

N° d'ordre :

REPUBLIQUE ALGERIENNE DEMOCRATIQUE & POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR & DE LA RECHERCHE
SCIENTIFIQUE



UNIVERSITE DJILLALI LIABES
FACULTE DES SCIENCES EXACTES
SIDI BEL ABBES

THESE DE DOCTORAT EN SCIENCES

Présentée par

Cheikh MOUILAH

Spécialité : Informatique

Option : Intelligence Artificielle

Intitulée

Techniques d'optimisation à base d'Approche bio-inspirée

Soutenue le : 17/12/2020

Devant le jury composé de :

Président : Dr. Boukfi hacene Sofiane, M.C.A à l'université de Sidi Bel Abbès

Examineurs :

Pr. Mimoun Malki, Professeur à l'E.S.I de Sidi Bel Abbès

Pr. Benslimane Sidi Mohammed, Professeur à l'E.S.I de Sidi Bel Abbès

Dr. Gafour abdelkader, M.C.A à l'université de Sidi Bel Abbès

Dr. Adjouj Réda, M.C.A à l'université de Sidi Bel Abbès

Directeur de thèse :

Pr. Rahmoun Abdellatif, Professeur à l'E.S.I de Sidi Bel Abbès

Année universitaire : 2019-2020

Remerciements

Tout d'abord, je tiens à remercier Dieu, qui nous a donné la force, la volonté, la conscience et la patience d'accomplir ce modeste travail.

Nous sommes incapables d'exprimer notre gratitude envers notre encadreur : Mr : Abdellatif RAHMOUN et le moins que nous pouvons traduire c'est une profonde reconnaissance pour l'aide scientifique et pédagogique qu'il nous ait apporté durant l'élaboration de ce projet.

Nous remercions le président et les membres de jury d'avoir accepté de juger ce modeste travail.

Nous exprimons notre reconnaissance à messieurs : Prof. A.LAKMECHE, Prof. R. ADJOUJ, Dr : M.HELAL et les autres enseignants du laboratoire « Biomathématique » qui grâce à leurs services ont participé activement à la bonne marche de nos travaux.

Mr. MOUILAH Cheikh

Dédicace

À la mémoire de mon cher père et ma grand-mère,
à ma mère,
à mon épouse,
à mes sœurs et mes frères,
à mes enfants et
à tous les membres de ma famille et toutes les personnes
qui ont contribué à travers leurs soutiens moraux ou
matériels.

Je dédie ce travail.

Mr. MOUILLAH Cheikh

Table des matières

	Page
Liste des figures et des tableaux	8
Introduction générale	12
Chapitre 1 : Analyse, Modélisation et Numérisation du réseau routier urbain	
1. Introduction	16
2. Terminologies et définitions	17
3. Etude de l'espace urbain	18
3.1. Analyse spatiale	18
3.1.1. Le système des coordonnées géographiques	20
3.1.2. Le système des coordonnées projetées	20
3.2. La modélisation de l'espace urbain	21
3.2.1. Introduction	21
3.2.2. La modélisation urbaine basée sur la géométrie de l'espace : Modèle de Lynch	22
3.2.3. La modélisation urbaine basée sur la fonctionnalité de l'espace	23
a. Le modèle cristallin	23
b. La théorie de la zone concentrique	23
c. La théorie de secteurs	24
d. Modèle des noyaux multiples	24
3.2.4. La modélisation urbaine basée sur la topologie de l'espace	25
4. La modélisation numérique de l'espace urbain	27
4.1. Introduction	27
4.2. La dualité Raster / Vecteur	28
4.3. La topologie	28
4.4. L'information géographique	31
4.4.1. Définitions	31
4.4.2. La représentation de l'information géographique	32
4.4.3. Classification des informations géographiques	33
5. Conclusion	34
Chapitre 2 : Le chemin optimal multiobjectif dans un réseau urbain	
1. Introduction	38
2. Terminologies et définitions	39
3. Problème d'optimisation « multiobjectif »	39
3.1. Définition :	39
3.2. Méthodes d'optimisation « multiobjectif »	41
3.2.1. Les méthodes agrégées	42
3.2.2. Méthodes non-agrégées et non Pareto	43
3.2.3. Les méthodes basées sur l'optimum de Pareto	44
a. Méthodes de première génération	44
b. Méthodes élitistes	45
4. Les chemins optimaux dans un graphe pondéré	51
4.1. Introduction	51
4.2. Présentation du problème	51
4.3. Le plus court chemin dans un graphe statique	52

4.4. Le plus court chemin dans un graphe dynamique	54
4.5. Le problème du plus court chemin stochastique	55
5. Conclusion :	57
Chapitre 3 : Les animaux marquants et le plus court chemin multiobjectif	
1. Introduction	60
2. Démarches de l'algorithme proposé	61
2.1. Principe	61
2.2. Représentation de l'algorithme	63
2.2.1. La reproduction des individus	63
2.2.2. L'élimination des individus	64
2.2.3. Le marquage	64
2.2.4. La concurrence entre des individus	66
2.2.5. Les types de recherche	67
2.2.6. La recherche d'une solution	68
2.2.7. Enrichir la solution	70
2.2.8. La condition d'arrêt de l'algorithme	72
2.3. L'algorithme proposé et le plus court chemin multiobjectif	72
2.3.1. Présentation	72
2.3.2. Définition	73
2.3.3. Application	74
3. Conclusion	76
Chapitre 4 : Implémentation et résultats	
1. Introduction	80
2. Présentation de l'algorithme	81
2.1. Principe et objectifs	81
2.2. Aspect physique et logiciel	81
2.2.1. Aspect physique	81
2.2.2. Aspect logiciels	81
a. Présentation du projet PgRouting	81
b. Présentation des données spatiales OpenStreetMap (OSM) et osm2PgRouting	83
c. Présentation de l'application cartographique	84
c.1. L'application cartographique « serveur »	84
c.2. L'application cartographique « cliente »	88
c.3. La programmation de la métaheuristique proposée	89
c.4. Installer l'algorithme dans le noyau de Postgresql	91
3. Résultats et interprétations	92
3.1. Présentation du lieu d'implémentation	92
3.2. Estimation de la base des données	94
3.3. Interprétations des résultats	94
4. Conclusion	97
Conclusion générale et perspectives	100
Bibliographie	104
Annexe	116

*Liste des figures et
des tableaux*

Liste des figures et des tableaux :**1. Liste des figures :**

Figure 1.1 : Le système de coordonnées géographiques.

Figure 1.2 : Le découpage UTM planétaire (il commence au parallèle 80 ° au Sud et se termine au parallèle 84 ° au Nord. Les zones polaires ne sont donc pas couvertes par la projection UTM).

Figure 1.3 : Un graphe simple.

Figure 1.4 : Un graphe orienté est pondéré.

Figure 1.5 : Mode Raster et Vecteur.

Figure 2.1 : Représentation de l'espace de recherche et de l'espace des objectifs.

Figure 2.2 : Un graphe statique déterministe.

Figure 2.3 : Un graphe dynamique déterministe.

Figure 2.4 : Un graphe statique stochastique.

Figure 3.1 : Les deux ondes progressent l'une vers l'autre.

Figure 3.2 : Les quatre sous-populations.

Figure 3.3 : Le marquage des rues par les individus.

Figure 3.4 : La recherche par alternance de 4 rubriques de marquage.

Figure 3.5 : La recherche par alternance de 2 rubriques de marquage (à gauche, les deux sous-populations qui forment un itinéraire : BPA — RPB partagent la même rubrique et RPA — BPB partagent l'autre, à droite, les deux sous-populations qui circulent dans le même sens : BPA — RPA partagent la même rubrique et RPB — BPB partagent l'autre).

Figure 3.6 : La rencontre de deux individus de deux populations concurrentes forme une solution.

Figure 3.7 : L'organigramme de composition d'un parcours des individus.

Figure 3.8 : L'itinéraire 1-5-8 est le chemin le plus court par rapport aux nombres d'arcs, et les parcours des individus bloqués (avec des têtes rouges dans la figure) donnent d'autres solutions réalisables.

Figure 3.9 : Le modèle logique des données.

Figure 4.1 : Les architectures utilisées dans le projet PgRouting.

Figure 4.2 : L'interface client « carte de sidi bel abbes ».

Figure 4.3 : La topologie de la ville de sidi bel abbes.

Figure 4.4 : Le chemin le plus court « multiobjectif » dans la ville de Sidi Bel Abbès.

Figure 4.5 : La ville de Sidi Bel Abbès.

Figure 4.6 : Les nœuds de la ville de Sidi Bel Abbès.

Figure 4.7 : Les segments de la ville de Sidi Bel Abbès.

Figure 4.8 : Nombre des solutions trouvées par les trois techniques.

Figure 4.9 : La rapidité entre les deux techniques.

Figure 4.10 : Les nombres de saturation du réseau dans la technique par alternance.

Figure 4.11 : Les taux d'échec de la méthode par alternance 02 rubriques de marquage.

Figure 4.12 : Influence du nombre des objectifs sur le temps d'exécution.

2. Liste des tableaux :

Tableau 3.1 : Les rencontres possibles entre les individus des quatre sous-populations.

Tableau 3.2 : La recherche de la racine dans la table généalogique.

Tableau 3.3 : Recensement des chemins trouvés.

Tableau 3.4 : La fusion entre les chemins trouvés forme plusieurs itinéraires réalisables.

Tableau 3.5 : On commence par le tri de la première objective, puis la rubrique $Sum_{S_{index_C}}$ calcule la somme des index.

Tableau 3.6 : Le tri de la deuxième objective, et puis la rubrique $Sum_{S_{index_C}}$ calcule la somme des index.

Tableau 3.7 : Le tri de la rubrique $Sum_{S_{index_C}}$ qui contient la somme des index.

Tableau 4.1 : La création de la topologie des trois villes d'expérimentation.

Tableau 4.2 : Le temps nécessaire pour créer un graphe avec les techniques de programmation.

Introduction générale

La ville devient ainsi un objet d'étude pour les experts en ingénierie et en traitement d'informations. Face à la complexité de l'environnement urbain, qui comporte des éléments imprévisibles et dynamiques, et pas toujours faciles à modéliser, plusieurs solutions méthodologiques ont été conçues dans le but de mieux comprendre les fonctions de la ville et d'analyser les comportements des personnes dans l'espace et le temps. L'étude des activités urbaines et de la mobilité dans la ville ont particulièrement eu un grand intérêt pour de nouvelles applications telles que le développement des systèmes de transport ou des systèmes de génération d'itinéraires. Ces systèmes ont pour objectif de proposer un usage de la ville adapté aux besoins de l'utilisateur, mais aussi d'acquérir des données représentant les divers usages de la ville [1].

Le transport urbain est un sujet très complexe à traiter et il possède différentes facettes. Une de ces facettes est la problématique de l'optimisation du transport urbain qui concerne le choix de l'itinéraire. En effet, pour un automobiliste ou un chauffeur d'ambulance, un choix judicieux de l'itinéraire peut représenter une économie financière et un gain de temps important et même il peut sauver des vies. C'est dans ce contexte que s'inscrit ce travail. Plusieurs chercheurs se sont penchés sur la problématique du calcul d'un itinéraire optimal entre deux points d'un réseau routier depuis plusieurs années. Néanmoins, ce sujet est toujours un thème de recherche d'actualité.

