

N° d'ordre : 14 / 2014 - M / INF

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE**  
**SCIENTIFIQUE**  
**UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI**  
**BOUMEDIENE**  
**FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE**



## **Mémoire**

Présenté pour l'obtention du diplôme de Magister

En : **Informatique**

**Option : Traitement d'Images et Systèmes d'Information**  
**Géographique (SIG)**

**Par : BELAROSSI Asma**

THEME

# **Analyse de réseaux routiers et détermination de chemins optimaux dans un SIG**

Soutenu publiquement le 09 Juillet 2014, devant le jury composé de :

M.	Y. SMARA	Professeur	à l'USTHB	Président
M.	M. BELAHADJAÏSSA	Professeur	à l'USTHB	Directeur de mémoire
M <sup>elle</sup>	S. AOUAT	Maitre de conférences / A	à l'USTHB	Examinatrice
M.	K. BOUKHALFA	Maitre de conférences / A	à l'USTHB	Examineur
M.	B. BESSAA	Maitre-Assistant	à l'USTHB	Invité

## ***Remerciements***

*J'adresse mes vifs remerciements et toute ma gratitude à Monsieur M. Belhadj-Aissa, Professeur à l'USTHB, et à Monsieur B. Bessaa, Maître Assistant, pour m'avoir encadré et dirigé dans mes travaux.*

*Je remercie M Y. SMARA, Professeur à l'USTHB pour l'honneur qu'il me fait en acceptant de présider le jury de cette soutenance.*

*Je remercie les membres du jury qui ont accepté de prendre de leur temps pour examiner mon travail : Mademoiselle S. AOUAT, Maitre de Conférences à l'USTHB et Monsieur K. BOUKHALFA, Maitre de Conférences à l'USTHB.*

*Enfin, j'exprime ma gratitude à Madame A. Belhadj-Aissa et à Monsieur Y. Smara, Professeurs à l'USTHB et à tous les membres du Laboratoire de Traitement d'Images et Rayonnement (LTIR) de la Faculté de d'Electronique et d'Informatique (FEI) de l'USTHB pour les conseils et l'aide qu'ils m'ont prodigués.*

## *Dédicaces*

*A mes parents,*

*A mon mari,*

*A ma fille,*

*A mes frères et sœurs,*

*A toute ma famille,*

*A mes amis,*

*A tous ceux qui, de près ou de loin, ont contribué à l'élaboration  
de ce modeste mémoire.*

## **Table des matières**

<i>Table des matières</i>	<b>01</b>
<i>Table des figures</i>	<b>04</b>
<i>Introduction</i>	<b>07</b>
<b>Chapitre I : Information géographique et Concepts de base dans les SIG.</b>	<b>09</b>
<i>I.1. Système d'Information Géographique.</i>	<b>09</b>
<i>I.2. Information Géographique.</i>	<b>10</b>
<i>I.2.1. Définition</i>	<b>10</b>
<i>I.2.2. Sources de données géographiques</i>	<b>10</b>
<i>I.2.3. Modélisation d'une Information Géographique</i>	<b>11</b>
<i>I.2.4. Mesure et représentation de la localisation d'une Information                 Géographique</i>	<b>18</b>
<i>I.2.5. Organisation de l'Information Géographique dans un SIG</i>	<b>18</b>
<i>I.3. Conclusion</i>	<b>19</b>
<b>Chapitre II : Détermination de Chemins Optimaux</b>	<b>20</b>
<i>II.1. Notation</i>	<b>21</b>
<i>II.2. Algorithme de recherche de chemins optimaux</i>	<b>21</b>
<i>II.2.1. Algorithme à fixation /correction d'étiquettes</i>	<b>22</b>
<i>II.2.2. Accélération des algorithmes d'étiquetage.</i>	<b>24</b>
<i>II.3. Modélisation des mouvements directionnels routiers.</i>	<b>29</b>
<i>II.4. Détermination de chemins optimaux avec séquences d'arcs pénalisées.</i>	<b>30</b>
<i>II.4.1. Approche de résolution naïve</i>	<b>31</b>
<i>II.4.2. Expansion du graphe</i>	<b>32</b>
<i>II.4.3. Passage au graphe dual</i>	<b>32</b>
<i>II.5. Conclusion</i>	<b>34</b>
<b>Chapitre III : Analyse et modélisation du réseau routier</b>	<b>35</b>

III.1. Analyse du réseau routier	35
III.2. Modélisation du réseau routier	38
III.2.1. Codification du réseau routier	38
III.2.2. Modèle conceptuel de données	39
III.2.3. Problématiques et discussion	42
III.2.4. Solution proposée : Modèle topologique 3D	43
III.3. Conclusion.	50
<b>Chapitre IV : Fonctionnalités et modalités de mise en œuvre des SIG en ligne.</b>	<b>51</b>
IV.1. Cartographie sur le web et SIG en ligne	51
IV.2. Architecture des SIG en ligne	52
IV.2.1. Architecture des Systèmes d'Information.	52
IV.2.2. Architecture Web.	57
IV.2.3. Architecture des SIG en ligne	58
1. SIG en ligne du point de vue services offert.	58
2. SIG en ligne du point de vue architecture.	58
3. SIG en ligne du point de vue mise en ligne.	60
IV.3. Webmapping et Concept de l'open source.	61
IV.4. Approches libres existantes pour la mise en ligne des SIG.	62
IV.4.1. Approche carte statique (HTML simple).	62
IV.4.2. Solution vectorielle SVG.	63
IV.4.3. Applet java.	64
IV.4.4. Serveurs cartographiques dédiés.	66
IV.5. Conclusion.	67
<b>Chapitre V : Réalisation du système et présentation des résultats.</b>	<b>68</b>
V.1. Réalisation du SIG en ligne.	68
V.1.1. Architecture générale de la solution.	68
V.1.2. Composants et outils open source utilisés.	69

V.1.3. <i>Présentation du SIG en ligne réalisé.</i>	75
1. <i>Interface client.</i>	75
2. <i>Fonctionnement du SIG en ligne.</i>	76
V.2. <i>Intégration de l'outil de guidage.</i>	
V.2.1. <i>Acquisition de données.</i>	77
V.2.2. <i>Intégration des données dans PostgreSQL/PostGIS.</i>	79
V.2.3. <i>Détermination des chemins optimaux.</i>	81
V.3. <i>Conclusion.</i>	87
<b>Conclusion</b>	

## Table des figures :

<b>Figure 1.1</b> : Fonctionnalités d'un SIG.	09
<b>Figure 1.2</b> : Composantes d'un objet géographique.	12
<b>Figure 1.3</b> : Représentation Raster.	13
<b>Figure 1.4</b> : Représentation vecteur.	14
<b>Figure 1.5</b> : Diagramme de classes du modèle spaghetti.	15
<b>Figure 1.6</b> : Diagramme de classes du modèle Réseau.	16
<b>Figure 1.7</b> : Diagramme de classes du modèle topologique	17
<b>Figure 1.8</b> : Etat/Transition de l'Information Géographique à travers les niveaux de modélisation dans un SI.	17
<b>Figure 1.9</b> : Organisation de l'Information Géographique dans un SIG.	18
<b>Figure 2.1</b> : Problème de recherche de plus court chemin.	21
<b>Figure 2.2</b> : Différents algorithmes de recherche de plus court chemin.	28
<b>Figure 2.3</b> : Exemples Mouvements directionnels interdits.	30
<b>Figure 2.4</b> : Exemples de solutions incluant des cycles.	31
<b>Figure 2.5</b> : Expansion du graphe au niveau d'une intersection.	32
<b>Figure 3.1</b> : Codification des obstacles et/ou de changement de valeur d'un attribut de tronçon.	38
<b>Figure 3.2</b> : Codification d'un carrefour.	39
<b>Figure 3.3</b> : Codification d'un rond-point.	39
<b>Figure 3.4</b> : Diagramme de classe du réseau routier fondé sur le modèle vectoriel topologique.	40
<b>Figure 3.5</b> : Affectation des niveaux aux arcs.	44
<b>Figure 3.6</b> : Modèle Topologique 2D et Modèle 3D utilisé.	45
<b>Figure 3.7</b> : Diagramme de classe du modèle topologique 3D.	45
<b>Figure 3.8</b> : Passage du modèle 2D au modèle 3D.	46

<b>Figure 3.9 :</b> Calcul des altitudes des points intermédiaires.	48
<b>Figure 3.10:</b> Diagramme de classe du réseau routier selon le modèle vectoriel topologique 3D.	49
<b>Figure 4.1 :</b> Trois composantes d'un SI et en l'occurrence d'un SIG.	53
<b>Figure 4.2 :</b> Architecture du modèle Client/serveur selon la classification Présentation- Traitement- Données.	55
<b>Figure 4.3 :</b> Exemple d'une architecture à trois niveaux.	56
<b>Figure 4.4 :</b> Fonctionnement d'un SIG en ligne orienté Serveur.	59
<b>Figure 4.5 :</b> Fonctionnement d'un SIG en ligne orienté client.	60
<b>Figure 4.6 :</b> Fonctionnement de la solution « Carte Statique ».	63
<b>Figure 4.7:</b> Effet de pixellisation ou aliasing.	63
<b>Figure 4.8 :</b> Fonctionnement de la solution vecteur.	64
<b>Figure 4.9:</b> Etapes de fonctionnement de la solution Java.	65
<b>Figure 5.1 :</b> Architecture générale d'un SIG en ligne basé sur l'approche serveur cartographique.	68
<b>Figure 5.2 :</b> Architecture détaillée du SIG en ligne réalisé.	75
<b>Figure 5.3 :</b> Interface du SIG en ligne.	75
<b>Figure 5.4 :</b> Zone d'étude.	78
<b>Figure 5.5 :</b> Vectorisation du réseau routier.	79
<b>Figure 5.6 :</b> Nouer un graphe.	79
<b>Figure 5.7 :</b> Passage au modèle topologique 3D.	81
<b>Figure 5.8 :</b> formulation de la requête de recherche de l'itinéraire optimal.	82
<b>Figure 5.9 :</b> Exploration de l'itinéraire optimal.	82

## **Tableaux :**

<b>Tableau 1.1 :</b> Différents niveaux de modélisation de données au sein d'un SI.	11
<b>Tableau 3.1:</b> Liste des attributs des classes du modèle conceptuel.	42
<b>Tableau 4.1:</b> Tableau comparatif des fonctionnalités offertes par chaque solution.	67
<b>Tableau 5.1 :</b> Comparaison de PostgreSQL/PostGIS à MySQL.	73

## **Organigrammes :**

<b>Organigramme 5.1 :</b> Processus de création de la topologie 3D.	80
<b>Organigramme 5.2 :</b> Processus de détermination de chemin optimal dans un SIG en ligne.	85

## Introduction

Les distances parcourues par habitant et par jour suivent une tendance à la hausse, cette hausse est en partie expliquée par la forte augmentation du nombre de véhicules, ainsi que par la construction de nouvelles infrastructures routières permettant de se déplacer rapidement, cela a, en effet, permis d'allonger les distances domicile-travail et donc l'étalement des zones urbanisées. Cette mobilité supplémentaire a été majoritairement absorbée par la voiture. Aujourd'hui, le réseau routier arrive à saturation et subit des congestions récurrentes, il paraît donc indispensable d'optimiser la mobilité en adoptant de nouvelles manières d'utiliser l'infrastructure routière en répartissant au mieux la demande de déplacement dans l'espace et dans le temps. Cette perspective se fonde principalement sur une analyse du réseau routier par le biais de la modélisation.

Dans ce contexte, les Systèmes d'Informations Géographiques (SIG) jouent un rôle fondamental. Ils permettent de constituer, organiser et mutualiser les bases de données indispensables à la représentation de l'infrastructure routière, de plus en plus ouvertes sur le web.

Internet constitue, en effet, un vecteur privilégié de diffusion d'information sur le transport. De nombreux portails-web d'information sur le déplacement ont vu le jour au cours de ces dernières années. Ils diffusent généralement l'information sur les conditions de circulation et proposent le calcul d'itinéraire. Google-Map ([Map.Google.com](http://Map.Google.com)) ou V-Trafic ([www.v-traffic.com](http://www.v-traffic.com)) couvrent un grand nombre d'agglomérations dans le monde. En Grande Bretagne, une initiative prometteuse a vu le jour, à l'échelle nationale, avec le site Transport Direct ([www.transportdirect.info](http://www.transportdirect.info)) qui recense l'ensemble de l'offre de transport en commun du pays. A l'échelle continentale, en Europe, le projet Eu\_Spirit ([www.vbb-fahrinfo.de](http://www.vbb-fahrinfo.de)) a pour objectif d'unir les systèmes d'information sur les transports en commun de différents pays afin d'offrir un calcul d'itinéraire sans rupture. La Suède, le Luxembourg, le Danemark et l'Allemagne ont actuellement adhéré à ce programme. C'est dans cette optique que s'inscrivent nos travaux de recherche dont l'objectif est de concevoir et de mettre en œuvre un système d'information et de guidage permettant le calcul d'itinéraire porte à porte (entre deux points distincts) selon

différents critères (distances, temps de parcours) en prenant en compte les restrictions de circulation. Ce système d'information couvrant l'ensemble des données, des tests et des résultats est intégré dans un SIG en ligne en utilisant des outils open source.

Notre travail se divise en quatre étapes distinctes. La première étape consiste en l'analyse du réseau routier par la définition de sa géographie et par la collecte de l'ensemble des données descriptives en construisant un Dictionnaire de Données. L'information géographique, quant à elle, est acquise par la procédure de vectorisation manuelle qui génère un fichier de formes (shape file) où seront stockées les caractéristiques géométriques du réseau routier.

La deuxième étape consiste en la modélisation du réseau routier en lui définissant une codification et en élaborant le modèle conceptuel de données (diagramme de classe) fondé principalement sur le modèle topologique. Comme nous avons proposé une amélioration de ce dernier par le passage du modèle topologique classique (2D) vers un modèle topologique 3D développé et proposé par B. Bessaa [5].

La troisième étape consiste en l'élaboration du modèle de calcul du chemin optimal précédée par une étude des problèmes classiques de plus court chemin. Et dans la dernière étape nous développons et nous intégrons ce système dans un SIG en ligne par l'utilisation des outils open source. Cependant, une recherche et une étude des solutions de cartographie en ligne disponibles est nécessaire pour circonscrire l'approche la plus adaptée à notre domaine d'application.

Hormis l'introduction et la conclusion, notre mémoire se présente en cinq chapitres. Le premier chapitre, présente quelques définitions et notions concernant les SIG, l'information géographique et les différentes transformations qu'elle subit au cours de son parcours réalité – Informatisation. Dans le deuxième chapitre nous avons étudié les différentes stratégies de détermination du chemin optimal. Nous abordons, ensuite dans le troisième chapitre l'analyse, la codification et la modélisation du réseau routier selon le modèle topologique 3D. Le quatrième chapitre se propose de présenter les SIG en ligne, leurs utilités, leur architecture ainsi qu'une étude comparative des différentes solutions de la cartographie web. Finalement, dans le dernier chapitre, nous décrivons les étapes de réalisation de notre système en présentant son architecture et en introduisant les outils *open source* utilisés.

## Chapitre I :

### **Information Géographique et Concepts de base dans les SIG :**

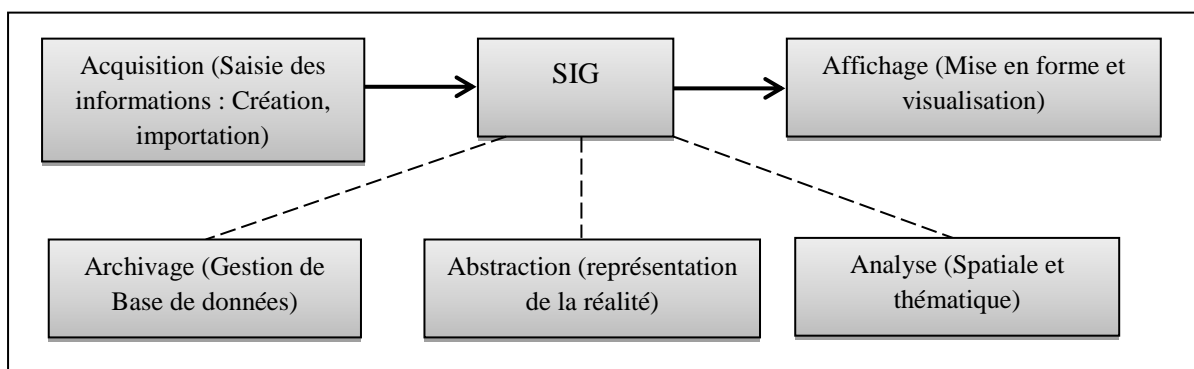
*Il est estimé qu'environ 80% des données stockées dans des bases de données d'entreprises intègrent des informations à caractère géographique [61]. Ces données deviennent de plus en plus stratégiques, quel qu'en soit le domaine.*

*Dans ce premier chapitre, nous commençons par définir ce qui est un Système d'information géographique. Ce qui nous amène ensuite à évoquer l'information géographique et ses différents concepts de base.*

#### **I.1. Système d'Information Géographique (SIG) :**

Depuis une quinzaine d'années, les Systèmes d'Information Géographique (SIG) sont utilisés de façon révolutionnaire grâce aux progrès informatiques et aux possibilités offertes en matière de gestion et d'analyse.

Un SIG est un Système d'Information au sens large capable de répondre à toute la chaîne de traitement des données géographiques. Il se caractérise par cinq fonctionnalités connues sous le nom des 5A (figure 1.1), à savoir : Acquisition, Abstraction, Archivage, Analyse et Affichage des données géographiques [61].



**Figure 1.1 :** Fonctionnalités d'un SIG

Les systèmes d'information géographique servent principalement à :

- Enregistrer l'information sur le territoire (Acquisition, Archivage) ;
- Questionner l'information sur le territoire (Analyse) ;
- Produire des cartographies thématiques (Abstraction, Affichage) ;
- Analyser l'information sur le territoire (Analyse) ;
- Effectuer des simulations (Analyse, Prévision).

## **I.2. Information Géographique :**

### **I.2.1. Définition :**

En informatique, le mot information laisse supposer un certain nombre de définitions : La donnée, l'information et la connaissance, classées par degré d'interprétation et difficulté de partage :

- La donnée, étant un nombre, un texte ou un symbole (en général neutre et indépendante du contexte) est une mesure brute et sans interprétation.
- L'information est généralement dédiée à un sujet ou soumise à un certain degré d'interprétation.
- La connaissance, quant à elle, est une information interprétée par rapport à un contexte particulier, à une expérience ou à un objectif donné.

Le mot géographique associé à l'information en plus d'une thématique (sa nature, son aspect, ses caractéristiques diverses...) un positionnement sur la surface terrestre.

Et en ajoutant la dimension temporelle (ce qui est très utile dans les systèmes décisionnels), nous définissons l'information géographique par « *une représentation d'un objet ou d'un phénomène réel, localisé dans l'espace à un moment donné* » [61].

### **I.2.2. Principales sources de données géographiques :**

On distingue deux principales méthodes d'acquisition de données géographiques [61] :

- Acquisition de l'information par création de données :
  - Relevé de terrain ou levés topographiques, GPS, etc.
  - Enquêtes et recensement, registres administratifs, état civil, etc.
- Acquisition de l'information par importation de données :

- Photographie aérienne et photogrammétrie.
- Télédétection spatiale.
- Cartes scannées et/ou vectorisées.
- Modèles numériques de terrain.

L'acquisition de données peut représenter plus de 80% du coût d'un projet SIG [61].

### I.2.3. Modéliser une information géographique :

Modéliser une information géographique revient à créer un ensemble de règles pour représenter des objets et des comportements du monde réel dans le cadre logique d'un ordinateur. Il existe quatre niveaux de définition/modélisation des données au sein d'un Système d'Information (SI) (tableau 1.1) [6] :

Niveaux	Acteur	Abstraction de la réalité
Niveau externe	Utilisateur	Aucune
Niveau conceptuel	Concepteur	Modélisation conceptuelle de la réalité
Niveau logique	SGBD	Organisation du modèle liée à l'informatisation
Niveau interne (physique)	Système de fichiers	Organisation interne à l'application

**Tableau 1.1 :** Différents niveaux de modélisation de données au sein d'un SI.

En parcourant ces niveaux-là, l'information géographique passe de la réalité vers l'informatisation en adoptant plusieurs formes, des règles de représentations ainsi que différents modèles de données sont dégagés.

#### a. Niveau Conceptuel :

##### 1. De la réalité à la géographie : Création/Définition des objets géographiques.

Un objet géographique est un *objet* comme défini en théorie de l'information « *un ensemble encapsulé d'attributs et de méthodes permettant de décrire la connaissance et le comportement pour une vision contextuelle de la réalité* » [61] comportant trois composantes principales :

- une description sémantique : Des informations textuelles, qualitatives et quantitatives sur une entité.
- une localisation géographique : Des informations décrivant la forme et le positionnement de l'objet sur la surface terrestre, exprimées dans un système de coordonnées explicite.
- et un comportement.

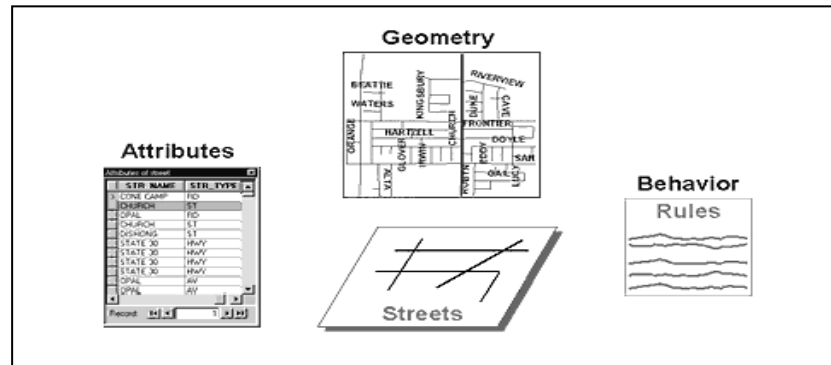


Figure 1.2 : Composantes d'un objet géographique.

En d'autres termes, un objet géographique concrétise la relation entre une définition sémantique (attributs descriptifs) et une précision de la description de la géométrie et de la localisation.

## 2. De la géographie à la géométrie :

C'est la deuxième étape du niveau conceptuel, elle représente la schématisation de la localisation.

On distingue deux types de schématisation :

- Schématisation cartographique classique, où on définit trois géométries chacune progressivement plus complexe et s'appuyant sur la précédente :
  - Le point est utilisé pour définir la localisation des objets ponctuels ou trop petits, par rapport à une échelle donnée.
  - La ligne est utilisée pour représenter les structures linéaires trop étroites pour être représentés par des surfaces comme les routes et les rivières.
  - Le polygone (zone) est utilisé pour représenter les objets surfaciques.

- Schématisation en pixel, où un seul type de géométrie d'objet est défini qui est la maille ou le pixel.

### b. Niveau Logique :

#### De la géométrie à l'informatisation :

A ce niveau, le modèle logique de représentation de données au sein d'un ordinateur est dégagé, on distingue deux types de représentations :

#### - Représentation raster :

Dans ce modèle, on attache tous les attributs descriptifs à la seule géométrie d'objet définie (pixel) (figure 1.3).

Les objets sont différenciés par la valeur radiométrique ou le niveau de gris associé à chaque pixel.

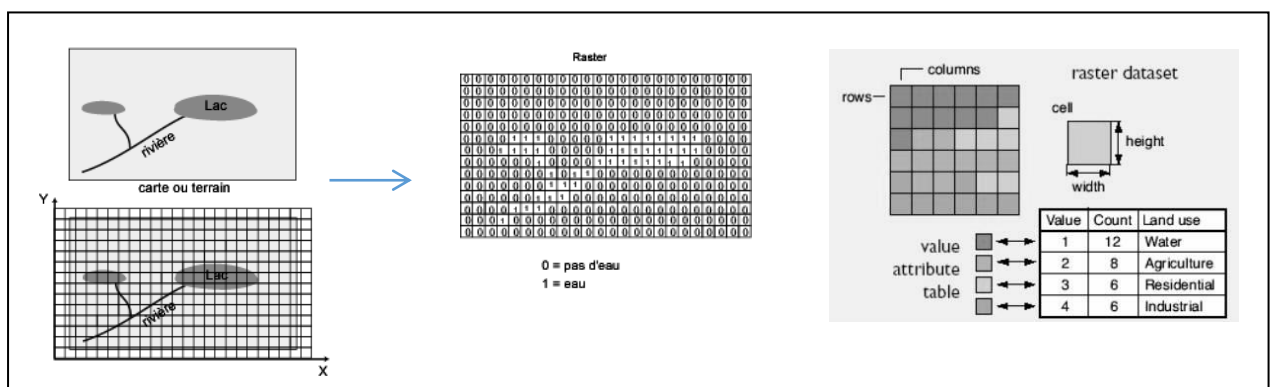


Figure 1.3 : Représentation Raster.

Les données raster remplissent tout l'espace à représenter. La précision est fixée une fois pour toute, ce qui peut induire une dégradation de l'information (en cas de passage à une plus haute résolution).

La mise en œuvre des algorithmes pour les opérations d'analyse est facile, mais la difficulté de l'analyse spatiale reste entière.

#### - Représentation vecteur :

Dans cette représentation on conserve la définition géométrique des objets de la schématisation cartographique classique (point, ligne, polygone), en passant d'une description mathématique (dans  $R^2$  ou  $R^3$ ) à une description informatique simple dans un ensemble discret (figure 1.4).

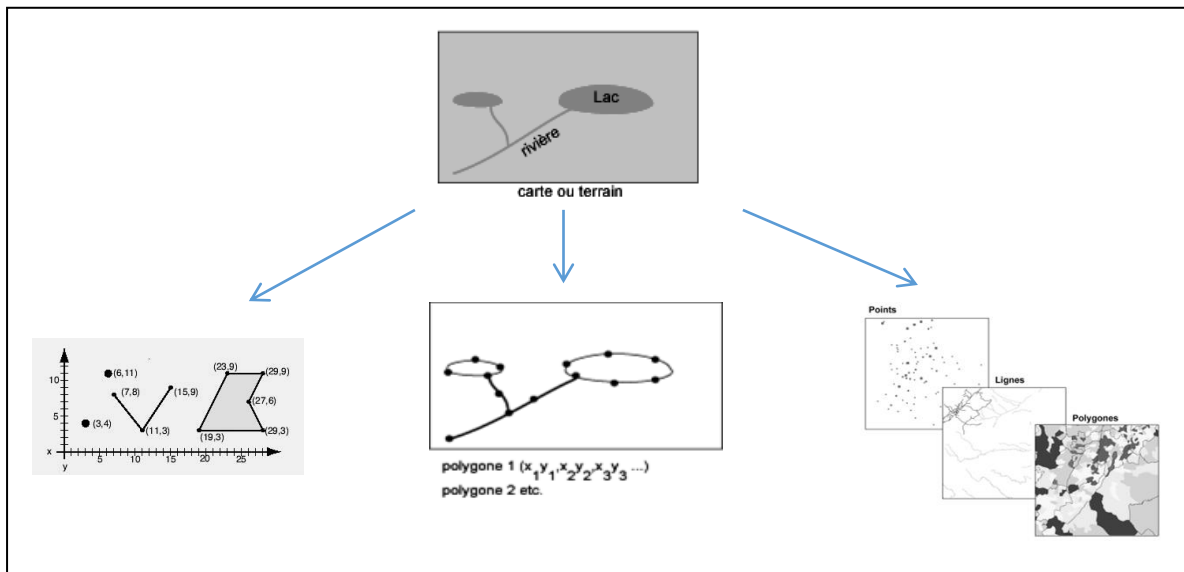


Figure 1.4 : Représentation vecteur.

Dans la représentation en vecteur :

- Les objets du modèle conceptuel ne sont pas modifiés.
- Le rapport graphique-descriptif n'est pas perturbé.
- L'espace de stockage est faible.
- La structure permet l'indexation bidimensionnelle.

Il n'y a pas une unique manière de représentation des géométries (point, ligne, polygone) trois différents types de modèles existent [4] :

- Modèle spaghetti,
- Modèle réseau,
- Modèle topologique.

