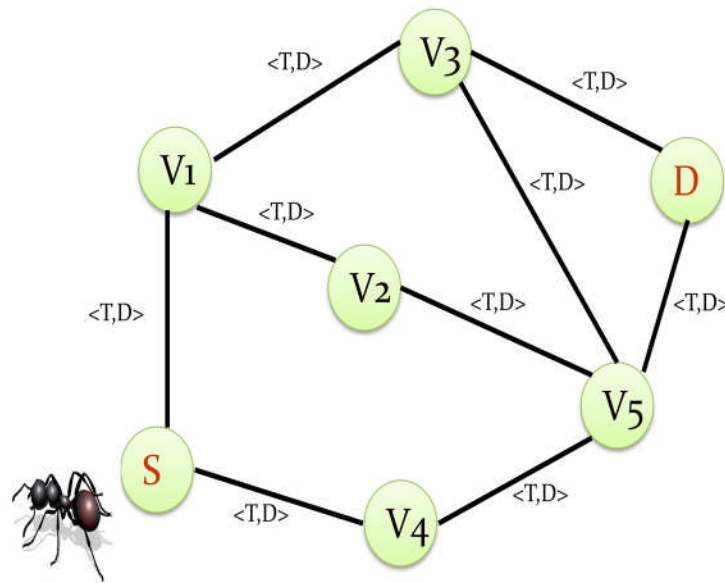


Figure 4.1. représentation d'une carte routière sous forme d'un graphe

3.1. Structures des tables :

Chaque nœud i du graphe maintient deux tables :

La table T est définie comme une matrice de phéromones. Chaque ligne dans la table correspond à une destination d et chaque colonne correspond à un voisin du nœud i . La table T_i contient un élément τ_{ijd} indique l'opportunité apprise pour une fourmi dans le nœud i avec destination d pour passer par le nœud j :

$$\sum_{j \in N_i} \tau_{ijd} = 1, \quad d \in [1, n]$$

✓ Où N_i est l'ensemble des voisins du nœud i

La table T_i :

τ_{i11}	τ_{i12}	τ_{i1n}
τ_{i21}	τ_{i22}	τ_{i2n}
....
τ_{iN_i1}	τ_{iN_i2}	τ_{iN_in}

Figure 4.2 : la matrice de phéromones $T_i = [\tau_{ijd}]$ au niveau du nœud i

La table M_i contient des statistiques sur la topologie de carte routière et sa charge vue par le nœud i .

Pour chaque destination d , M_i contient une fenêtre d'observation mobile W_{id} , un temps moyen μ_{id} estimé et une variance estimée σ_{id}^2 . La fenêtre d'observation mobile W_{id} de taille W_{\max} , représente un vecteur contenant les temps de parcours des dernières W_{\max} VAnts (fourmi qui simule le comportement d'un véhicule) qui voyagent du nœud i à la destination d .

La fenêtre d'observation mobile W_{id} est utilisée pour stocker la valeur $t_{\text{best}_{id}}$ qui représente une mémoire à court terme exprimant une estimation du temps optimal pour aller du nœud courant i à la destination d . c.a.d. le meilleur temps de parcours obtenu par une VAnt, parmi les W_{\max} dernières.

μ_d et σ_d^2 représentent la moyenne et la variance du temps de parcours effectués par les VAnts pour se déplacer du nœud i au nœud destination d

$$\mu_{id} \leftarrow \mu_{id} + \zeta(t_{i \rightarrow d} - \mu_{id}) \quad (1)$$

$$\sigma_{id}^2 \leftarrow \sigma_{id}^2 + \zeta((t_{i \rightarrow d} - \mu_d)^2 - \sigma_d^2) \quad (2)$$

Où :

- $T_{i \rightarrow d}$ représente le dernier temps de parcours de la VAnt pour passer du nœud i au nœud destination d .
- $\zeta \in [0, 1]$ est un facteur. Di Caro et Dorigo associent ζ à la taille maximale de la fenêtre d'observation W_{\max} de la manière suivante :

$$W_{max} = \frac{5c}{\varsigma} \text{ where } c < 1 \quad (3)$$

$M_i =$

1	2	...	n
μ_1, σ_1^2, W_1	μ_2, σ_2^2, W_2	...	μ_n, σ_n^2, W_n

Figure 4.3 : La table de statistique $M_i(\mu_{id}, \sigma_{id}^2, W_{id})$

4. Description de l'algorithme Vants :

1. Chaque Vant choisit un nouveau nœud à visiter parmi ses voisins en se basant sur une probabilité dite probabilité de transition. cette probabilité de transition du nœud i vers un nœud j est en fonction d'une part, de la quantité de phéromone se trouvant sur l'arc (i, j) et d'autre part de la visibilité local, cette probabilité est défini par l'équation (3)

Calcul de la probabilité P

$$P_{ijd} = \frac{\tau_{ijd} + \alpha\eta_{ij}}{1 + \alpha(|N_i| - 1)} \quad (3)$$

Avec :

N_i : représente l'ensemble des voisins du nœud courant i , $|N_i|$ le cardinal de cet ensemble.

La valeur α doit être choisie dans l'intervalle $\in]0; 1[$.

τ_{ijd} : La quantité de phéromone déposée sur l'arc connectant les nœuds i et j .

η_{ijd} : est la qualité local d'un arc. Elle représente le facteur de visibilité de la fourmi. Il est donné par l'équation (4) :

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (4)$$

d_{ij} : est la distance entre le nœud i et j

2. Chaque fourmi a une mémoire interne (pile) où elle sauvegarde deux informations : le chemin qu'elle prend pour se rendre de s vers d , L'identifiant de chaque nœud visité k et le temps qui s'est écoulé depuis le lancement de la fourmi VAnt pour arriver à ce $k^{ième}$ nœud sont empilés dans cette mémoire ($S_{s \rightarrow d}$).

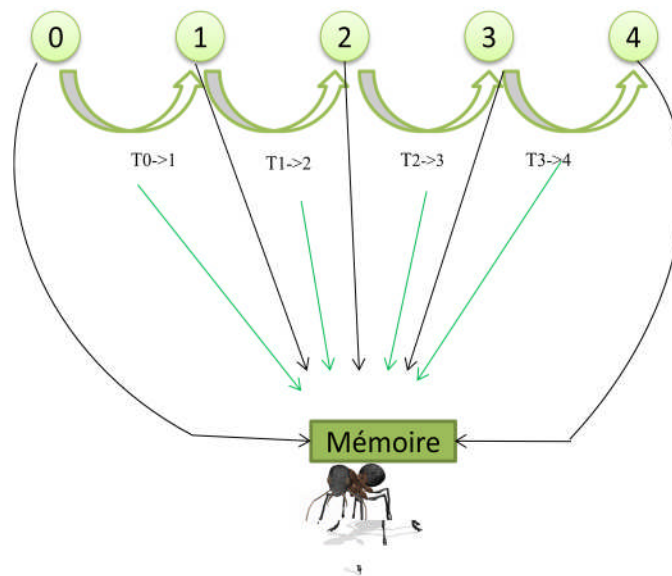


Figure4.4 : Déplacement d'une Vant

2.1 Si un cycle est détecté, c-à-d., si la fourmi est forcée de retourner vers un nœud déjà visité, les nœuds du cycle sont dépilés de la mémoire de la VAnt, comme illustré dans la Figure 4.5.