Ce travail de thèse s'articule autour de plusieurs objectifs émanant d'axes de recherche croisés entre l'étude spatiale et le transport dans un milieu urbain. La complexité de la ville se manifeste clairement par la structure multimodale du transport et par la diversité et le nombre important des impacts environnementaux, organisationnels pris en compte dans les calculs des itinéraires dans ce milieu spatial condensé.

L'objectif de notre recherche est de développer un système d'aide à la décision pour le choix d'un itinéraire dans un réseau routier urbain. Le coût et la durée du déplacement ne sont plus les seuls critères de choix d'un itinéraire. Dans ce cas, l'aspect environnemental n'est plus intégré comme une contrainte, mais comme un objectif à atteindre. Le système proposé doit guider le décideur dans le choix d'un itinéraire dans une ville (succession de nœuds et de chemins) avec le meilleur compromis possible sur le plan organisationnel, environnemental et économique.

Parmi les problèmes les plus étudiés dans l'optimisation combinatoire, le problème de plus court chemin reste toujours un sujet le plus fréquent, vu à son importance dans nos choix et nos décisions. Le gain du temps, de l'argent et de vie (pour les chauffeurs d'ambulance) est un grand souci pour un exploitant d'un réseau de transport. De nombreux problèmes de plus court chemin peuvent être modélisés sous la forme d'un problème combinatoire. Ce problème consiste à construire un chemin minimisant un objectif dans un graphe modélisant un réseau routier urbain. Au vu du nombre de critères à prendre en compte, nous modéliserons notre problème sous la forme d'un problème de plus court chemin « multiobjectif ».

Dans la suite, ce mémoire s'organise de la manière suivante :

Le chapitre 1 définit le milieu urbain, son analyse et sa modélisation dans le domaine géomatique, puis nous présenterons la numérisation du milieu urbain, la représentation de l'information géographique, le système d'information géographique et ses fonctionnalités.

Le chapitre 2 : nous représenterons le problème de plus court chemin avec les différents types de graphes, nous expliquerons comment ce problème classique deviendra un problème d'optimisation combinatoire qui exige des outils algorithmiques précis pour le résoudre, nous montrerons que les métaheuristiques sont des outils puissants pour résoudre le problème de plus court chemin « multiobjectif » .

Dans le chapitre 3 : nous nous intéresserons par la mise en œuvre d'une métaheuristique inspirée du comportement des vivants, où chaque animal et même l'être humain marque un territoire pour indiquer sa possession d'une zone de chasse, de vivre, d'attirer une femelle et de dissuader un autre male de pénétrer chez lui. Cette métaheuristique exploite ce comportement naturel pour résoudre des problèmes d'optimisation combinatoire sur le réseau routier urbain avec ses contraintes dites « Objectifs ».

Et enfin, le dernier chapitre est dédié à l'implémentation de la solution choisie, et les outils nécessaires pour la mise en application puis l'intégrée dans le noyau de l'extension PostGIS fonctionnant sous le SGBD (Système de gestion de bases de données) relationnel PostgreSQL et nous exploitons les algorithmes proposés par le projet PgRouting pour évaluer notre algorithme.

Chapitre 1 :

*Analyse, Modélisation et
Numérisation du réseau routier urbain*

1. Introduction :

L'environnement fait, actuellement débat et suscite les inquiétudes du grand public. Sa gestion est devenue une affaire de survie quotidienne. La croissance urbaine importante a souvent engendré des problèmes de maîtrise et de gestion technique de l'environnement urbain. Cependant, elle devient de plus en plus primordiale, faisant appel à des approches scientifiques pour apporter des solutions aux problèmes qui menacent notre environnement et par conséquent notre vie.

La relation entre l'homme et son environnement prend sa forme selon la valeur qu'on donne à l'espace, les différentes approches du design urbain le considèrent comme un acteur dans la vie humaine qui peut influencer les cours des choses comme il peut être influencé [2].

L'étude et la compréhension de l'espace urbain, ses différentes dimensions et les méthodes d'analyse, la connaissance des formes d'évolution de cet espace et les acteurs qui interviennent dans le changement spatial deviennent une étape prépondérante pour traiter cet espace [2].

Les déplacements quotidiens en véhicule représentent aujourd'hui un des principaux problèmes dans la gestion l'environnement surtout dans les milieux urbains. De nos jours, se déplacer est devenu un aspect essentiel de la vie quotidienne : qu'il s'agisse de transports en commun ou de véhicules personnels, le vaste réseau formé de ces moyens de locomotion est immensément complexe à gérer. Sa gestion recouvre l'ensemble des techniques humaines et automatisées permettant d'assurer la surveillance des transports, au mieux un gain de performance dans l'acheminement des différents flux.

Une bonne gestion de l'environnement passe par une exploitation poussée des systèmes d'information modernes. Aujourd'hui, il n'est plus nécessaire de présenter l'utilisation de l'informatique standard dans la gestion de base des données alphanumériques et dans la présentation graphique. Elle est devenue omniprésente et à la portée des utilisateurs concernés par l'environnement partout dans le monde, y compris dans les pays en développement. Il semble plus important d'examiner les nouveaux développements concernant les systèmes d'information et l'aide à la décision qui concernent les paramètres fondamentaux de l'environnement, comme les dimensions spatiales et plus généralement, géographiques [3].

L'objectif de ce chapitre est de définir l'information géographique et ses caractéristiques, de présenter les principales fonctionnalités des systèmes d'information géographiques (S.I.G) et son domaine d'application.

2. Terminologies et définitions:

- Urbain : Plusieurs définitions sont proposées, on choisit quelques une :
 - De la ville ; qui est relatif, qui appartient à la ville, aux villes [4].
 - La notion d'unité urbaine repose sur la continuité du bâti et le nombre d'habitants. On appelle unité urbaine une commune ou un ensemble de communes présentant une zone de bâti continu (pas de coupure de plus de 200 mètres entre deux constructions) qui compte au moins 2 000 habitants [5].

Il vient du latin « urbs, urbis », signifiant « la ville » et par extension « la population de la ville », les urbains. Il a donné lieu notamment à une discipline, l'urbanisme, étude du phénomène urbain et de son organisation [6].

- Agglomération : Du latin médiéval *agglomeratio*, qui signifie accumulation. Elle comprend un nombre plus ou moins grand de quartiers et de rues dont la population se livre à des tâches non agricoles [7], mais le code de la route français définit la notion d'agglomération comme un «espace sur lequel sont groupés des immeubles bâtis rapprochés et dont l'entrée et la sortie sont signalées par des panneaux placés à cet effet le long de la route qui le traverse ou qui le borde» (article R.110-2) [8].
- La rue et la route : La distinction entre ces deux mots tient compte de différence contextuelle, la route est «une voie de communication terrestre aménagée, plus importante que le chemin, située hors d'une agglomération ou reliant une agglomération à une autre» (opposé à rue), alors qu'une rue est une «voie bordée, au moins en partie, de maisons dans une agglomération» [9].
- Le réseau routier, viaire ou de transport : selon l'encyclopédie libre Wikipédia, le réseau routier est un ensemble des voies de circulation terrestres permettant le transport par véhicules routiers, et en particulier, les véhicules motorisés (automobiles, motos, autocars, poids lourds...)[10], mais en Québec, ce terme veut dire ensemble des routes aménagées et entretenues par l'administration publique pour l'utilisation au profit du public [11].

3. Étude de l'espace urbain :

3.1. Analyse spatiale :

Plusieurs définitions sont proposées dans la littérature, dans les dictionnaires de géographie, le terme « d'analyse spatiale » veut dire :

- «Ensemble de méthodes mathématiques et statistiques visant à préciser la nature, la qualité, la quantité attachées aux lieux et aux relations qu'ils entretiennent — l'ensemble constituant l'espace —, en étudiant simultanément attributs et localisations» [12] .
- Les auteurs De Smith, Goodchild et Longley [13] ont donné une autre définition plus claire : L'analyse spatiale désigne l'ensemble des méthodes de traitement et de mesure applicables aux données géoréférencées, et fournissant une information sur des propriétés géométriques et/ou géographiques variant en fonction de la position de ces données.

En résumé, on peut définir l'analyse spatiale comme étant « un ensemble de démarches qui visent à décrire et à restituer l'organisation des structures matérielles de l'espace et les manières dont il est occupé et/ou représenté »

Pour le géographe, l'expression « analyse spatiale » évoque toute un ensemble de méthodes mathématiques ou statistiques qui permettent de valider ou révéler un ordre estimé sur les données. Cette même expression est utilisée par les cartographes [9] pour désigner l'ensemble des mesures qui permettent de vérifier si les données à cartographier peuvent l'être telles quelles dans l'espace de la feuille ou si, localement trop denses, trop confuses, trop discrètes, elles nécessitent quelque adaptation ou transformation. L'activité générale que laisse entrevoir le premier cas, le questionnement des données, nous l'appellerons « analyse spatiale exploratoire » dans la suite de cette communication ; l'autre activité, l'acceptation du sens qu'imposent les données, nous l'appellerons « analyse spatiale constitutive » [14].

D'après ces définitions, il est question d'analyse spatiale des données, lorsque ces données sont d'une part localisées dans l'espace et d'autre part que cette organisation spatiale est considérée importante, explicitement, dans l'analyse ou l'interprétation des résultats.

Selon MORENCY [15], le concept d'analyse spatiale s'articule autour de trois composantes :

- La visualisation et la représentation des données spatiales;
- L'exploration et la synthétisation des données spatiales afin de révéler des configurations spatiales particulières (patrons, relations spatiales);

- La modélisation spatiale qui tente de représenter l'organisation spatiale sous forme des modèles et estimer ses paramètres.

Les auteurs Bailey, Gatrell [16] et Cressie [17] ont classifié les données spatiales dans trois classes fondamentales [15]:

- Les nuages de points (point patterns) : les nuages de points représentent un ensemble d'événements se produisant en différentes localisations. C'est l'organisation spatiale de ces événements qui est d'intérêt et l'étude de ces données porte sur l'identification de configurations spatiales particulières ;
- Les données géostatistiques (geostatistical data) : les données géostatistiques réfèrent généralement à un échantillon de points (x, y) , provenant d'une surface continue, pour lesquels un attribut (z) a été mesuré. La finalité de cet échantillonnage est souvent de modéliser les variations spatiales afin de prédire la valeur de l'attribut aux endroits non échantillonnée à l'aide d'un variogramme et de méthodes d'interpolation spatiale ;
- Les données agrégées (regional data) : ces données résultent de l'agrégation de données discrètes et sont représentées par des points ou des zones. Généralement, plusieurs attributs ont été mesurés pour ce découpage territorial et l'enjeu consiste à comprendre l'organisation spatiale de ces attributs afin d'identifier des tendances ainsi que les relations entre ceux-ci.

Selon la définition de Miller en 2003 [18], ces données spatiales ont trois types d'existence :

- Une existence spatiale : toute donnée à caractère spatial est géo-référencée selon un système de projection cartographique ;
- Une existence temporelle : les données utilisées proviennent de deux séries mixtes puisqu'elles permettent d'étudier à la fois les tendances spatiales et temporelles. Chaque donnée est liée à la période à laquelle elle réfère : l'année de réalisation de l'enquête, le jour et l'heure de déplacement ;
- Une existence sémantique : toute donnée révèle un état. L'existence sémantique est la valeur, l'intensité, l'attribut révélé par la donnée (le genre d'un individu), le nombre d'unités de logement d'un secteur de dénombrement [15].