### 1. Modèle spaghetti

Il est considéré comme le modèle le plus simple. Les entités spatiales sont géométriquement indépendantes entre elles, aucune relation explicite ne les lie. Les géométries peuvent être des points, des lignes ou des polygones. Il n'existe aucune contrainte de positionnement de ces géométries : deux lignes peuvent se couper sans marquer leur point d'intersection ou deux polygones peuvent se chevaucher sans restriction. Un point dans un modèle spaghetti est représenté par une paire de coordonnées appelées abscisse et ordonnée qui représente la

localisation du point dans le système. Les lignes et les polygones sont représentés par des listes des points les constituant (Figure 1.5).

Point = [abscisse, ordonnée]

Arc = [<abscisse, ordonnée>]

Poly = [<abscisse, ordonnée>]

Le modèle spaghetti peut offrir plusieurs points forts comme sa simplicité et sa facilité d'édition. Ses points faibles sont la sauvegarde redondante des zones d'intersection et la difficulté de déterminer les relations topologiques entre entités.

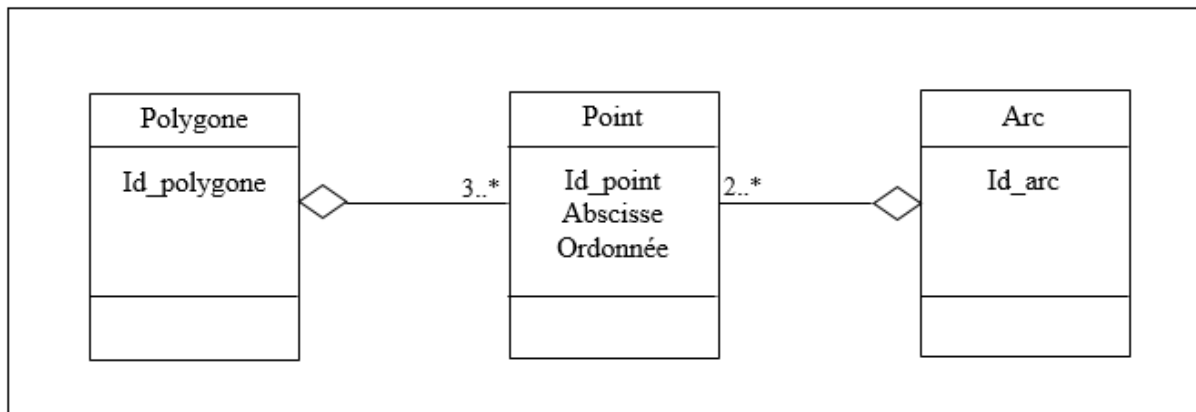


Figure 1.5 : Diagramme de classes du modèle spaghetti.

## 2. Modèle réseau

Les réseaux sont utilisés pour modéliser le transport des personnes et des ressources comme l'eau, l'électricité, le gaz et les télécommunications. Les réseaux sont une collection unidimensionnelle des points et des lignes topologiquement interconnectés [4]. Les points sont appelés nœuds et les lignes sont appelés arcs. Un arc connecte deux nœuds (nœud d'origine et nœud de destination). Les réseaux facilitent la représentation de flot le long des arcs (comme les routes et les rivières) et à travers les points d'intersections.

Les attributs des éléments de réseaux (nœuds et arcs) peuvent être utilisés dans des algorithmes de synthèse comme l'algorithme de plus court chemin ou l'algorithme du flot maximum. Un nœud est représenté par ses coordonnées et la liste des arcs sous-tendus. Un arc est représenté par son nœud d'origine, son nœud de destination et la liste des points intermédiaires. La représentation d'un polygone dans un modèle réseau est la même que celle du modèle spaghetti (figure 1.6).

Nœud = [abscisse, ordonnée, <Arc sous-tendue>] .

Arc = [nœud origine, nœud destination, <abscisse, ordonnée>] .

Polygone = [<abscisse, ordonnée>].

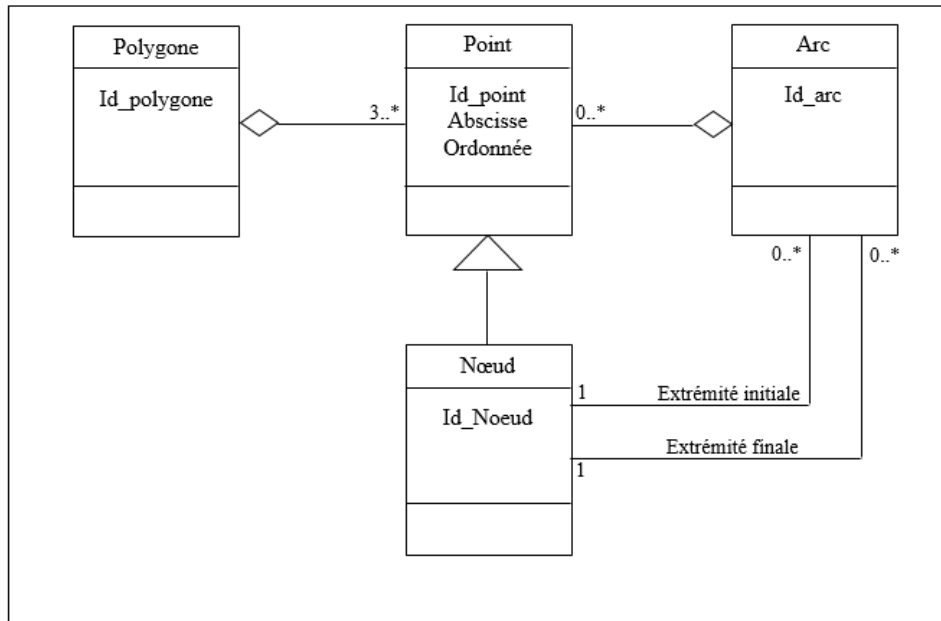


Figure 1.6 : Diagramme de classes du modèle Réseau.

### 3. Modèle topologique 2D

La topologie concerne les relations géométriques entre objets graphiques (adjacence, fermeture, connectivité...). Elle permet de :

- Décrire des relations spatiales entre éléments géométriques.
- Maintenir la cohérence d'un ensemble.
- Faciliter les algorithmes d'analyse spatiale.

Le modèle topologique enregistre explicitement les relations topologiques comme l'adjacence et la connectivité. Un nœud est représenté par ses coordonnées. Un arc est représenté par ses deux nœuds d'extrémités, ses polygones voisins (gauche et droit) et la liste des points intermédiaires. Un polygone est représenté par la liste des arcs le constituant (figure 1.7).

Nœud = [abscisse, ordonnée, liste Arc sous-tendue]

Arc = [nœud origine, nœud destination, polygone droit, polygone gauche, <abscisse, ordonnée>]

Polygone = [<Arc>]

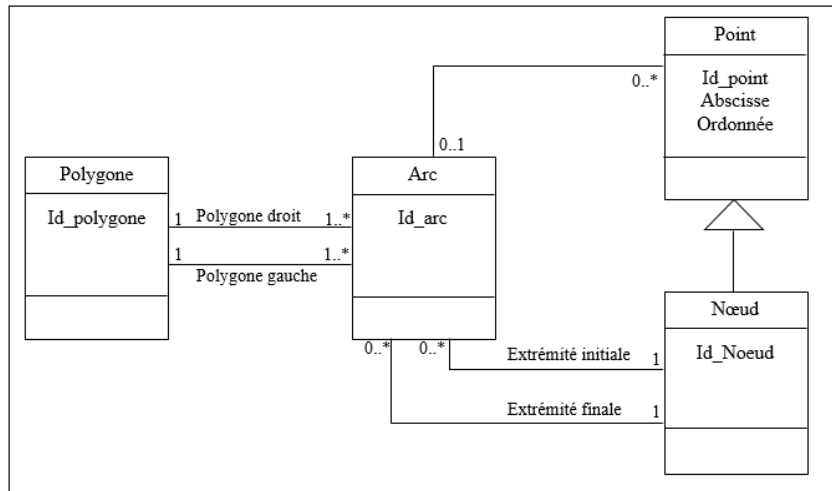


Figure 1.7 : Diagramme de classes du modèle topologique.

**c. Niveau interne :**

Le modèle interne est la manière de stocker, dans des fichiers, la description logique d'un ensemble d'objets géométriques, en assurant une certaine cohérence [61].

Afin d'avoir une vision globale, nous résumons dans le diagramme ci-dessous (figure 1.8) les états et les transitions de l'information géographique en passant à travers les différents niveaux de modélisation :

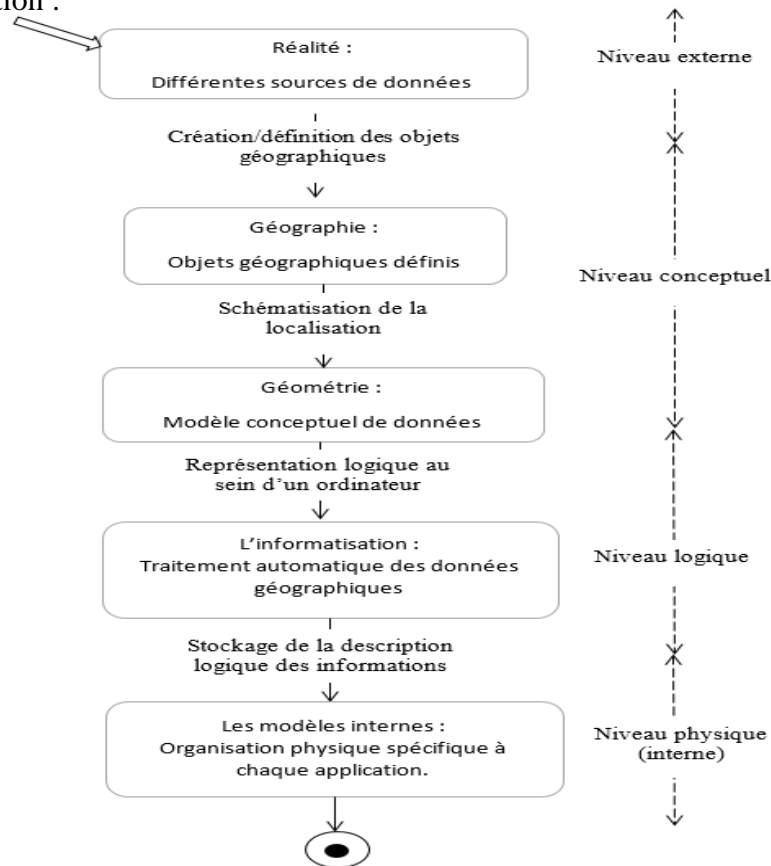


Figure 1.8 : Etat/Transition de l'information Géographique à travers les niveaux de modélisation dans un SI

#### I.2.4. Mesure et représentation de la localisation dans une information géographique :

L'attribut de localisation assure l'objectif de mise en relation des objets géographiques, ces derniers nécessitent une représentation topographique de la surface de la terre (géoréférencement), ce qui impose un référentiel commun et des précisions connues pour cet attribut.

##### 1. Datum : La forme et la position de la terre

C'est la définition d'une forme de référence pour décrire la position d'un point par des coordonnées sphériques (longitude, latitude, altitude) cette forme est appelée ellipsoïde.

Il existe de nombreux modèles d'ellipsoïdes caractérisés par leur position dans l'univers (Centre et inclinaison) ex : *WGS84* (World Geodesic System of 84).

Toutes les coordonnées des objets d'une même base de données doivent être exprimées dans le même système pour pouvoir être comparées.

##### 2. Projection : représentation d'un Plan sur un ellipsoïde

C'est une opération mathématique permettant de représenter un plan sur une portion de l'ellipsoïde à l'aide d'un système de coordonnées géographiques basé sur des lignes de latitude et de longitude, en estimant les déformations induites par cette opération sur les distances, les angles, les direction, les surfaces... [61]

#### I.2.5. Organisation de l'information géographique dans un SIG :

Un SIG organise les informations géographiques sous forme de thèmes (Appelés aussi couches, calques, plans, ...).Ce sont des collections homogènes d'objets. Chaque couche représente un ensemble de données similaires.

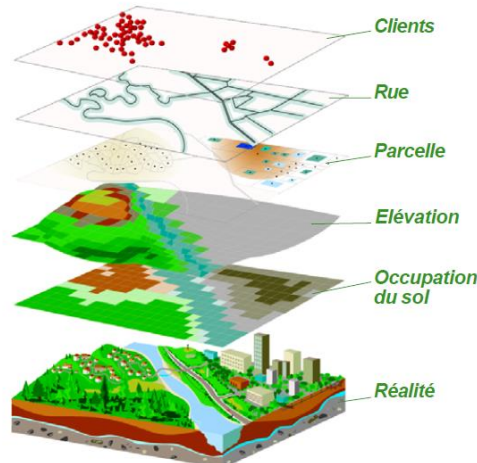


Figure 1.9 : Organisation de l'Information Géographique dans un SIG

La plupart des relations spatiales entre les couches peuvent être déduites facilement de leur emplacement géographique commun.

### **I.3. Conclusion :**

Dans ce premier chapitre, nous avons jugé utile d'étudier les principaux concepts de base des SIG en se focalisant sur l'information géographique et ses différents aspects dans le but de déterminer les outils nécessaires ainsi que la méthode la plus adaptée pour mener à bien notre étude théorique du problème de recherche de plus court chemin, notre analyse du réseau routier et par la suite notre modélisation du problème de recherche des itinéraires routiers dans un SIG.

## **Chapitre II :**

### **Détermination de chemins optimaux**

*Les problèmes de calcul de chemin optimal sont classiquement traités à l'aide de stratégies de recherche de chemins dans un graphe. La recherche d'un plus court chemin fait partie des problèmes d'optimisation dans les graphes les plus étudiés. Les méthodes pionnières de résolution datent des années 1950 [7]. Encore aujourd'hui, le sujet fait l'objet d'une littérature abondante, qui présente une volonté d'adapter les algorithmes classiques aux spécificités des problèmes rencontrés en pratique, notamment à des structures de graphe particulières et à la nature des coûts.*

*Dans ce chapitre nous nous intéressons à l'optimisation d'itinéraire dans les réseaux routiers. De ce fait, une synthèse bibliographique sur des problèmes classiques de recherche de plus court chemin est faite. Les méthodes classiques de résolution qui procèdent par étiquetage des nœuds du graphe ainsi que des méthodes d'accélération de ces algorithmes sont exposées. Le support de déplacement utilisé est un réseau routier représenté par un graphe. Dans ce réseau, il existe des restrictions de déplacement qui doivent être prise en compte pour l'optimisation des itinéraires : interdiction de tourner à gauche interdiction de faire demi-tour, le poids et la hauteur maximaux autorisés...etc. nous verrons, ainsi, que le problème de calcul d'itinéraires dans le réseau routier se formule comme un problème de plus court chemins avec séquences d'arcs pénalisées.*

## II.1. Notations :

Dans la suite de ce chapitre, nous utilisons les notations suivantes : dans un graphe orienté  $G = (N, A)$  où  $N$  est l'ensemble des nœuds, de cardinalité  $n$ , et  $A$  l'ensemble des arcs, de cardinalité  $a$ :

$\delta^+(u) = \{v \in N \mid (u, v) \in A\}$ : ensemble des nœuds successeurs du nœud  $u$ ,

$\delta^-(u) = \{v \in N \mid (v, u) \in A\}$ : ensemble des nœuds prédécesseurs du nœud  $u$ ,

$FS(u) = \{(u, v) \in A\}$ : ensemble des arcs successeurs du nœud  $u$ ,

$BS(u) = \{(v, u) \in A\}$ : ensemble des arcs prédécesseurs du nœud  $u$ ,

$O(e) = u$  : nœud origine de l'arc  $e = (u, v)$ ,

$D(e) = v$  : nœud destination de l'arc  $e = (u, v)$ .

## II.2. Algorithmes de recherche de plus court chemin

Dans un graphe  $G = (N, A)$ , on considère un coût  $c_{uv} \in R$  associé à chaque arc  $(u, v) \in A$ . Ce coût peut représenter une distance, un temps de parcours, un coût financier ou un coût généralisé combinant plusieurs critères...etc. Le problème de plus court chemin entre deux nœuds  $o$  et  $d$  de  $G$  cherche un chemin dont la somme des coûts associés aux arcs est minimum (figure 2.1) [7].

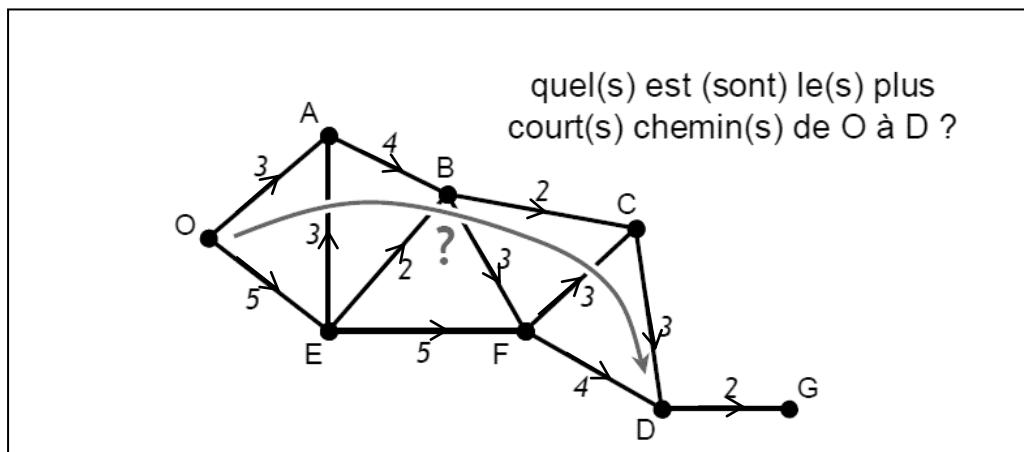


Figure 2.1 : Problème de recherche de plus court chemin.

Les premiers algorithmes de plus court chemin proposés dans la littérature suivent une stratégie d'étiquetage des nœuds du graphe. Ils consistent à trouver l'ensemble des plus courts chemins entre un nœud origine  $o$  et chaque nœud du graphe, ce qui revient à trouver l'arbre de racine  $o$  des plus courts chemins. Ce problème admet une solution si et seulement si le graphe ne comporte aucun cycle de coût négatif. C'est le cas de la plupart des graphes

représentant des réseaux routiers, les coûts étant dans la grande majorité des cas des valeurs positives (distances, temps de parcours).

Le problème de recherche de l'arbre GT des plus courts chemins peut être écrit sous la forme d'un programme linéaire (équation (2.1)) [7]:

$$\left\{ \begin{array}{l} \text{Minimiser } \sum_{(u,v) \in A} c_{uv} x_{uv} \\ \text{s.c. } \sum_{v \in \delta^-(u)} x_{vu} - \sum_{v \in \delta^+(u)} x_{uv} = \begin{cases} -(n-1) & \text{si } u = o \\ 1 & \text{sinon} \end{cases} \\ x_{uv} \in \mathbf{R}, \forall (u,v) \in A \end{array} \right. \quad (2.1)$$

Où  $x_{uv}$  correspond au nombre de chemins de GT empruntant l'arc  $(u, v)$ .

À ce problème primal correspond un problème dual qui s'écrit de la façon suivante (équation (2.2)) [7]:

$$\left\{ \begin{array}{l} \text{Maximiser } (1-n)\pi_o + \sum_{v \neq o} \pi_v \\ \text{s.c. } \pi_v - \pi_u \leq c_{uv}, \forall (u,v) \in A \end{array} \right. \quad (2.2)$$

Où les  $\pi_u, u \in N$  sont les variables duales des  $x_e, e \in A$ .

La contrainte  $\pi_v - \pi_u \leq c_{uv}$  est appelée *Condition de Bellman*.

### II.2.1. Algorithmes à fixation/correction d'étiquettes

Les algorithmes classiques pour résoudre le problème de recherche de plus court chemin suivent une approche primale. Leur schéma consiste en la construction itérative d'une arborescence couvrante GT de racine  $o$  et à mettre à jour les valeurs de potentiel  $\pi_u$  associées aux nœuds  $u$  de GT. Le potentiel  $\pi_u$  représente une borne supérieure du coût du plus court chemin de  $o$  à  $u$ . La procédure algorithmique teste, à chaque itération, la condition de Bellman  $\pi_v - \pi_u \leq c_{uv}$ , pour un nœud  $u$  et pour tous les arcs  $(u,v) \in FS(u)$  et met éventuellement à jour les potentiels  $\pi_u$ .

On appelle cette opération *l'exploration du nœud u*. Le nœud  $u$  est choisi parmi un ensemble de nœuds candidats  $Q$ . Si un arc  $(u, v)$  ne respecte pas la condition de Bellman, le potentiel  $\pi_u$  est mis à jour par la valeur  $\pi_u + c_{uv}$  et  $v$  est inséré dans  $Q$ . On dit alors que  $v$  est étiqueté

depuis  $u$ .  $pred_v$ , qui représente le prédécesseur du nœud  $v$  dans le plus court chemin de  $o$  à  $v$  trouvé jusqu'ici, pointe alors sur  $u$ . On n'arrête la procédure algorithmique que lorsque les étiquettes de potentiel constituent une solution duale réalisable, c'est-à-dire lorsque la condition de Bellman est vérifiée pour tous les arcs du graphe. L'arbre des plus courts chemins est alors reconstitué à partir des étiquettes désignant les prédécesseurs.

Les algorithmes qui suivent ce schéma sont appelés algorithmes d'étiquetage. Ils sont basés sur le principe de la programmation dynamique, selon lequel si le plus court chemin entre deux nœuds  $u$  et  $v$  passe par un nœud  $w$ , alors il correspond à la concaténation de deux plus courts chemins, entre  $u$  et  $w$  et entre  $w$  et  $v$ .

On distingue les méthodes dites à *fixation d'étiquettes* qui à chaque itération choisissent d'examiner le nœud de  $Q$  (ensemble de nœuds candidats) ayant le plus petit potentiel, des méthodes dites à *correction d'étiquettes* qui choisissent d'examiner un nœud de  $Q$  choisi selon une autre règle.

Dijkstra [14] a proposé le premier algorithme à fixation d'étiquettes. Dans sa version la plus simple, cet algorithme implémente  $Q$  comme une liste chaînée non ordonnée. Plusieurs modifications de cet algorithme ont été proposées, implémentant par exemple  $Q$  comme un tas binaire (algorithme S-heap [35]) ou comme une structure de buckets (algorithme S-bucket[12]). Ces méthodes ne permettent de résoudre le problème de plus court chemin que sur des graphes à coûts positifs. Elles présentent en effet la particularité que lorsqu'un nœud est retiré de  $Q$ , son potentiel n'est plus modifié au cours de l'algorithme.

Les algorithmes à correction d'étiquettes sélectionnent un nœud candidat au moyen de stratégies différentes. Bellman [3], Ford [18] et Moore [43] proposent un algorithme basé sur la règle de sélection FIFO. Cet algorithme implémente  $Q$  comme une file. Une structure appelée *deque*, qui est vue comme une pile  $Q'$  et une file  $Q''$  connectées en série, peut également être utilisée : la première fois qu'un nœud est inséré dans la structure, il est ajouté à la fin de  $Q''$  ; quand le même nœud, après avoir été retiré de  $Q$ , devient candidat à nouveau, il est ajouté à la tête de  $Q'$  ; les nœuds sont retirés de la tête de  $Q'$ , si  $Q'$  n'est pas vide ; sinon, le premier élément de  $Q''$  est retiré. L'algorithme correspondant, appelé L-deque, a prouvé son efficacité en pratique particulièrement sur des graphes peu denses et presque planaires [49], ce qui est le cas de beaucoup de graphes représentant des réseaux routier.

Les algorithmes d'étiquetage de nœuds sont conçus pour résoudre le problème **one-to-all**, autrement dit, entre une origine et l'ensemble des nœuds destinations possibles dans le graphe.

On peut considérer trois autres types de problèmes de plus court chemin. Le problème **all-to-one** cherche l'arbre des plus courts chemins entre tous les nœuds du graphe et une destination. Il peut être résolu en appliquant les algorithmes conçus pour le problème one-to-all vers l'arrière, c'est-à-dire en partant de la destination et en inversant le sens des arcs.

Le problème **one-to-one** cherche le plus court chemin entre une origine  $o$  et une destination  $d$ . Il est résolu par les mêmes algorithmes que le problème one-to-all. Les algorithmes à fixation d'étiquettes sont particulièrement adaptables pour ce problème puisqu'ils peuvent être stoppés après l'exploration du nœud destination  $d$ , dont le potentiel est alors définitif. Dans ce cas, l'arbre des plus courts chemins n'est pas construit dans sa totalité et le temps d'exécution de l'algorithme est réduit.

Le problème **all-to-all** cherche le plus court chemin entre chaque paire de sommets du graphe et est résolu par des algorithmes spécifiques (algorithme de Floyd[7] par exemple) ou par plusieurs applications (une pour chaque origine) d'un algorithme adapté au problème one-to-all.

### II.2.2. Accélération des algorithmes d'étiquetage

Nous venons de voir que le problème de plus court chemin peut être résolu par des algorithmes d'étiquetage, dont un état de l'art très complet peut être trouvé dans [49]. Les temps d'exécution de ces algorithmes peuvent être importants sur des instances de grande taille et dans le cadre d'applications devant fournir des solutions en temps réel, il peut être souhaitable d'accélérer leur exécution. Les techniques d'accélération proposées dans la littérature concernent, généralement, le problème **one-to-one** entre une origine  $o \in N$  et une destination  $d \in N$ . En effet, les méthodes à fixation d'étiquettes ne sont pas très efficaces pour ce problème, puisqu'elles effectuent une recherche en cercle autour de l'origine. L'ajout d'une information heuristique permet de limiter le nombre de nœuds examinés dans le graphe. Fu et al. [19] proposent un état de l'art des méthodes heuristiques appliquées au problème de plus court chemin **one-to-one**. Parmi ces méthodes, certaines peuvent être appliquées directement sur le graphe et d'autres nécessitent un précalcul.

Les méthodes d'accélération qui ne nécessitent aucun précalcul cherchent à réduire la zone explorée en décomposant le problème en sous-problèmes qui peuvent être résolus parallèlement ou en utilisant une information directionnelle telle que la stratégie  $A^*$ [41,44]. L'algorithme alloue aux nœuds candidats une priorité d'autant plus forte que la probabilité qu'ils appartiennent à un plus court chemin est élevée. Dans cet algorithme, les nœuds ne sont plus classés dans  $Q$  en fonction de leur potentiel  $\pi_u$  mais en fonction de la valeur  $\pi_u + D_{u,d}$

où  $D_{u,d}$  est une estimation du coût du plus court chemin entre  $u$  et  $d$ . Les nœuds  $u$  examinés avant l'arrêt de l'algorithme satisfont l'équation (2.3) :

$$\pi_u + D_{u,d} \leq \pi_d \quad (2.3)$$

A\* fournit une solution optimale si la fonction heuristique  $D_{u,d}$  ne surestime jamais le coût minimal d'un chemin entre  $u$  et  $d$  et respecte la propriété suivante (équation (2.4)):

$$\forall (u, v) \in A, D_{u,d} - D_{v,d} \leq c_{uv} \quad (2.4)$$

Une adaptation de l'algorithme A\* a été proposée par Sedgewick et Vitter [60]. Lorsque les coûts correspondent à des temps de parcours, on peut utiliser la distance à vol d'oiseau entre  $u$  et  $d$  et la vitesse maximale de déplacement observée sur l'ensemble des arcs du graphe pour construire l'estimation  $D_{u,d}$ .

La recherche bidirectionnelle constitue une autre stratégie d'accélération des algorithmes d'étiquetage. Celle-ci décompose le problème de plus court chemin en sous-problèmes. Elle déploie un étiquetage depuis l'origine et la destination simultanément [19]. Le critère d'arrêt de cette méthode est fourni par Nicholson [48] (équation (2.5)) :

$$\pi_u^o + \pi_u^d \leq \min_{v \in N} \{ \pi_v^o \} + \min_{v \in N} \{ \pi_v^d \} \quad (2.5)$$

Où  $\pi_v^o$  et  $\pi_v^d$  représentent respectivement le potentiel du nœud  $v$  étiqueté depuis  $o$  et depuis  $d$ . L'utilisation de nœuds intermédiaires ayant une forte probabilité d'appartenir au plus court chemin permet aussi d'accélérer les stratégies d'étiquetage, et ce, en considérant plusieurs problèmes de plus court chemin successifs : entre l'origine et le point intermédiaire puis entre le point intermédiaire et la destination [24].