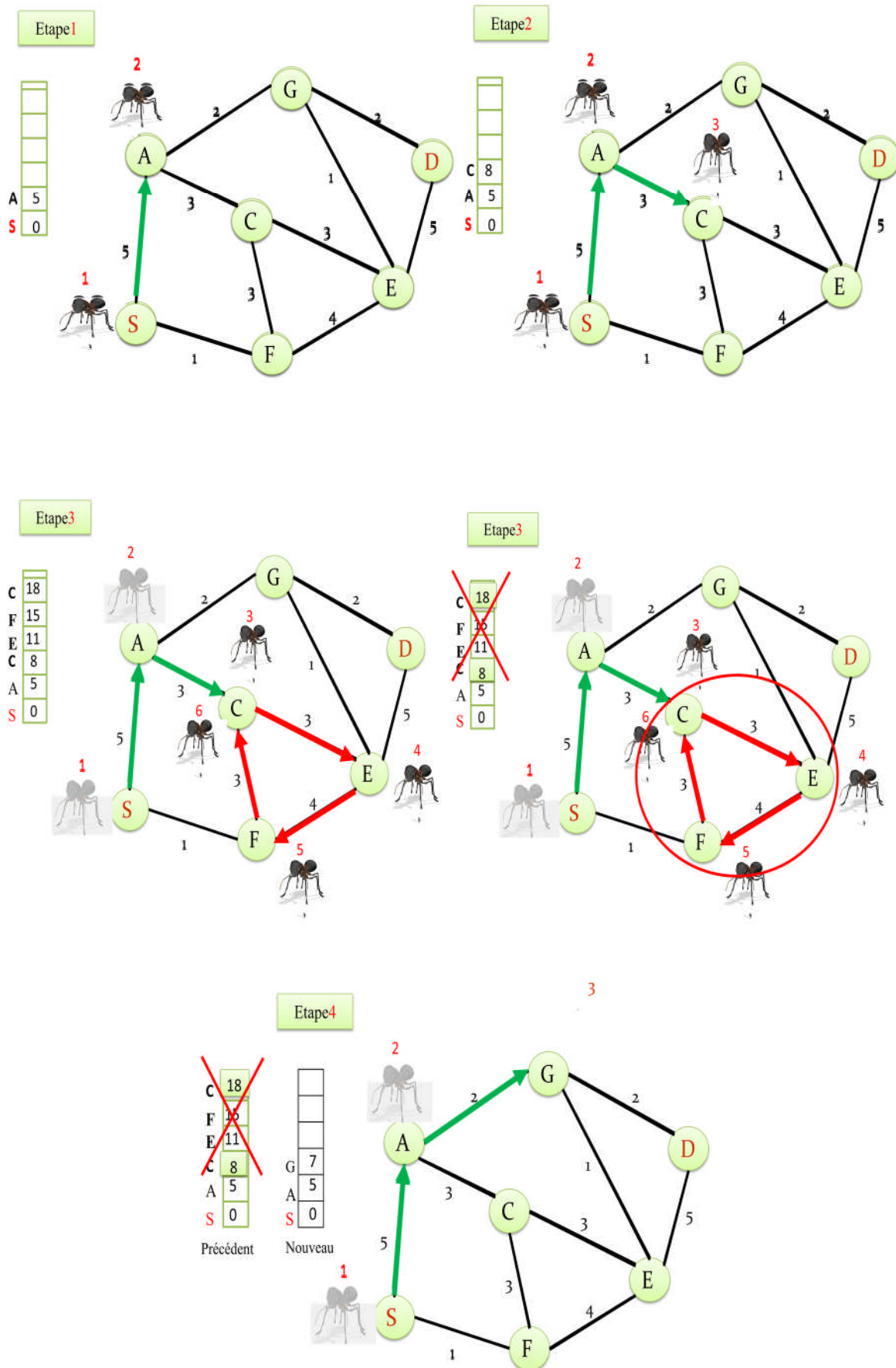


Figure 4.5 : Détection d'un cycle par une VAnt.

3. A chaque fois Vant Arrive à un certain nœud **d**, les deux tables du nœud **i** (**i** = l'ensemble des nœuds sont déjà visités par la Vant avant **d**) sont mises à jour, la table **M** et la table **T**, pour toutes les entrées correspondant au nœud destination **d** :
 - Les valeurs μ_d et σ_d^2 dans la table **M_i** sont modifiées en utilisant les formules (1) et (2).
 - La meilleure valeur $t_{best_{id}}$ du temps de parcours des VAnts du nœud **i** au nœud **d** sauvegardée dans la fenêtre d'observation mobile **W_{id}** est également mise à jour.
 - Si le temps $t_{i \rightarrow d}$ de parcours de la VAnt du nœud **i** à la destination **d** est plus petit que $t_{best_{id}}$ alors $t_{best_{id}}$ est remplacé par le $t_{i \rightarrow d}$.
 - De plus, la table **T_i** est changée en incrémentant la probabilité $p_{hd'}$ (probabilité de choisir le voisin **h** quand la destination est **d'**). Elle est modifiée par la formule suivante :

$$\tau_{ihd'} \leftarrow \tau_{ihd'} + r(1 - \tau_{ihd'}) \quad (7)$$

- De cette façon pour une même valeur **r**, les petites valeurs de phéromone sont augmentées proportionnellement plus que les grandes valeurs favorisant l'exploration rapide des bons chemins
- Les autres probabilités sont décrémentés par la normalisation Autrement dit, les valeurs sont réduites de sorte que la somme des phéromones sur les liens sortant du nœud **i** = 1 :

$$\tau_{ijd'} \leftarrow \tau_{ijd'} - r\tau_{ijd'}, \quad \forall j \neq h, j \in N_i \quad (8)$$

- ✓ Où : **h** est le voisin d'un nœud **i** accédé par la VAnt et **d'** représente tout les nœuds entre le nœud destination **d** et le nœud courant **i**
- Chaque chemin trouvé reçoit un taux de phéromone (**r**) compris entre dans l'intervalle [0, 1] et calculée comme suit:

$$r = c_1 \frac{t_{best_{id}}}{t_{i \rightarrow d}} + c_2 \frac{t_{sup} - t_{best_{id}}}{(t_{sup} - t_{best_{id}}) + (t_{i \rightarrow d} - t_{best_{id}})} \quad (9)$$

Où $t_{i \rightarrow d}$ est le temps de parcours observé par la dernière VAnt du nœud k à la destination d et $t_{best_{id}}$ est le meilleur temps de parcours fait par les VAnts voyageant vers la destination d sur la fenêtre d'observation W_d .

La valeur de t_{sup} est calculée comme suit:

$$t_{sup} = \mu_d + \frac{\sigma_d}{\sqrt{1 - \gamma} \sqrt{|W_{max}|}} \quad (10)$$

Où γ est le niveau de confiance.