Pour se localiser sur la terre, il est nécessaire d'utiliser un système géodésique duquel découlent les coordonnées géographiques figurant sur les cartes. Celles-ci peuvent être exprimées soit sous la

forme de longitude et latitude (coordonnées dites géographiques) (Fig : 1.1), soit en représentation cartographique plane (coordonnées dites en projection).

Il existe trois systèmes pour exprimer une coordonnée sur terre : les systèmes de coordonnées cartésiennes, géographiques et projetées. Le système de coordonnées cartésiennes est très peu utilisé, il consiste à exprimer une position en X, Y, Z par rapport au centre O de la terre. Intéressons-nous donc aux deux autres systèmes très répandus.

3.1.1. Le système des coordonnées géographiques : Ayant pour référence le centre O du système géodésique utilisé, les coordonnées sont déclinées sous forme de deux (2) angles : latitude et longitude. L'angle 0 de la latitude correspond à l'équateur, et l'angle 0 de la longitude au Méridien¹ de Greenwich².

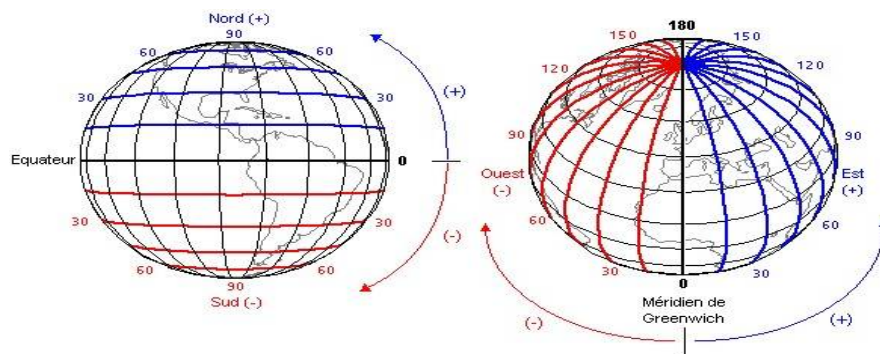


Figure 1.1 : Le système de coordonnées géographiques [21].

Plusieurs unités d'angle peuvent être utilisées, mais les 2 principales sont les degrés décimaux et les degrés sexagésimaux (degré, minute, seconde).

3.1.2. Le système des coordonnées projetées : Comme chacun le sait, la terre est ronde (ou presque...) et les cartes de papiers sont plates ! Pour représenter une surface courbe sur une surface plate il y a une formule mathématique qui, une fois appliquée permet d'exprimer une position dans un repère 2D (o, x, y) en unités de longueur (mètres...).

Ce système est très pratique à utiliser avec une carte, car il permet de calculer très facilement des distances. Le type de projection plane de la terre utilisé couramment est la projection

^[1] En géographie, un méridien est un demi-grand cercle imaginaire tracé sur le globe terrestre reliant les pôles géographiques. Tous les points de la terre situés sur un même méridien ont la même longitude. On parle également d'arc de méridien entre deux latitudes [19].

^[2] Un méridien où la longitude est définie comme égale à 0°. Il passe à travers l'Observatoire royal de Greenwich, à Greenwich (la banlieue de Londres), au Royaume-Uni. Avec le 180° méridien qui lui est directement opposé, il définit les hémisphères est et ouest [20].

cylindrique de Mercator³, d'où découle le système de coordonnées UTM⁴ (Universal Transverse Mercator). Celui-ci est notamment retenu pour le système GPS. Voici la grille mondiale correspondante (Fig : 1.2) :

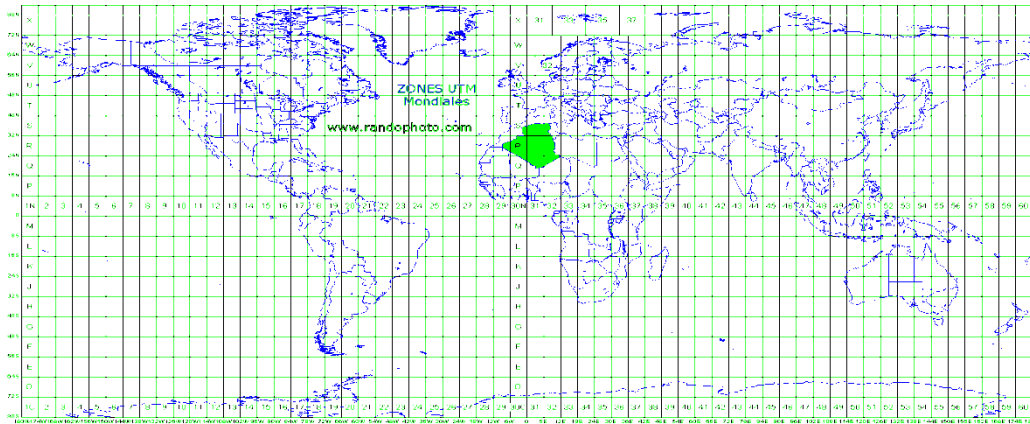


Figure 1.2 : Le découpage UTM planétaire (il commence au parallèle 80 ° au Sud et se termine au parallèle 84 ° au Nord. Les zones polaires ne sont donc pas couvertes par la projection UTM)[24].

L'Algérie est située sur 4 fuseaux UTM Nord (Fig. 1.2) : entre le fuseau 29 et le fuseau 32. La wilaya de Sidi Bel Abbas se trouve couverte par la zone UTM 30 S E : 653833.11 N : 3861336.08.

3.2. La modélisation de l'espace urbain :

3.2.1. Introduction :

Le mot modèle est défini d'abord dans le dictionnaire « le Petit Robert » comme étant « ce qui sert ou doit servir d'objet d'imitation pour faire ou reproduire quelque chose ».

L'auteur Varenne [25] a donné une autre définition plus claire « un modèle facilite la recherche d'information au sujet d'un système réel ou fictif, cela dans le cadre d'un processus à visée de représentation, de connaissance, de conceptualisation, de conception ou encore de transformation. »

^[3] La projection de Mercator est une projection cylindrique tangente à l'équateur du globe terrestre sur une carte plane formalisée par Mercator en 1569 [22].

^[4] UTM est un type de projection conforme de la surface de la Terre. Cette projection est une projection cylindrique où l'axe du cylindre croise perpendiculairement l'axe des pôles de l'ellipsoïde terrestre au centre de l'ellipsoïde. L'UTM est également un système de référence géospatiale permettant d'identifier n'importe quel point sur notre planète [23].

La modélisation urbaine, en tant que représentation simplifiée de la ville et de son fonctionnement, est finalement une notion relativement ancienne, même si sa théorisation et sa mise en pratique ne s'affirment que plus tardivement. Les modèles se distinguent selon leurs finalités et selon les disciplines au sein desquelles ils situent leurs réflexions [26].

Plusieurs techniques et concepts ont été exploités afin de modéliser la structure de la ville dont nous pouvons citer la théorie des graphes.

L'auteur Jguirim [1] a reparti ces méthodes en trois classes. Chaque classe se base sur un critère précis : la géométrie, la topologie et la fonction.

3.2.2. La modélisation urbaine basée sur la géométrie de l'espace : Modèle de Lynch

La théorie de l'image urbaine de Kevin Lynch [27] est certainement la plus importante dans le développement de cette approche, basée sur les cinq éléments constitutifs du paysage urbain : les voies, les limites, les nœuds, les points de repère et les quartiers :

- Voies : Les voies (paths) sont des éléments linéaires du paysage urbain permettant l'organisation du mouvement. Comme les rues, les trottoirs, les sentiers, les lignes de transport en commun, les voies ferrées. Enfin, ce sont les chemins où les personnes se déplacent ;
- Nœuds : Les nœuds (nodes) sont des éléments ponctuels dans la perception du paysage urbain. Ils présentent des transitions lors de déplacements dans la ville comme les carrefours et les stations de transport. Ils représentent les points stratégiques de la ville accessibles par les individus. Les nœuds offrent des fonctions dans l'espace permettant d'évaluer la ville ;
- Points de repère : Les points de repère («landmarks») sont d'autres éléments ponctuels du paysage urbain. Ce sont des points de référence dans la ville. Ils peuvent être repérés facilement dans l'espace. Leurs natures peuvent être très variées : un bâtiment remarquable, un élément végétal singulier, un monument, un équipement technique... etc. Comme le nom l'indique, ces éléments permettent à l'utilisateur de se situer (au moins de façon relative) et de s'orienter dans l'espace urbain ;
- Limites : Les limites (edges) se sont les éléments linéaires du paysage urbain, susceptibles d'être parcourus visuellement et qui constituent le bord d'éléments surfaciques, qui

représentent les frontières entre les quartiers. Les limites représentent les changements de la topographie dans l'espace par exemple les bords de la mer et les rues ;

- Quartiers : Les quartiers (districts) sont des éléments surfaciques de la ville, caractérisés par un certain degré d'homogénéité (perceptive, fonctionnelle, etc.) et permettant à l'utilisateur d'avoir la sensation d'entrer, de sortir ou d'être dans un espace. Plusieurs paramètres définissent le caractère d'un quartier tel que l'identité sociale, économique, historique ou fonctionnel (par exemple la zone résidentielle, la zone industrielle).

3.2.3. La modélisation urbaine basée sur la fonctionnalité de l'espace

Les méthodes conçues pour modéliser la ville en se basant sur les fonctions offertes par différents secteurs sont :

a. Le modèle cristallérien :

Le modèle cristallérien a été formulé par Christaller (1933) en Allemagne, est une théorie spatiale dite «la théorie des lieux centraux» cherchant à expliquer la hiérarchie des villes, selon leurs tailles, leurs localisations et leurs fonctions [28]. Selon cette théorie, la ville est une place centrale ou un nœud qui renferme les équipements, les services, la production matérielle qu'elle diffuse et le réseau de circulation qui les relie dans une aire plus ou moins étendue [29].

L'homogénéité de l'espace (homogénéité physique, sociale, économique, etc.) et la rationalité des comportements humains en matière de consommation et de pratiques spatiales sont les deux facteurs qui définissent la qualité de cette centralité ou de vie.

b. La théorie de la zone concentrique :

Le modèle concentrique a été conçu en 1923 par Burgess. L'idée derrière le modèle concentrique est que l'organisation urbaine y est exprimée par un processus basé sur l'ascension sociale et la migration des groupes aisés, du centre vers la périphérie, modifiant progressivement le statut des zones concentriques [30].

Les cercles présentent des quartiers qui regroupent des classes sociales différentes. Le noyau de ce modèle est le quartier central d'affaires. Autour du centre d'affaires, la ville se développe selon différents critères. La zone suivante est une zone transitoire entre le centre d'affaires de la ville et les zones résidentielles. Vient ensuite la zone caractérisée par une forte

présence de logements ouvriers (petits appartements). Les zones suivantes de la ville sont plutôt réservées aux classes aisées et supérieures [29].