On peut également accélérer les algorithmes de recherche de plus courts chemins au moyen d'une recherche par niveau hiérarchique du graphe. Cette stratégie est basée sur l'opération réalisée naturellement par un conducteur en cherchant sa route sur une carte routière : il commence par chercher un axe majeur passant à proximité de son origine et de sa destination puis il cherche des connexions secondaires lui permettant de relier cet axe aux points origine et destination [7]. Dans les réseaux routiers, une hiérarchisation naturelle du réseau repose sur les vitesses moyennes pratiquées sur les différents arcs ou encore sur la longueur de ces arcs. La méthode d'étiquetage associée à la hiérarchisation est basée sur une modification de l'algorithme de Dijkstra, qui considère les nœuds de niveau faible seulement lorsque ceux-ci

sont proches de l'origine ou de la destination. Une heuristique de ce type, qui ne garantit plus l'optimalité, peut se révéler intéressante lorsque des stratégies de régulation tendent à favoriser la circulation sur les axes principaux aux dépens des axes secondaires [7].

Les méthodes qui utilisent un prétraitement cherchent à transférer une partie de la complexité du problème dans une phase initiale de calcul. Cette phase se révèle relativement longue mais elle n'est relancée que lorsque le graphe (structure ou coûts) est modifié.

Une méthode d'accélération basée sur la notion de *conteneur géométrique* a été proposée par Sedgewick et Vitter [60]. Les nœuds du graphe sont munis de coordonnées dans le plan. Pour chaque arc, la phase de prétraitement détermine un objet géométrique contenant l'ensemble des nœuds pouvant être atteints par un plus court chemin commençant par cet arc. Au cours du processus d'étiquetage, tout arc dont l'objet géométrique associé ne contient pas le nœud destination est ignoré.

Möhring et al. [42] proposent, également pour un graphe dont les nœuds sont munis de coordonnées géographiques, de travailler à partir d'une partition géométrique  $I$  du graphe. Pour chaque élément  $i$  de la partition et pour chaque arc  $e = (u, v) \in A$ , ils définissent deux valeurs booléennes associées aux deux extrémités de l'arc  $e$ :  $f_{e,i}^v$  (resp.  $f_{e,i}^u$ ) qui vaut 1 si et seulement s'il existe un plus court chemin partant du nœud  $v$  (resp.  $u$ ) passant par  $e$  et par la région  $i$ . Dans l'algorithme d'étiquetage, l'exploration d'un nœud  $u$  est modifiée de la manière suivante : pour tout nœud  $v \in \delta^+(u)$ , l'étiquetage n'est réalisé que si  $e: f_{(u,v)i^*}^u = 1$ , où  $i^*$  est la région à laquelle appartient la destination du chemin.

Goldberg et Harrelson [23] proposent une variante de l'algorithme  $A^*$  appelé l'algorithme ALT ( $A^*$ , Landmarks with Triangle inequality), il utilise des bornes inférieures précalculées du temps de parcours entre le nœud courant et des nœuds de référence (*landmarks*) pour modifier le potentiel des nœuds étiquetés. Étant donné un petit nombre de nœuds de référence dont l'ensemble est noté  $L \subset N$ , l'inégalité triangulaire permet de calculer une borne inférieure du coût entre n'importe quel nœud  $u \in N$  et la destination  $d \in N$ . En effet,  $\forall l \in L, \forall u, v \in N$  et pour une fonction heuristique  $D_{u,v}$  qui soit une borne inférieure du coût minimum d'un chemin entre  $u$  et  $v$  respectant l'équation 2.4, on a (équation (2.6)):

$$D_{u,v} + D_{v,l} \geq D_{u,l} \text{ et } D_{l,u} + D_{u,v} \geq D_{l,v} \quad (\text{inégalité triangulaire}) \quad (2.6)$$

Par conséquent, pour  $v = d$ , on obtient que  $\max_{l \in L} \{ \max(D_{u,l} - D_{d,l}, D_{l,d} - D_{l,u}) \}$  est une borne inférieure du coût du plus court chemin entre  $u$  et  $d$ . La qualité de cette borne inférieure dépend étroitement du choix des nœuds de  $L$ , qui peut être réalisé au moyen de différentes

heuristiques présentées notamment par Goldberg [25]. L'idée générale de ces heuristiques est d'utiliser des nœuds de référence dont la dispersion dans le graphe est maximale. La phase de précalcul comprend donc le choix des nœuds de référence  $l \in L$  ainsi que le calcul des coûts des plus courts chemins entre  $u$  et  $l$ ,  $\forall u \in N$  et  $\forall l \in L$ .

Afin de présenter une vue claire des différents algorithmes cités nous les résumons à travers le schéma suivant (figure 2.2) :

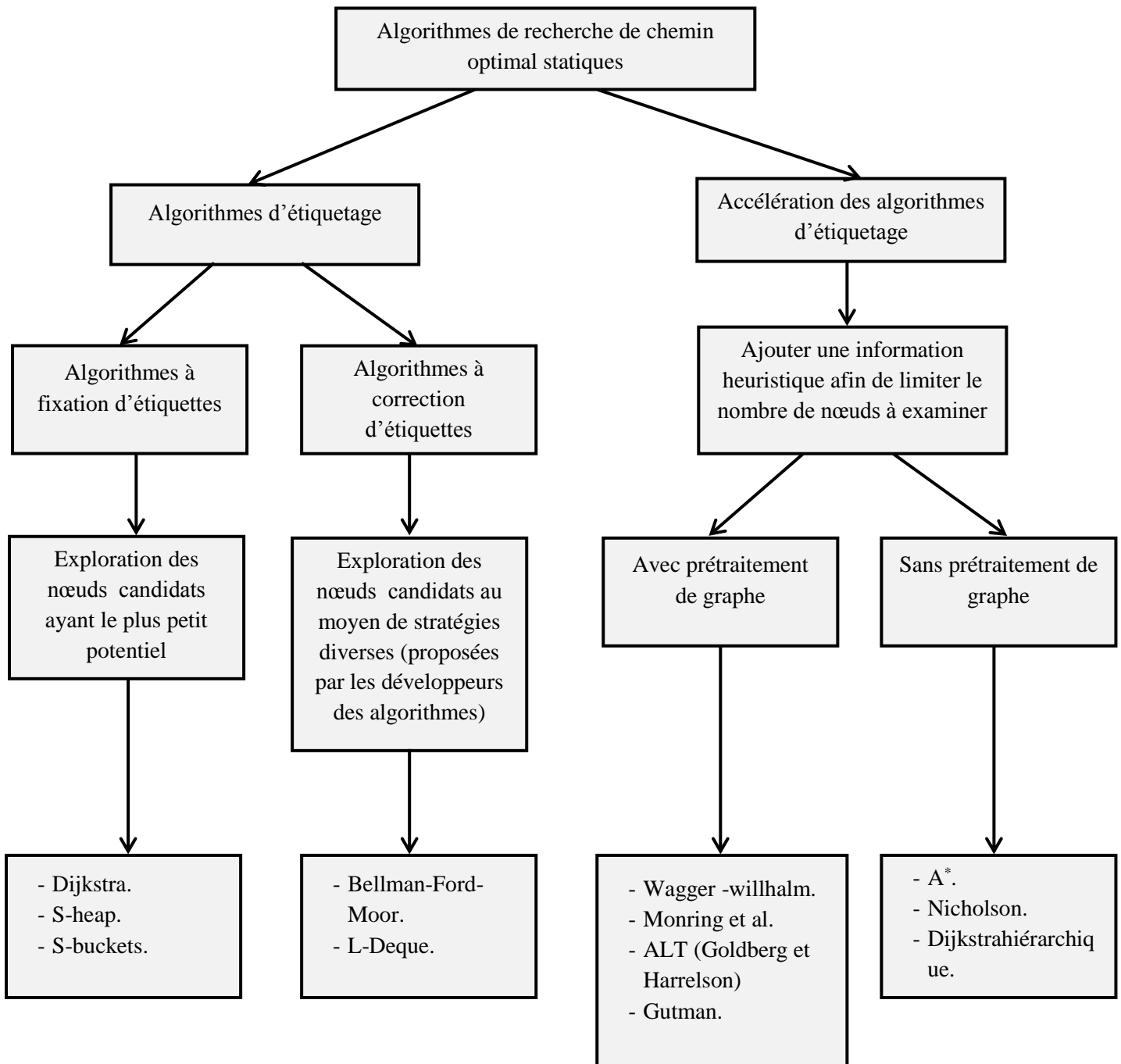


Figure 2.2 : Différents algorithmes de recherche de plus court chemin

### II.3. Modélisation des mouvements directionnels routiers :

Nous nous intéressons dans notre travail à l'optimisation d'itinéraire dans le réseau routier. Dans ce réseau, nous devons prendre en compte les restrictions de déplacement pour l'optimisation des itinéraires (interdiction de tourner à gauche, interdiction de faire demi-tour, poids maximal autorisé...) (figure 2.3 a).

Par ailleurs, les mouvements directionnels autorisés -dans le cas de recherche de plus court chemin selon le temps de parcours- font le plus souvent l'objet d'un temps de traversée qui peut dépendre de la direction du mouvement (vers la gauche, vers la droite ou tout droit) et doit être pris en compte dans le temps de parcours. Le problème de calcul d'itinéraire dans le réseau routier se formule alors comme *un problème de plus court chemin avec séquences d'arcs pénalisées*.

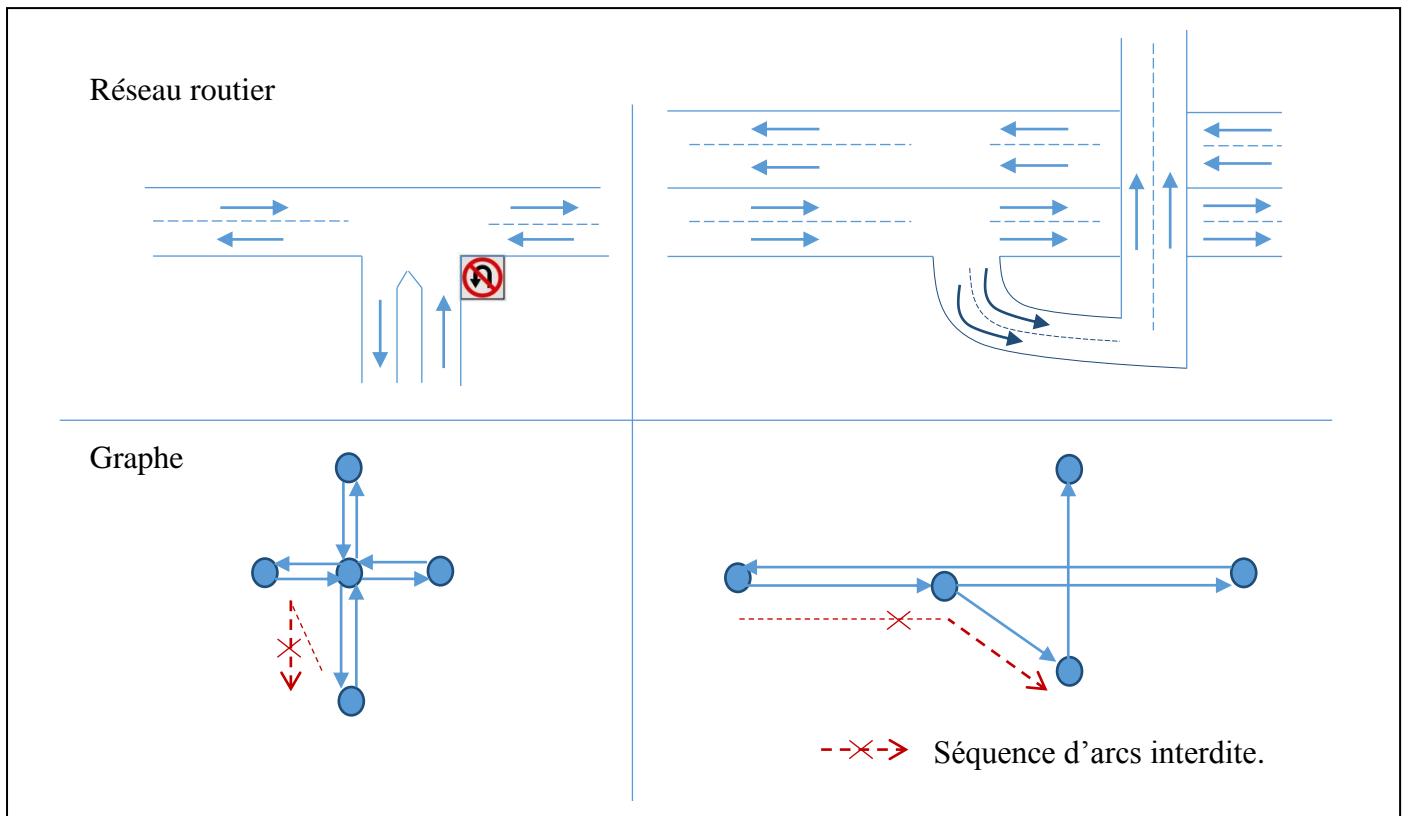
Dans le graphe servant de support au problème de recherche d'itinéraires routiers, les tronçons de route sont représentés par des arcs et les intersections par des nœuds. Les critères d'optimisation retenus sont :

- La minimisation de la distance parcourue
- La minimisation du temps de parcours

La longueur d'un itinéraire correspond à la somme des longueurs associées aux tronçons de route. Le temps de parcours, quant à lui, correspond à la somme des temps de parcours des tronçons de route et des temps associés à la traversée des intersections, variables selon la nature du mouvement directionnel (tourne à gauche, tourne à droite, tout droit...).

Les mouvements directionnels interdits sont représentés par des séquences de deux arcs auxquelles on associe un coût infini (figure 2.3 b).

Les mouvements directionnels sont représentés par des séquences de deux arcs dans la majorité des bases de données actuellement disponibles pour décrire le réseau routier (Navteq, TeleAtlas...) [7]. Le problème de plus court chemins avec coût associés à des séquences d'arcs a été largement traité dans la littérature, connus sous le nom de *Problème de plus court chemin avec séquences d'arcs pénalisées* [7].



**Figure 2.3:** Exemples de mouvements directionnels interdits : Mouvement interdit par le code de la route : a- demi-tour interdit (à gauche) et b- mouvement directionnel interdit en cas de dépassement de poids ou de hauteur autorisés (à droite).

#### II.4. Détermination de chemins optimaux avec séquences d'arcs pénalisées :

Soit un graphe  $G(N,A)$  dans lequel des coûts  $c_e$  sont associés aux arcs  $e \in A$ . Des coûts  $\sigma_l > 0$  sont associés à certaines séquences  $l$  de deux arcs consécutifs, dont l'ensemble est noté  $C$ .  $C$  est en fait un ensemble de chemins de  $G$ .

Un chemin  $l \in C$  de coût  $\sigma_l = \infty$  est un chemin interdit. Le coût d'un chemin solution  $P$  correspond à la somme des coûts des arcs et des coûts associés aux sous-chemins  $l \in C$  tel que  $l \subset P$ . Un chemin comportant un sous chemin interdit a donc un coût infini.

Les méthodes classiques d'étiquetage, présentées au début de ce chapitre, sont uniquement capables de prendre en compte des coûts portants sur les arcs du graphe. Il est donc nécessaire de chercher des solutions qui prennent en compte également les coûts portants sur des séquences d'arcs.

### II.4.1. Approche de résolution naïve :

Nous étudions le problème de plus court chemin avec sous chemin pénalisé sans contraindre le chemin solution à être élémentaire. Donc, le chemin solution peut comporter plusieurs fois le même nœud. En effet, dans une application de guidage les chemins solution peuvent comporter des cycles [7] (figure 2.4).

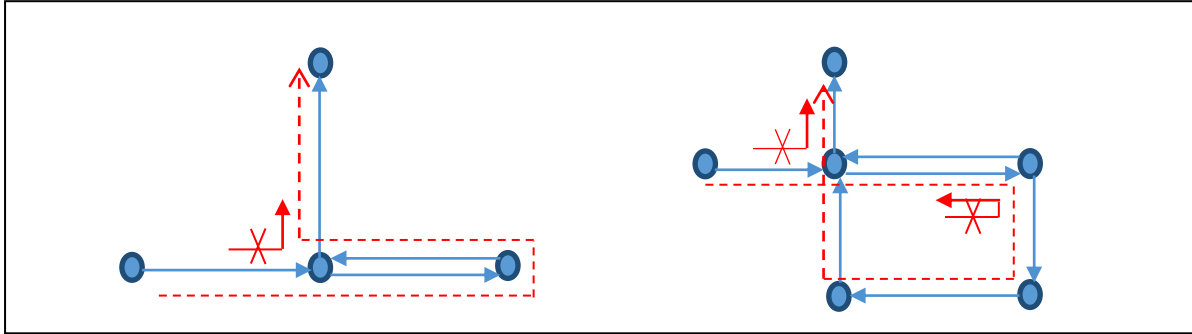


Figure 2.4 : Exemples de solutions incluant des cycles

Dans le problème de plus court chemin avec mouvements pénalisés, on note  $\sigma_e$  le coût associé au parcours de deux arcs consécutifs  $e = (u,v)$  et  $f = (v,w)$  [7]. Une approche naïve de résolution de ce problème consiste à modifier les stratégies classiques d'étiquetage, permettant de trouver un plus court chemin entre un nœud  $o$  et tous les nœuds du graphe, de la manière suivante : lors de l'examen d'un nœud  $v$  ayant pour prédécesseur le nœud  $u$  dans le plus court chemin entre  $o$  et  $v$  trouvé jusqu'ici, la mise à jour des potentiels des nœuds  $w \in \delta^+(v)$  est réalisée grâce à la formule (équation (2.7)) :

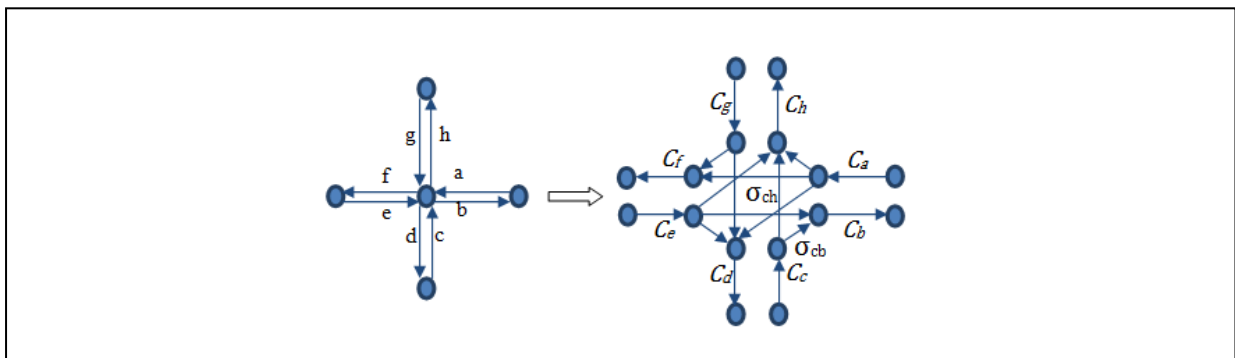
$$\pi_w \leftarrow \min(\pi_w, \pi_v + \sigma_{(u,v)(v,w)} + C_{vw}), \forall w \in \delta^+(v). \quad (2.7)$$

Namkoong et al. [46] mettent en évidence qu'une telle stratégie conduit à la construction d'une structure arborescente, par définition sans cycles. Or, dans le contexte des réseaux routiers, les solutions optimales peuvent comporter des cycles. Un tel algorithme fournit donc des solutions sans garantir l'optimalité.

Les méthodes exactes développées dans la littérature pour résoudre le problème de plus court chemin avec coûts associés à des séquences d'arcs sont de deux natures [7] : les méthodes qui travaillent sur une expansion  $G'$  du graphe  $G$  et celles qui travaillent sur le graphe  $G_D$ , dual de  $G$ .

### II.4.2 Expansion du graphe

L'expansion du graphe  $G$  au niveau des nœuds consiste à construire un graphe  $G'$ , dans lequel chaque nœud est reproduit autant de fois qu'il a d'arcs incidents dans  $G$ . Des arcs fictifs relient les arcs  $e$  et  $f$  dont le parcours consécutif est autorisé (c'est-à-dire que  $\sigma_{ef} < \infty$ ) et le coût associé à la séquence qu'ils représentent leur est attribué (figure 2.5). Cette modélisation a été largement utilisée dans les applications sur les réseaux routiers, notamment par Gutierrez et Medaglia [30] pour représenter les délais de traversée des intersections dans un réseau routier ou par Schulz [59] pour représenter les délais de correspondance dans les réseaux de transport en commun. Elle est coûteuse en espace mémoire à cause de la grande taille de  $G'$  et en temps de calcul, à la fois pour la construction de  $G'$  et pour l'application de l'algorithme d'étiquetage. Elle ne peut donc être utilisée que sur de petites instances de graphe.



**Figure 2.5 :** Expansion du graphe au niveau d'une intersection avec interdiction sur les mouvements tournant vers la gauche (c,f) et (g,b).

### II.4.3. Passage au graphe dual :

D'autres travaux considèrent le graphe dual de  $G$ , noté  $GD$ . Les nœuds de  $GD$  correspondent alors aux arcs de  $G$ . Les arcs de  $GD$  représentent les séquences de deux arcs de  $G$  pouvant être parcourus consécutivement. Le coût associé à un arc  $(e, f)$  de  $GD$ ,  $e, f \in A$ , correspond au coût  $c_f$  de l'arc  $f$  dans  $G$ , auquel est ajouté le coût de traversée  $\sigma_{ef}$ . Une séquence d'arcs interdite dans le primal est modélisée par une absence d'arc dans le dual (ou par la présence d'un arc de coût infini). Un algorithme classique d'étiquetage appliqué sur  $GD$  fournit une solution optimale au problème de plus court chemin avec mouvements pénalisés. La structure générée est arborescente sur le dual, le chemin associé dans le primal ne peut donc contenir qu'une seule fois le même arc mais peut contenir plusieurs fois le même nœud.

Comme pour la méthode d'expansion de  $G$  au niveau des nœuds, la construction explicite de  $GD$  amène à manipuler un objet de grande taille par rapport à celle de  $G$ , ce qui rend la méthode difficilement applicable pour des instances de grande taille.

Kirby et Potts [38] proposent une utilisation implicite du dual. Cette solution permet de s'affranchir des difficultés constatées lors de l'utilisation des solutions précédentes. Elle consiste à décrire un chemin non plus comme une suite de nœuds mais comme une suite d'arcs et à appliquer la procédure d'étiquetage sur les arcs au lieu des nœuds. Kirby et Potts ont proposé une adaptation duale de la condition d'optimalité de Bellman (équation (2.8)) pour deux arcs consécutifs  $e$  et  $f$ :

$$\pi_e + \sigma_{ef} + c_f \geq \pi_f \quad (2.8)$$

Où  $\pi_e$  correspond à l'étiquette de l'arc  $e$ , représentant le coût du chemin allant de l'origine  $o \in V$  jusqu'à l'extrémité finale de l'arc  $e$ . De nombreux algorithmes, dits d'étiquetage des arcs sont inspirés de cette formulation de la condition d'optimalité. Ziliaskopoulos et Mahmassani [65] résolvent par exemple le problème one-to-all avec mouvements pénalisés au moyen d'un algorithme à correction d'étiquettes portant sur les arcs. Cette approche est comparée à un algorithme d'étiquetage portant sur les nœuds d'un graphe ayant fait l'objet d'une expansion au niveau des nœuds. Les applications numériques montrent que l'approche par étiquetage des arcs présente un temps de calcul inférieur, la différence étant d'autant plus grande que la taille et la densité du graphe sont importantes. Dans leurs travaux, Ziliaskopoulos et Wardell [66] modélisent également les coûts associés aux mouvements directionnels sur le réseau routier au moyen de séquences de deux arcs et résolvent le problème correspondant par étiquetage des arcs. Namkoong et al. [45] ont étudié ce même problème et proposé une comparaison entre l'approche d'étiquetage des arcs et une approche algorithmique d'étiquetage des nœuds appelée *vine building algorithm*. Cette méthode propose d'étiqueter non plus le nœud adjacent au nœud courant mais les nœuds situés à une distance de deux arcs du nœud courant. Les auteurs mettent en évidence que cette approche ne garantit pas l'optimalité des solutions et qu'elle est moins performante en termes de temps de calcul et d'espace mémoire. Ils montrent ainsi la supériorité de l'approche par étiquetage des arcs.

Della Valle et Tartaro [29] proposent un algorithme d'étiquetage hybride nœuds-arcs pour le problème de plus court chemin avec mouvements interdits. Il permet de limiter le temps de calcul pour des réseaux comportant des interdictions uniquement sur certaines intersections. Il utilise à la fois des étiquettes de nœuds lorsque les intersections considérées sont sans

interdictions, et des étiquettes d'arcs pour les autres intersections. Les auteurs montrent sur des réseaux réels que des économies en temps de calcul et en espace mémoire peuvent être réalisées grâce à cette stratégie par rapport à un étiquetage systématique des arcs.

## **II.5 Conclusion**

Pour optimiser les itinéraires routiers, la nécessité de modéliser des retards ou des interdictions sur certains mouvements directionnels au niveau des intersections nous a amené à considérer un problème de plus court chemins avec pénalités associées à certaines séquences de deux arcs. Il découle de l'étude bibliographique faite, que les pénalités définies par des séquences d'arcs peuvent être prises en considération par les algorithmes d'étiquetage en étiquetant non plus les nœuds mais les arcs du graphe.

## **Chapitre III :**

### **Analyse et modélisation du réseau routier**

*Le réseau routier peut être analysé selon différentes approches classées selon le niveau de détail lié au nombre de véhicules et à leur comportement. Les descriptions sub-microscopiques prennent en compte toutes les composantes des véhicules et leurs interactions avec l'environnement. Les modèles microscopiques décrivent le comportement de chaque véhicule et conducteur dans un environnement modélisé avec lequel ils interagissent. L'approche mésoscopique décompose le trafic en pelotons de véhicules. Enfin, l'approche macroscopique [52] consiste à modéliser l'écoulement de flux de véhicules dans les axes de circulation du réseau routier.*

*Nous adoptons pour notre analyse l'approche macroscopique. Celle-ci va nous permettre d'alimenter le graphe servant de support au problème par des informations logiques permettant de définir la topologie du graphe et par des informations attributaires qui associent à chaque élément du graphe un certain nombre de caractéristiques.*

#### **III.1. Analyse du réseau routier :**

L'analyse et/ou la description d'un réseau routier pour les besoins de modélisation nécessite la description de sa structure (définition de ses composants), des propriétés de ses composants (longueur, vitesse...) et des relations entre ces éléments et les flux de déplacements [22].

Le réseau routier est structuré en routes. Chaque route est partitionnée en un ensemble de tronçons de route, chacun d'eux correspondant à une portion de route homogène (même nombre de voies, même vitesse libre, etc. tout au long du tronçon) et sans carrefour intermédiaire. Un nouveau tronçon de route est donc créé dès qu'il y a une intersection mais également dès qu'un des attributs de la route change.

##### **1. Tronçon de route :**

Il représente une portion connexe de route homogène pour les attributs qu'elle porte.

Propriétés :a. Propriétés statiques :

Elles sont dites statiques parce qu'elles sont rarement mises à jours.

- Vocation : cette propriété matérialise une hiérarchisation du réseau routier basée sur l'importance des tronçons de route pour le trafic routier, ça peut être de type autoroutier, liaison principale, liaison secondaire, une bretelle une piste ...
- Nombre de chaussées.
- Nombre de voies.
- Etat physique : revêtue, non revêtue, en construction, ...
- Position par rapport au sol : normale, sur pont, en tunnel.
- Sens de circulation : sens unique suivant le sens du tronçon, sens unique suivant le sens inverse du tronçon, double sens.
- Vitesse limitée.
- Véhicules autorisés.
- Hauteur maximale autorisée
- Poids maximal autorisé
- ...

b. Propriétés temporelles : Permettent de décrire l'état du trafic routier dans l'espace et dans le temps. Elles sont représentées par les équations de la mécanique des fluides [52].

- **Le débit** : traduit le nombre de véhicules qui passent en un point, pendant un intervalle de temps  $\delta t$ , il s'exprime en véhicules par unité de temps, généralement des heures ou des minutes.
- **La concentration** : ou densité de trafic, traduit la répartition des véhicules le long du tronçon. Elle est définie à l'instant  $t$  sur un tronçon routier donné par le rapport entre le nombre de véhicules présents à cet instant sur ce tronçon et la longueur du tronçon. La concentration s'exprime en véhicules par kilomètre.