Le premier terme dans (9) évalue le rapport entre le temps de parcours de la VAnt courante et le meilleur temps de parcours observé sur la fenêtre d'observation mobile. Le second terme est un facteur de correction qui indique l'éloignement entre les valeurs $t_{i \rightarrow d}$ et $t_{best_{id}}$. Les valeurs de C_1 et C_2 indiquent l'importance relative de chaque terme. Il est logique de supposer que le premier terme dans (9) est plus important que le deuxième terme. En conséquence la valeur de C_1 doit être choisie plus grande que la valeur de C_2 .

Exemple

Pour bien illustrer cette idée voici un exemple déroulé sur le graphe de la figure 4.6, avec pour objectif de chercher un chemin allant du nœud 1 au nœud 4 :

La recherche d'un itinéraire optimal (moins d'embouteillage) se fait en parcourant successivement les nœuds du graphe, un calcul détermine à chaque fois le nœud suivant.

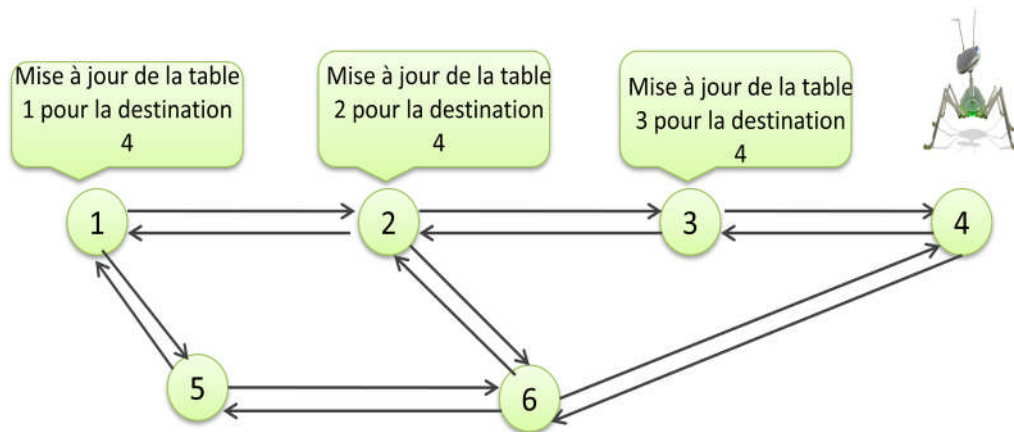


Figure4.6 : Mise à jour des tables de phéromones correspondant à chaque nœud visité.

Commençons par le nœud 1, un choix s'offre alors à nous, soit le nœud 2 ou le nœud 5, choisissons le nœud 2 selon la probabilité (3),

arrivant au nœud 2 la Vant utilise sa mémoire pour mettre à jour les tables du nœud 1 (M_1 et $T_1[2][2] \rightarrow$ incrémentation, $T_1[2][5] \rightarrow$ décrémentation) selon les équations (1), (2), (7), (8) au niveau de ce nœud un autre choix à faire, soit le nœud 3 ou le nœud 6, choisissons le nœud 3 selon la probabilité (3),

arrivant au nœud 3 la Vant utilise sa mémoire pour mettre à jour les tables du nœud 1 et 2 (M_1, M_2 et $(T_1[3][2]; T_2[3][3]) \rightarrow$ incrémentation ; $(T_2[3][6]) \rightarrow$ décrémentation) selon les équations (1), (2), (7), (8).

Puis le nœud 3, ce n'est qu'au nœud 4 qu'un choix est possible, arrivant à destination nœud 4, la Vant utilise sa mémoire pour mettre à jour les tables du nœud 1, 2 et 3 (M_1, M_2, M_3 et $(T_1[2][4]; T_2[3][4]; T_3[4][4]) \rightarrow$ incrémentation, $(T_1[4][5]; T_2[4][6]) \rightarrow$ décrémentation) selon les équations (1), (2), (7), (8).

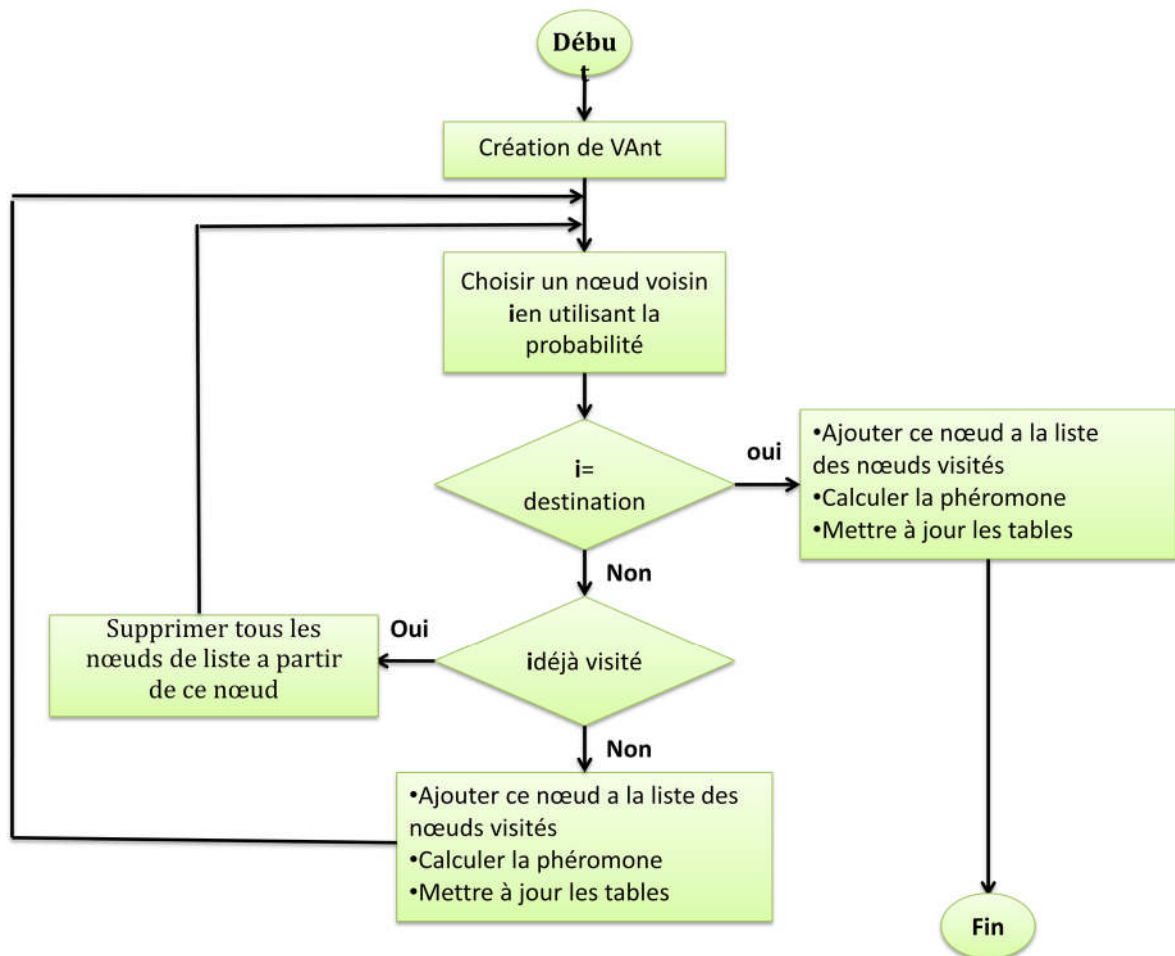


Figure 4.7 : organigramme du chemin optimal

5. Comportement de l'algorithme au niveau des différents nœuds :

Au niveau de la ville source V_s :