L'organisation spatiale de la ville selon Burgess est donc consécutive à une évolution lente et historique et les découpages spatiaux correspondent en fait, aux différentes phases de cycle de vie [29].

Une des limites de ce modèle est son manque de généralité. En effet, conçu particulièrement pour la ville de Chicago, il est difficile à utiliser pour représenter la structure d'autres villes [1].

c. La théorie de secteurs :

Le schéma concentrique ne tient pas compte de la présence des axes de transport et néglige donc l'effet structurant de ces voies sur les zones avoisinantes qu'elles rendent plus attractives. Ainsi, Hoyt en 1939 complète le modèle précédent par la prise en compte des effets «structurants» des réseaux de transport sur les zones avoisinantes qu'elles rendent plus attractives [31].

L'évolution de la ville d'après Hoyt est due au développement urbain autour des réseaux de transport (les rues et les arrêts publics). Cette évolution prend une autre forme celle de secteurs qui se chevauchent avec le quartier d'affaires situé en partie centrale. Les zones résidentielles de la classe ouvrière se situent à proximité des zones industrielles de la ville alors que la classe aisée s'installe plus loin.

d. Modèle des noyaux multiples

Cette théorie a été proposée dans une phase de véritable transformation des villes américaines, avec une motorisation galopante et l'aménagement d'espaces spécialisés comme les parcs industriels ou les centres commerciaux.

Le schéma sectoriel montre que la ville dispose de multiples centres d'attraction, reliés entre eux par des axes de transport. C'est dans ce sens que Harris et Ullman (1945) proposent un modèle qui valide l'hypothèse qu'une ville ne possède pas un seul centre d'affaires comme indiqué dans le modèle de la zone concentrique et le modèle de secteurs. Le modèle des noyaux multiples repose sur le fait que la ville peut se développer autour de plusieurs centres avec une densité homogène. Le modèle prend en compte les agrégations et ses effets sur la configuration de l'espace [32]. La théorie des noyaux multiples a contribué à

la préfiguration de ce qui va se passer en Europe, et peut être ailleurs également avec un certain décalage, au dernier quart du 20^{ème} siècle...[33].

3.2.4. La modélisation urbaine basée sur la topologie de l'espace

Un espace urbain est représenté par un ensemble d'éléments tels que des lieux et des zones connectées par un réseau routier linéaire. Le réseau routier représente le substrat de développement de la ville. Comme tout objet dans l'espace, les éléments urbains sont caractérisés par des relations topologiques qui les relient. Les graphes sont parmi les concepts les plus utilisés pour modéliser un espace urbain.

La théorie des graphes est une théorie mathématique utilisée dans l'analyse des réseaux. Elle permet de modéliser les relations entre plusieurs entités dans l'espace. La théorie des graphes est très utilisée en géographie pour analyser et modéliser les réseaux routiers et de télécommunication.

En général, la représentation utilisée pour modéliser un réseau de transport est un graphe $G=(V, A)$ où l'ensemble des arcs A représente les chemins possibles du réseau et l'ensemble des nœuds V représente les intersections entre ces chemins; en d'autres termes, les nœuds représentent des intersections des rues dans une agglomération et les arcs sont les segments qui relient ces intersections entre elles; ces chemins peuvent être des routes, rues, boulevards, autoroutes,... etc.

Un graphe G peut être représenté graphiquement en symbolisant les sommets par des points et les arêtes par des lignes. Cette représentation offre une visualisation aisée de la structure générale du graphe (Fig : 1.3).

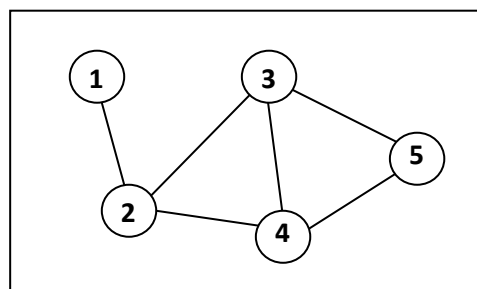


Figure 1.3: Un graphe simple.

Il est également courant de rencontrer des graphes pour lesquels on distingue l'extrémité initiale de l'extrémité terminale des arêtes, c'est-à-dire que l'on ne considère plus des paires, mais

des couples de sommets ; on parle alors d'arcs (arcs) et non plus d'arêtes. Un tel graphe est qualifié d'orienté (directed graph ou digraph) [34].

Enfin, on définit un graphe pondéré (weighted graphs), comme un graphe où chaque arête et/ou chaque sommet est affecté d'un nombre réel, appelé poids (Fig : 1.4).

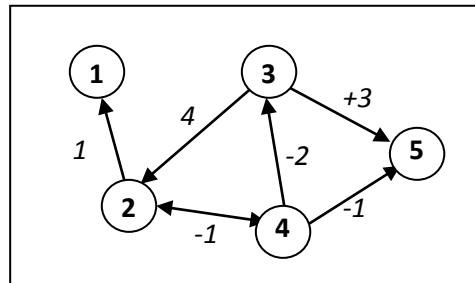


Figure 1.4 : Un graphe orienté est pondéré.

Le nombre de nœuds contenus dans un graphe G est appelé ordre de G . Par convention, il est noté n . De même, le nombre d'arcs (ou d'arêtes, selon l'orientation du graphe) contenus dans un graphe G est noté m . La taille d'un graphe G est $|G| = n + m$.

Un graphe possède un ensemble de propriétés dont :

- L'adjacence : s'il existe une connexion directe entre deux nœuds alors les deux nœuds sont adjacents ;
- La connectivité est le nombre de nœuds directement connecté à un nœud donné ;
- Dans un graphe orienté $G = (V, A)$, un chemin allant d'un nœud $i \in V$ à un nœud $j \in V$ est une séquence d'arcs consécutifs ayant les nœuds i et j pour extrémités : $((i, k), (k, l), \dots, (p, j))$;
- La distance topologique est la distance du plus court chemin qui sépare un nœud de départ et un nœud d'arrivée. La distance est calculée comme le nombre de nœuds intermédiaires entre les deux nœuds ;
- Un circuit est un chemin dont les sommets d'origine et de fin coïncident, i.e., une séquence d'arcs de la forme $((i, k), (k, l), \dots, (p, i))$. Un chemin est dit élémentaire s'il ne passe pas deux fois par le même sommet. Par défaut, tous les chemins considérés seront implicitement élémentaires ;
- Un graphe est connexe si et seulement s'il existe un chemin entre toute paire de nœuds. Dans un graphe, chaque nœud appartient à exactement une composante connexe.

- Un arbre est un graphe non orienté $G = (V, E)$ connexe et sans cycle. Il satisfait l'égalité suivante : $|E| = |V| - 1$. Une arborescence enracinée en r est un graphe de flot $G(r)$ connexe et sans cycle. Par conséquent, chaque nœud de $V \setminus \{r\}$ a exactement un prédécesseur et r n'a aucun prédécesseur. On a donc $|A| = |V| - 1$. Une anti-arborescence enracinée en r est un graphe orienté G tel que G^{-1} est une arborescence enracinée en r [35].
- Le degré $\text{deg}(v)$ d'un sommet v est le nombre de ses voisins, c'est-à-dire le nombre de sommets partageant une arête avec lui : $\text{deg}(v) = |\text{Voisins}(v)|$. Le degré moyen z d'un graphe est la moyenne de cette quantité sur l'ensemble de ses sommets [36].

4. La modélisation numérique de l'espace urbain :

4.1. Introduction :

Une caractéristique essentielle d'un SIG est sa capacité de modélisation pour spécifier des objets, ce qui peut signifier :

- Définir des objets géographiques de référence ;
- Définir des objets utilisateurs ;
- Et définir des liens entre les objets utilisateurs et les objets géographiques.

La modélisation spatiale définit de quelle manière où seront stockées et structurées les primitives géométriques, que l'on souhaite manipuler dans le système.

Les SIG se distinguent des autres systèmes d'information par plusieurs aspects, au niveau des données comme au niveau du système lui-même. Du point de vue des données, on peut ainsi relever les points suivants [37] :

- Une quantité importante de données (jusqu'à des téra-octets dans le cas de gros systèmes contenant des données Raster), qui nécessite des systèmes de gestion robuste ;
- La coexistence de nombreux formats ou modèles géographiques (dans des systèmes distincts, mais la cohabitation de modèles différents, intervient de plus en plus au sein d'un même système, notamment dans le cas de fédération de bases de données) ;
- La complexité des structures de données géographiques (par opposition aux types de données classiques des SGBD) ;
- La tendance de plus en plus importante à la réutilisabilité (échange de fichiers, interopérabilité) ;

- Les interactions entre les entités géographiques (relations spatiales, que de simples jointures ne peuvent représenter).

La complexité du monde réel oblige à le simplifier pour pouvoir le manipuler dans un processus d'abstraction [37].

4.2. La dualité Raster / Vecteur :

Il existe deux grands modèles de représentation des données géographiques sous forme géométrique « le modèle à structure objet ou vecteur » et « le modèle à structure image ou maillé » (en anglais « Vector » et « Raster ») (Fig : 1.5) :

- Le mode vecteur, où la géométrie de l'image cartographique est décrite par des points composant le contour de chaque objet, que celui-ci soit ponctuel, linéaire ou zonal. Ces points sont reliés par des segments de droite orientés ou par toute autre ligne définie mathématiquement ;
- Le maillé ou Raster, également appelé matriciel, où la surface de l'image cartographique est décrite selon une logique de balayage ligne par ligne, chaque ligne étant composée de points élémentaires jointifs ou pixels.

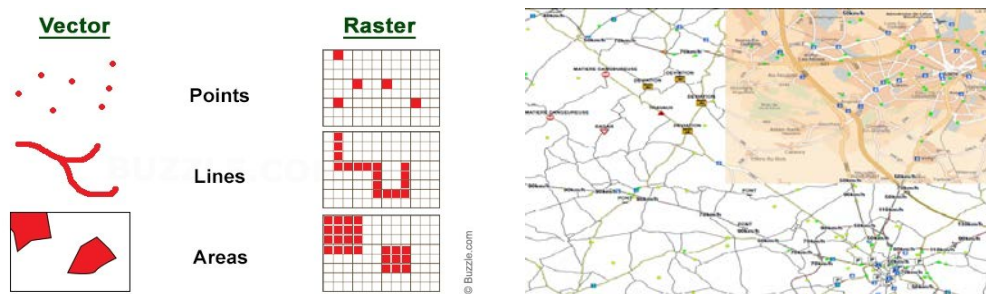


Figure 1.5 : Mode Raster et Vecteur. [38]

Ces deux modes présentent de grandes différences, en ce qui concerne leurs caractéristiques. Beaucoup de SIG disposent d'algorithmes de conversion d'un mode à l'autre, mais le mode Vector reste toujours privilégié [39], ces deux modes de représentation sont complémentaires. Une même information localisée à une position géographique pourrait être traduite dans les deux modes. Ses différentes représentations se superposent, laissant à l'utilisateur le soin de choisir en fonction de son usage la représentation qui lui convient le mieux.

4.3. La topologie :

On rappelle la définition de la topologie générale par Bouvier, George, Le Lionnais et Bros [40] : c'est la branche des mathématiques née de l'étude sur les surfaces et dans l'espace usuel des propriétés conservées par une déformation continue est liées à la notion de voisinage.