- **Le taux d'occupation** : Il est égal au rapport entre le temps de passage d'un véhicule à un point et l'intervalle de temps de la mesure. Il est proportionnel à la concentration. C'est une grandeur sans dimension (pourcentage).
- **La vitesse moyenne** : définie sur un tronçon routier à l'instant  $t$ , par la moyenne arithmétique des vitesses des véhicules présents sur le tronçon à cet instant. Elle s'exprime en Km/h.

## 2. Nœud routier :

Correspond à l'extrémité d'un tronçon de route, il traduit un changement de condition de circulation. Il peut représenter une intersection, un obstacle ou un changement de valeur d'attribut.

Propriétés :

- Type : un nœud routier peut être :
  - Carrefour à feu ;
  - Carrefour à priorité à droite ;
  - Rondpoint ;
  - Barrière interdisant la communication libre entre deux tronçons de route (ex : barrage de police, de gendarmerie, de douane ... ) ;
  - Changement d'attribut (ex : vitesse libre limitée) ;
  - ...

## 3. La route :

C'est un parcours classé par l'autorité compétente, pouvant être une route nationale, un chemin de wilaya ou bien un chemin communal, reliant entre eux des villes ou des pôles d'attraction (port, aéroport, hôpitaux, lieux touristiques...)

Propriétés :

Une route est caractérisée par :

- Un classement (route nationale RN, chemin de wilaya CW, chemin Communal CC).
- Un numéro

Après avoir défini les principaux composants d'un réseau routier, la deuxième partie de ce chapitre, est consacrée à la modélisation du réseau routier.

### III.2. Modélisation du réseau routier :

La modélisation d'un réseau routier consiste à construire une abstraction des éléments de l'infrastructure. Il existe différents types de modélisations [22] : graphes des routes et de leurs intersections, modèles analytiques ou géométriques filaires représentant les axes de circulation, modèles géométriques en deux ou trois dimensions, modélisation symbolique, etc.

#### III.2.1. Codification du réseau routier :

Le réseau routier est représenté par un graphe, appelé *graphe routier principal*, -composé d'arcs et de nœuds.

Le nœud de réseau correspond aux extrémités des tronçons du réseau, il n'a pas forcément une traduction physique directe il représente les intersections, les obstacles ou un changement de valeur d'un attribut de tronçon.

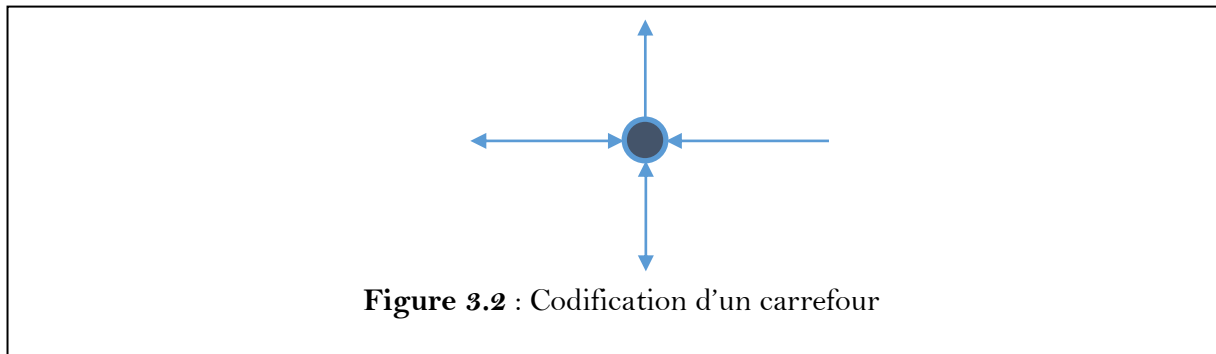
Les obstacles et/ou le changement de valeurs d'un attribut de tronçon (figure 3.1) sont représenté par un simple nœud qui coupe l'axe de circulation en deux tronçons distincts portant chacun des propriétés décrivant les conditions de circulation.



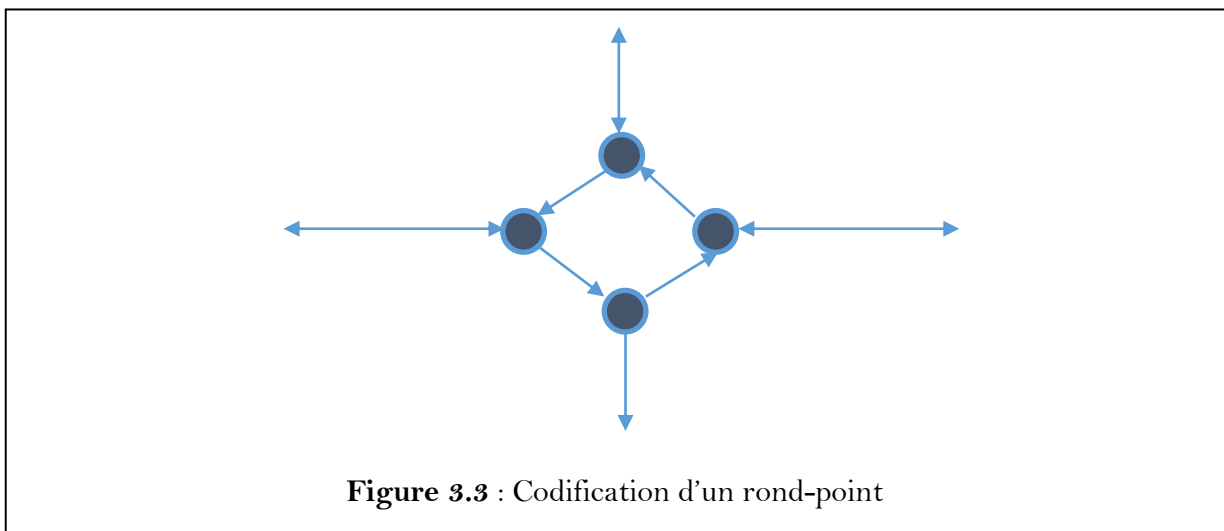
**Figure 3.1** : Codification des obstacles et/ou de changement de valeur d'un attribut de tronçon.

Quant aux intersections, on peut distinguer deux types, les carrefours et les ronds-points.

Un carrefour est un nœud qui interconnecte plus de deux axes de circulation portant chacun des propriétés permettant de définir les restrictions de circulation (carrefour à feux, carrefour à priorité à droite, restriction d'accès ...).



Un rond-point est une intersection particulière où les véhicules ne sont pas dotés du libre parcours, de ce fait, il est représenté par plusieurs nœuds gérant le sens de circulation.



L'arc de réseau correspond aux infrastructures de transport (tronçons de route), il représente une traduction très proche de la réalité, il est doté d'attributs permettant de traduire la fonctionnalité de la partie de réseau qu'il représente. En particulier, ces attributs doivent permettre de calculer le coût (distance, temps de parcours) nécessaire pour parcourir l'arc.

### III.2.2. Modèle conceptuel de données :

En adoptant la codification présentée ci-dessus, en considérant le problème de plus courts chemin avec séquences d'arcs pénalisés et en se basant sur le modèle vectoriel topologique notre modèle conceptuelle de données se présente comme suit (figure 3.4):

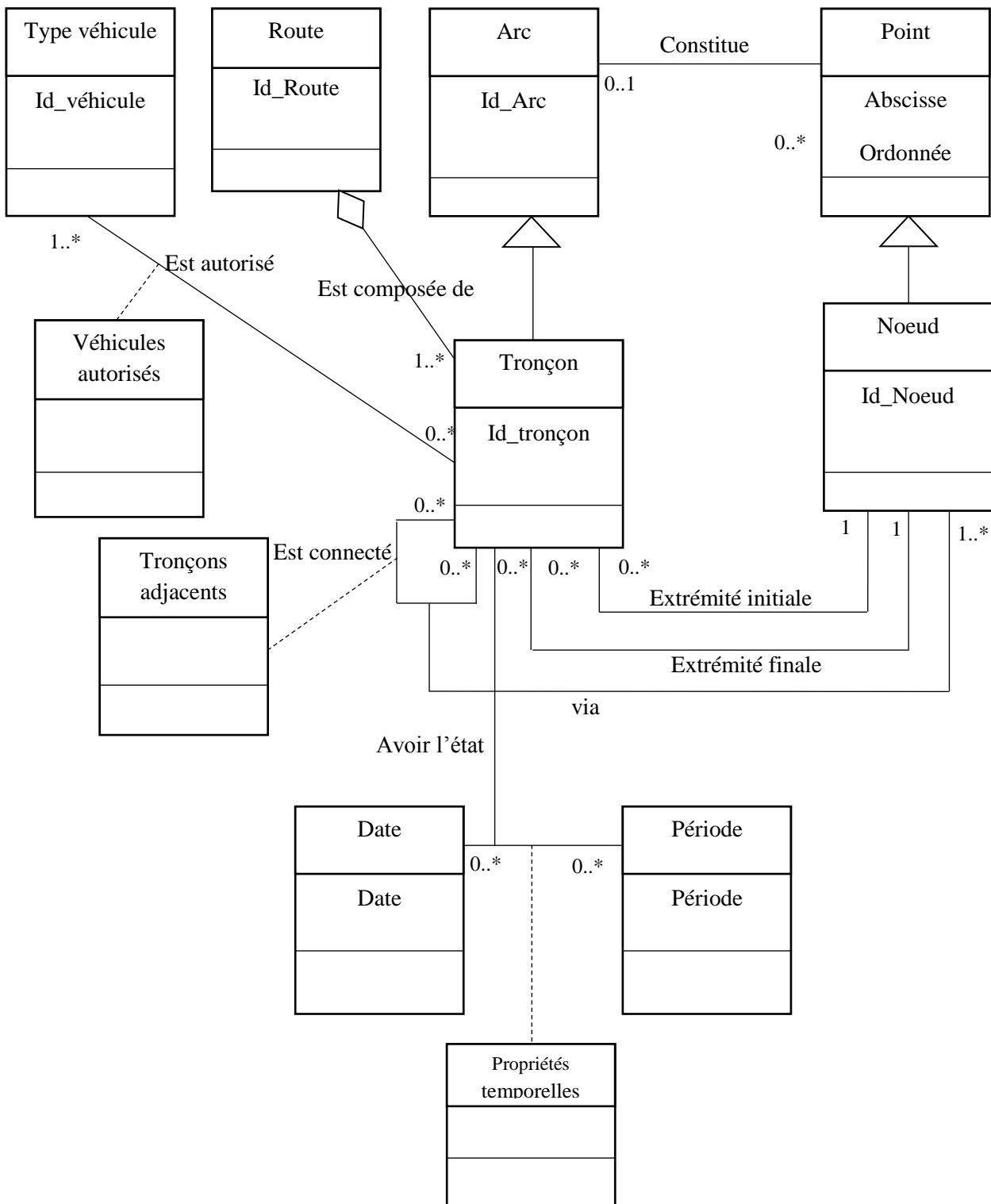


Figure 3.4: Diagramme de classe du réseau routier fondé sur le modèle vectoriel topologique.

Listes des attributs des classes du modèle de conception :

<b>Classe</b>	<b>Attributs</b>
<b>Route</b>	<p>Id_route</p> <p>Numero_route</p> <p>Classement_route</p>
<b>Tronçon</b>	<p>Id_tronçon</p> <p>Vocation</p> <p>Nombre_de_chaussees.</p> <p>Nombre_de_voies.</p> <p>Etat_physique</p> <p>Position_sol</p> <p>Sens_de_circulation</p> <p>Vitesse_libre_limitée.</p> <p>Hauteur_maximale_autorisée</p> <p>Poids_maximal_autorisé</p>
<b>Propriétés temporelles</b>	<p>Debit_Tronçon</p> <p>Concentration_tronçon</p> <p>Taux_occupation_tronçon</p> <p>Vitesse_moyenne_tronçon</p>

<b>Nœud</b>	<p>Id_noeud</p> <p>Type_noeud</p>
<b>Type Véhicule</b>	<p>Code_vehicule</p> <p>Poids_vehicule</p> <p>Hauteur_vehicule</p>
<b>Date</b>	Date
<b>Période</b>	Periode
<b>Véhicules autorisés</b>	
<b>Tronçons adjacent</b>	Pénalité
<b>Arc</b>	Id_arc
<b>Point</b>	<p>Abscisse</p> <p>Ordonnée</p>

**Tableau 3.1:** Liste des attributs des classes du modèle conceptuel.

### III.2.3. Problématique et Discussion:

L'objectif de la modélisation du réseau routier est de construire une abstraction des éléments réels de l'infrastructure de ce dernier, plus cette abstraction est proche de la réalité, plus la pertinence du modèle sera grande.

De nos jours, et avec les réseaux routier actuels, la notion de niveaux est devenue incontournable, puisqu'on ne parle pas de réseau routier sans évoquer l'existence des ponts des échangeurs, des tunnels, etc. Ces structures routières peuvent se couper en formant des intersections "*en l'air*".

Lors de la modélisation topologique classique, ce type d'intersection est pris en compte même s'il n'est pas réel, d'où la nécessité d'introduire la modélisation topologique tridimensionnelle (3D).

Dans de nombreux domaines, de nouvelles possibilités d'applications pour des SIG 3D ou multidimensionnels ont pu être définies, comme pour la géologie, l'aménagement urbain etc.[5]. Toutes ces disciplines ont depuis longtemps cherché des outils permettant d'analyser et d'administrer des données aussi proches que possible de la réalité, dans l'espace 3D.

Cependant, les méthodes de modélisation 3D utilisées dans l'environnement de conception assistée par ordinateur (CAO), ne répondent pas, ou répondent mal, aux besoins des SIG vu les spécificités de ces derniers (orientés plus vers l'analyse, les possibilités d'édition, la topologie,...), ainsi il est nécessaire d'utiliser des modèles adéquats pour l'administration des objets 3D par un SIG.

Plusieurs modèles ont été proposés pour la modélisation 3D [5]. Chacun de ces modèles implique des techniques de construction et de stockage particulières.

Dans ce contexte, notre contribution porte sur l'utilisation du modèle topologique 3D proposé par B.Bessaa [5] pour le réseau routier. Pour se faire, nous passons avec nos données d'une modélisation topologique 2D vers une modélisation topologique 3D.

### **III.2.4. Solution proposée : Modèle topologique 3D**

Vu les particularités de la couche réseau routier par rapport aux autres couches telles que le bâtis, le Modèle Numérique de Terrain...[5], il est nécessaire d'utiliser un modèle 3D spécifique au réseau routier.

Le modèle que nous utilisons présente la particularité d'être bien adapté aux caractéristiques du réseau routier.

#### **a. Principe :**

Il découle de l'analyse du réseau routier que ce dernier est composé d'un ensemble de routes, où chaque route est constituée d'un ensemble de tronçons de routes pouvant avoir des positions différentes par rapport au sol (normal, sur pont, en tunnel,...). Les tronçons de

routes peuvent se couper formant ainsi des intersections. Ces intersections peuvent être réelles, comme elles peuvent être virtuelles lorsqu'il s'agit d'une intersection entre des tronçons de niveaux différents (intersection entre un pont et une route qui passe en dessous).

En passant à la troisième dimension, ce type d'intersection "en l'air" ne doit pas exister. Néanmoins il nous donne une information sur la topologie (un tronçon est en dessous d'un autre).

Afin de prendre en considération ces caractéristiques, la notion de niveau est introduite aux arcs et aux nœuds. Au modèle 2D, un attribut niveau est rajouté aux éléments Arc et Nœud. L'ajout de cette information donnera l'aspect 3D aux données.

### Niveau d'arc :

L'affectation de niveaux aux arcs se fait d'une manière manuelle en procédant de la manière suivante :

- 1- Les arcs qui sont sur le sol (la terre) prennent toujours le niveau 0.
- 2- Les arcs qui sont totalement en l'air prennent des valeurs pairs (2, 4, 6,...). Pour les arcs en sous-sol, on prend des valeurs négatives (-2, -4, ...)
- 3- Les arcs intermédiaires qui lient les arcs de niveaux pairs prennent la valeur impair intermédiaires (1, 3, 5,..). Une illustration des niveaux d'arcs est donnée dans la figure suivante (figure 3.5). Le choix des tronçons, pendant la vectorisation, est assez important pour avoir des résultats cohérents.

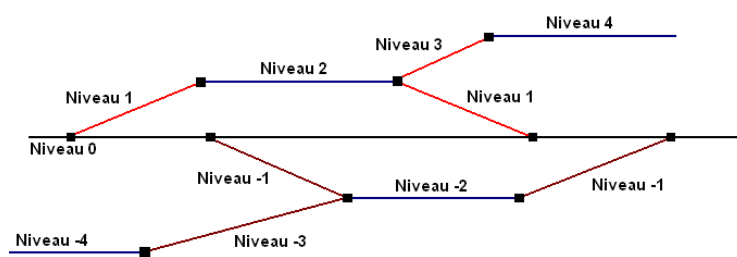


Figure 3.5: Affectation des niveaux aux arcs.

### Niveau de nœud :

L'affectation de niveaux aux nœuds se fait d'une manière automatique comme suit:

- 1- Les deux nœuds extrémités d'un arc de niveau pair prennent le niveau de l'arc.

2- Les nœuds reliant des arcs de niveaux différents prennent la valeur minimale des différents niveaux.

L'affectation des niveaux aux arcs et aux nœuds, permet d'éliminer les nœuds inutiles (intersection virtuelle). Ensuite en ayant les altitudes des nœuds de niveaux non nuls, on peut calculer l'altitude des différents points constituant les arcs ce qui permet d'avoir une vue 3D du réseau (figure 3.6).

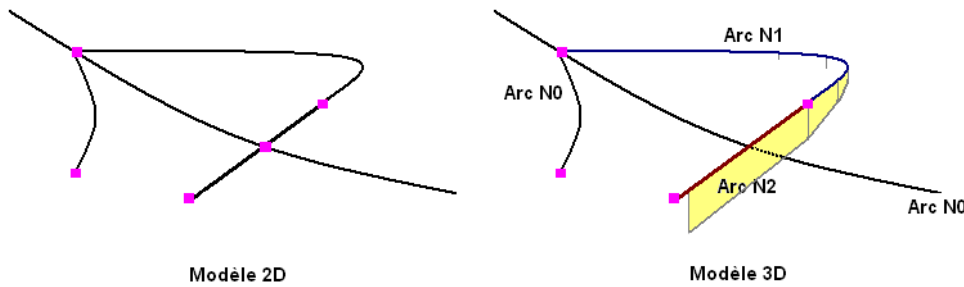


Figure 3.6 : Modèle Topologique 2D et Modèle 3D utilisé.

Suite aux changements introduits, le diagramme de classe du modèle 3D se présente comme suit (figure 3.7) :

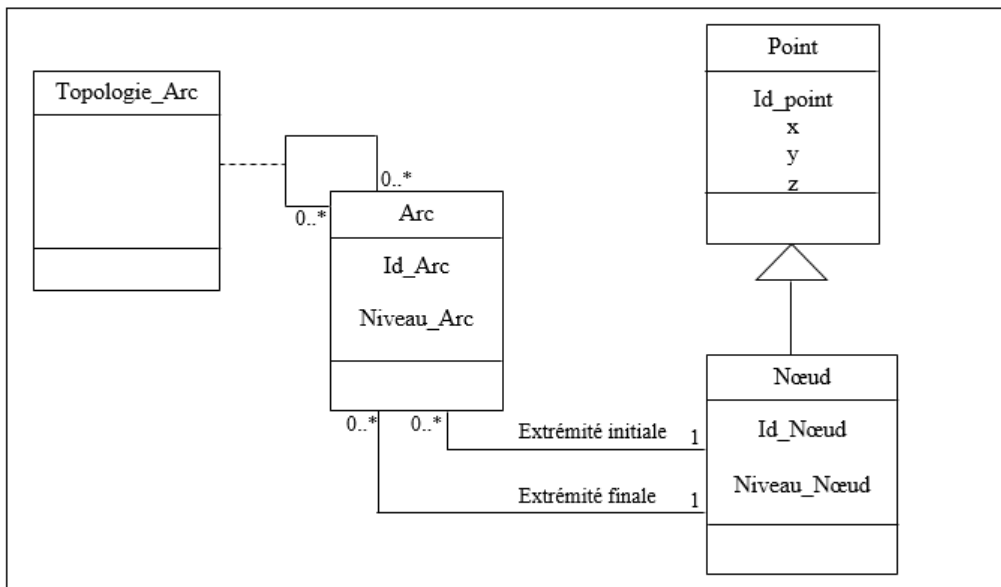
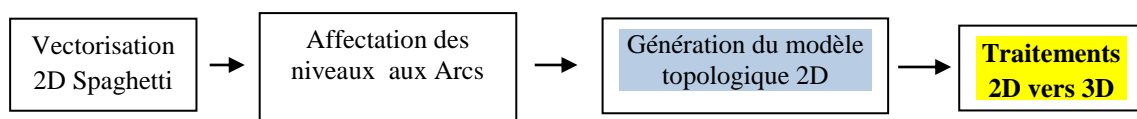


Figure 3.7: Diagramme de classe du modèle topologique 3D.

### b. Passage du modèle 2D au modèle 3D :

Parmi les problèmes rencontrés après la modélisation 3D est l'acquisition des données. En effet il est intéressant de développer des modèles 3D qui exploitent les données 2D déjà existantes pour passer au 3D avec un moindre coût [27].

L'avantage du modèle que nous utilisons est qu'à partir d'un modèle 2D existant, nous pouvons générer un modèle 3D. La figure suivante (figure 3.8) montre les étapes permettant de passer du modèle 2D (spaghetti) vers le modèle 3D (topologique).



**Figure 3.8 :** Passage du modèle 2D au modèle 3D.

#### **Première étape : Affectation des niveaux aux arcs**

A partir d'un modèle 2D (couche vecteur en spaghetti), nous affectons les niveaux aux différents arcs suivant le principe décrit précédemment.

#### **Deuxième étape : Génération du modèle topologique 2D**

C'est une étape automatique, elle consiste à générer les relations topologiques entre points et arcs. Le résultat de cette étape est la décomposition de certains arcs au niveau des points d'intersection (réel ou virtuel).

#### **Troisième étape : Traitements 2D vers 3D**

C'est une étape automatique permettant de compléter le modèle par les informations 3D (altitude et topologie). Cette étape passe par les phases suivantes :

- Affectation des niveaux aux nœuds : en suivant le principe décrit précédemment.
- Elimination des intersections virtuelles et mise à jour de la topologie.
- Affectation des altitudes aux différents points.

Elimination des intersections virtuelles et mise à jour de la topologie :

Une intersection virtuelle est une intersection entre deux arcs de niveaux différents. Son élimination consiste à annuler l'intersection et relier les arcs découpés par celle-ci lors de la génération du modèle topologique. Pour garder la topologie, nous sauvegardons les arcs concernés et le point d'intersection.

Affectation des altitudes aux différents points :

Pour l'affectation automatique des altitudes aux différents points, on pose l'hypothèse suivante : « *Les arcs de niveau non nul (autre que les arcs se situant sur le sol) ont une pente constante* ».

- Pour les altitudes des différents nœuds, elles peuvent être issues des différentes techniques de géoréférencement.
- Les altitudes des points appartenant aux arcs de niveau 0 (sol) : sont acquises directement à partir d'un MNT à haute résolution (métrique).
- Pour les altitudes des points restants : en se basant sur l'hypothèse posée, le calcul des altitudes des points intermédiaires constituant un arc se fait comme suit:

$$H_i = \frac{\sum_{j=1}^i L_j H}{L}$$

Où H est la hauteur entre le nœud de départ et le nœud de destination.

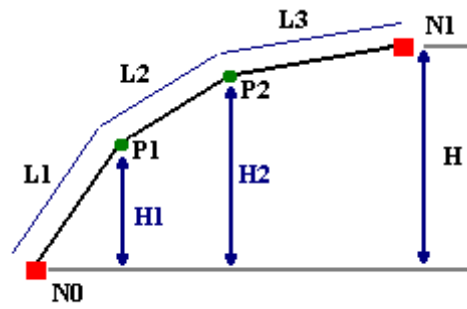
L la longueur de l'arc.

$L_i$  est la longueur d'un tronçon  $i$  de l'arc.

Un exemple de calcul est illustré par la figure suivante figure:

Soit un arc de niveau 1 composé de deux points intermédiaires P1 et P2 et de deux nœuds extrémités N0 et N1.

$$H_1 = \frac{L_1 H}{L} \qquad H_2 = \frac{(L_1 + L_2) H}{L}$$



**Figure 3.9 :** Calcul des altitudes des points intermédiaires.

Après l'application du modèle topologique 3D notre diagramme de classe se présente comme suit (figure 3.10):

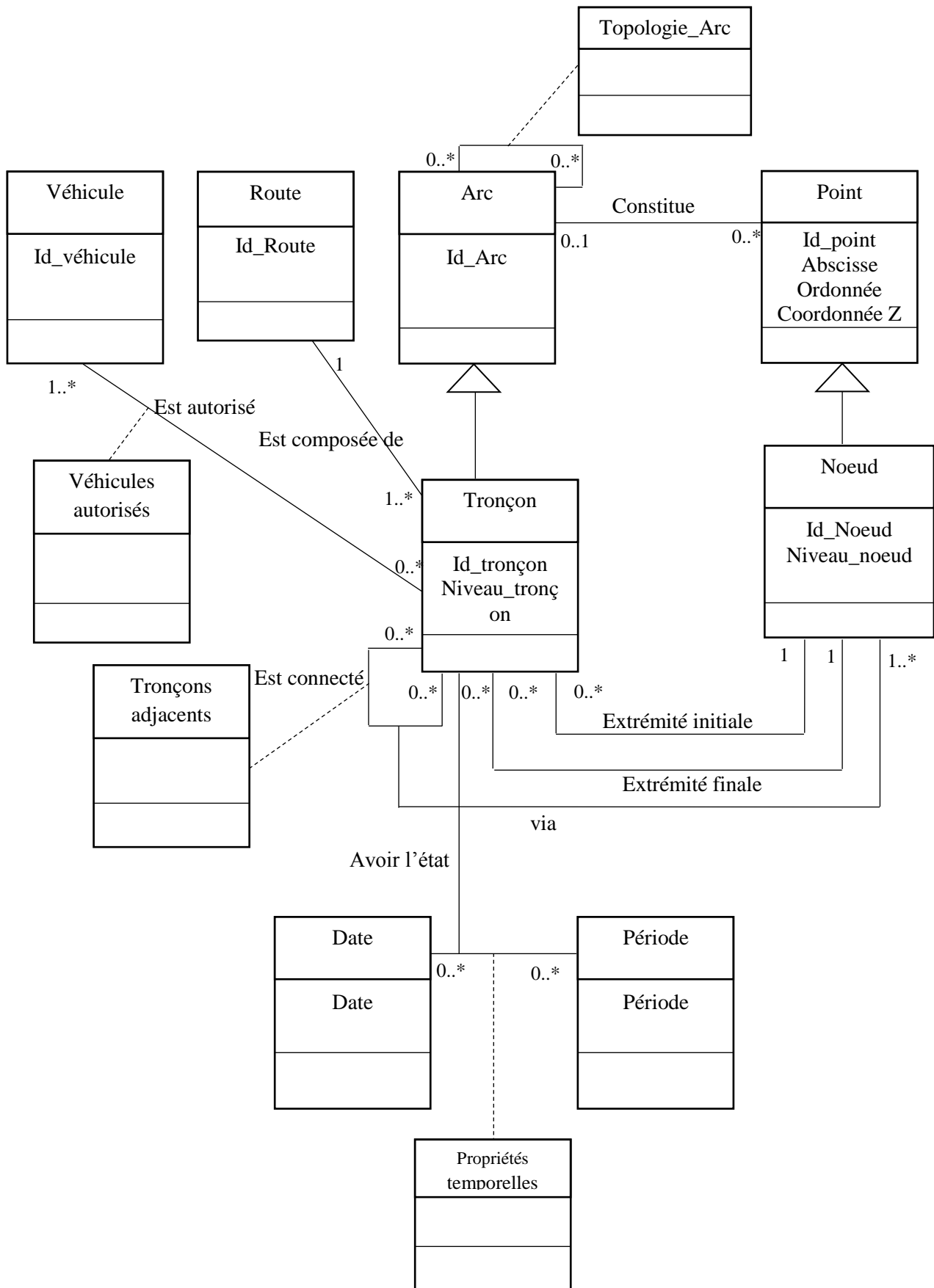


Figure 3.10: Diagramme de classe du réseau routier selon le modèle vectoriel topologique 3D.