Si un client situé dans une ville V_s demande un chemin vers une destination V_d **alors**

Récupération de longitude et altitude de la ville V_s

Création de la fourmi V_{ants} ;

Calcule de la probabilité pour choisir la ville voisine

Choix du nœud voisin V_{voisin} (ville voisine)

Finsi

Au niveau de la ville de destination V_d ou ville voisine V_{voisin}

Si V_{choisi} (ville chose $\notin V_{visité}$ (ville déjà visité) alors

Ajouter V_{voisin} a la mémoire fourmi

Ajouter le temps de parcour pour arrivé à V_{voisin} à la mémoire de la fourmi

Mettre à jour les champs des tables M correspondant aux villes déjà visité

$$\mu_{id} \leftarrow \mu_{id} + \varsigma(t_{i \rightarrow d} - \mu_{id})$$

$$\sigma_{id}^2 \leftarrow \sigma_{id}^2 + \varsigma((t_{i \rightarrow d} - \mu_d)^2 - \sigma_d^2)$$

Si temps de parcours ($t_{i \rightarrow d}$) $< t_{best}$

$$t_{best} \leftarrow t_{i \rightarrow d}$$

Fins

Calculer de t_{sup}

$$t_{sup} = \mu_d + \frac{\sigma_d}{\sqrt{1 - \gamma \sqrt{|W_{max}|}}}$$

Calculer de r

$$r = c_1 \frac{t_{best_d}}{t_{k \rightarrow d}} + c_2 \frac{t_{sup} - t_{best_d}}{(t_{sup} - t_{best_d}) + (t_{k \rightarrow d} - t_{best_d})}$$

Mettre à jour les champs des tables T correspondant aux villes déjà visité

Pour ville choisie

$$t_{ihd'} \leftarrow t_{ihd'} + r(1 - t_{ihd'})$$

Pour les villes voisines non choisies

$$t_{ijd'} \leftarrow t_{ijd'} - r t_{ijd'}$$

Sinon

Supprimer le cycle et et choisir une autre ville selon la probabilité

finsi

Si $V_{choisi} = V_D$ **alors**

Tuer la fourmi **Vant**

Sinon

Choisir ville voisine selon la probabilité

finsi

6. Conception de la base de données :

Dans cette partie nous décrivons la manière de concevoir la structure de la base avec un model *Entité-Association* pour optimiser l'implémentation de la base de données.

6.1. Dictionnaire de données :

Attribut	Type	désignation
Id_Noead	Entier	Identifiant d'un nœud
Latitude	Double	Latitude d'un nœud
Longitude	Double	Longitude d'un nœud
Nom	String	Le nom d'un nœud il en possède un
Longueur	Double	Longueur d'un arc en Km
Moyenne	Double	Temps moyens
Variance	Double	valeur pour mesurer la dispersions
Temps	Double	Temps entre deux nœud en Heure (H)

6.2. Modèle Entité Association :

Les entités que nous décrivons sont les arcs et les noeuds, qui modélisent respectivement les tables Arc et Noeud au niveau données. Tandis que l'association est la relation entre ces deux entités.

Le modèle *Entité-Association* de notre domaine d'étude est décrit par la figure (3.4).

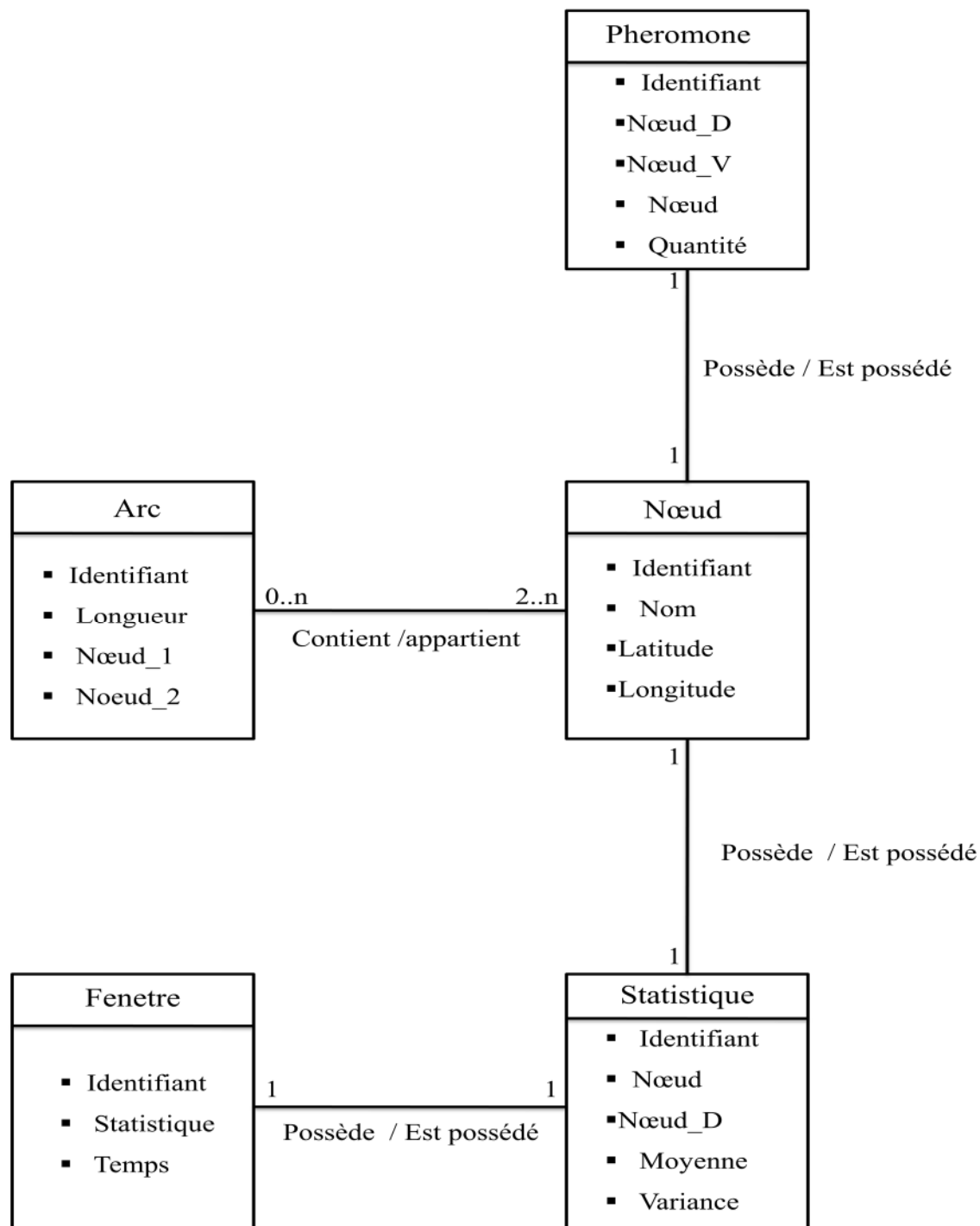


Figure 4.8 modèle Entité-Association décrivant la base de données.