La topologie est étymologiquement grecque : «topos-logos», ce mot composé signifie littéralement «l'étude de lieu», la topologie comme discipline mathématique a connu ses débuts avec la théorie des graphes, en 1736 grâce aux travaux d'Euler, sur «les ponts de Königsberg» celui-ci a défini la topologie comme : «la part de la géométrie ne relevant pas des mesures et des distances». D'après Augustus Möbius, cette discipline mathématique s'intéresse à l'étude des propriétés de figures invariantes sous des transformations topologiques [41], [42].

La topologie de données vectorielles permet de traiter les problèmes de déplacements et de distances sur les réseaux par des calculs sur les matrices associées : recherche du plus court chemin au sein d'une succession d'arcs pondérés, de la localisation optimale d'équipements sous contraintes spécifiques, etc. [43].

L'intérêt de la topologie se porte également sur [44] :

- Une non-redondance des coordonnées géographiques, soit un gain de place ;
- Un accès plus rapide à l'information.

Il existe divers niveaux de topologie selon les logiciels :

- Pas de topologie dite (spaghetti) : duplication des points et des contours des polygones ;
- Modalité : les coordonnées des nœuds sont stockées une seule fois, mais en duplication des contours ;
- Topologie complète : les coordonnées des nœuds et des arcs sont stockées une seule fois, les relations entre arcs et polygones sont codées. On obtient un réseau topologique : arcs connectés entre eux par des nœuds, formant parfois des polygones [44].

Les auteurs Schneuwly et Calo [45] ont fait une petite comparaison entre les deux modèles, le premier modèle (non topologique), dans sa version la plus simple dans ce qui suit :

- Structure simple et ancienne ;
- Facilité de transfert des données (format DXF⁵) ;

⁵ Les fichiers AutoCAD DXF (Drawing Interchange Format ou Drawing Exchange Format) ont été proposés en 1982 par Autodesk (l'entreprise qui a fait AutoCAD) pour permettre l'interopérabilité des données entre AutoCAD et d'autres programmes [46].

- Suffisante pour cartographie numérique ;
- Analyse spatiale très réduite ;
- Doublement des limites de polygones adjacents (esthétique, manque de rigueur de gestion).

Il existe une version plus perfectionnée de cette structure (Modèle Réseau) ajoute la notion de polygones sous la forme d'un dictionnaire permettant de retirer des informations sur la taille des unités (surface et périmètre) et d'affecter des attributs graphiques tels qu'une trame ou une couleur.

Par contre, la structure topologique a été posée par Peucker et Chrisman en 1975. Ces auteurs ont proposé un nouveau format qu'ils ont dénommés POLYVRT [47], elle se caractérise par :

- Structure complexe ;
- Relations spatiales introduites lors de la saisie ;
- Analyse spatiale performantes ;
- Indispensable pour la gestion de réseau ;
- Difficulté de transfert de l'information topologique (incompatibilité des formats) ;
- Aucun dédoublement, ni d'information, ni de dessin, même pour les des couches différentes ;
- Certains SIG ne proposent pas des modèles topologiques explicites, mais offrent cependant des outils d'analyse de la connectivité, comme Geomedia.

Pour résumer, la topologie apporte aux données des informations qualitatives qui facilitent certains traitements et qui assurent une meilleure cohérence permettant une exploitation des bases de données géographiques. Son stockage permet d'optimiser certaines applications [48].

La création de la topologie peut se révéler une opération très fastidieuse et coûteuse dans le cas de données sources très « sales ». Une nouvelle digitalisation peut parfois s'avérer plus avantageuse. Les fonctionnalités et les performances de création de la topologie varient considérablement d'un système géomatique à l'autre :

- MapInfo : assistance au nettoyage et à l'agrégation des données (module externe) ; pas de structures de données topologiques ;
- MGE : « topology builder » a posteriori (base pour l'analyse spatiale) ;
- Vision, Argis, SpatialWare, ArcInfo : contraintes topologiques à la saisie et « topology builder » a posteriori.

Les auteurs De Oliveira, Pires et Medeiros [49] précisent que la fonctionnalité de modélisation reste très peu développée dans la plupart des SIG. Les modèles SIG sont souvent dédiés à la cartographie, qui n'est qu'une première exploitation possible de données géographiques. Un atout certain concernant la modélisation est de supporter la modélisation objet. TIGRIS d'Intergraph a été un précurseur, mais a été retiré du marché à cause de son manque de robustesse. Smallworld est également un des premiers SIG orienté — objet. C'est un SIG offrant d'excellentes capacités de modélisation, mais il a longtemps été handicapé par de longs temps de traitement. Lamps2 est un SIG orienté-objet plus récent, utilisé à l'IGN. Il offre des capacités de modélisation extrêmement évoluées permettant d'attacher des comportements à un objet modélisé par l'utilisateur. Enfin, le serveur Oracle peut être associé à une sur-couche objet [50].

4.4. L'information géographique :

4.4.1. Définitions :

Selon Bordin [51], elle peut porter plusieurs noms : information géographique, information localisée ou information à référence spatiale⁶. La composante spatiale est leur point commun.

De son nom indique, l'information géographique est une entité (naturelle ou artificielle, comme les infrastructures) distribuée sur le territoire [37]; Denègre et Salgé [53] proposent une définition plus claire « L'information géographique est l'information recueillie dans une base de données géographique qui comprend une information relative à un phénomène du monde représenté par son aspect, ses caractéristiques, et sa localisation décrite dans un système de coordonnées de référence » [54].

L'information géographique est très utile pour décrire, visualiser et analyser efficacement les phénomènes réels localisés. De nos jours, elle est devenue un élément important manipulé par de nombreuses applications nécessitant des données localisées. Plusieurs systèmes sollicitent aujourd'hui l'information géographique dans des domaines très variés, tels que l'aménagement du territoire, l'agriculture, l'environnement, la planification, la gestion des risques naturels, le géomarketing, etc. L'utilisation de données géographiques n'est plus limitée aux géographes, mais

⁶ Le terme de « système d'information à référence spatiale » (SIRS) a été proposé comme étant plus générique, il s'est notamment largement répandu au Canada [54].

s'est répandue avec l'informatique moderne aux utilisateurs du grand public non experts en géomatique⁷.

Une information géographique est définie par [57]:

- Une description géométrique (point, ligne, surface, grille, etc.) ;
- Une description attributaire (qualitative ou quantitative) ;
- Un ensemble de relations, spatiales ou non (appartenance, inclusion, voisinage, jointure...)

L'attribut de localisation des données géographiques peut être exprimé de manière directe (par des coordonnées) ou par des références indirectes (adresse postale, abscisses curvilignes le long d'un cheminement).

4.4.2. La représentation de l'information géographique

La localisation d'un objet géographique peut-être décrite de manière absolue, relative ou indirecte :

- La manière absolue définit l'objet géographique par ses coordonnées dans l'espace ;
- La manière relative définit la localisation de l'objet en exprimant ses relations spatiales (l'adjacence, intersection, etc.) avec d'autres objets ;
- Indirectement, l'objet peut-être localisé par des propriétés descriptives comme l'adresse.

Une information géographique renvoie à un objet localisé sur la surface terrestre et qui comporte plusieurs attributs. On le localise à l'aide d'un système de référence dans l'espace ; par la localisation relative, on peut comparer cet objet par rapport à d'autres objets. L'une des caractéristiques d'une information géographique est d'être tridimensionnelle (Fig. 1.6) selon la fonction suivante (Equ : 1.1) [58] :

$$Z=f(X, Y) \quad (1.1)$$

Avec Z= attribut et X, Y = paire de coordonnées par rapport à un système de référence.

L'objet est associé à une localisation représentée par une paire de coordonnées. Cette information géographique peut être aussi multidimensionnelle [58]. Dans ce cas, on associe plusieurs attributs à l'endroit (un lieu) suivant (Equ : 1.2) :

⁷ Le nom « géomatique », proposé en 1968 par le géomètre français Bernard Dubuisson [55], provient de la contraction de « géographie » et « informatique », qui désigne l'ensemble des utilisations techniques de l'informatique en géographie : les outils et méthodes d'observation et de représentation des données géographiques, ainsi que la transformation de ces mesures en informations utiles à la société [56].

$$Z_1, Z_2, \dots, Z_n = f(X, Y) \quad (1.2)$$

Avec Z_1, Z_2, \dots, Z_n = attributs et X, Y = paires de coordonnées par rapport à un système de référence.

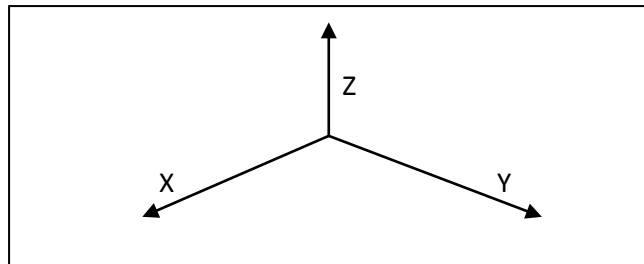


Figure 1.6 : Matrices géographiques.

4.4.3. Classification des informations géographiques :

Toute tentative de classification des informations nous conduit à une représentation dans un système quadripolaire comprenant :

- Les informations dites « topographiques », comme les cartes de base, les plans cadastraux... etc ;
- Les informations dites « thématiques », comme les plans de secteur, les plans d'aménagement, les cartes pédologiques, géologiques,... etc ;
- Les informations dites « modèle numérique », comme les modèles numériques de terrain (MNT) ;
- Les informations dites « images », comme les ortho-photos numériques, les données satellitaires fournies par « Landsat »⁸ ou « Spot »⁹ par exemple.

Dans un environnement informatique graphique, l'unité formelle de ces quatre pôles prend un format bipolaire dite la « Dualité Raster — Vecteur ».

- Le domaine vectoriel recouvre les informations « topographique », « thématique » et « modèles numériques » ;

^[8] Le programme Landsat est un des plus anciens programmes américains d'observation par satellite de la surface terrestre. Sept satellites Landsat ont été lancés depuis le 23 juillet 1972. Le dernier, Landsat 7, a été lancé le 15 avril 1999 [59].

^[9] SPOT (Système Probatoire d'Observation de la Terre ou Satellite pour l'Observation de la Terre) est une famille de satellites de télédétection Français civils d'observation de la terre. Cinq exemplaires ont été lancés entre 1985 et 2002. Deux nouveaux satellites devraient être lancés à compter de 2013 [60].

- Le domaine Raster (image numérique) recouvre les informations « topographiques », « thématiques » et « image ». Il comprend à la fois des données à traiter qualitativement (photos, pour certaines applications) et quantitativement (images classées, cartes thématiques).

5. Conclusion :

Un réseau est un ensemble de points géographiques reliés entre eux par des voies de transport. On fait ici référence au réseau routier (ensemble de lieux reliés entre eux par des voies routières).

Un S.I.G fait appel à une connaissance technique et à divers savoir-faire. Et pour concevoir un produit « S.I.G », on doit mobiliser des compétences en géodésie (connaissance des concepts de système de référence et de système de projection), en analyse des données, des processus et des différentes modélisations (la géométrie de l'espace, la fonctionnalité de l'espace, la topologie de l'espace), en traitement statistique ou graphique.