Le modèle relationnel correspondant au modèle vectoriel topologique 3D utilisé est le suivant :

- Route (Id\_route, Numero\_route, Classement\_route)
- Tronçon (Id\_tronçon, niveau\_tronçon, Vocation, Nombre\_de\_chaussees, Nombre\_de\_voies, Etat\_physique, Position\_sol, Sens\_de\_circulation, Vitesse\_limitée, Hauteur\_maximale\_autorisée, Poids\_maximal\_autorisé, Id\_route\*, Id\_noeud\_init\*, Id\_noeud\_final\*, Id\_arc\* ).
- Date (Date).
- Période (Periode).
- Proprietes\_temporelle (Id\_tronçon , Date, periode, Débit\_Tronçon, Concentration\_tronçon, Taux\_occupation\_tronçon, Vitesse\_moyenne\_tronçon).
- Nœud (Id\_noeud, Type\_noeud, niveau\_noeud, Id\_point\*).
- Véhicule (Code\_vehicule, Poids\_vehicule, Hauteur\_vehicule).
- Véhicules autorisés (code\_vehicule, id\_tronçon).
- Tronçons autorisés (id\_tronçon, id\_tronçon\_adjacent, pénalité).
- Arc (Id\_arc, niveau\_arc).
- Point (Abscisse, Ordonnee, coordonnee Z, id\_arc\*).
- Topologie\_arc (id\_arc\_dessus, Id\_arcdessous, Id Noeud).

Les attributs soulignés représentent les clés primaires et les attributs étoilés (\*) représentent les clés étrangères.

### III.3. Conclusion :

Ce chapitre a été consacré à l'analyse du réseau routier selon un niveau de granularité (niveau de détail) macroscopique où nous avons décrit l'infrastructure routière par l'analyse de ses différentes composantes et leurs propriétés. Nous avons, par la suite, codifié la géométrie du réseau routier par un graphe (arc et nœud). Enfin nous l'avons modélisé selon le modèle vectoriel topologique où nous avons dégagé certaines problématiques qui nous ont menées à proposer comme solution l'utilisation du modèle topologique 3D qui peut être vu comme une extension du modèle topologique classique 2D. Dans la suite de ce mémoire, nous passons à un autre volet de notre travail, qui consiste en l'intégration de notre solution dans un SIG en ligne cette partie est précédée par une étude des fonctionnalités et des modalités de mise en ligne des SIG par l'utilisation des outils open source.

## Chapitre IV

# Fonctionnalités et modalités de mise en œuvre des SIG en ligne

*Depuis leur apparition, les SIG de bureau (classiques) ne cessent de se développer jusqu'à devenir un outil incontournable touchant aux différents domaines tels que la gestion des réseaux, la gestion de l'environnement l'agriculture.... Cependant le développement de ces outils rencontre plusieurs problèmes, l'un des plus gros problèmes est la collecte d'information. Vu la nature spécialisée des données, les développeurs sont obligés de les collecter eux même et cela demande du temps et nécessite beaucoup d'efforts. D'un autre point de vue, les SIG professionnels présentent l'inconvénient de coûter cher. De plus, leur utilisation nécessite une familiarisation si ce n'est une formation pour une bonne maîtrise de l'outil et du langage de programmation l'accompagnant [1].*

*Parallèlement à une offre éditeur de plus en plus riche, des solutions libres souvent robustes et éprouvées permettent de concevoir des outils sur mesure et ouverts sur le web.*

*Ce chapitre se propose de présenter les SIG en ligne, d'introduire la philosophie de l'univers de l'open source et d'exposer les différentes approches libres de la cartographie sur le web.*

### IV.1. Cartographie sur le web et SIG en ligne :

La cartographie sur le web ou *webmapping* désigne la diffusion de cartes dynamiques ou statiques sur un réseau. Plusieurs définitions sont rencontrées dans la littérature nous citons les plus importantes :

Neumann[48], définit Le *Webmapping* ou la cartographie en ligne comme étant le processus de conception, d'implémentation, de génération et de diffusion de cartes sur le Web. Hächler [31] dissocie le webmapping du produit lui-même (c'est à dire la carte), le webmapping représente donc, pour lui, l'application ou l'outil utilisé pour la visualisation des cartes. Pornon et al. [58] le définissent comme étant "*Un ensemble d'applications cartographiques*

*dynamiques et interactives disponibles sur le Web permettant principalement à un utilisateur de visualiser des cartes contenant plus ou moins d'informations géographiques".*

Les SIG en ligne (*Web GIS*) sont un domaine d'application et de recherche utilisant Internet et d'autres systèmes d'interconnexion pour faciliter l'accès, le traitement et la diffusion d'informations géographiques ou d'analyse spatiale [52]. Une application *Web GIS* est en relation avec le *webmapping*, elle intègre des fonctionnalités permettant à l'utilisateur d'effectuer des requêtes, des recherches, de mesurer des distances ou de sélectionner des zones [31].

Donc d'après ces définitions, nous rejoignons Neumann[47] qui différencie le terme *webmapping* de celui de *SIG en ligne* même si selon lui les deux termes sont souvent associés dans la littérature et parfois confondus.

Le premier site web géographique a été créé par Steve Puts[1] de la société *Xerox*, En juin 1993, Présenté sous la forme d'une mappemonde, il permet de zoomer à différents niveaux n'importe quelle partie du monde. C'est l'émergence d'un nouveau concept: le *webmapping*.

Quatre ans après, en 1997, une autre société, *Intergraph* crée *Geomedia Web Maps*, un logiciel SIG destiné à la mise à jour des données SIG en utilisant le Web [1]. Le *SIG en ligne* est donc né. Les données sont ainsi consultables à partir de postes clients. Elles sont en général stockées dans un système de gestion de base de données (SGBD) sur un ou plusieurs serveurs et administrables de façon centralisée. Evoluant rapidement, les *SIG en ligne* sont souvent présentés comme l'avenir des SIG.

## **IV.2 Architecture des SIG en ligne :**

Un SIG en ligne, peut être vu comme un Système d'Information (SI) qui utilise le web, il s'avère donc important d'étudier l'architecture des SI et utile de connaître l'architecture Web.

### **IV.2.1 Architecture des Systèmes d'Information**

Deux concepts différents, physique et logique sont pris en compte lors de l'étude de l'architecture informatique ou plus précisément l'architecture des SI [17]. L'architecture physique concerne le côté matériel, elle représente les terminaux ainsi que l'interconnexion entre eux. L'architecture logique, quant à elle, décrit les modules et les composants logiciels qui composent le système ainsi que leur répartition.

Un SI est une application informatique au sens large qui est composée de trois couches principales : L'interface ou la présentation, les traitements et les données [17] [8].

- a. **La couche présentation** : c'est la partie responsable de l'interaction entre l'utilisateur et le système, elle s'occupe du transfert des informations nécessaires au traitement introduites par l'utilisateur et du formatage des données en sortie pour l'affichage à l'écran ou l'impression papier. La couche présentation se situe sur le post client [17].
- b. **La couche Traitement ou application** : cette couche représente le fer de lance du système, elle permet de réaliser les traitements sur les informations et d'effectuer les calculs à partir des données de base. Il existe deux types de traitements [1]:
  - les traitements locaux : effectués au niveau de l'interface homme-machine pour le contrôle et l'aide à la saisie.
  - Les traitements globaux : également appelés « couche métier » constituent l'application elle-même.
- c. **La couche Données** : contient les composants et mécanismes de représentation d'accès aux ressources données (accès aux bases de données), aux ressources matérielles (unités d'entrées ou de sortie de données...) ou aux ressources permettant la communication (les emails...) [41]. Elle s'occupe également de l'intégrité des données (aspects sécuritaires, partage...) [8]

Un système séparé en différentes couches ou composants présente l'avantage de délimiter les différentes tâches à assumer par chacune d'elles. Ainsi, le partage du travail entre les différentes équipes de développement en est facilité [17].

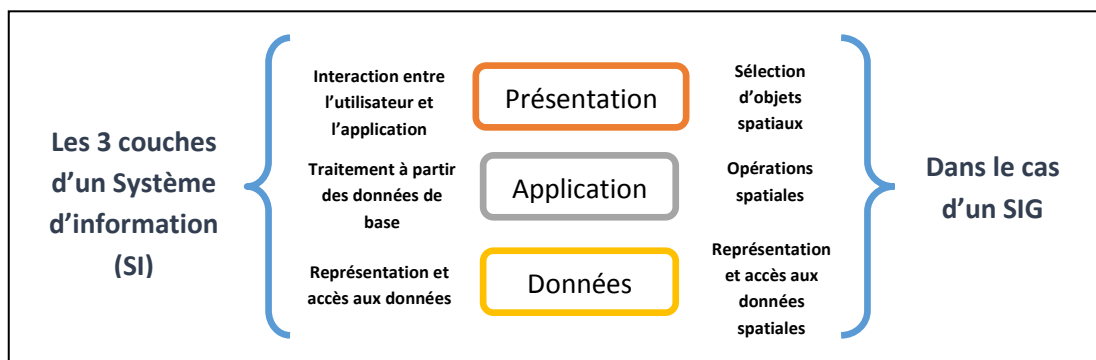


Figure4.1 : Trois composantes d'un SI et en l'occurrence d'un SIG.

La répartition physique des différentes couches : *présentation, traitement et données* est basée sur l'architecture Client/serveur, il est donc nécessaire de donner un aperçu sur cet environnement.

### **Architecture Client/serveur :**

L'architecture des SIG en ligne repose sur le concept de communication client/serveur [44] et c'est la répartition, entre le client et le serveur, des trois couches vue précédemment qui établit le lien entre un système d'information et l'architecture client/serveur [17]. Dans une application reposant sur cette architecture, les clients contactent le serveur dans le but d'obtenir un service (fichier, logiciel à partager, calcul...). Le serveur leur répond en leur fournissant le service demandé.

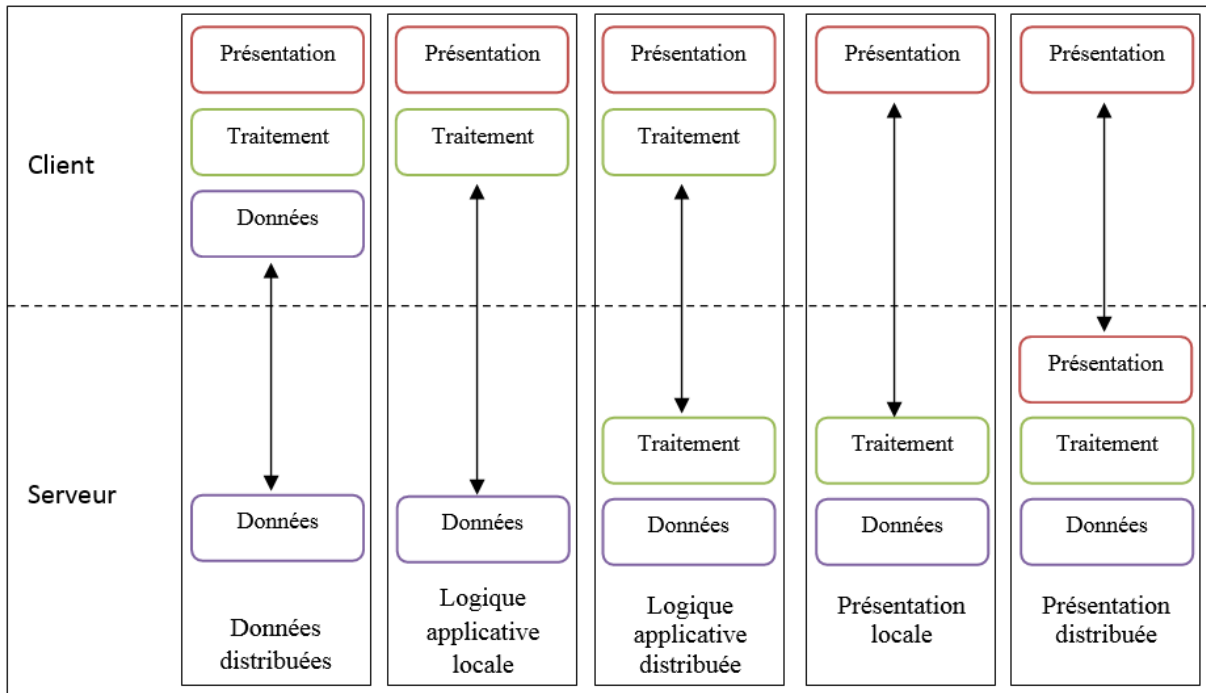
Le modèle client/serveur se caractérise par une interface client simple et robuste permettant la formulation des requêtes et l'affichage du résultat, une rapidité de transmission des données entre le client et le serveur et un système puissant et centralisé, constituant le serveur, pour traiter et sauvegarder les données.

McFarland [41] cite deux façons de voir l'architecture client/serveur. L'architecture physique, vu comme un ensemble d'ordinateurs (PC de bureau, PC portables, Smartphone,...) représentant les clients connectés à un réseau à travers lequel ils communiquent avec le serveur qui est dans la majorité des cas plus puissant. L'architecture logique, quant à elle, décrit le partitionnement de l'application en plusieurs processus (*process*) ou modules. Ces modules partagent des ressources en vue de résoudre un problème. Dans ce contexte, clients et serveur représentent des modules et non plus des machines. Ces modules peuvent résider dans une même machine ainsi, clients et serveur peuvent se retrouver sur le même poste.

L'administration des données et des utilisateurs dans l'architecture client/serveur se fait sur le serveur, de ce fait, la sécurité est renforcée et l'accès aux données est limité. Le système évolue en toute indépendance par rapport au concept physique (hardware) et logique (software) des différents terminaux (clients). Cependant, les coûts liés à l'administration du serveur demeurent élevés sans évoquer qu'en cas de panne de celui-ci, c'est tout le réseau qui est paralysé

**Classification des architectures Client/serveur :** McFarland [41] présente trois classifications basées sur l'architecture logique :

- Présentation – Traitement – Données** : Elle décrit la manière avec laquelle l'application distribue ses trois composantes (figure 4.2) : la présentation, le traitement et les données en précisant les rôles que jouent le serveur et le client dans le traitement des données.



**Figure 4.2** : Architecture du modèle Client/serveur selon la classification Présentation- Traitement- Données.

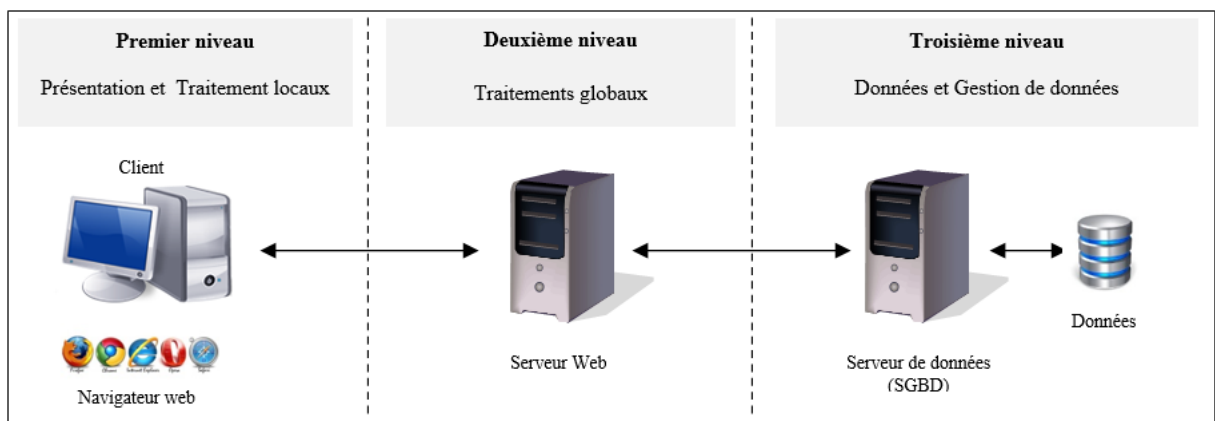
Bigand et al.[8] définissent six types de clients/serveurs. En plus des cinq cas représentés dans la figure, un sixième cas, appelé "*Données et Traitements Distribués*" est rajouté. Dans celui-ci, la charge entre le client et le serveur est répartie.

Un autre cas de cette dernière classification, appelé *Gestion Distante des Données* ou *Client/serveur de données* correspond au cas *Logique applicative locale* de la figure. Selon Bigand et al., il exploite une architecture à niveaux.

Dans le cas d'une architecture à deux niveaux, les clients communiquent avec un serveur de données sur lequel se situent les données et le système de gestion de base de données. Clients et serveur ne peuvent dialoguer que grâce à l'existence d'une couche logicielle appelée *middleware*. Le middleware permet d'assurer une liaison transparente entre le client et le serveur à travers le réseau.

L'inconvénient dans cette architecture est un problème de standard. Un poste client peut ne pas avoir la même architecture que celle d'un serveur. De plus, le middleware est, dans la majorité des cas, développé pour un type particulier d'architecture de poste client, donc il n'est pas standard pour les autres architectures [8].

Comme solution à ces inconvénients, l'architecture à trois niveaux (figure 4.3) répartie l'application en trois niveaux distincts : Le premier niveau est représenté par le poste client où on s'occupe de l'affichage et des traitements locaux (mise en forme des données, contrôle de saisie, etc.), Le deuxième niveau est chargé d'effectuer les traitements applicatifs globaux et le troisième niveau s'occupe de la gestion des données (SGBD) [8]. L'indépendance de ces trois niveaux est l'avantage majeur d'une telle architecture. Ce qui permet de les implanter indifféremment sur la même machine (SGBD et serveur d'application sur la même machine) ou sur des machines différentes. D'autre part, le client effectue moins de traitements (plus léger) car une partie des traitements (traitements globaux) est transféré vers le second niveau.



**Figure 4.3** : Exemple d'une architecture à trois niveaux.

Une architecture à *n-niveaux* est une extension de la précédente. Le nombre de couches reste inchangé (trois couches) mais c'est la répartition des différents services qui est faite sur plusieurs serveurs [8].

- **Classification par tiers** : Elle est basée sur le nombre de *processes* ou modules dont est partitionnée l'application client/serveur. Chaque *process* est considéré comme un composant (appelé *tiers* ou *module* ou encore *couche*). On dit qu'un programme est *monolithique* lorsque l'application n'est composée que d'un seul tiers. S'il est à deux tiers, le client communique directement avec le serveur, dans un système à trois tiers,

le client communique avec un serveur à travers un *process* intermédiaire (*middle tier*). Des traitements complexes sont supportés dans ce type d'architecture [17]. Une architecture à *n-tier* est une extension de l'architecture à *trois-tiers*. Ainsi, un client communique avec le serveur à travers plusieurs *processes* intermédiaires.

- **Classification basée sur les fonctionnalités du serveur :** Elle est basée sur le rôle du serveur pour décrire l'architecture client/serveur. Ce rôle est généralement défini par le type de traitements qu'effectue le serveur.

#### IV.2.2. Architecture web :

L'architecture Web décrit l'organisation interne d'un système de communication utilisant le Web et à la façon dont l'information circule à travers ses composants [57]. Les architectures de communication Internet se basent essentiellement sur l'architecture client/serveur [16].

Furht [20] a proposé une des architectures Web qui est basé sur l'architecture software des applications. Il dégage quatre modèles :

- **Le modèle client :** dans ce modèle, l'ensemble de l'application se trouve du côté du client et l'exécution est faite localement. On parle alors de client *lourd* ;
- **Le modèle client/serveur :** le traitement se fait du côté client mais les ressources sont partagées entre le serveur et/ou un ou plusieurs clients ;
- **Le modèle réseau :** l'application est dynamiquement téléchargée à partir du serveur et ce via le réseau. Dans ce modèle, le client est considéré aussi comme un client *lourd* ;
- **Le modèle serveur :** l'application se situe sur le serveur. Elle est accessible par les différents clients sans qu'ils n'aient besoin de la télécharger.

D'autres classifications utilisent la décomposition d'une application Web en nombre de tiers [1]. Elle peut être à deux ou trois tiers [57]. Ou encore *hybride* qui combine les deux décompositions précédemment citées [34].

### **IV.2.3 Architecture des SIG en ligne**

Plusieurs approches de classification des SIG en ligne existent dans la littérature. Certaines classifications se basent sur les services offerts. D'autres se basent plutôt sur l'architecture interne des SIG en ligne ou encore selon la technologie utilisée [1].

#### **1. Les SIG en ligne du point de vue services offerts**

Les SIG en ligne sont classés selon différentes façons. Se basant sur les services que peuvent offrir les SIG en ligne, Green et Bossomaier [26] les classent dans deux catégories:

- Des systèmes dédiés à la formulation de requêtes spatiales. C'est une des fonctionnalités les plus répandues des SIG en ligne, elle consiste à sélectionner les objets spatiaux en cliquant directement sur la carte (points de départ et d'arrivée), en les sélectionnant dans une liste liée à une base de données (région) ou en saisissant directement leurs coordonnées géographiques.
- Des systèmes dédiés à la conception et la diffusion de cartes. Ici, l'utilisateur peut avoir la faculté de créer librement ses propres cartes (définitions d'arcs, polygones, etc.) et le résultat peut être, par la suite, diffusé en ligne. Ainsi, l'utilisateur peut modifier la base de données contrairement à la première catégorie où l'utilisateur est restreint à utiliser les options et les données disponibles.

#### **2. Les SIG en ligne du point de vue architecture**

Dans ce genre de classification, on s'intéresse à l'architecture du SIG en ligne et la répartition de ses différents composants entre le client et le serveur [1].

Cette classification engendre deux catégories de SIG en ligne: ceux qui sont orientés serveur et ceux qui sont orientés client [26] [50].

##### **a. SIG en ligne orienté serveur**

Dans ce genre de solution, le serveur se charge de l'analyse et du traitement. L'utilisateur envoie sa requête au serveur distant via Internet par le biais d'un navigateur Web. Le serveur traite la requête et renvoie, à son tour, le résultat au poste client. L'utilisateur voit s'afficher alors ce résultat sur son navigateur Web.

Dans ce genre de solution, le langage HTML peut être utilisé pour l'envoi de la requête et une interface CGI (Common Gateway Interface) pour le passage de paramètres vers le serveur web, ce dernier incapable d'interpréter la requête spatiale, il transmet les informations relatives à celle-ci au serveur SIG via l'interface CGI. Le serveur SIG exécute la requête et renvoie, à son tour, le résultat vers le serveur Web en utilisant la même interface (CGI), celle-ci est nécessaire dans le sens où le serveur Web et le serveur SIG ne peuvent pas communiquer directement. Enfin, le serveur Web retourne les résultats vers le client (voir figure 2.6), les résultats étant, dans la majorité des cas, des images (PNG, GIF, JPEG...) peuvent être affichés dans n'importe quel navigateur Web.

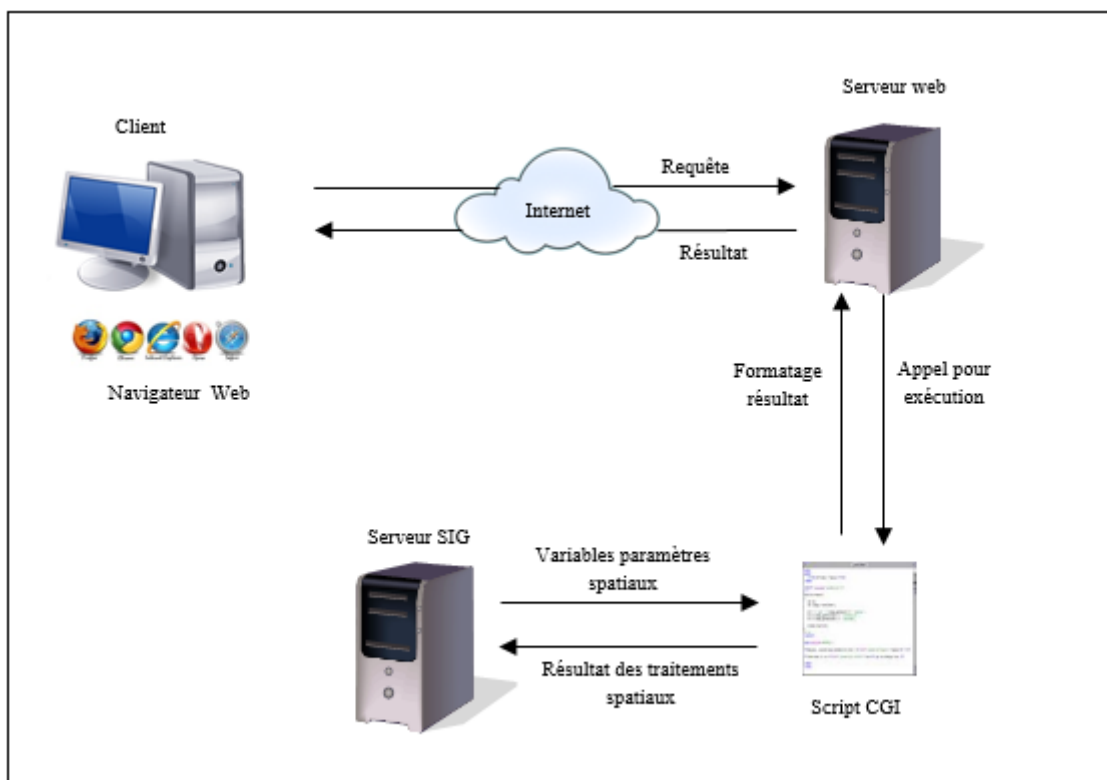


Figure 4.4 : Fonctionnement d'un SIG en ligne orienté Serveur.

Dans ce type de solution, le client reste limité par les fonctionnalités offertes. En effet, le champ d'action de l'utilisateur se restreint aux seules fonctionnalités offertes par le navigateur Web. En revanche, ce type de solution est exploitable sur n'importe quelle machine dotée de n'importe quel système d'exploitation [1].

#### b. SIG en ligne orienté client

Dans cette seconde catégorie, l'analyse et le traitement des données SIG se font sur le poste client et plus précisément sur le navigateur Web. Se trouvant initialement sur le serveur, les

outils d'analyse et de traitement sont renvoyés au client, lors de l'émission d'une requête par ce dernier lui permettant, ainsi, de traiter et d'analyser cette requête par lui-même.

Le point fort d'une telle solution est la richesse de l'interface du post client, elle offre des fonctionnalités telles que la rotation d'image, le zoom ou la formulation de requêtes. Cela réduit la communication entre le client et le serveur [62]. Cependant, il est nécessaire de charger les données et les modules permettant le traitement (*ActiveX*, *plugin* ou *applet*) sur le poste client. La portabilité de cette solution sur les différents types de machines et systèmes reste limitée [33].