6.3. Description textuelle :

TABLE Nœud :

Cette table contient tous les nœuds de la carte :

Id_Noed : Identifiant unique pour chaque nœud.

Longitude et Latitude : Coordonnées GPS du nœud afin de le situer sur la carte.

Nom : contient le nom d'un endroit précis.

TABLE Arc :

Id_Arc: Identifiant unique pour chaque Arc.

Neoud_1: Identifiant du nœud de la première extrémité de l'arc.

Neoud_2: Identifiant du nœud de la deuxième extrémité de l'arc.

Longueur: La longueur du chemin réel entre les deux extrémités en Kilomètres.

TABLE Pheromone :

Identifiant :Identifiant unique pour chaque quantité de phéromone

Nœud_D : Identifiant du nœud destination.

Nœud_V : Identifiant du nœud voision.

Nœud : identifiant du nœud au quel la table de phéromeone appartient .

Quantité : Quantité de phéromone déposée entre Nœud et Nœud_V pour atteindre Nœud_D.

TABLE Statistique:

Identifiant : Identifiant unique pour chaque échantillant

Nœud_D : Identifiant du nœud destination.

Nœud : Identifiant du nœud decrit par les statistiques

Moyenne : Temps moyens pour atteindre Nœud_D à partir du Noeud

Variance : Mesure permet de caractériser la dispersion des valeurs de temps par rapport à la Moyenne.

TABLE Fenetre :

Identifiant : Identifiant unique pour chaque temps .

Statistique : Identifiant pour chaque échantillant .

Temps : Durée entre deux nœuds ..

7. fonctionnement de l'application :

Après avoir implémenté l'algorithme fournis pour résoudre le problème d'embouteillage, on s'intéresse dans cette partie au fonctionnement général de l'application

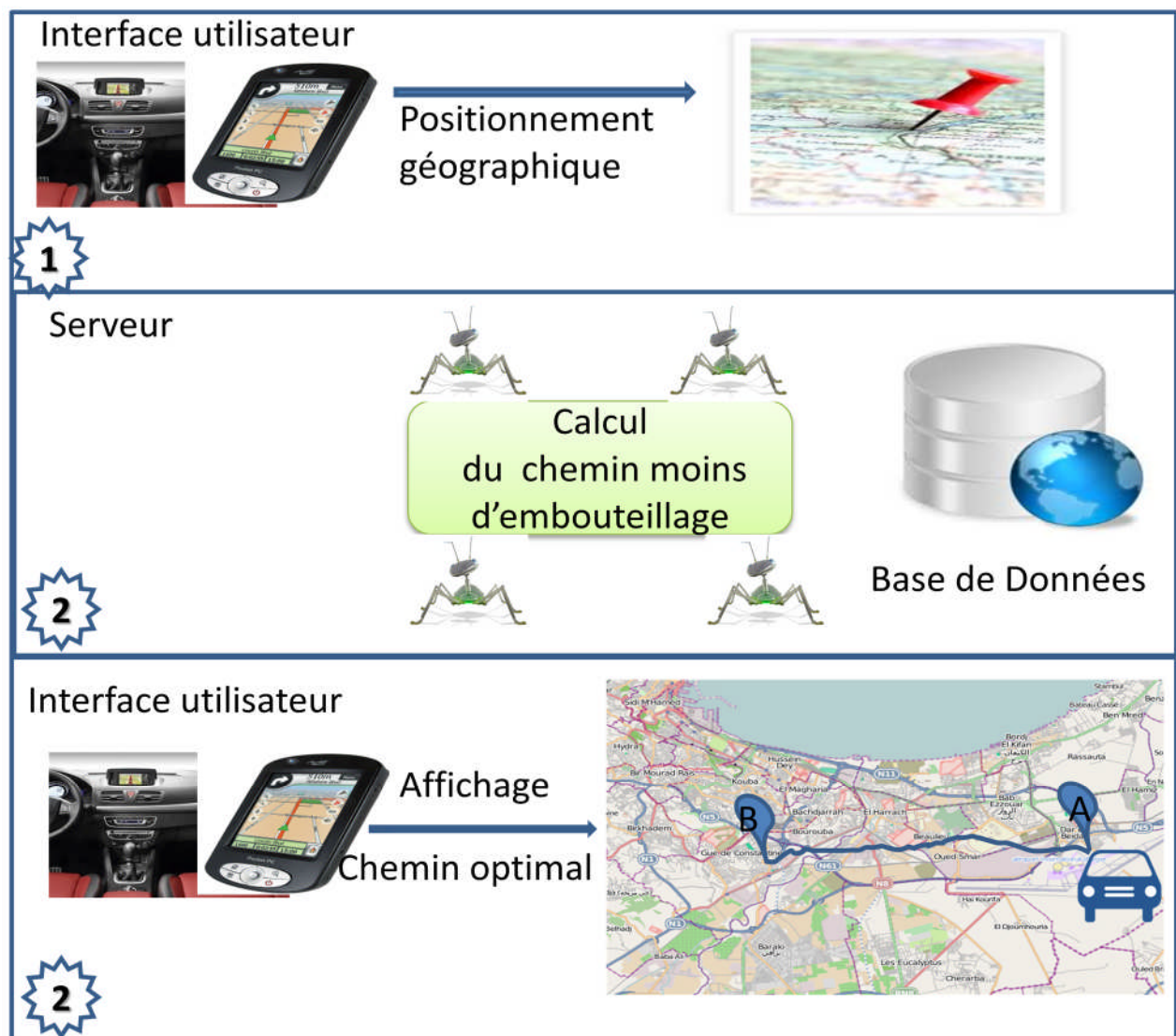


Figure 4.7: fonctionnement général de l'application

- ❖ Etapes1: demande d'un client utilisant l'application un acheminement en saisissant une destination précise.
- ❖ Etape 2 : récupération de coordonnées géographiques (longitude, altitude) de la source avec GPS.
- ❖ Etape 3 : calcul du chemin en appliquant algorithme fourni, en se basant sur les informations stockées dans la base de données .Ces information sont modifiées constamment par les mutilateurs d'application.
- ❖ Etape 4 : arrivant à destination on mit à jour les tables de la base de donnée (mise a jour de la phéromone, la moyenne et la variance estimé en utilisant de temps de parcoure).