La modélisation des systèmes d'information géographique (S.I.G), est une tâche rigoureuse qui demande beaucoup de précision quant à l'organisation des données, leur nature et leur forme de représentation.

Dans ce chapitre, nous avons décrit les SIG, et leurs fonctionnalités de base. Celles-ci sont nombreuses, multidisciplinaires, liées et influencées les unes par les autres. Parmi elles, la modélisation qui est considérée comme le cœur et le centre de masse d'un SIG, ainsi que l'analyse spatiale qui représente la raison d'être du SIG, dans laquelle réside sa puissance. La divergence qui existe, entre le milieu extérieur qui alimente les SIG par les données et le milieu intérieur d'un SIG, qui intègre stocke et gère l'information géographique, est reflétée dans la modélisation par l'existence de deux modes différents de représentations des données spatiales, dont le mode Raster est dicté par un milieu externe, qui est la plus grande source d'acquisition des données, et le mode vectoriel le plus répandu et plus avantageux, car il représente l'information géographique de façon concise et appropriée aux besoins des SIG. D'un autre côté, l'importance de l'analyse spatiale qui représente une spécificité des SIG, réside dans sa capacité à satisfaire les différents besoins de l'utilisateur en termes de production de l'information géographique utile à partir des données acquises.

Dans le chapitre suivant, nous représenterons le problème de plus court chemin dans les graphes orientés et pondérés. Ces pondérations représentent des objectifs, la diversité de ces derniers rend la modélisation du réseau plus complexe et le nombre de données à traiter devient très important; d'où la nécessité d'utiliser des algorithmes capables de gérer ces données importantes, de les traiter dans un temps acceptable, et de résulter des itinéraires optimaux qui répondent aux exigences des usagers du réseau.

Nous expliquerons comment ce problème classique deviendra un problème d'optimisation combinatoire qui exige des outils algorithmiques précis pour le résoudre. Nous montrerons que les métaheuristiques sont des outils puissants pour résoudre le problème de plus court chemin « multiobjectif ».

Chapitre 2 :

Le chemin optimal multiobjectif

dans un réseau routier urbain

1. Introduction :

La théorie des graphes — et l’algorithmique qui lui est liée — est un des outils privilégiés de modélisation et de résolution de problèmes dans un grand nombre de domaines allant de la science fondamentale aux applications technologiques concrètes.

En général, la représentation utilisée pour modéliser un réseau routier, est un graphe $G = (V, A)$ où l’ensemble des arcs A représente les chemins ou les rues possibles du réseau et l’ensemble des nœuds V représente les intersections entre ces chemins; en d’autres termes, les nœuds représentent des intersections des rues dans une agglomération et les arcs sont les segments qui relient ces intersections entre elles; ces chemins peuvent être des routes, rues, boulevards, autoroutes... etc.

La particularité des réseaux de transport et la diversité des objectifs de ses usagers rendent la modélisation du réseau plus complexe et le nombre de données à traiter devient très important; d’où la nécessité d’utiliser des algorithmes capables de gérer ces données importantes, de les traiter dans un temps acceptable, et de résulter des itinéraires optimaux qui répondent aux exigences des usagers du réseau [61].

Dans ce chapitre, nous débuterons avec la représentation du problème de plus court chemin dans de différents types de graphes. Nous expliquerons comment ce problème classique deviendra un problème d’optimisation combinatoire qui exige des outils algorithmiques précis pour le résoudre. Nous montrerons que les métaheuristiques sont des outils puissants pour résoudre le problème de plus court chemin « multiobjectif ».

2. Terminologies et définitions :

- « L'optimisation est une branche des mathématiques, cherchant à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à déterminer le meilleur élément d'un ensemble, au sens d'un critère quantitatif donné. Ce mot vient du latin optimum qui signifie le meilleur » [62].
- « L'optimisation consiste à construire des algorithmes pour calculer le minimum d'une fonction sur un ensemble » [63].
- Une fonction objective est définie comme une fonction qui modélise l'objectif à atteindre dans le problème d'optimisation sur l'ensemble des critères dits « contraintes ». Il s'agit de la fonction qui doit être optimisée. Elle est notée $f(x)$. De manière générale, $f(x)$ est un vecteur $f(x) = (f_1(x); f_2(x); \dots; f_n(x))$.
- Le but d'un problème d'optimisation est de trouver une solution maximisant (resp. minimisant) une fonction « objectif » donnée. À chaque problème d'optimisation, on peut l'associer à un problème de décision dont le but est de déterminer s'il existe une solution pour laquelle la fonction « objectif » soit supérieure (resp. inférieure) ou égale à une valeur donnée ;
- Un problème d'optimisation combinatoire est un problème de maximisation ou de minimisation spécifié par l'ensemble de ses instances [64].
- Un problème d'optimisation combinatoire est un problème qui peut se mettre sous la forme :

$\underset{x \in X}{\text{Max}} f(x)$ où x est un ensemble fini et où f est une fonction à valeur entière positive ou nulle [65].

3. Problème d'optimisation « multiobjectif » :

3.1. Définition :

L'optimisation permet de minimiser ou maximiser des fonctions des systèmes dans lesquels peut intervenir un grand nombre de paramètres. On peut déterminer aisément ces objectifs grâce à l'application des outils mathématiques. Ces outils nous permettent de trouver les extremums des fonctions mono ou plusieurs variables, incluant les minimums ou les maximums. Les méthodes que nous allons étudier sont considérées comme des principes généraux pour

comprendre les problèmes de l'optimisation. Les principes sont les mêmes, mais l'objectif principal converge vers la recherche d'une solution adéquate [66].

Un problème d'optimisation est souvent présenté par une ou plusieurs fonction(s) ou objectif(s) (un ensemble de contraintes). Il consiste à trouver une solution qui satisfait les contraintes et qui minimise (ou maximise) la fonction « objectif » [67].

- Les variables sont les inconnues du problème. La solution recherchée est présentée par des valeurs exactes de ces variables exprimant des données quantitatives ou qualitatives. Elle s'appelle solution optimale. Ces inconnus peuvent être de nature diverse (réelle, entière, booléenne, etc.) et l'ensemble des domaines de définition des variables du problème définit un espace restreint appelé l'espace d'état. Cette limitation n'est pas problématique, car lorsqu'un problème est posé, le décideur précise un domaine de valeurs envisageable à chacune des variables ;
- La fonction « objectif » représente l'intérêt du décideur (minimisation de coût, de durée, d'erreur, etc.). Elle définit avec les limitations des contraintes un espace de solutions potentielles au problème ;
- Les contraintes représentent des conditions à satisfaire lors de la recherche d'une solution optimale. Ces conditions sont souvent données sous forme des équations d'inégalité ou d'égalité et permettent en général de limiter l'espace de recherche [67].

La résolution optimale du problème consiste à trouver le point ou un ensemble de points de l'espace de recherche qui satisfait au mieux la fonction objectif. Le résultat est appelé valeur optimale ou optimum. Néanmoins en raison de la taille des problèmes réels, la résolution optimale s'est souvent montrée impossible dans un temps raisonnable. Cette impossibilité technique impose la résolution approchée du problème, qui consiste à trouver une solution de bonne qualité (la plus proche possible de l'optimum). Il est vital pour déterminer si une solution est meilleure qu'une autre, que le problème introduise un critère de comparaison (une relation d'ordre).

Les problèmes d'optimisations sont souvent considérés comme des problèmes NP-difficile où il n'existe pas un algorithme optimal qui donne des solutions en temps polynomial en fonction de la taille du problème et le nombre d'objectifs à optimiser.

D'une façon générale, un problème d'optimisation « multiobjectif » peut être défini par la minimisation ou la maximisation d'une solution $x = (x_1, \dots, x_k)$ selon n fonctions objectifs. Chaque fonction objectif $f_i(x)$ (avec $i \in [1, n]$ et $n \geq 2$, le nombre d'objectifs) représente un critère du problème qu'il s'agit d'optimiser. $f(x) = (f_1(x), \dots, f_n(x))$ est alors le vecteur des fonctions objectifs et les f_i sont les objectifs à optimiser, appelées aussi critères de décision. Un problème peut avoir un ensemble de contraintes à satisfaire. Nous appelons l'espace de recherche l'ensemble des valeurs qui peuvent être prises par les variables et l'espace réalisable, le sous-espace des valeurs des variables satisfaisant les contraintes. L'image de l'espace de recherche par la fonction f est appelée l'espace des objectifs ou l'espace des critères [68] comme le montre la Figure suivante (Fig : 2.1) :

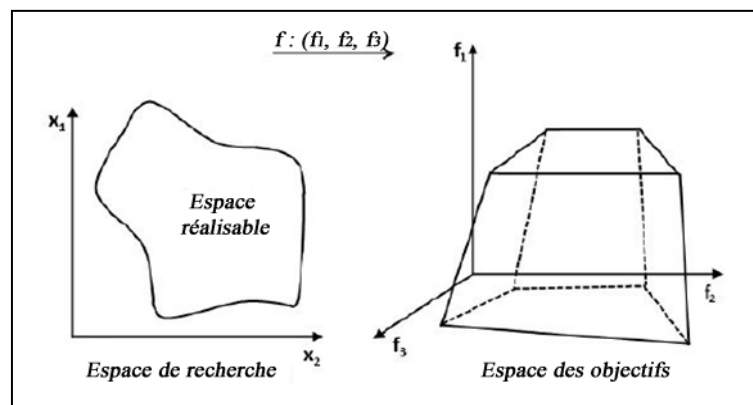


Figure 2.1. Représentation de l'espace de recherche et de l'espace des objectifs [69].

3.2. Méthodes d'optimisation « multiobjectifs » :

Dans la littérature, nous trouvons deux classifications différentes des méthodes de résolution des problèmes « multiobjectifs » (PMO). La première classe les méthodes suivant leurs façons de traiter les fonctions « objectifs ». Par contre, la deuxième classification adopte un point de vue décideur :

Les approches de la première classe, qui sont utilisées pour la résolution des problèmes d'optimisation « multiobjectifs » peuvent être classées en trois catégories [70] :

- *Méthodes agrégées* : Elle consiste à ramener le problème « multiobjectif » au problème de l'optimisation d'une combinaison linéaire des objectifs initiaux. Cette classe d'approches comprend par exemple les méthodes basées sur l'agrégation qui combine entre les différentes fonctions « objectif » f_i en une seule fonction « objectif » f . Ces approches

nécessitent d'avoir une bonne connaissance du problème. Cette méthode de résolution de PMO est la plus évidente et, probablement, la plus largement utilisée en pratique [70];

- *Approches non Pareto* : Elles ne transforment pas les problèmes d'optimisation « multiobjectifs » en un problème « uniobjectif », mais elles utilisent des opérations de recherche qui traitent séparément les différents objectifs [70];
- *Approches Pareto* : Elles utilisent directement « la notion d'optimalité Pareto » dans leur processus de recherche. Le processus de sélection des solutions générées est basé sur la notion de non-dominance [70].