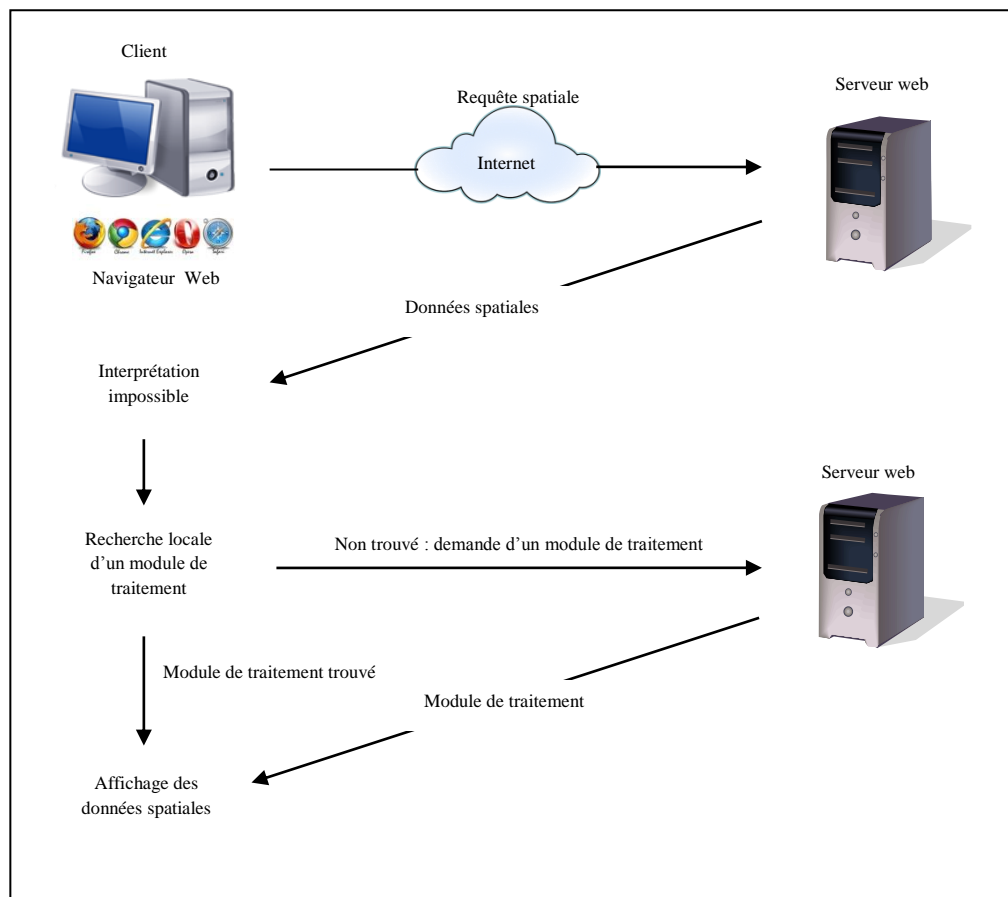


Figure 4.5 : Fonctionnement d'un SIG en ligne orienté client.

### 3. SIG en ligne du point de vue mise en ligne

Deux principales approches existent lors de la mise en ligne des SIG [26].

La première approche considère un poste client doté d'un logiciel (SIG de bureau) offrant toutes les fonctionnalités (ou presque toutes) tout en étant connecté à Internet. Ainsi il aura la possibilité d'accéder à d'autres données situées sur d'autres postes eux même connecté à Internet. Cette solution offre à l'utilisateur la puissance et l'ensemble des fonctionnalités d'un SIG de bureau, de plus, l'utilisateur pourra utiliser un logiciel qui lui est familier. Cependant,

cette solution peut coûter cher puisqu'elle nécessite l'installation d'un logiciel et éventuellement d'un matériel spécialisé assez coûteux, de plus, la disponibilité de ce genre de logiciels est restreinte et leur portabilité sur d'autre architecture reste réduite.

Dans la deuxième approche, le navigateur Web est utilisé comme interface du logiciel SIG, par conséquent, la majorité des données se situent du côté du serveur. Celui-ci se charge également des traitements. Cette solution se caractérise par une certaine accessibilité et par sa facilité d'implémentation. Toutefois, les fonctionnalités demeurent limitées et les données doivent être converties vers des formats utilisables par le navigateur Web.

### **VI.3. Webmapping et Concept de l'open source :**

La progressive généralisation des SIG en ligne au sein des organismes gestionnaires, est accélérée par le recours à des technologies *Open Source* éprouvées. Ces solutions permettent, en effet, de faire l'économie d'une offre éditeur tout en permettant d'envisager des développements ultérieurs dans le but de répondre aux besoins spécifiques et constatés des utilisateurs.

Le concept du logiciel libre a été défini, en premier, par Richard Stallman [46]. Le caractère **open source** d'une application implique, selon lui, non seulement la libre diffusion en totalité ou en partie de son code source mais aussi, sa libre redistribution et la possibilité de développer librement des applications dérivées ou des fonctionnalités complémentaires. Elle ne doit pas non plus impliquer de discrimination ni entre personnes ou groupes de personnes ni entre domaines d'application.

La **licence GNU** (licence publique générale - General Public License « GNU GPL ») est la licence open source la plus répandue. Elle prévoit la liberté d'usage, d'étude, de modification et de distribution de tout ou partie des applications couvertes. La clause « copyleft » est une de ses caractéristiques. Elle consiste à octroyer à l'utilisateur un droit total de redistribution conditionné par la mise à disposition du code source de la version modifiée [1].

Plusieurs organismes sont en relation avec l'open source :

- **World Wide Web Consortium (W3C)**: Créé en 1994, cela concerne la spécification des normes, les orientations, les logiciels et les outils permettant de répondre aux standards Web. (site officiel : <http://www.w3.org/>).
- **International Organization for Standardization (ISO)**: Créé en 1947 avec l'objectif de définir des normes dans les domaines industriel et commercial. Il est

composé de représentants d'organisations nationales de normalisation de 158 pays. (site officiel : [www.iso.org](http://www.iso.org)).

- **Open Geospatial Consortium (OGC):** Anciennement Open GIS Consortium, est une organisation à but non lucratif consacrée au développement des solutions libres en géomatique et à l'interopérabilité des systèmes. Elle regroupe plusieurs centaines d'organismes commerciaux, gouvernementaux et de recherche dans le monde afin de promouvoir les systèmes ouverts, définir des normes et des standards en matière de technologies et services géospatiaux. L'OGC publie des normes comme le format *GML (Geographic Markup Language)*, les protocoles *WMS (Web Map Server)* et *WFS (Web Feature Server)*. (site officiel : [www.opengeospatial.org](http://www.opengeospatial.org)).
- **Open Source Geospatial Foundation (OSGeo):** C'est une fondation américaine dont la mission est d'aider et de promouvoir le développement collaboratif des données et des technologies géospatiales ouvertes. Elle fournit une aide financière, organisationnelle et légale à la communauté géospatiale libre. Elle sert aussi d'entité légale indépendante dans laquelle peuvent contribuer les membres de la communauté pour la programmation, le financement et l'acquisition des ressources. Elle s'assure également que leurs contributions seront maintenues au bénéfice du public. *OSGeo* est aussi vue comme une organisation d'assistance et de défense pour la communauté géospatiale libre, elle fournit une infrastructure partagée pour améliorer la collaboration entre projets. (Site officiel : <http://www.osgeo.org>).

#### **IV.4. Approches libres existantes pour la mise en ligne des SIG :**

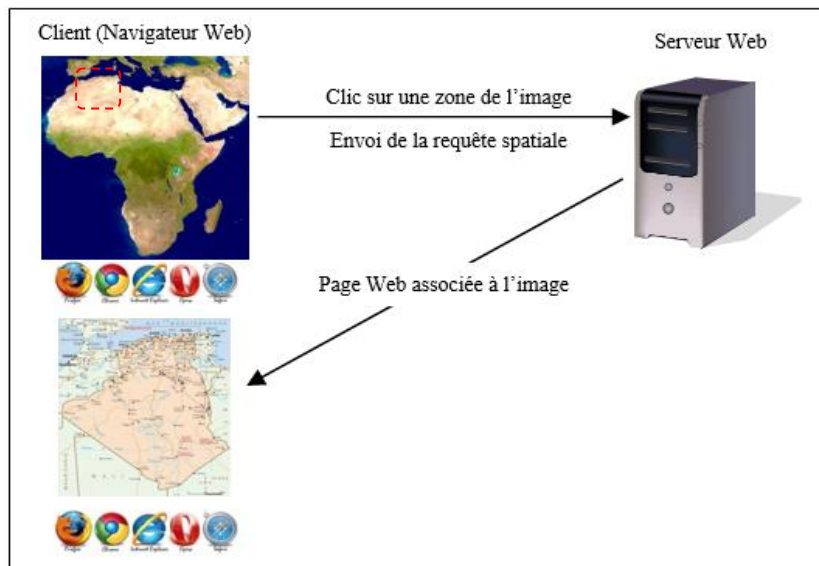
Il existe actuellement quatre principales solutions techniques libres permettant de publier en ligne des données cartographiques. Le choix entre celles-ci est facilité par leurs limites respectives ainsi que par la lourdeur de leur déploiement qui permettent de circonscrire chacune à un type d'usage et de finalité particulier.

##### **IV.4.1. Approche carte statique (HTML simple) :**

Il s'agit de la solution la plus légère et la plus facilement mise en place. Elle consiste à insérer une simple image (JPG, PNG, GIF, BMP, etc.) : des cartes et des archives photographiques dans une page HTML classique. Le choix du format dépend des performances de celui-ci [1].

Le HTML ne permet, cependant, que de définir des zones cliquables pointant vers une autre page (fonctionnalité utilisable par exemple pour proposer à la navigation un zoom sur zone

prédéfinie) (figure 4.6) ou associées à un événement JavaScript permettant par exemple l'affichage de données attributaires au passage de la souris.



**Figure 4.6 :** Fonctionnement de la solution « Carte Statique ».

Cette solution ne gère pas les sources vectorielles ni la projection et demeure très limitée en termes d'outil de navigation. Elle ne permet pas, non plus, l'intégration d'outil d'analyse et de recherche attributaire ou spatiale.

#### IV.4.2. Solution vectorielle SVG (Scalable Vector Graphics):

Il s'agit d'un format image vectoriel ouvert (basé sur la norme XML) permettant un encodage et un affichage sans perte de qualité de formes géométriques (figure 4.7).



**Figure 4.7:** Effet de pixellisation ou aliasing.

Cet effet est visible lors de l'agrandissement de l'image raster (à gauche).

Les données peuvent être stockées comme table (.tab ou .shp par exemple) avant d'être encodées en SVG et envoyées au poste client. Elles sont alors chargées dans la mémoire cache du navigateur pour pouvoir être visualisées par l'utilisateur, ce qui permet une grande fluidité dans la navigation mais peut être lourd en temps et en ressource pour le poste client (figure 4.8).

Cette solution permet notamment de développer des outils de requête spatiale et de procéder à la mise à jour des données (notamment géométriques si le fichier source est une table SIG).

En revanche, elle nécessite l'installation d'un outil complémentaire appelé plugin côté client. Certains navigateurs (excepté Microsoft Internet Explorer) supportent toutefois le SVG en natif selon leur version.

Il est courant de présenter un SIG en ligne avec le format SVG tant que les spécialistes administrent le système et que les utilisateurs travaillent sur un ordinateur suffisamment puissant pour rendre la navigation fluide et le filtrage rapide des données malgré une connexion Internet à faible débit. Le point fort de ce format est sa possibilité d'intégration dans un grand nombre de sites Internet existants [36].

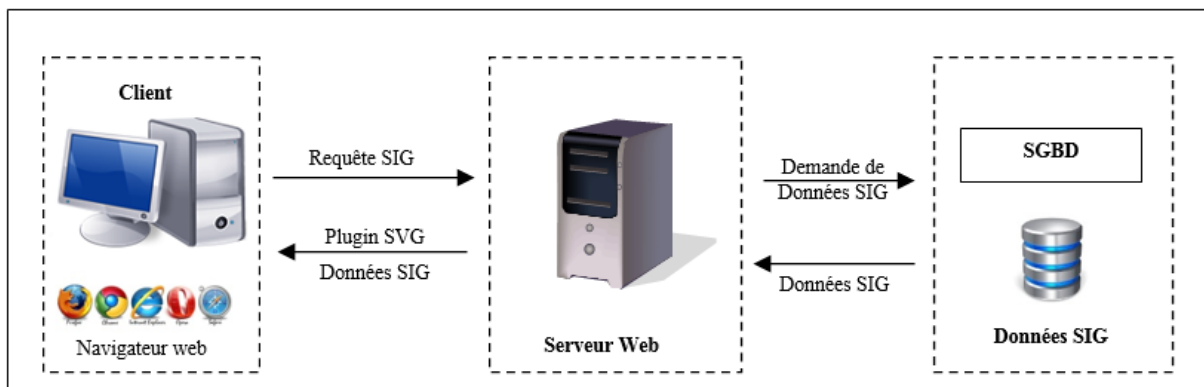


Figure 4.8 : Fonctionnement de la solution vecteur.

#### IV.4.3. Applet Java :

L'applet Java est l'une des techniques les plus utilisées pour les SIG en ligne. Java est un langage de programmation orienté objet qui offre une large gamme d'outils de création et d'affichage graphiques. Il peut être vu comme le langage le plus approprié pour la manipulation des données SIG sur le Web [1].

Une *applet* est une mini application qui s'exécute dans un navigateur Web pouvant supporter les applications Java (Java-enabled).

En plus d'afficher des images vectorielles, l'applet Java, développé par Sun Microsystem, offre la possibilité de transformer des données contenues dans des tables SQL au format de données cartographiques (SHP,...).

La visualisation des cartes et des données à partir d'une page web est possible grâce au plug-in Java qu'il faut installer sur le navigateur Internet. Le chargement de la totalité des données pouvant être long, il est préférable de laisser l'utilisateur choisir celles qu'il souhaite afficher avant qu'il ne les mette en mémoire (figure 4.9).

Tout comme le format SVG, cette application ne permet pas aux utilisateurs d'administrer le contenu du SIG. Cependant l'Applet Java permet de travailler avec des ordinateurs peu puissants tant que la connexion à internet offre un bon taux de téléchargement.

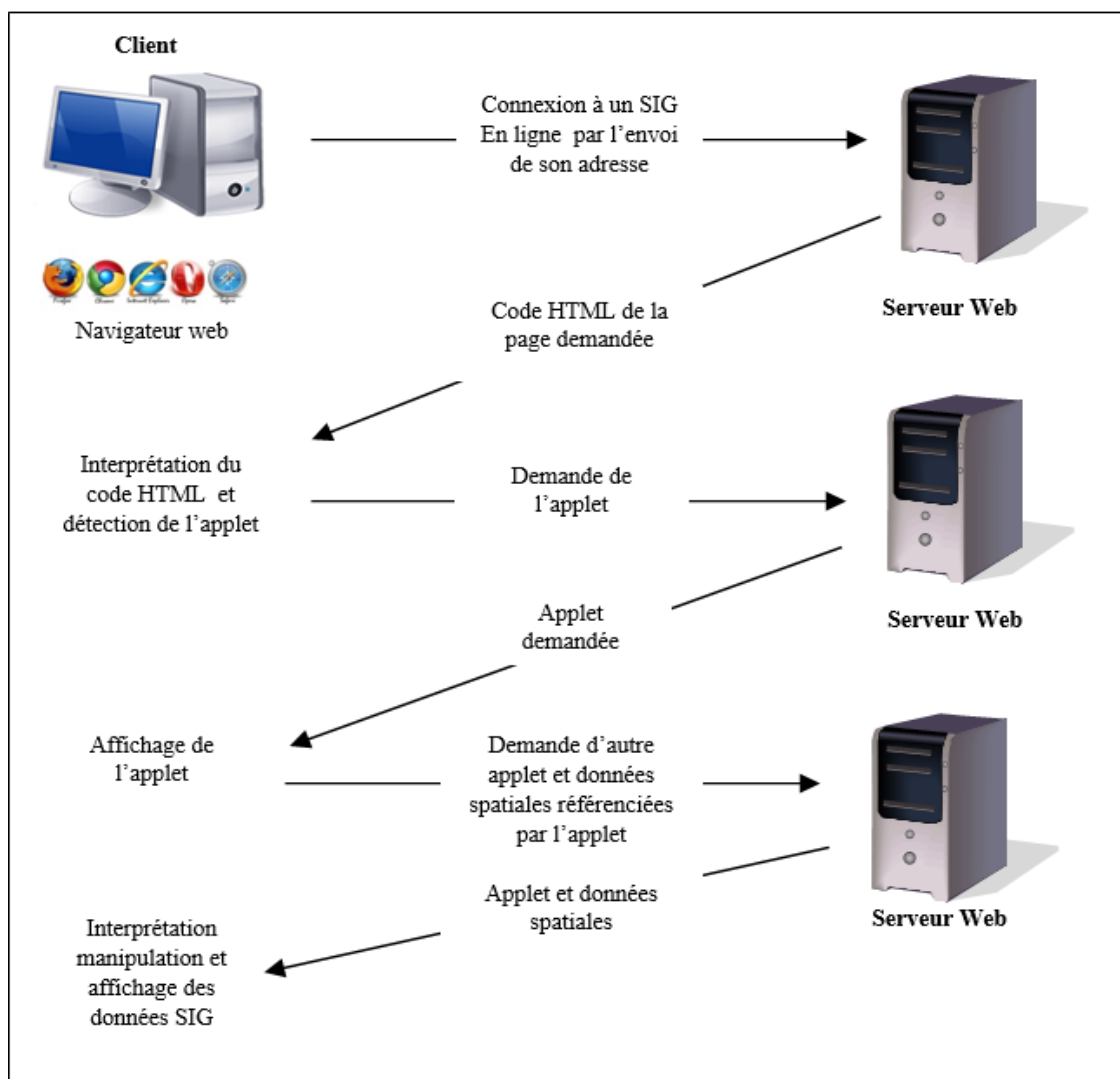


Figure 4.9: Etapes de fonctionnement de la solution Java.

### IV.4.3. Serveurs cartographiques dédiés :

Cette solution webmapping est la plus répandue car les fonctionnalités de tels systèmes sont réellement comparables avec celles de logiciels SIG de bureau (classiques). Plus complète, leur mise en place implique un plus grand investissement en matière de développement de l'architecture réseau et logicielle par rapport aux applications présentées précédemment.

L'utilisation de ce type de système induit donc l'installation d'un serveur http dédié aux calculs et aux mises en relation des cartes et des données selon les requêtes utilisateurs et les bibliothèques de fonctions informatiques disponibles. On remarque que dédier à une machine spécifique une tâche telle que le stockage des bases de données ou des composants cartographiques, l'affichage d'un texte ou d'une image ou encore la gestion des requêtes ou de la navigation permet d'accélérer et d'améliorer l'utilisation professionnelle d'un SIG en ligne.

Un développeur peut facilement adapter les interfaces et les outils disponibles via le monde du SIG en ligne open source aux besoins des acquéreurs.

L'incorporation de nouvelles fonctions personnalisées et liées aux requêtes ainsi qu'à la visualisation des données est possible. Comportant une administration en ligne et une gestion des droits d'accès poussée, ces outils permettent de visualiser des cartes complexes et interactives reliées à des bases de données mises à jour régulièrement [36].

Le point fort de toutes ces solutions se base sur leur compatibilité avec les formats des fichiers générés par d'autres applications. En effet, pour être compétitifs, la plupart des logiciels de webmapping sous licence libre offrent la possibilité de récupérer et d'utiliser des cartes développées à partir de programmes propriétaires. Ainsi, l'installation d'un nouveau système n'induit pas forcément une refonte des archives cartographiques avec un nouveau format d'utilisation.

Les solutions citées ci-dessus offrent des fonctionnalités plus au moins riches. Chacune adaptée à un domaine d'application précis et dépendant étroitement de l'architecture matérielle et logicielle adoptée, le tableau suivant (tableau 4.1) donne une abstraction du niveau de difficulté d'implémentation de fonctionnalités (liste non exhaustive) pour chacune de ces solutions :

Solution/ Fonctionnalités	Installation plug-in côté client	Source matricielle (raster)	Source vectorielle	Zoom	Déplacement	Gestion des couches	Gestion de l'échelle	Projection	Requêtes attributaires	Administration en ligne
HTML (Carte statique)		F	I	D	I	D	D	I	I	I
SVG	N	F	F	F	F	F	D	D	F	D
Applet Java	N	F	F	F	F	F	D	D	F	D
Serveur cartographique		F	F	F	F	F	F	F	F	F

Intégration    I : Impossible            D : Difficile            F : Facile            N : Nécessaire

**Tableau 4.1:** Tableau comparatif des fonctionnalités offertes par chaque solution [36].

#### IV.5. Conclusion :

L'usage d'outils de webmapping connaît un développement parallèle à celui d'internet et des réseaux en général. La possibilité de mettre en ligne les SIG a été facilitée par l'existence de technologies open source qui constituent aujourd'hui la référence dans le domaine du webmapping. A travers ce chapitre nous avons étudié les SIG en ligne, leur architecture ainsi que les différentes approches libres de mise en ligne des SIG tout en présentant leurs limites respectives afin de permettre de faire un choix pertinent de la solution à adopter.

Il est à noter qu'avec le développement rapide des outils open source, il est possible d'associer plusieurs techniques (serveur cartographique, applet java...) de webmapping dans le but d'enrichir de plus en plus en fonctionnalités l'interface du client et d'optimiser le temps de réponse du serveur. Dans la partie suivante, nous passons à la réalisation de notre système qui basé essentiellement sur la technologie serveur cartographique, nous y intégrons nos données vectorielles représentées dans un modèle topologique 3D pour le réseau routier et nous y implémentons le moteur de recherche de plus court chemin entre deux points distincts.

## Chapitre V :

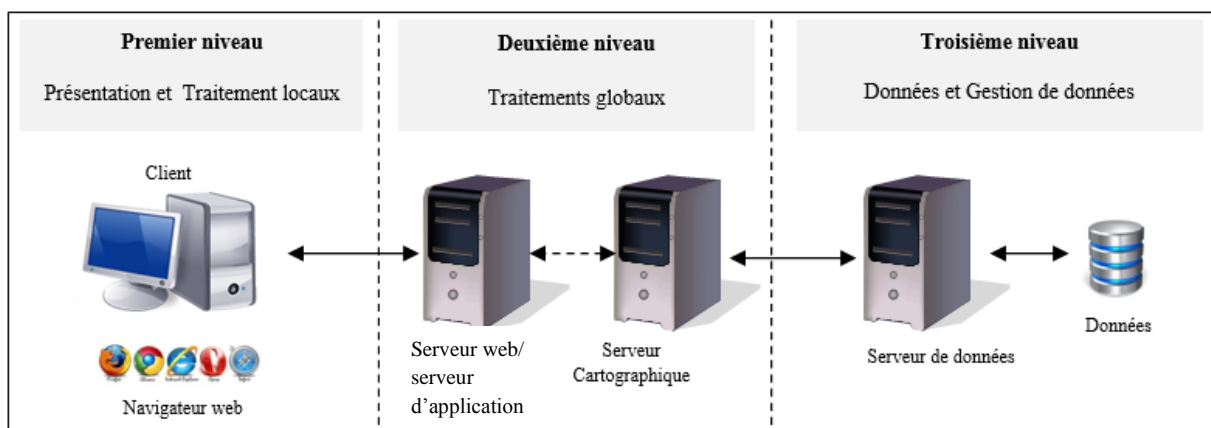
### Réalisation du système et présentation des résultats.

*Dans ce chapitre nous décrivons les étapes d'implémentation d'un système d'information géographique (SIG en ligne) fondé essentiellement sur l'approche serveur cartographique. Cette approche est considérée comme la plus complète en termes de fonctionnalités. Nous présentons, par la suite, la réalisation et l'intégration de l'outil de guidage. La dernière partie de ce chapitre est consacrée à l'injection des tests et à la présentation des résultats.*

#### V.1. Réalisation d'un SIG en ligne :

##### V.1.1. Architecture générale de la solution :

La mise en ligne d'un SIG fondée sur l'approche serveur cartographique est basée sur l'architecture Client/serveur à trois niveaux. Le premier niveau est représenté par un client distant. Un serveur web couplé à un serveur d'application et à un serveur cartographique constitue le deuxième niveau. Le troisième niveau est représenté par un serveur de données (figure 5.1). Les données peuvent parvenir de différentes sources (fichiers de formes, fichiers tabulaires, Images, cartes...) et sont éventuellement stockées dans une base de données spatiale.



**Figure 5.1 :** Architecture générale d'un SIG en ligne basé sur l'approche serveur cartographique.

### V.1.2. Composants et outils open source utilisés :

L'utilisation en ligne des Systèmes d'Informations Géographiques sous licence libre est en pleine expansion. Pour l'instant, l'installation de tels systèmes reste complexe et demande un nombre important d'heures de travail. De telles technologies requièrent une bonne maîtrise pour configurer le serveur, développer le moteur de mise en page, exporter et traiter les données, gérer les utilisateurs ou encore les systèmes de stockage.

Pour la réalisation de notre système, le choix des outils open source a été basé sur une plage de critères commençant par la robustesse en allant vers la popularité.

#### a. Client (navigateur Web) :

Dans l'architecture citée précédemment le client est une application distante représenté par un navigateur web. Il permet d'envoyer des requêtes http au serveur Web et d'en interpréter la réponse. Les navigateurs sont aujourd'hui capables de travailler également avec le protocole FTP et d'afficher plusieurs formats notamment les formats géographique tels que (XML, SVG, GeoJson...).

#### b. Serveurs Web et Serveurs d'application :

Un serveur web est un logiciel chargé d'interpréter les requêtes du client par la transmission des pages HTML et leur contenu statique (Images, feuilles de style...) présents sur le serveur. Si la requête porte sur un contenu dynamique, elle est aiguillée vers le serveur d'application afin de bâtir une réponse paramétrée, si cette réponse est constituée d'informations géographiques le serveur d'application sollicite le serveur cartographique pour la construction dynamique des cartes. La réponse, une fois construite, est renvoyée au client par le serveur web.

Le serveur web utilisé dans notre système est le serveur web *Apache*. Il est développé par « *The Apache Software Fondation* » et est doté d'une licence libre portant le nom de licence Apache [2]. Etant le serveur web le plus répandu, il est réputé par sa robustesse, sa rapidité et sa facilité de configuration [2].

Le serveur d'application, quant à lui, se charge aussi bien des contrôles au niveau de l'Interface Homme-Machine (IHM) que de la logique applicative et des règles de gestion de l'application. Nous avons utilisé dans la réalisation de notre système le PHP (acronyme récursif pour « *PHP : Hypertext Preprocessor* »). Il témoigne un grand

succès auprès de la communauté du monde Unix, car il utilise un langage proche des scripts Shell.

### c. Serveurs cartographiques :

Il existe plusieurs plateformes open source pour mettre en ligne les données spatiales et des applications interactives sur le web. Nous étudions à travers une approche comparative deux serveurs cartographiques les plus répandus dans le monde open source : MapServer et GeoServer. Cette étude vise à faire ressortir les potentialités offertes par chaque serveur et à dégager leurs forces et leurs faiblesses.

- **GeoServer :**

GeoServer est un serveur cartographique libre lancé en 2001 par le projet « *The Open Planning Project* » sous licence GNU General Public License. Il assure l'édition et le stockage des objets spatiaux et il les rend accessibles aux travers d'un réseau. Il permet de publier et de modifier une grande variété de formats ouverts sous forme de cartes, d'images ou encore de données géographiques. Parmi ses qualités, on peut noter sa facilité d'utilisation et sa compatibilité avec différentes bases de données (Oracle spatial, PostGIS, etc.) protocoles (WFS, WMS, etc.) et fichiers cartographiques (SVG, SHP, etc.). Les capacités transactionnelles présentent ainsi un support solide pour l'édition partagée de cartes. Développé intégralement en Java, GeoServer peut intégrer des bibliothèques qui facilitent et accélèrent les opérations complexes telles que la prise en charge de nombreux formats de données cartographiques ou encore les transformations et traductions de systèmes de coordonnées spatiales [21].

La bibliothèque GeoTools offre une large panoplie d'outils qui se rapprochent des fonctionnalités fondamentales de tout Système d'Informations Géographiques. Cette bibliothèque de fonctions algorithmiques est destinée à la projection cartographique, à la conversion des images matricielles et vectorielles ainsi qu'à l'adaptation des systèmes de coordonnées spatiales en fonction de la courbure terrestre et des mesures de références. GeoTools complète la liste de compatibilité avec les formats développés par les applications de SIG disponibles sur le marché. De nombreux formats libres et propriétaires sont pris en charge. Ainsi, il est possible de structurer et d'organiser de vastes bases de données et des fichiers géographiques enregistrés au format Shapefile, VPF ou encore GML [21].

La fluidité des interfaces issues du langage Java rend la gestion des objets géométriques et de leurs attributs intrinsèques intuitive. De plus, ce langage permet d'intégrer une grande diversité de formats. Il faut finalement noter qu'étant un produit récent, de nombreuses mises à jour sortent pour améliorer la stabilité et les fonctionnalités du programme [21].

- **MapServer :**

C'est un serveur cartographique développé originellement au milieu des années 90 par l'université de Minnesota en coopération avec la NASA et a été publié sous licence de type *MIT Style–licence*. MapServer est maintenant un projet de l'OSGeo, et est maintenu par un nombre croissant de développeurs tout autour du monde. Il a permis de grandes avancées aux niveaux des outils de gestion et de diffusion des cartes dynamiques sur Internet. Spécialisé dans l'affichage et l'édition de données géographiques, il n'est pas à lui seul un SIG complet. Il s'intègre au sein d'un système de serveurs dédiés au stockage, à l'affichage ou encore à la compilation des algorithmes [40].