8. Conclusion :

Nous avons vu dans ce chapitre la conception de notre application. Une stratégie a été proposée. Dans le chapitre qui va suivre, nous allons présenter les tests et les résultats de simulation de notre application

Chapitre 5

Réalisation

Dans ce chapitre, nous présentons dans un premier temps les différentes classes de l'algorithme fournis. Puis, les outils logiciels et quelques interfaces graphiques

1. Introduction

Après avoir présenté la partie conception, nous entamons, dans ce chapitre, la réalisation du projet. Cette partie met en évidence les différents choix techniques (choix du langage, IDE, bibliothèque logicielle, outils matériels, ...) utilisés

2. Les différentes classes de l'algorithme fournis :

La classe fourmi: contient des variables et des méthodes concernant la fourmi. Exemple de méthodes : accéder aux informations d'un nœud k déjà visité,....

La classe fourmifonc : le but de cette classe est de faire tous les traitements pour une fourmi. par exemple, des méthodes pour détecter la présence de cycles et leur destruction, pour la sélection du prochain voisin à visiter, pour stocker dans la mémoire de la fourmi les informations relatives au nœud, pour modifier la table de phéromone, pour détecter si la fourmi a atteint sa destination ou sa source ou s'il faut détruire cette fourmi, calculer le temps total que met une fourmi pour atteindre une destination.

La classe Table : fait les opérations de lecture et écriture et d'affichage de la table de la base de données.

La classe statistique : calculs statistiques : calcul de la variance, du temps moyen.

La classe distance : calcule de la distance.

3. La variation de la phéromone en fonction de temps

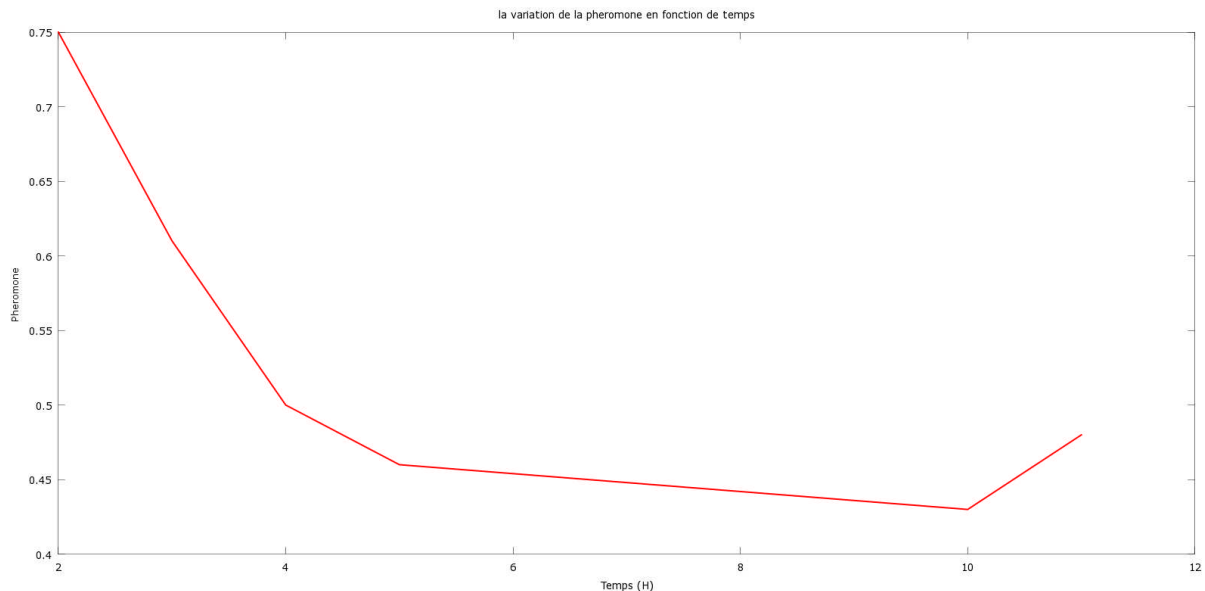


Figure 5.1 : La variation de la phéromone en fonction de temps.

4. Pré requis Logiciels :

4.1. langage de programmation C :

C est un langage de programmation impératif généraliste. Inventé au début des années 1970 pour réécrire UNIX, C est devenu un des langages les plus utilisés. De nombreux langages plus modernes comme C++, Java et PHP reprennent des aspects de C.

4.2. IDE Eclipse :

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT,

D'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions. La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

4.3. Android studio :

Android Studio permet principalement d'éditer les fichiers JAVA et les fichiers de configuration d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément.