3.2.1. Les méthodes agrégées

Plusieurs variantes sont proposées dans la littérature, on cite les suivants :

- **Somme pondérée** : Cette méthode consiste à additionner tous les objectifs en leur affectant un coefficient de poids importance relative que le décideur attribue à l'objectif. Les auteurs Jin, Olhofer et Sendhoff [71] ont proposé une agrégation dynamique des poids par contre les auteurs Kim et De Weck [72] ont présenté une méthode de somme pondérée adaptative [73] ;
- **Goal programming** : est connue aussi sous le nom de "target vector optimisation". Elle cherche à minimiser la somme des écarts entre les résultats des objectifs initiaux (ou individuels) et des valeurs qui ont été fixées comme buts idéaux à atteindre pour chaque objectif. La nouvelle fonction objectif est modifiée de façon à minimiser la somme des écarts entre les résultats et les buts à atteindre [74];
- **Le min-max** : selon Coello [75], Il minimise le maximum de l'écart relatif entre un objectif et son but associé par le décideur [73] ;
- **Atteindre un point cible** : atteindre un point cible dans cette approche [76] [77] [78], le décideur spécifie l'ensemble des buts T_i qu'il souhaite atteindre et les poids associés w_i . La solution optimale est trouvée en résolvant le problème suivant (Equ : 2.1) :

$$\text{Minimiser } a \text{ tel que } T_i + a.w_i \geq f_i(x) \quad (2.1)$$

Les objectifs T_i représentent le point de départ de la recherche dans l'espace et les poids w_i indiquent la direction de recherche dans l'espace [73]; L'algorithme tente de minimiser l'écart entre la solution trouvée et ses buts déclarés. Cette approche utilise un point de référence et une direction de recherche pour guider le processus de résolution [79].

- **Méthode ϵ -contrainte** : consiste à transformer les (n-1) objectifs en contraintes. L'objectif restant, qui peut être choisi arbitrairement, est la fonction d'objectif du problème d'optimisation mono-objectif qui en résulte [80], elle basée sur la minimisation d'un « objectif » f_i en considérant que les autres objectifs f_j avec $j \neq i$ doivent être inférieurs à une valeur ϵ_j . En général, l'objectif choisi est celui que le décideur souhaite optimiser en priorité (Equ : 2.2) :

$$\text{Minimiser } f_i(x) \text{ avec } f_j(x) \leq \epsilon_j, \quad \forall j \neq i \quad (2.2)$$

De cette manière, un problème simple objectif sous contraintes peut être résolu. Le décideur peut ensuite réitérer ce processus sur un objectif différent jusqu'à ce qu'il trouve une solution satisfaisante [73]. Grâce à cette méthode, on peut résoudre des problèmes convexes et non convexes. L'inconvénient est que l'on doit connaître à priori les intervalles pour les valeurs de ϵ_i pour toutes les fonctions « objectif » [80].

Selon Pinville [73], Il a recensé plusieurs inconvénients quant à l'utilisation de méthodes agrégées :

- La difficulté de déterminer les différents paramètres ;
- Elles nécessitent de nombreuses connaissances a priori, notamment lors de l'affectation d'un coefficient de poids à chaque objectif ;
- Il est de plus nécessaire d'effectuer de nombreux tests pour déterminer l'influence de chaque objectif.

3.2.2. Méthodes non agrégées et non Pareto :

- **Vector Evaluated Genetic Algorithm (VEGA)** : en 1985, Schaffer propose la première extension d'un algorithme génétique simple pour la résolution d'un problème « multiobjectif » [81]. La différence avec un algorithme génétique simple se situe au niveau de la sélection. Pour k objectifs et une population de n individus, une sélection de n/k individus est effectuée pour chaque objectif. k sous-populations seront créées contenant chacune les n/k meilleurs individus pour un objectif particulier. Les k sous-populations sont ensuite mélangées afin d'obtenir une nouvelle population de taille n [82] ;
- **La méthode lexicographique** : Fourman [83] a proposé une méthode dans laquelle les objectifs sont classés par ordre de préférence. Plusieurs optimisations seront mises en place

en suivant le niveau d'importance du critère. Comme cette approche se concentre trop sur des solutions extrêmes sans aucun compromis, elle très rarement appliquée. Mais, même avec cet inconvénient, elle est appliquée dans le problème du plus court chemin. Par exemple, dans le cas où le critère est de faire le moins de changement, cette approche implique la solution faisable même si la distance de parcours est trop importante. [84].

3.2.3. Les méthodes basées sur l'optimum de Pareto :

En 1897, Pareto [85] a formulé le concept suivant : dans un problème « multiobjectif », il existe un équilibre tel que l'on ne peut pas améliorer un critère sans détériorer au moins un des autres critères. Cet équilibre a été défini comme l'optimum de Pareto.

La première utilisation de la dominance au sens de Pareto remonte à Goldberg en 1989 [86] pour résoudre des problèmes proposés par Schaffer [81].

Définition 1 : La convergence permet de mesurer à quelle distance les solutions trouvées sont du vrai front de Pareto.

Définition 2 : La distribution (*Dispersion*) indique comment les solutions sont bien réparties sur le vrai front de Pareto (ou son approximation).

Résoudre un problème « multiobjectif » consiste à converger le plus rapidement possible vers le front de Pareto (convergence), tout en gardant une bonne distribution des solutions sur le front de Pareto (dispersion).

a. Méthodes de première génération : Du fait de la non-conservation des individus Pareto-optimaux de génération en génération dans ces méthodes dites non élitistes, la convergence vers le front de Pareto est lente. De plus, elles maintiennent difficilement de la diversité sur ce front de Pareto. Pour contrecarrer ces limitations, des méthodes dites « élitistes » se sont développées.

- **Multiple Objective Genetic Algorithm (MOGA)** : le premier algorithme qui utilise directement la notion de dominance de Pareto a été proposée par Fonseca et Fleming [87], il utilise une procédure de ranking dans laquelle, pour chaque solution i , le nombre n_i de solutions du dominant est calculé, et le rang $r_i = (1 + n_i)$ lui est associé. Ainsi, le rang de chaque solution non dominée est égal à 1 et le rang maximal ne peut pas être plus grand que la taille de la population N . La performance (fitness) F_i de chaque individu est basée sur son rang. La

performance est calculée de sorte que tous les individus du même rang aient la même performance et que cette performance soit à maximiser [88] ;

- **Non dominated Sorting Genetic Algorithm (NSGA)** : publié en 1994 par Srinivas et Deb [89], il utilise la première procédure de ranking qui a été initialement introduite par Goldberg [90]. Tous les individus non dominés de la population possèdent le rang 1. Ces individus sont ensuite enlevés de la population et l'ensemble suivant d'individus non dominés est identifié et on leur attribue le rang 2. Ce processus est réitéré jusqu'à ce que tous les individus de la population aient un rang [88]. La NSGA a connu un succès pendant plusieurs années [91], bien que plusieurs études comparatives de l'époque [92] a indiqué qu'il a été devancé par MOGA et NPGA. NSGA est également un algorithme inefficace en raison de la façon dont il classe les individus. Les auteurs de [93] [94] ont proposé une nouvelle version élitiste de cet algorithme, nommée Non Sorting Genetic Algorithm II (NSGA-II) [70];
- **Niched Pareto Genetic Algorithm (NPGA)** : Horn, Nafpliotis, et Goldberg [95] ont proposé en 1994 la méthode NPGA utilise un tournoi basé sur la notion de dominance de Pareto. La comparaison se fait sur deux individus pris au hasard avec une sous-population de petite taille également choisie au hasard. Si un seul de ces deux individus domine la sous- population, il est alors positionné dans la population suivante. Dans les autres cas, une fonction de sharing est appliquée pour sélectionner l'individu [73].

b. Méthodes élitistes

- **Strength Pareto Evolutionary Algorithm (SPEA II)** : est proposé par Zitzler, Laumanns et Thiele [96], ont été développé comme une amélioration de SPEA [97]. Cette méthode utilise le concept de Pareto pour comparer les solutions. Une des caractéristiques de SPEA II est de maintenir un ensemble de solutions Pareto-optimales dans une archive. La fitness de chaque individu est calculée en fonction des solutions stockées dans celle-ci. De plus, toutes les solutions de l'archive participent à la sélection. Afin de maintenir une taille d'archive raisonnable, une méthode de clustering est utilisée visant à garder seulement les solutions les plus représentatives. SPEA2 diffère de son prédécesseur en plusieurs aspects (la taille de l'archive est fixe, ..

etc.), [88] ; Une nouvelle méthode de niche, basée sur Pareto, est utilisée afin de préserver la diversité. L'avantage essentiel est qu'elle n'exige pas de réglage de paramètres de sharing [73].

- **Pareto Archived Evolution Strategy (PAES)** : La méthode PAES, conçue et mise en oeuvre par D. Knowles et Joshua David W. Corne [98] est une méthode du simple objective fournissait des résultats supérieurs aux méthodes de recherche basées sur une population. Son principe n'est pas basé sur une population car elle n'utilise qu'un seul individu à la fois pour la recherche des solutions. Elle utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto-optimales. Les particularités de cette méthode sont les suivantes [99] :
 - Elle n'est pas basée sur une population. Elle n'utilise qu'un seul individu à la fois pour la recherche des solutions ;
 - Elle utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto-optimales ;
 - L'algorithme utilisé est très simple et inspiré d'une stratégie d'évolution ($i+1$) ;
 - Elle utilise une technique de « Crowding » basé sur un découpage en hypercubes de l'espace des objectifs.
- **Pareto Envelope based Selection Algorithm (PESA)** : Horn, Nafpliotis et Goldberg [100] utilisent une petite population interne et une population externe plus large. le PESA reprend approximativement le principe de crowding développé dans PAES et définit un paramètre appelé squeezeFactor qui représente la mesure d'encombrement d'une zone de l'espace. Alors que PAES est basé sur une stratégie d'évolution, PESA est une méthode basée sur les algorithmes génétiques qui définit deux paramètres concernant la taille des populations d'individus: POPI (taille de la population interne) et POPE (taille de la population externe ou archive) [99]. Une division en grille de l'espace des phénotypes est utilisée pour maintenir une diversité (application d'une mesure de densité) durant le processus. De plus, cette mesure de densité est utilisée pour permettre à des solutions d'être retenues dans une archive externe de la même manière que dans PAES [100]. La différence entre PESA et PESA-II [101], se situe au niveau de la sélection. La sélection se fait d'abord sur une zone. Un individu est

ensuite sélectionné dans cette zone. Le but de cette approche étant de réduire le coût computationnel associé au classement de Pareto [73] ;

- **Non-dominated Sorting Genetic Algorithm II (NSGA-II)** : a été présentée dans [94]. NSGA-II intègre un opérateur de sélection, basé sur un calcul de la distance de « Crowding » (ou « surpeuplement ») qui estime la densité de chaque individu dans la population. Comparativement à NSGA, NSGA-II obtient de meilleurs résultats sur toutes les instances présentées dans les travaux de Deb, Pratap, Agarwal et Meyarivan [94], ce qui fait de cet algorithme un des plus utilisés aujourd'hui. Pour gérer l'élitisme, NSGA-II n'utilise pas d'archive externe pour stocker l'élite, comme le font d'habitude les autres AGs. NSGA-II assure qu'à chaque nouvelle génération, les meilleurs individus rencontrés soient conservés [88] ;
- **ϵ - Multiobjective Evolutionary Algorithm, (ϵ -MOEA)** : L' ϵ -MOEA [102] [103] repose sur une version modifiée de la relation de dominance : [102] [104] [103]. Soit ϵ un vecteur quelconque de n réels strictement positifs, n étant le nombre d'objectifs optimisés, on dit que la solution x_1 ϵ -domine la solution x_2 si (Equ : 2.3) :

$$\begin{aligned} \forall i \in [1, n], f_i(x_1) + \epsilon_i &\geq f_i(x_2) \\ \exists i \in [1, n], f_i(x_1) + \epsilon_i &> f_i(x_2) \end{aligned} \quad (2.3)$$