Dans sa forme la plus basique, MapServer est un programme *CGI* qui reste inactif sur le serveur Web. Quand une requête est envoyée à MapServer, il utilise les informations passées par celle-ci et le fichier *Mapfile* pour créer une image de la carte demandée. La requête peut aussi renvoyer des images pour les légendes, les barres d'échelle, les cartes de référence, etc. Il peut aussi être étendu et personnalisé par l'utilisation du module *MapScript*. Ce dernier fournit une interface de Scripting pour MapServer pour la construction d'application Web. C'est une extension qui met à disposition les capacités de MapServer dans un langage de Scripting. Il existe actuellement pour *PHP*, *Perl*, *Python*, *Ruby*, *Tcl*, *Java* et *.NET* [40].

MapServer supporte en entrées de nombreux formats *vecteur* (*ESRI shapfiles*, *PostGIS*, *ESRI Arc SDE*, *Oracle Spatial*, *MySQL...*) à l'aide de la bibliothèque OGR (OpenGis Simple Features Reference) et *raster* (TIFF/GeoTIFF, EPPL7,...) via la bibliothèque GDAL (Geospatial Data Abstract Library) et il génère une multitude de formats *de sortie*. Il gère de nombreux standards de l'OGC : WMS (en client et en serveur), WFS (en client et en serveur), WMC, WCS, Filter Encoding, SLD, GML, SOS et OM. MapServer supporte la projection et la re-projection des cartes grâce à la bibliothèque PROJ4. Il génère, également, des cartographies thématiques en utilisant des logiques de classification ou d'expressions régulières [40].

N'ayant pas d'interface graphique prédéfinie, la création de celle-ci est un réel investissement en temps et en personnes. Il faut noter que le travail de plusieurs personnes sur un même projet à partir du réseau Internet est à l'origine difficile, voire impossible avec MapServer. Cependant, une bonne architecture réseau peut pallier ce problème d'administration multi-utilisateur à distance. Très efficace pour ce qu'elle a été conçu, cette solution requiert une bonne connaissance en langage de programmation. Le point faible reste que pour fonctionner, l'installation doit être faite sur une machine dédiée au sein du système.

Après cette brève étude ressortant les potentialités de chaque logiciel serveur cartographique, nous avons choisi d'utiliser pour notre système la plateforme MapServer. Malgré les problèmes liés aux compatibilités matérielles, MapServer est la solution la plus complète en termes de fonctionnalités qui sont de plus en plus nombreuses, variées et ouvertes à de nouveaux formats.

Nous ne sommes pas arrêtés à l'utilisation basique de MapServer nous l'avons également configuré afin qu'il diffuse des données répliant à la norme WMS (Web Map Service). Cela induit l'utilisation d'une syntaxe spécifique à cette norme dans la rédaction du *MapFile*. Nous avons, par la suite interfacé ce serveur par le client WMS OpenLayers.

OpenLayers est une bibliothèque JavaScript open source *coté client*, cela veut dire que le navigateur du client n'a besoin que du code de cette bibliothèque pour l'interpréter. Ce code peut être téléchargé et cohabiter avec le serveur cartographique ou bien être référencé directement à partir du site officiel du projet OpenLayers ([www.openlayers.org](http://www.openlayers.org)).

#### **d. Serveur de données :**

Les données étant principalement gérées par des serveurs dédiés, le serveur d'application et le serveur cartographique offrent des moyens d'interagir avec eux. Les Systèmes de Gestion de Bases de Données (SGBD) permettent d'interroger les données et de les mettre à jour. Le langage de manipulation de données adopté par la majorité des SGBD est le SQL (Structured Query Language).

Plusieurs SGBD Relationnels et/ou objets (MySQL, SQL server, PostgreSQL, Maria DB, Mongo DB,...) sont en concurrence dans le monde des SGBD libres [13]. Toutefois, dans le domaine cartographique le choix est restreint. MySQL supporte les

données géographiques et géoréférencées depuis sa version 4.1 [13]. Une autre solution est représentée par PostgreSQL et son extension géographique PostGIS.

Pour notre système nous avons utilisé le couple PostgreSQL/PostGIS. Nous argumentons notre choix par la comparaison des principales fonctionnalités géographiques de celui-ci à celles du très populaire MySQL qui est la référence des SGBD dans le monde de l'open source (Tableau 5.1) :

	<b>MySQL</b>	<b>PostgreSQL/PostGIS</b>
<b>Modèle Objet</b>	Point, Linestring, Polygon, GeometryCollection, MultiPoint, MultiLineString, MultiPolygon,	Point, Linestring, Polygon, GeometryCollection, Multipoint, MultiLineString, MultiPolygon, CircularString, Compound Curve, CurvePolygon, MultiCurve.
<b>Système de référence spatial (SRS)</b>	<ul style="list-style-type: none"> <li>• Ne gère pas les SRS</li> <li>• Permet le stockage des SRID</li> </ul>	<ul style="list-style-type: none"> <li>• Doté d'un moteur de projection Proj4</li> <li>• Définitions issues d'EPSG (plus de 2670 systèmes de projection)</li> <li>• Ajout de systèmes de projection possible</li> <li>• Faible support des coordonnées géocentriques</li> </ul>
<b>Opérateurs (Union, intersection, difference, symetricDifference, Buffer, Convex Hull...) et prédicats (Crosse, Touche, Within, Disjoint...).</b>	<ul style="list-style-type: none"> <li>• Les prédicats ne sont pas totalement implémentés.</li> <li>• Pas de support des opérateurs.</li> </ul>	<ul style="list-style-type: none"> <li>• Les prédicats sont totalement implémentés avec respect du nommage OGC et sont caractérisés par leur robustesse.</li> <li>• Support de tous les opérateurs</li> </ul>
<b>Métadonnées</b>	<ul style="list-style-type: none"> <li>• Pas de gestion des métadonnées</li> </ul>	<ul style="list-style-type: none"> <li>• Gestion des métadonnées avec respect des règles du nommage OGC</li> </ul>

**Tableau 5.1** : Comparaison de PostgrSQL/PostGIS à MySQL.

En effet, PostgreSQL/PostGIS est spécifiquement utilisé dans le domaine de la cartographie et est souvent associé à un serveur cartographique. PostGIS est la

cartouche spatiale qui confère au SGBD PostgreSQL le statut de base de données spatiale, la première version a été publiée en 2001 par « *Refractions Research* » sous la licence libre GNU General Public Licence. En 2006, PostGIS a été enregistré conforme aux standards établis par l'OGC. Il connaît un grand succès et ce dès les premières versions. En effet, en peu de temps, il est devenu un composant incontournable pour le fonctionnement d'un grand nombre de projets (MapServer, GeoServer, Open Street Map...) [2].

PostGIS permet de stocker et de manipuler l'information spatiale (en s'appuyant sur les bibliothèques GDAL pour le raster et OGR pour le vecteur) d'une manière complètement transparente et unifiée par de simples requêtes SQL en suivant la norme « *Simple Features Specification For SQL* » de l'OGC. Il gère plusieurs types de géométries et supporte des milliers de systèmes de projection cartographiques (grâce à la bibliothèque Proj4).

Plusieurs classifications universelles des systèmes de projection existent [54], la classification qui est très souvent utilisée par les outils open source, notamment par la librairie Proj4, est la classification EPSG (European Petroleum Survey Group). L'EPSG est une organisation scientifique impliquée dans l'exploration pétrolière, qui en 2005 a été reprise par l'OGP (International Association of Oil and Gas Producers) [32].

PostgreSQL/PostGIS offre, également, une large panoplie de fonctions (mille fonctions) dont la majorité offre des possibilités d'analyse très poussées. Cette liste peut être étendue par l'écriture de nouvelles fonctions avec n'importe quel langage de programmation, notamment le langage PL/PgSQL.

Les données géographiques sont intégrées dans PostgreSQL/PostGIS par des utilitaires d'importation de données, par exemple l'outil « *shp2pgsql* » intègre des fichiers shapefiles dans la base de données PostgreSQL/PostGIS. Ces données étant initialement sous le modèle spaghetti, PostgreSQL propose le calcul de la topologie par le biais de son extension PgRouting.

En utilisant les outils décrits précédemment, l'architecture détaillée de notre application se présente comme suit (Figure 5.2):

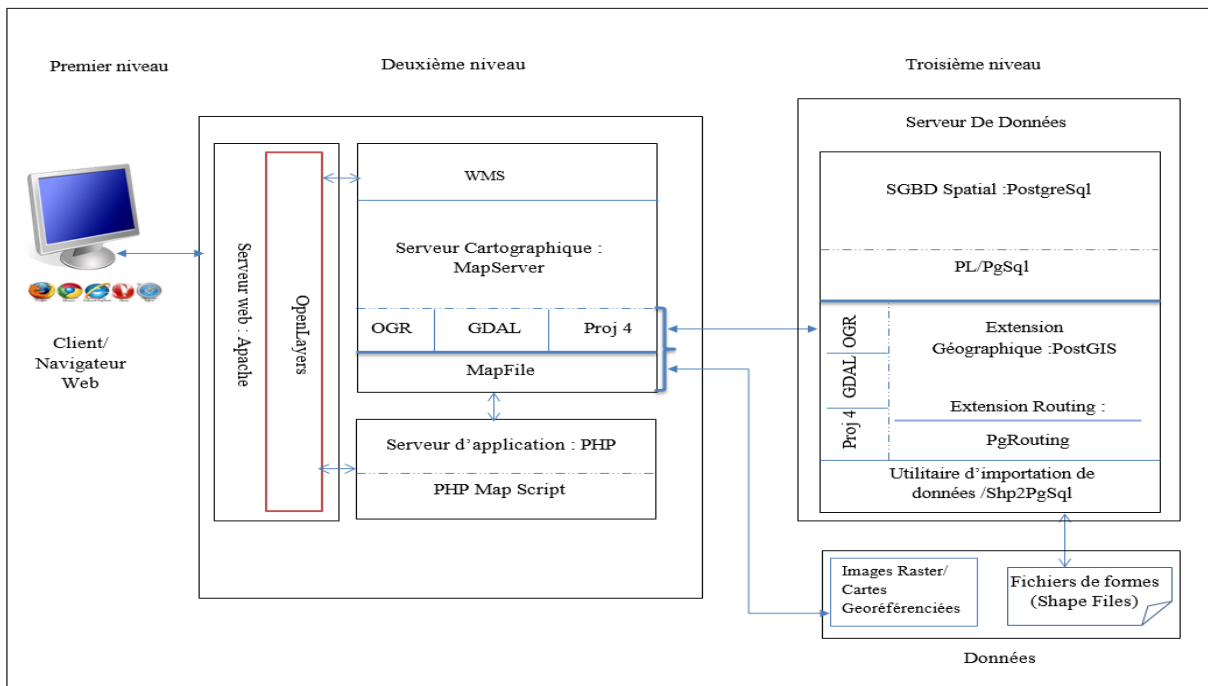


Figure 5.2 : Architecture détaillée du SIG en ligne réalisé.

### V.1.3. Présentation du SIG en ligne réalisé :

#### V.1.3.1. Interface Client :

Le SIG en ligne que nous avons réalisé est doté d'une interface client proposant les fonctionnalités suivantes (figure 5.3):

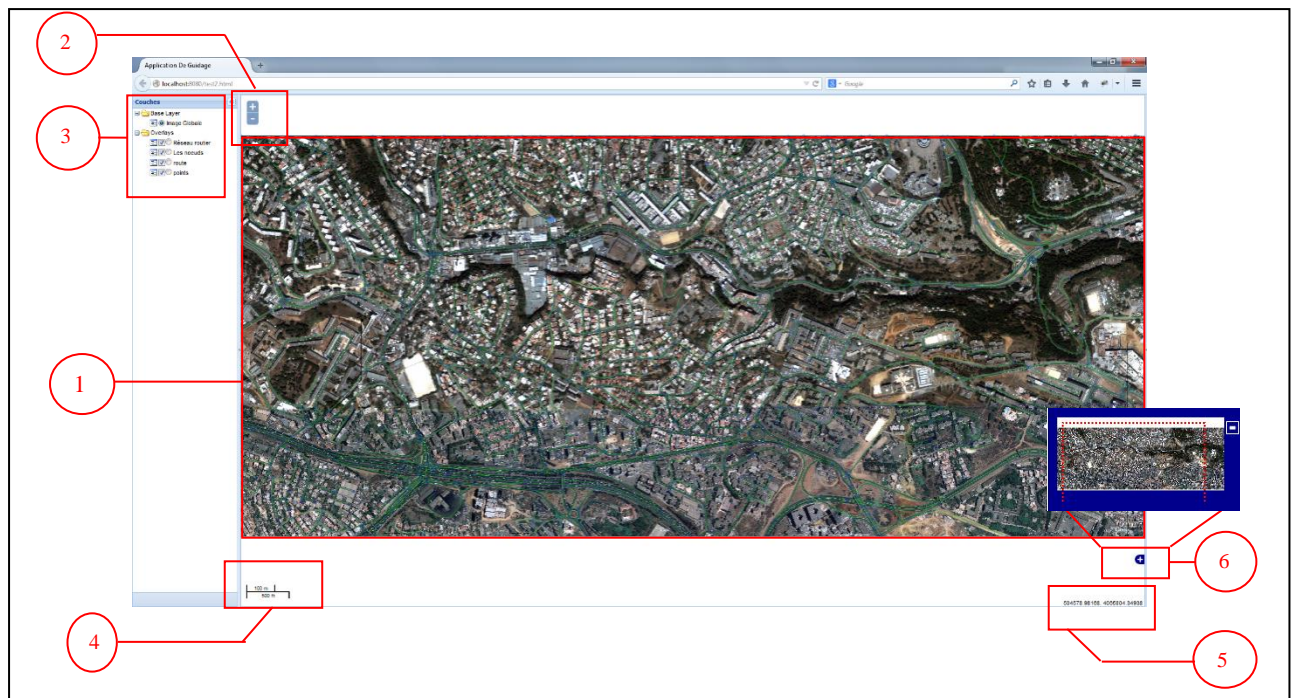


Figure 5.3 : Interface du SIG en ligne.

1. Visualisation des données géographiques : Une large partie de l'interface de notre système est réservée à la visualisation de la carte géographique, des données géographiques de différents types (raster et vecteur) sont affichées.
2. Zoom et déplacement : On peut zoomer sur n'importe quelle région de la carte affichée, soit en utilisant les boutons de zoom in / zoom out soit en utilisant la roulette de la souris. Le déplacement, quant à lui, se fait par l'opération « *cliquer-glisser* » de la souris ou en utilisant les boutons d'orientation du clavier.
3. Gestion des couches thématiques : les informations issues de la carte affichée sont réparties en couches. L'interface client assure la gestion de l'affichage de celles-ci, on peut afficher plusieurs couches simultanément comme on peut n'afficher que le fond de la carte.
4. Affichage d'une barre d'échelle.
5. Affichage des coordonnées géographiques des objets au survol de la souris.
6. Affichage d'une carte de référence : cette fonctionnalité donne une vue globale de la carte. Elle délimite la région affichée par rapport à la carte entière. Cette fonctionnalité facilite la navigation sur la carte.

### V.1.3.2. Fonctionnement du SIG en ligne :

#### 1. Génération des cartes :

En utilisant MapServer en tant que serveur WMS, les cartes produites par ce dernier peuvent être utilisées de manière pratique par n'importe quel client communiquant par le protocole WMS. Etant des cartes qui respectent les spécifications de l'interface WMS imposées par l'OGC, elles peuvent être combinées avec d'autres cartes sans avoir à être générées au même temps ou par le même serveur.

#### 2. Gestion des couches thématiques :

La couche thématique est décrite au niveau du fichier *mapfile*. Chaque couche est caractérisée, entre autres, par son nom, son type de données géographique, son système de projection, sa source de données (Image raster, fichier de formes, base de données géographique,...) et son statut (ON, OFF, DEFAULT). Les couches ayant un statut DEFAULT seront toujours envoyées au client tandis que les couches ayant un statut ON ou OFF peuvent être demandées par le client. Nous avons utilisé le client WMS OpenLayers, il conçoit deux types de couches les couches de fond appelées « *Base Layer* » et les couches couvrantes appelées « *Overlay Layer* ». Une

couche de fond est par défaut affichée. Deux couches de fond ne peuvent pas être affichées au même temps tandis qu'il n'y a pas de restrictions d'affichage pour les couches couvrantes on peut tout afficher comme on peut ne rien afficher.

### **3. Zoom et déplacement :**

Dans son fonctionnement basique, MapServer génère la carte et l'envoie entièrement au client. S'il est sollicité pour un zoom avant, il agrandit l'image et la renvoie au client. Ainsi, on se retrouve rapidement avec une image de très grande taille et difficile à gérer. Afin de remédier à ce problème OpenLayers utilise le principe de *tuiles*. Il divise l'image en un tas d'images de petite taille [32]. Donc, que ce soit pour un zoom ou un déplacement, MapServer est sollicité pour générer une série d'images de petite taille. La taille de la tuile est spécifiée lors de la création de la carte.

### **4. Gestion de la projection :**

MapServer gère la projection cartographique en s'appuyant sur la librairie Proj4, il associe à chaque carte le code EPSG du système de projection cartographique utilisé. Les couches thématiques héritent systématiquement ce système de projection si on ne leur définit pas explicitement leur code EPSG.

### **5. Requêtes spatiales :**

Notre système gère un cas particulier de requête spatiale, celui de l'optimisation d'itinéraires routiers entre deux points distincts. Et ce, depuis la formulation de la requête par l'introduction des paramètres et des données nécessaires jusqu'à l'affichage des résultats. La deuxième partie de ce chapitre se propose de détailler cet outil d'analyse.

## **V.2. Intégration de l'outil de guidage :**

La réalisation de l'outil de guidage représente une grande partie de notre travail. Cet outil permet l'affichage des données concernant le réseau routier et propose la détermination de l'itinéraire optimal entre deux points sur la carte.

### **V.2.1. Acquisition de données :**

L'acquisition de données constitue le point de départ de tout projet d'analyse SIG. Plusieurs méthodes d'acquisition de données existent soit par importation de données décrivant le

réseau routier en modèle spaghetti ou par la création de celles-ci (voir chapitre1). Pour acquérir les données pour notre système nous utilisons la méthode de digitalisation de la zone d'étude pour en extraire les données du réseau routier.

**a. Zone pilote d'étude :**

La zone pilote sur laquelle se base notre étude représente une partie du réseau routier de la région ouest d'Alger (503640.304 4064939.718, 507620.395 4066414.078) (figure 5.4). Cette zone est caractérisée par la diversité de son infrastructure routière.



**Figure 5.4 :** Zone d'étude.

Le système de projection utilisé pour cette image est le WGS 84, UTM, Zone 31 Nord. Son code EPSG est 32631.

**b. Extraction des données spatiales :**

Pour extraire les données spatiales (réseau routier) issue de la carte nous procédons par l'opération de vectorisation (figure 5.5). C'est une des fonctions essentielles dans le domaine cartographique car elle permet d'intégrer les données géographiques dans les SIG. Pour ce, nous reproduisons manuellement les formes des structures du réseau routier. La principale difficulté de cette phase réside dans la perception visuelle des limites des entités et dans le temps opératoire [4].

Le logiciel utilisé pour la vectorisation est le SIG de bureau open source **Quantum GIS** (QGIS). C'est un Système d'Information Géographique convivial distribué sous licence publique générale GNU. C'est l'un des projets officiels de la fondation Open Source Geospatial (OSGeo) [54].

L'opération de vectorisation du réseau routier revient à créer une couche de données vecteur de type *polyline* cette couche doit être projeté dans le même système cartographique de la

couche de fond afin de garantir leur superposition. Nous associons pour chaque arc vectorisé une liste de données attributaires (niveau, orientation, vocation,...). La couche réseau routier une fois créée, elle est enregistrée sous le modèle Spaghetti dans un fichier de formes *Shapefile* (.shp).



Figure 5.5 : Vectorisation du réseau routier.

### V.2.2. Intégration des données dans PostgreSQL/PostGIS :

Pour intégrer les données dans notre système nous utilisons l'adaptateur de données « *shp2pgsql* » qui convertit les données en modèle spaghetti (fichier de forme/shape file) en données adaptées (requêtes SQL) puis, il les insère dans la base de données PostgreSQL/PostGIS.

#### a- Calcul de la topologie du réseau routier :

Les données étant initialement sous le modèle spaghetti, aucune information sur la topologie ne leur est associée. Pour créer la topologie du réseau routier, nous devons passer par deux phases. La première phase consiste à « *nouer* » le graphe. Cela signifie que lorsque deux ou plusieurs arcs forment une intersection, un nœud doit être créé au niveau de cette intersection là et les arcs doivent être rompus pour permettre la navigation de l'un d'eux vers tous les autres connectés à cette intersection (figure 5.6). Cette opération est réalisée par l'exécution de la fonction « *Pgr\_nodeNetwork* () » offerte par l'extension PgRouting. Cette fonction va créer des nœuds au niveau de tous les croisements des arcs même ceux qui ne forment pas réellement une intersection (cas d'une route et d'un pont qui passe par-dessus).

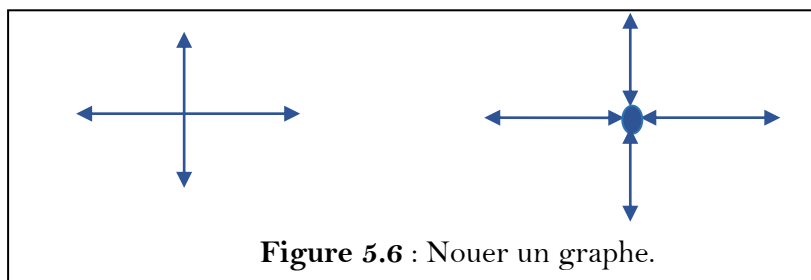
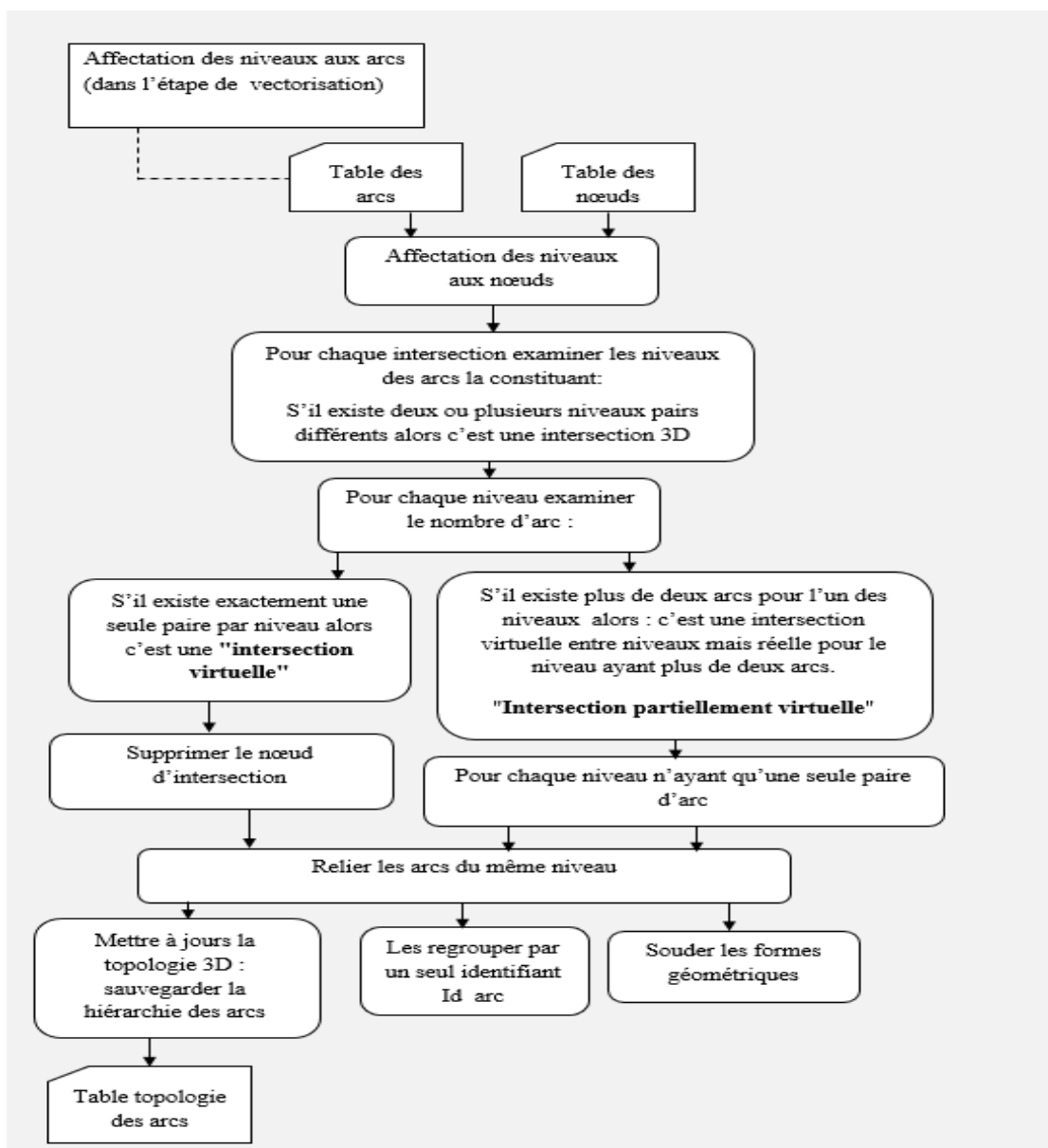


Figure 5.6 : Nouer un graphe.

Nous passons ensuite à la création de la topologie. Ici nos données passent du modèle spaghetti au modèle topologique 2D. PgRouting permet ce passage grâce à la fonction « *pgr\_createTopology()* ». Avant de lancer cette fonction des colonnes 'source' et 'destination' doivent être ajoutées à la table des tronçons. Cette fonction va créer la table des nœuds et remplir ces deux colonnes.

### b. Passage au modèle topologique 3D :

En passant au modèle topologique 3D (Solution proposée, chapitre3) nous allons éliminer les intersections virtuelles produites lors de la création de la topologie 2D (figure 5.7) cette opération est décrite par l'organigramme suivant (organigramme 5.1) :



Organigramme 5.1 : Processus de création de la topologie 3D.

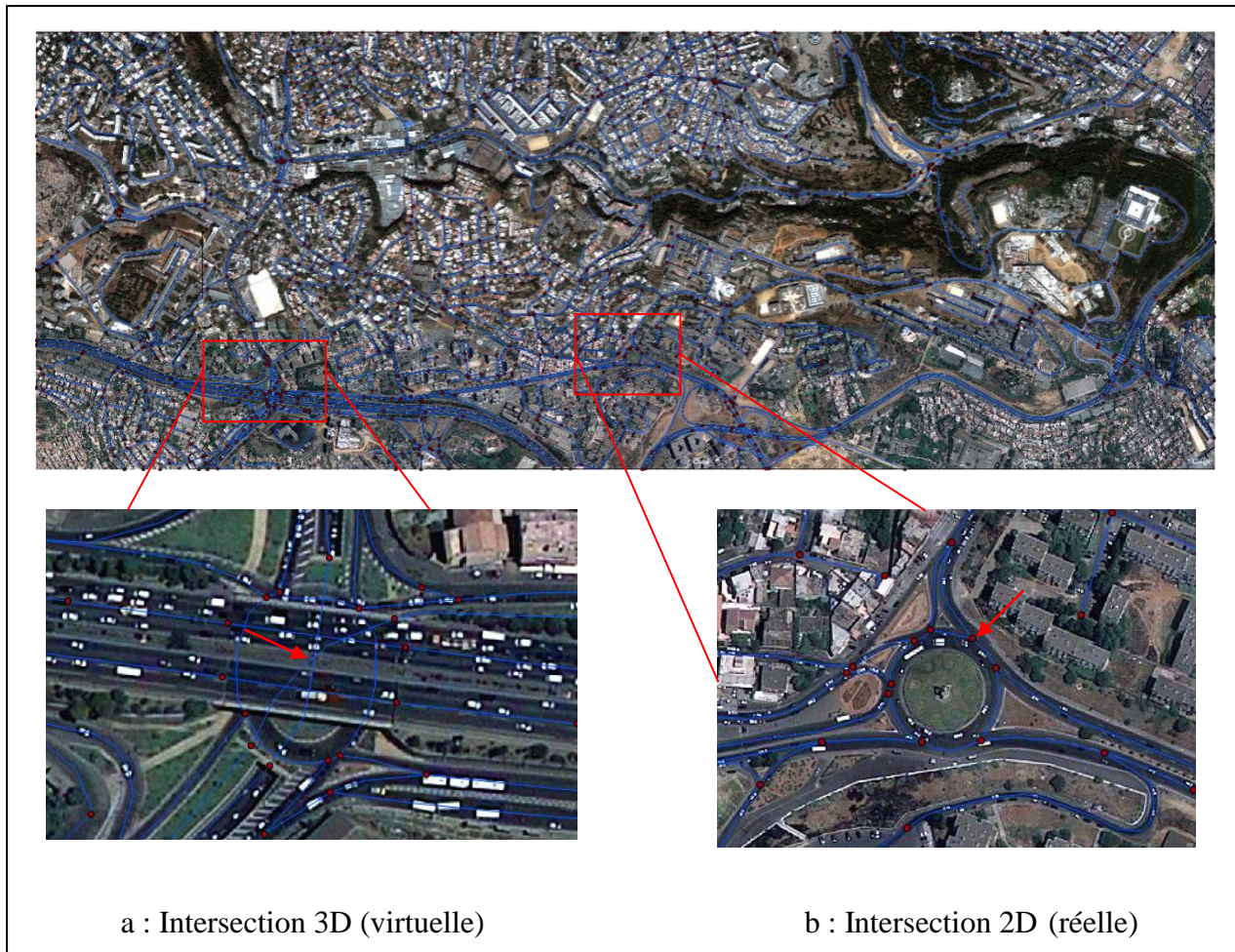


Figure 5.7 : Passage au modèle topologique 3D.