5. Présentation de l'application :

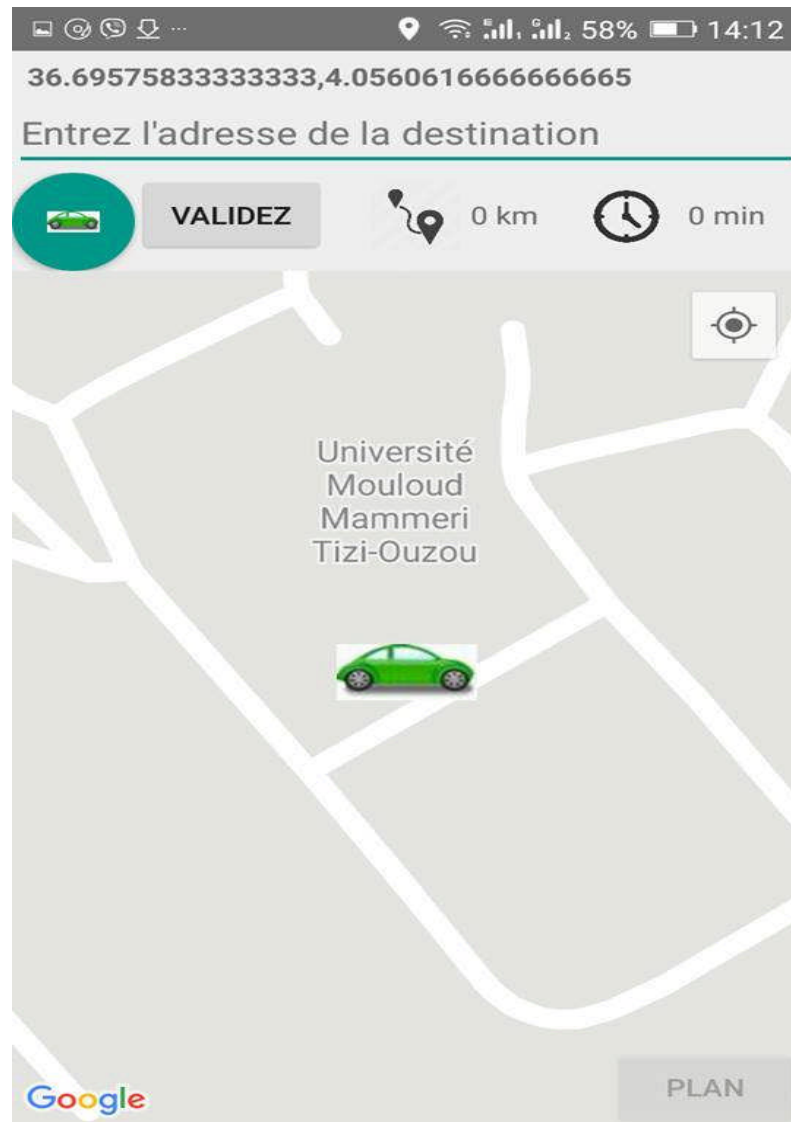


Figure 5.2 : Interface graphique de l'application

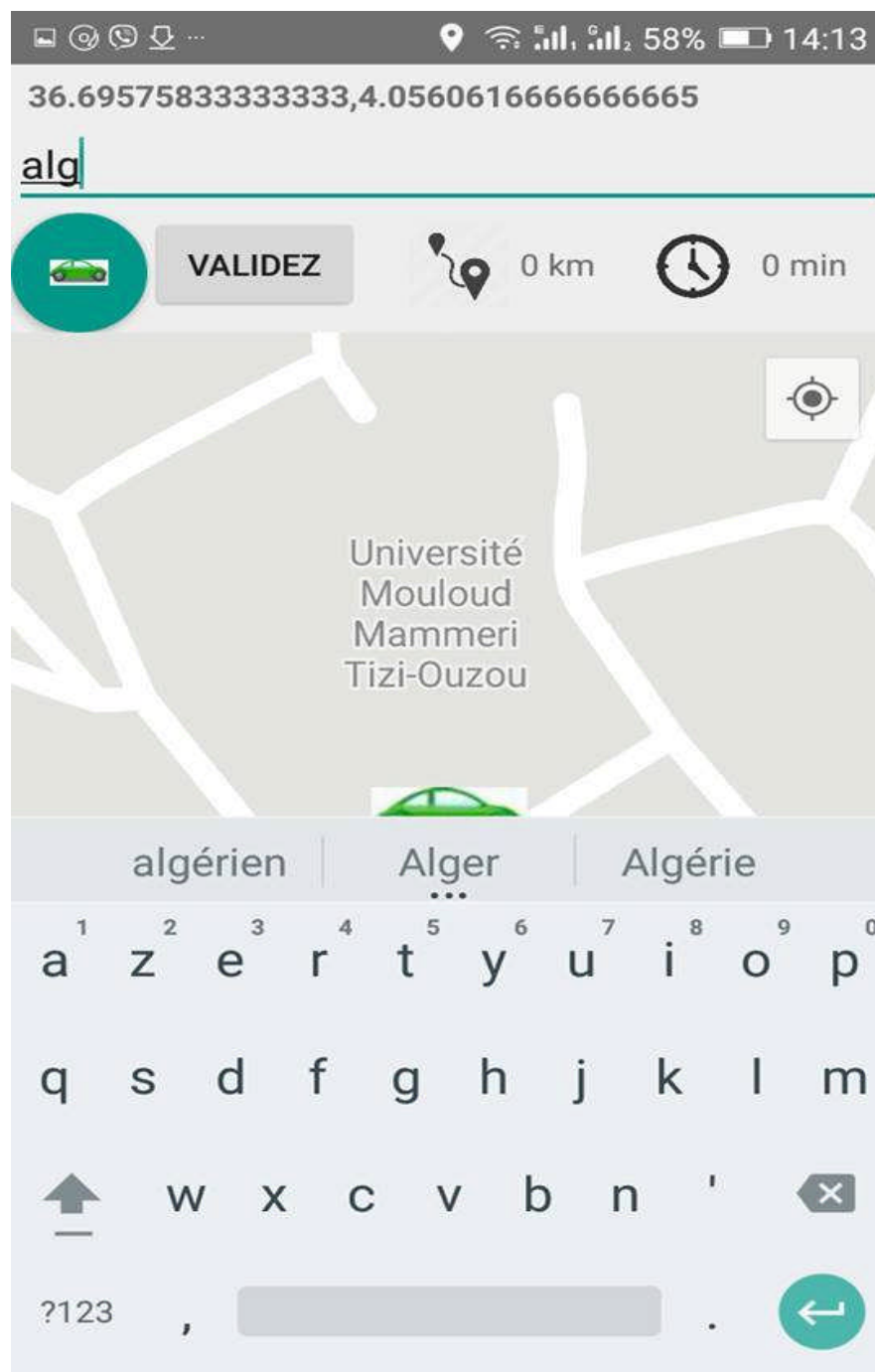


Figure 5.3 : Interface graphique de l'application.

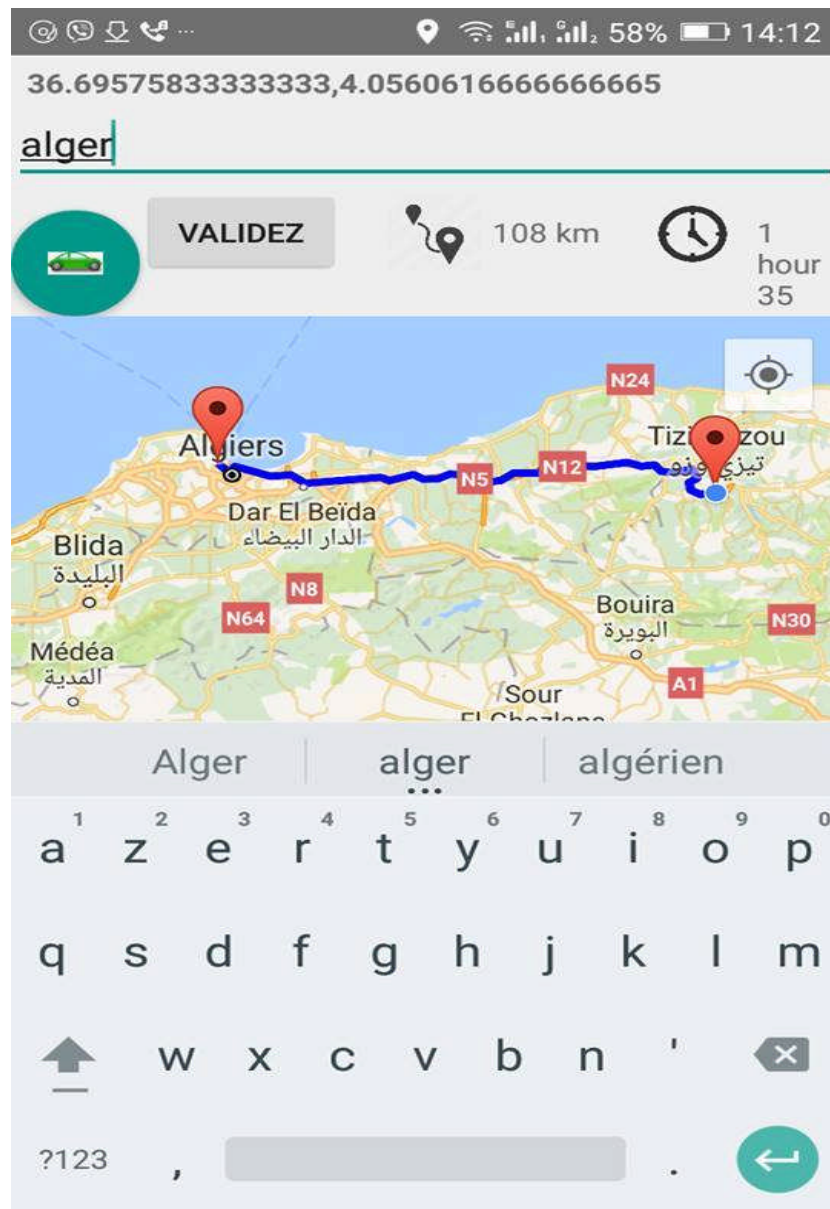


Figure 5.4 : Interface graphique de l'application.