### V.2.3. Détermination des chemins optimaux :

#### a- Formulation de la requête :

Pour lancer une requête de recherche d'itinéraire optimal. Une partie de l'interface est réservée à l'introduction des paramètres de recherche (figure 5.8). L'utilisateur doit tout d'abord choisir le critère d'optimisation (distance, temps de parcours). Si le critère est « *le temps de parcours* » l'utilisateur doit introduire (choisir à partir d'une liste déroulante) la date et la période de déplacement. Ensuite, Il est amené à choisir le point de départ et le point d'arrivée en cliquant sur la carte. Le serveur est, alors, interrogé afin d'afficher le chemin optimal. L'utilisateur pourra, ensuite, déplacer son point de départ et/ou son point d'arrivée pour une mise à jour automatique du résultat. Cette partie de l'interface ainsi que la partie de gestion des couches thématique peuvent être réduites afin de mieux explorer le chemin optimal (figure 5.9).

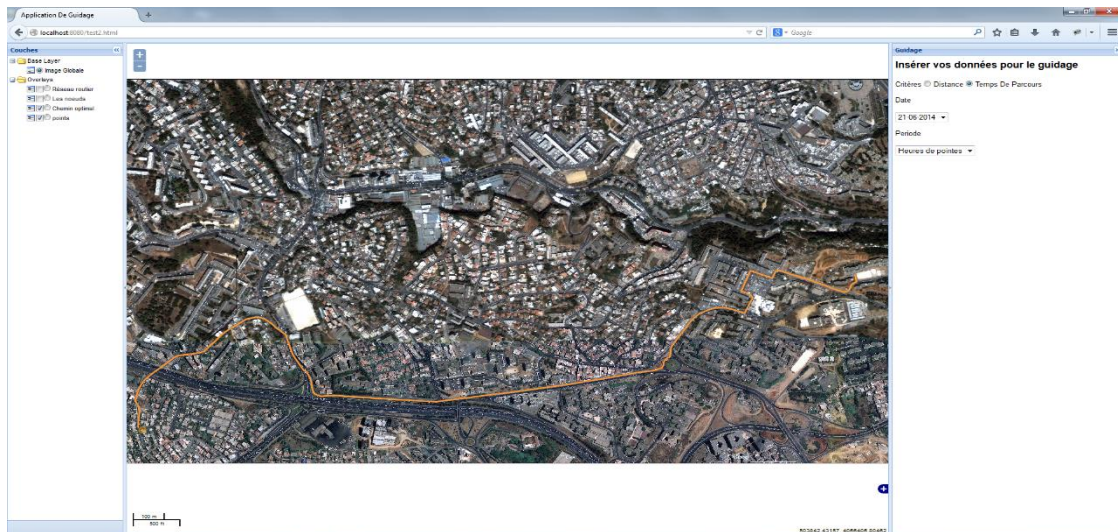


Figure 5.8 : formulation de la requête de recherche de l'itinéraire optimal.

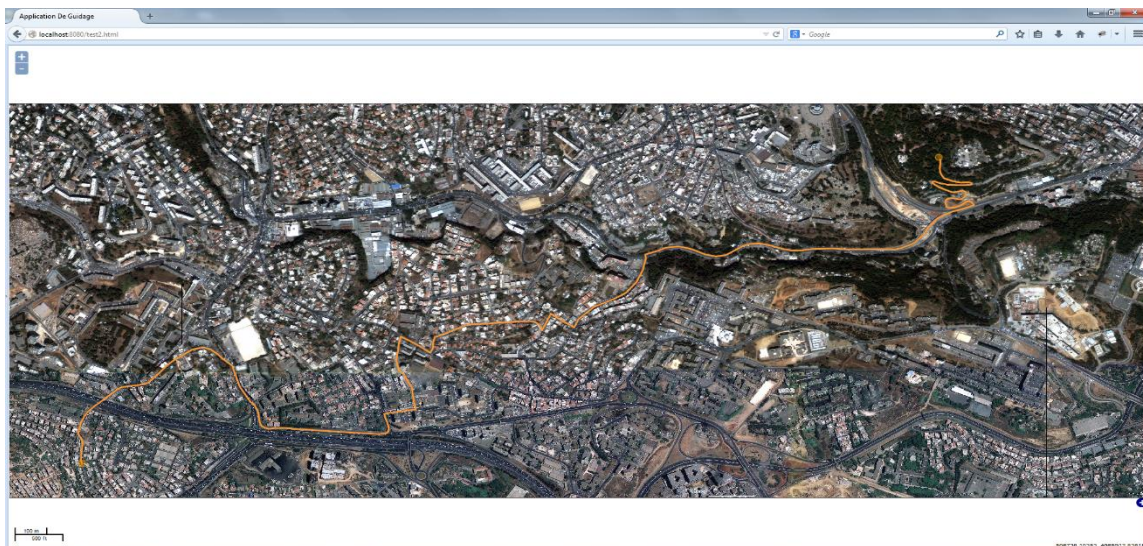


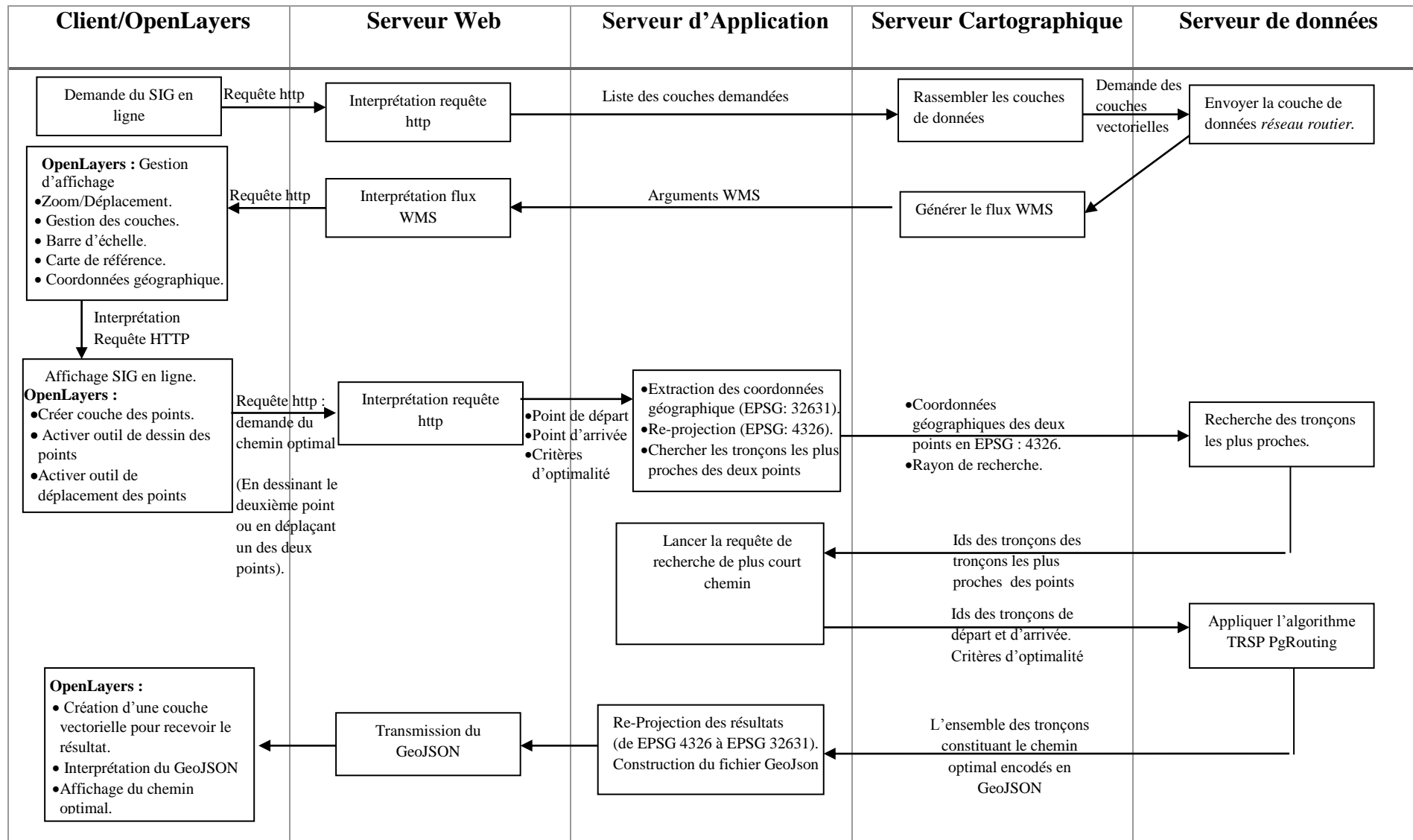
Figure 5.9 : Exploration de l'itinéraire optimal.

### b- Fonctionnement :

Pour permettre de dessiner les points et de les déplacer sur la carte nous utilisons les outils offerts par OpenLayers. Trois outils sont nécessaires : l'outil de dessin des points, l'outil de déplacement des points et une couche vectorielle pour contenir ces points. Les paramètres une fois introduits, ils sont envoyés au serveur Web par une requête http. Cette opération est déclenchée par l'insertion du point d'arrivée et ensuite, par le déplacement de l'un des deux points insérés. Le serveur Web reçoit et interprète la requête http et envoie les arguments de la

requête au serveur d'application. A ce niveau-là, le serveur d'application extrait les coordonnées géographiques des deux points projetés dans le système EPSG : 32631 et il les projette dans le système utilisé par PostGIS (WGS 84) dont le code EPSG est 4326. Les points insérés n'étant, généralement, pas des points du réseau routier une recherche des tronçons les plus proches des deux points est lancée. Pour ce, le serveur d'application transmet un rayon de recherche autour du point au serveur de données. Si le serveur de données ne retourne aucun tronçon le serveur d'application augmente la taille du rayon et relance la recherche. Le tronçon de départ et celui d'arrivée une fois obtenus, ils sont retransmis par le serveur d'application en tant qu'arguments pour la recherche du chemin optimal en utilisant l'algorithme TRSP (*Turn Restriction ShortestPath*) proposé par l'extension PgRouting. C'est un algorithme qui prend en compte les restrictions de circulation trouvées dans les réseaux routiers réels. Il est aussi performant que l'algorithme A\*. Il représente une application de la stratégie de détermination du chemin optimal avec séquences d'arcs pénalisées (chapitre 2). Il procède dans la recherche du chemin optimal par étiquetage des arcs et non pas des nœuds. Il utilise en plus de la table des « tronçons » la table des « séquences autorisées » (Modélisation du réseau routier. chapitre 3). Il retourne l'ensemble des tronçons constituant le chemin optimal. Dans ce genre de solution, l'affichage des résultats se fait, dans la majorité des cas, par MapServer. En effet, MapServer récupère les données vectorielles, génère le flux WMS et l'envoi au serveur Web pour le transmettre au client via le protocole http. La transmission des résultats implique le rechargement de toute la page web, ce qui signifie le rechargement du SIG en ligne et toutes ces fonctionnalités ainsi que toutes les couches de données, cela induit un temps de réponse relativement long. Pour pallier ce problème, nous utilisons la technologie AJAX (*Asynchronous JavaScript and XML*). Cette technologie permet d'envoyer en arrière-plan des demandes au serveur web. Puis modifier le contenu de la page web actuellement affichée par le navigateur en fonction des résultats reçu du serveur évitant ainsi la transmission et l'affichage de la page complète. Dans cette technologie les échanges de données entre client et serveur utilisent principalement le XML mais d'autres formats peuvent être utilisés tels que JSON (*JavaScript Object Notation*). C'est un format de données inspiré de la syntaxe du langage JavaScript. Il est utilisé pour structurer les informations envoyées par le serveur Web [wiki]. Pour la manipulation des objets géographiques, le GeoJSON (*Geographic JSON*) existe. C'est une variante du JSON supportant des données géographiques (point, ligne, polygone). Ainsi, l'ensemble des tronçons constituant le chemin optimal est encodé en GeoJSON et sans passer par le serveur cartographique, est envoyé au serveur d'application pour la re-projection dans le système EPSG : 32631 puis au serveur web pour la transmission au client. OpenLayers étant une application JavaScript qui s'exécute au niveau du client, est capable de manipuler du

GeoJSON. A ce niveau, une couche vectorielle de type ligne est créée et les tronçons y sont affichés. Ce fonctionnement est résumé par l'organigramme suivant (organigramme 5.2) :



Organigramme 5.2 : Processus de détermination de chemin optimal dans un SIG en ligne.

**V.3. Conclusion :**

Dans ce chapitre, nous avons présenté le SIG en ligne réalisé. Nous avons commencé par décrire son architecture, la liste des outils open source utilisés au cours de son développement et son fonctionnement. La deuxième partie de ce chapitre est consacré à la détermination de chemins optimaux dans un SIG en ligne. Nous l'avons abordé par le choix d'une zone d'étude que nous avons vectorisée manuellement. Les données sont, par la suite, intégrées dans un SGBD spatial. Nous nous sommes ensuite penché sur l'intégration d'une application de guidage en décrivant la formulation de la requête et en détaillant le processus de détermination des chemins optimaux dans un SIG en ligne.

## Conclusion

Le travail réalisé dans le cadre de cette recherche a porté essentiellement sur la réalisation d'un SIG en ligne fondé sur l'approche serveur cartographique selon une architecture Client/serveur à trois niveaux. Le premier niveau est un client offrant une large gamme de fonctionnalités permettant la manipulation de l'information géographique. Les traitements se situent dans le deuxième niveau où nous avons couplé le serveur web/application à un serveur cartographique permettant ainsi la création dynamique des cartes. Le serveur cartographique crée la carte répondant à la requête du client et le serveur web se charge de la transférer à travers le réseau. Dans le troisième niveau les données géographiques sont stockées dans un serveur de données via un SGBD spatial. Ces données sont sollicitées par le serveur cartographique lors de la construction des cartes. Dans ce système d'information géographique, nous avons implémenté un outil de guidage permettant la détermination d'itinéraire optimal dans le réseau routier de la région ouest d'Alger. Les données décrivant le réseau routier sont importées dans la base de données et passent automatiquement du modèle topologique 2D au modèle topologique 3D, remédiant ainsi, à la prise en compte des intersections virtuelles engendrées par le croisement d'un tronçon de route avec un autre tronçon passant par-dessous ou par-dessus. L'outil de guidage implémenté détermine le chemin optimal entre deux points distincts selon la distance et le temps de parcours. Dans ce dernier cas, l'outil est alimenté par des estimations dynamiques des temps de parcours. Comme il prend en compte les restrictions de circulation par l'utilisation de l'algorithme TRSP (Turn Restriction Shortest Path). C'est un algorithme offert par PgRouting. Aussi performant que l'algorithme A\*, il procède par étiquetage des arcs et non pas des nœuds.

Les perspectives de recherche pour l'amélioration du système développé sont diverses. Elles concernent à la fois, l'ajout de nouvelles fonctionnalités au SIG en ligne développé et l'amélioration de la fiabilité des informations diffusées. Rendre la fonctionnalité de digitalisation du réseau routier interne au système, facilite l'intégration des données dans celui-ci.

Une extension très naturelle à l'outil de guidage intégré consiste à s'intéresser à l'optimisation d'itinéraires multimodaux qui offrent plusieurs modes de transport entre une origine et une destination. Par ailleurs, le système d'information géographique et de guidage réalisé pourrait regrouper plusieurs agglomérations, voir même couvrir l'ensemble du territoire national.

## Références Bibliographiques :

- [1] : R.Amara, " *Etude des fonctionnalités et modalités de mise en œuvre des SIG en ligne* ", mémoire de magistère, Université des Sciences et des Technologies Houari Boumediene, 2010.
- [2] : M. Arezki et N. Zerguini, " *Conception et réalisation d'un système d'information géographique (SIG) pour l'aide à la gestion des transports en commun*", Projet de fin d'étude, Institut National d'Informatique, 2011.
- [3] : R. Bellman, " *On a routing problem. Quarterly of Applied Mathematics*" 87-90, 1958.
- [4] : Y. Beskri, " *Traitement du MNT et extraction du réseau hydrologique d'un bassin versant*", mémoire de magistère, Université des Sciences et des Technologies Houari Boumediene, 2010.
- [5] : B. Bessaa, A. Belhadj-aissa, M. Belhadj-aissa. " *Modèle Topologique 3d pour le réseau routier*". XIIème Journées Scientifiques du Réseau Télédétection de l'A.U.F.- 2010 du 23 au 25 novembre 2010 à Monastir Tunisie. [www.js-auf-tun2010.org](http://www.js-auf-tun2010.org).
- [6] : P. Bourcier, " *Modélisation des données et des relations spatiales et temporelles*", Note de cours DEA : ENVP EDEL, 2008.
- [7] : A. Bousquet, " *Optimisation d'itinéraires multimodaux fondées sur les temps de parcours à l'échelle d'une agglomération urbaine dense*", thèse de doctorat, Ecole nationale des travaux publics de l'état, 2010.
- [8] : M. Bigand, J-P. Bourey, H. Camus, D. Corbeel, " *Conception des Systèmes d'information: Modélisation des données, études de cas*", TECHNIP Edt., 2006.
- [9] : Y.-L. Chou, H. E. Romeijn et R. L. Smith, " *Approximating shortest paths in large-scale networks with an application to intelligent transportation systems*". *INFORMS Journal on Computing*, 10(2):163\_179, 1998.
- [10] : A. De La Losa, " *Modélisation de la troisième dimension dans les bases de données géographiques*", Thèse de Doctorat, Université Marne-La-Vallée, 2000.
- [11] : G. Della Valle et D. Tartaro : " *Ricerca dei percorsi di minimo costo mediante un algoritmo ibrido in presenza di penalità di svolta* ". *Ricerca Operativa*, 72:5\_38, 1994.
- [12] : E. V. Denardo et B. L. Fox , " *Shortest-route methods : 1. reaching, pruning and buckets*". *Operations Research*, 27:161\_186, 1979.
- [13] : Club des développeurs, " *Quel SGBD choisir*", [www.facade.developpez.com/sgbdcmp/](http://www.facade.developpez.com/sgbdcmp/), dernière mise à jour 2010.
- [14] : E. W. Dijkstra , " *A note on two problems in connexion with graphs*". *Numerische mathematik*, 1959.
- [15] : J.-F. Dillenburg et P.-C. Nelson, " *Improving search efficiency using possible subgoals*", *Mathematical and Computer Modelling*, 22:397\_414, 1995.

- 
- [16] : D. Dromard, D. Seret, "*Architecture des réseaux*", Pearson Education France, 2006.
- [17] : P. Dugerdil, "*Impact des décisions informatiques: introduction à l'informatique pour décideurs non informaticien*", PPUR presses polytechniques, 2005.
- [18]: L. R. Ford, "*Network flow theory*". RAND, 1956.
- [19]: L. Fu, D. Sun, L.R. Rilett, "*Heuristic shortest path algorithms for transportation applications*" Computers and Operations Research, pp 24-43, 2006.
- [20] : B. Furht, "*Internet Architectures*", in computer engineering handbook, CRC Press, 2002.
- [21] : Geosever, [www.geoserver.org](http://www.geoserver.org). Dernière mise à jour 2014.
- [22] : C. Godinot et P. Bonnel , "*Phase d'affectation : mise en forme du réseau de routier sur l'Aire Urbaine de Lyon*" : Rapport intermédiaire n°8 du projet Simbad (Simuler les MoBilités pour une Agglomération Durable). 2008
- [23] : A. Goldberg et C. Harrelson, "*Computing the shortest path : A\* search meets graph theory*". Rapport technique, Microsoft Research, mars 2003.
- [24] : A. Goldberg et H. Kaplan et R. Werneck, "*Reach for A\*: efficient point-to point shortest path algorithms*". Rapport technique, Microsoft Research, Octobre 2005.
- [25] : A. V. Goldberg, "*Point-to-point shortest path algorithms with preprocessing*". LCNS, 4362:88\_102, 2007.
- [26] : D. Green, T. Bossomaier, "*Online GIS and Spatial Metadata*", Taylor & Francis Edt., 2002.
- [27] : G. Gröger et L. Plümer, "*How to Get 3-D for the Price of 2-DV Topology and Consistency of 3D Urban GIS*", *GeoInformatica* 9:2, 139–158, Springer, 2005.
- [28]: G. Gröger et L.Plümer, "*How to Achieve Consistency for 3D City Models In: Geoinformatica*", 14. Jg, 2010.
- [29] : A. Goldberg et C. Harrelson, "*Computing the shortest path: A\* search meets graph theory*", Microsoft Research, mars 2003.
- [30] : E. Gutiérrez et A. L. Medaglia, "*Labeling algorithm for the shortest path problem with turn prohibitions with application to large-scale road networks*". *Annals of Operations Research*, 157:169 \_ 182, 2008.
- [31] : T. Hächler, "*Online visualization of spatial data. A prototype of an open source internet map server with backend spatial database for the swiss national park*", Diploma Thesis, Department of Geography, University of Zürich, 2003.
- [32] : Hazzard, "*OpenLayers Beginner's Guide*", PACKT publishing, 2011.
- [33] : T. Horanont, N. K. Tripathi, and V. Raghavan, "*A Comparative Assessment of Internet GIS Server Systems*", dans *Map Asia*, 2002.
- [34] : W. Jia, W. Zhou, "*Distributed Network Systems: From Concepts to implementations*", Springer Edt., 2005.

- 
- [35] : D. B. Johnson, "*On shortest paths and sorting*". In Proceedings of the 25th ACM Annual conference, pages 510\_517, 1972.
- [36] : L. Kaddouri, "*Développement d'une application webmapping*", Rapport de stage, université d'Avignon et des pays de Vaucluse, 2007.
- [37] : A. Kamoun, "*Conception d'un système d'information pour l'aide au déplacement multimodal : une approche multi-agents pour la recherche et la composition des itinéraires en ligne*". Thèse de doctorat, Ecole Centrale de Lille et Université des Sciences et Technologies de Lille, 2007.
- [38] : R. F. Kirby et R. B. Potts, "*The minimum route problem for networks with turn penalties and prohibitions*". Transportation Research, 3:397\_408, 1969.
- [39] : B. Lachance, et J. Pouliot et D. Kirkwood, "*3D Topological analysis of geological models*". Geomatica, 2005.
- [40] : MapServer , [www.mapserver.org](http://www.mapserver.org), dernière modification 2013.
- [41] : D. McFarland, "*Client/Server Computing*", in The Internet Encyclopedia, Wiley ans Sons Edt., 2004, pp 194-203.
- [42] : R. H. Möhring et H. Schilling, B. Schütz, D. Wagner et T. Willhalm, "*Partitioning graphs to speed-up dijkstra's algorithm*". J. Exp. Algorithmics, 11, 2006.
- [43] : E. F. Moore, "*The shortest path through a maze*". In Harvard University Press, éditeur: Proceedings of international symposium on theory of switching, part 2, pp. 285\_292, 1959.
- [44] : D. Moretz, "*Internet GIS*", Department of Geography, University of Minnesota, Minneapolis, USA, in Encyclopedia of GIS, Springer Edt., 2007.
- [45] : S. Namkoong et J.H. Rho et J.U. Choi, "*Development of the tree-based link labeling algorithm for optimal path-finding in urban transportation networks*". Mathematical and computer modelling, 27:51\_65, 1998.
- [46] : M. Neteler, H. Mitasova, "*Open source GIS, a GRASS GIS Approach*", Springer Edt., Third Edition, 2008.
- [47] : A. Neumann, "*Web Mapping and Web Cartography*", Institute of Cartography, ETH Zurich, in Encyclopedia of GIS, Springer Edt., 2007.
- [48] : J.A. T. Nicholson, "*Finding the shortest route between two points in a network*". Computer Journal, pp 9:275\_280, 1966.
- [49] : S. Pallottino et M. G. Scutellà, "*Equilibrium and Advanced Transportation Modelling, chapitre Shortest path algorithms in transportation models: classical and innovative aspects*", pages 245 -281. Kluwer Academic Publishers, 1998.
- [50] : Z.R. Peng, "*An Assessment of The Development of Internet GIS*", Abstract for Journal of the Urban and Regional Information Systems Association, 1997.
- [51] : Z.R. Peng, M.S. Tsou, "*Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks*", Wiley Edt., New York, 2003.
- [52] : Y. PIGNÉ, "*modélisation et traitement décentralisé des graphes dynamiques : application aux réseaux mobiles ad hoc*". Thèse de doctorat en informatique, université du Havre, 2008.

- 
- [53] : H. Pornon et P. Yalamas et E. Pelegris, "*Services Web géographiques, état de l'art et perspectives*", Géomatique Expert, Oct/Nov 2008.
- [54] : QGIS, [www.qgis.org](http://www.qgis.org), dernière mise à jour 2014.
- [55] : USGS
- [56] : W. Qiang et X. Hua et Z. Wanfang, "*Development of a 3D GIS and its application to karst areas*". Environ Geol 54:1037–1045 Springer, 2008.
- [57] : S. M. Rahman, R. J. Bignall: "*Internet Commerce and Software agents: Cases, technologies and opportunities*", Idea Group Inc., 2001.
- [58] : F. Ramos, "*Modélisation et validation d'un système d'information géographique 3D opérationnel*". Thèse de doctorat, Université de Marne-La-Vallée, 2003.
- [59] : F. Schulz, "*Timetable information and shortest paths*". Thèse de doctorat, Fakultät für Informatik der Universität Fridericiana zu Karlsruhe, January 2005.
- [60] : 72 R. Sedgewick et J. S. Vitter, "*Shortest paths in euclidean graphs*". Algorithmica, 1(1):31\_48, 1986.
- [61] : M. Souris, "*Principe généraux des SIG*", Formation SIG : Institut de recherche pour le développement, 2008.
- [62] : C. de Souza Baptista, Z. Kemp, "*Spatial Information Systems and the World Wide Web*", Integrating Information Infrastructure with Geographic Information Technology, Innovation in GIS 6, University of Edinburgh, Bruse Gittings Edt., 2005.
- [63] : D.Wagner et T. Willhalm, "*Geometric speed-up techniques for finding shortest paths in large sparse graphs*". In 11th European Symposium on Algorithms (ESA 2003), 2003.
- [64] : A. K. Ziliaskopoulos, H. S. Mahmassani, "*Time-dependent shortest path algorithm for real-time intelligent vehicle highway system applications*". Transportation Research Record: Journal of the Transportation Research Board, p 94-100, 1993.
- [65] : A. Ziliaskopoulos et H. S. Mahmassani, "*A note on least time path computation considering delays and prohibitions for intersection movements*". Transportation Research Part B, 30(5):359 \_ 367, 1996.
- [66] : A. Ziliaskopoulos et W. Wardell, "*An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays*". European Journal of Operational Research, 125(3), 2000.