6. Conclusion

Dans ce chapitre nous avons présenté les classes de l'algorithme fournis, suivit des outils utilisés, a la fin du chapitre un bref aperçu de l'application.

1. Conclusion Général :

L'objectif de ce travail consiste en l'application de la méta-heuristique fourmi au problème d'embouteillage. Pour cela, nous avons donné un état de l'art sur le développement android et la géolocalisation. Nous avons consacré une partie de l'état de l'art à l'algorithme d'optimisation basé sur la colonie de fourmis dans laquelle, nous avons illustré le comportement des fourmis réelles. Cela montre l'utilité de l'intelligence collective, où l'on observe que les fourmis sont capables de trouver le plus court chemin entre la fourmilière et la source de nourriture.

Nous avons implémenté cet algorithme pour trouver le chemin qui contient moins d'embouteillage et puis nous avons présenté la réalisation de notre application.

2. Perspectives:

Notre but de départ était de développer une application android qui indique au conducteur le chemin ayant moins d'embouteillage mais par contrainte de temps l'objectif n'a pas été atteint à 100%.

Néanmoins le travail fait jusqu'ici est un début et une bonne expérience qui mérite d'être continué et développé par les binômes des promotions à venir.

Les perspectives de ce travail peuvent être résumées dans le point suivant :

Remplacer le GPS du téléphone mobile par une antenne de faible portée et un petit processeur qui seront incorporés à chaque voiture. Ce qui permet à une voiture prise dans un embouteillage, de transmettre le message à la voiture derrière elle, qui transmet elle-même cette information à la voiture précédente et ainsi de suite : de proche en proche, une chaîne d'information est créée.

Références bibliographiques

[DOR01] Dorigo M., Stützle T. "The Ant Colony Optimization metaheuristic: Algorithms, Applications, and Advances". Université Libre de Bruxelles, IRIDIA. Computer Science, Intellectics Group, 2001.

[DOR04] Dorigo M., Stützle T. "Ant Colony Optimization", The MIT Press, Cambridge, MA, 2004.

[DOR06] Dorigo M., Birattari M., Stützle T. "Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique", IEEE Computational Intelligence Magazine, vol. 1, no.4, pp 28-39, 2006.

[DRÉ01] Dréo J., Siarry P. "A new ant colony algorithm using direct interindividual communication aimed at continuous optimization". London : The MIT Press, 2001/11/28.

[LAA05] Laalaoui Y. "Ordonnancement temps réel hors-ligne à l'aide de Système de Colonie de Fourmis en environnement monoprocesseur ", Thèse de Magister en Informatique ESI 2005.

[DRE03] Dréo J., Petrowski A., Taillard E., Siarry P. "Métaheuristiques pour l'optimisation difficile", Éd. Eyrolles, Paris, septembre 2003, Broché, ISBN 2-212-11368-4.

[DOR92] Dorigo M. " Learning and Natural Algorithms (in Italian)". PhD thesis, DEI, Politecnico di Milano, Italy, 1992

[REI94] Reinelt G. "The traveling Salesman: Computational Solutions for TSP Application ", volume 840 of Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, 94.

[HAM03] Hamani N., Gharout S. "Résolution des problèmes Max-Sat et Partitionnement en utilisant l'Optimisation par Colonie de Fourmis". Mémoire Ingénieur. INI 2003.

[GUE07] Guerrou E. " Optimisation des Protocoles de Routage avec la méthode de Colonie de Fourmis", mémoire fin étude pour l'obtention du diplôme d'ingénieur d'état en informatique, ESI 2007.

[HAM05] Hamani N. "Application des fourmis quantiques au Partitionnement avec ordonnancement en Codesign", Thèse de Magister en Informatique ESI, Alger

[DOR 99] Marco Dorigo, Gianni Di Caro and L.M. Gambardella, Ant Algorithms for Discrete Optimization, Artificial Life, vol. 5, No. 2, pp. 137-172, 1999. 2005.

[DOR 99b] Marco Dorigo and Gianni Di Caro, "The Ant Colony Optimization Meta-Heuristic", McGraw-Hill Press, pp. 11 - 32, 1999.

[DOR 99] Marco Dorigo and Gianni Di Caro "The Ant Colony Optimisation Meta-Heuristic" IRIDIA Université Libre de Bruxelles 1999.**[DOR 99a]** M.Dorigo & G.Di Caro .Ant Colony Optimisation :A new meta-heuristic.IEEE 99.

[CiscoMag, 2006], CiscoMag, Géolocalisation WiFi & RFID décembre 2006.

[Ilyas & Ahson, 2006] Mohammad Ilyas, Syed A. Ahson, Smartphones, 2006.

[Kleusberg, 1990] Kleusberg, A., Comparing GPS and GLONASS, GPS World, Vol. 1, No. 6, November/December 1990, pp. 52-54.

[Mohan, 2013], Operating Systems I. Chandra Mohan 2013.

[Neviere, 2001], Philippe Neviere, Etude sur la co-localisation d'un réseau UMTS avec les réseaux GSM et DCS existants, 2001.

[Phillips, 1996] Phillips, B., GPS Field Applications in Forestry Consulting, Global

Positioning System in Forestry Workshop, Kelowna, British Columbia, Canada, November 25-28, 1996.

[Wiley & Sons, 2011], Next Generation Mobile Communications Ecosystem: Technology Management for Mobile Communications John Wiley & Sons, 25 févr. 2011.

[Wiley & Sons, 2008], Developing Software for Symbian OS 2nd Edition: A Beginner's Guide to Creating Symbian OS v9 Smartphone Applications John Wiley & Sons, 28 feb. 2008.

Webliographie

[1] Géolocalisation

<http://blog.nordnet.com/loeil-sur-le-web/decryptages-loeil-sur-le-web/la-geolocalisation.html>

[2] Géolocalisation par wifi

<http://www.cnil.fr/linstitution/actualite/article/article/la-geolocalisation-a-partir-des-points-d'accès-wi-fi/>

[3] Géolocalisation par adresse IP

<http://www.definitions-webmarketing.com/Definition-Geolocalisation-adresse-IP>

[4] Open Street Map

http://fr.flossmanuals.net/_booki/openstreetmap/openstreetmap.pdf

[5] Google Maps Marker

<https://support.google.com/mapmaker>

[6] Ushahidi

<http://www.ushahidi.com/>

[7] WikiMapia

<http://wikimapia.org/>