L'algorithme ϵ -MOEA [102] [103] utilise cette relation d' ϵ -dominance en plus de la relation de dominance classique. Il repose sur deux ensembles d'individus, une population et une archive contenant les solutions non dominées trouvées. À chaque étape, deux parents sont sélectionnés, un dans la population et un dans l'archive, et sont utilisés pour générer deux enfants. Ces nouveaux individus sont ensuite intégrés à la population selon la relation classique de dominance et ajoutés à l'archive selon la relation d' ϵ -dominance. L'idée principale est d'assurer une bonne dispersion des solutions non dominées dans l'espace des objectifs, en ne gardant que quelques solutions représentatives le long du front dans l'archive. L'algorithme ϵ -MOEA permet ainsi d'obtenir de bonnes performances en termes de convergence et de dispersion des solutions tout en ayant un coût computationnel moins élevé que SPEA et NSGA-II. Néanmoins, le choix des valeurs ϵ_i a une influence évidente sur l'approximation finale du front de Pareto. D'autre part, la densité de solutions peut

- également être variable le long du front et l'utilisation d'une constante par objectif risque d'entraîner un sous-échantillonnage certaines zones par rapport à d'autres [73] ;
- **Algorithmes utilisant l'indicateur d'hypervolume** : L'hypervolume [97] [105] est un indicateur permettant de mesurer et de comparer la qualité des solutions finales dans des algorithmes à base de population. Cet indicateur représente le volume d'espace à N dimensions dominé par une ou plusieurs solutions à un problème à N objectifs. Il est calculé relativement à un point arbitraire facilement accessible et dominé. L'hypervolume couvert entre ce point et les points positionnés sur le front de Pareto de l'expérience représente l'hypervolume de ces solutions. L'ajout d'une solution non dominée au front ajoute automatiquement de l'espace à l'hypervolume de ce front. L'indicateur d'hypervolume a été originellement proposé et employé comme mesure de performance pour comparer différents MOEA [97]. Il a par la suite été intégré dans le processus d'optimisation [105]. Dans ce cas, il décrivait une stratégie pour maintenir une archive séparée et bornée de solutions dominées basées sur cet indicateur [73].

D'après Wagner, Beume et Naujoks [106], NSGA-II et SPEA2 ont des difficultés quand le nombre d'objectifs augmente ($n > 3$), alors que ϵ -MOEA gère bien l'optimisation massivement « multiobjectif » [106]. L'utilisation d'un indicateur d'hypervolume [97] [105] semble prometteuse mais sa complexité d'exécution reste trop importante pour utiliser efficacement dans le cadre d'optimisation de contrôleur, qui nécessitent un grand nombre d'évaluations. D'autre part, l'utilisation d'une archive dans un algorithme évolutionniste, comme dans ϵ -MOEA, SPEA, SPEA2, PESA ou PESA- II, exclu la possibilité de traiter des objectifs dynamiques, c'est-à-dire dont la valeur change au cours du temps ou en fonction des individus présents dans la population. Un objectif devient dynamique notamment dans le cas où l'on désire maintenir une certaine diversité dans la population d'individus i a un objectif de diversité spécifique [73].

La deuxième classification des méthodes de résolution des PMO qui adopte un point de vue décideur sont aussi regroupées généralement en trois familles en fonction du moment d'intervention du décideur [107] [108] [109] :

- **Méthodes d'optimisation a priori** : Le décideur définit ses préférences et le compromis qu'il souhaite réaliser entre les différents critères du problème avant de

lancer l'optimisation. Cette famille regroupe la plupart des méthodes par agrégation qui consistent à fusionner toutes les fonctions « objectif » en une seule. L'avantage de ces méthodes est l'obtention rapide de la solution recherchée en une exécution. Cependant, elles peuvent avoir des inconvénients liés au temps de modélisation des préférences ou du compromis et la nécessité de tout relancer si la solution trouvée ne répond pas aux attentes du décideur [68] ;

- **Méthodes d'optimisation progressives :** Au fur et à mesure du déroulement de l'optimisation, le décideur fait ses choix de compromis à l'aide de méthodes interactives. Ces méthodes exigent alors la présence du décideur tout au long le processus d'optimisation et devra répondre à toutes les questions permettant d'orienter la recherche des solutions [68] ;
- **Méthodes d'optimisation a posteriori :** Ces méthodes fournissent une surface de compromis à la fin de l'optimisation. Le décideur peut alors sélectionner une solution de compromis parmi la liste complète des solutions générées par la méthode d'optimisation mise en œuvre. Dans cette famille, nous pouvons trouver également des méthodes d'optimisation a priori avec des préférences d'entrée choisies arbitrairement. A la fin de l'optimisation, le décideur aura un panel de solutions et il lui revient de décider de la solution de compromis [68].

D'autres auteurs comme Collette et Siarry [110] classifient ces méthodes en cinq groupes, en se basant sur les outils utilisés pour résoudre le problème « multiobjectif » :

- Les méthodes scalaires ;
- Les méthodes interactives ;
- Les méthodes floues ;
- Les méthodes qui exploitent une métaheuristique ;
- Les méthodes d'aide à la décision.

Dans notre travail, nous nous intéressons aux méthodes exploitant une métaheuristique pour résoudre un problème « multiobjectif », car ces méthodes génèrent un ensemble de solutions grâce à l'établissement d'une relation d'ordre entre les différents éléments. Les métaheuristicques ont initialement été proposées dans la recherche dédiée aux problèmes d'optimisation difficile. Le principal avantage de ces méthodes, constituant aussi leur principale source d'efficacité, se trouve

être leur capacité à éviter les pièges des minima locaux, et ce, contrairement aux méthodes d'optimisation classiques. En complément de la programmation mathématique, les heuristiques sont des méthodes de résolution purement algorithmiques qui permettent d'obtenir des solutions à n'importe quel problème décisionnel rapidement.

L'utilisation des méthodes exactes étant limitée à des problèmes des petites tailles, il convient, notamment pour pouvoir traiter des problèmes réels, de développer des méthodes heuristiques. Les métaheuristiques et en particulier les algorithmes évolutionnaires, travaillant sur des populations de solutions, sont bien adaptés à l'optimisation « multiobjectif ».

Le mot « métaheuristique », présenté la première fois par Glover en 1986 [111], dérive de la composition de deux mots grecs : Heuristique, dérive du verbe « heuriskein » qui signifie « trouver », et le suffixe « méta » qui signifie « au-delà » ou « dans un niveau supérieur ». Avant que ce terme ne soit adopté, les métaheuristiques étaient souvent appelées heuristiques modernes. Plusieurs définitions ont été proposées pour expliquer clairement ce qu'est une métaheuristique [112] [113] [114]. Aucune de ces définitions n'est universellement reconnue.

Les « métaheuristiques » d'optimisation sont des algorithmes généraux d'optimisation applicables à une grande variété de problèmes. Elles sont apparues à partir des années 80, dans le but de résoudre au mieux des problèmes d'optimisation.

Les métaheuristiques s'efforcent de résoudre tout type de problème d'optimisation. Elles sont caractérisées par leur caractère stochastique, ainsi que par leur origine discrète. Elles sont inspirées par des analogies avec la physique (recuit simulé, recuit micro-canonique), avec la biologie (algorithmes évolutionnaires) ou encore l'éthologie (colonies de fourmis, essaims particuliers). Cependant, elles ont l'inconvénient d'avoir plusieurs paramètres à régler. Il est à souligner que les métaheuristiques se prêtent à toutes sortes d'extensions, notamment en optimisation « monoobjectif » et « multiobjectif » [115].

Selon Jacquin [116], Il existe un grand nombre de métaheuristiques différentes qui peuvent être classifiées en deux groupes : les métaheuristiques à solution unique et les métaheuristiques à population de solutions :

- **Les métaheuristiques à solution unique :** Les métaheuristiques à solution unique sont des méthodes itératives qui partant d'une solution initiale puis l'améliore pas à pas en choisissant une nouvelle solution dans le voisinage de la solution courante [117], parmi ces

méthodes les plus connues sont les recherches locales ou méthodes de descente [118] (LS), les recherches locales itérées [119] (ILS), le recuit simulé [120] (RS) et la recherche, tabou [111] (TS) ;

- **Les métaheuristiques à population de solutions :** Les métaheuristiques à population de solutions améliorent une population de solutions au fur et à mesure des itérations. L'intérêt est d'utiliser la population comme un facteur de diversité. Parmi les métaheuristiques à population de solutions, les plus connues et les plus couramment utilisées sont les colonies de fourmis [121], les algorithmes à essaims de particules [122], les systèmes immunitaires artificiels [123] [124] et les algorithmes génétiques [125].

4. Les chemins optimaux dans un graphe pondéré :

4.1. Introduction :

Comme nous avons déjà défini la théorie des graphes dans l'introduction du chapitre, elle est un des outils privilégiés de modélisation et de résolution de problèmes dans un grand nombre de domaines allant de la science fondamentale aux applications technologiques concrètes.

On accorde à Euler l'origine de la théorie des graphes (un article du mathématicien suisse Leonhard Euler, présenté à l'Académie de Saint-Petersbourg en 1735 puis publié en 1741, traitait du problème des sept ponts de Königsberg), parce qu'il fut le premier à proposer un traitement mathématique de la question, suivi par Vandermonde. La théorie des graphes est la discipline mathématique et informatique qui étudie les graphes. Elle constitue un outil primordial pour résoudre les problèmes de cheminement [126].

En général, la représentation utilisée pour modéliser un réseau de transport est un graphe $G=(V, A)$ où l'ensemble des arcs A représente les chemins possibles du réseau et l'ensemble des nœuds V représente les intersections entre ces chemins; en d'autres termes, les nœuds représentent des intersections des rues dans une agglomération et les arcs sont les segments qui relient ces intersections entre elles; ces chemins peuvent être des routes, rues, boulevards, autoroutes... etc.

4.2. Présentation du problème :

Ce problème a de nombreuses applications pratiques, car la « longueur » peut s'interpréter aussi bien comme un coût de transport sur l'arc, comme les dépenses de construction de l'arc a ,

comme le temps nécessaire pour parcourir l'arc a , etc. Selon les propriétés du graphe traité (les longueurs sont quelconques, positives ou toutes égales, le graphe est quelconque ou sans circuit) et selon le problème considéré (recherche du plus court chemin d'un sommet à un autre, ou d'un sommet à tous les autres, ou entre tous les couples de sommets), il existe de nombreux algorithmes qui ont été mis au point et qui permettent l'obtention d'une solution. Nous définissons dans la suite le problème du plus court chemin selon chaque type de graphes, et à la description de quelques algorithmes existants [126].

Les problèmes de cheminement sont des problèmes classiques de la théorie des graphes. Ils consistent à chercher le chemin qui minimise la somme des valuations (poids) des arêtes traversées. Mais, lorsqu'on ajoute des contraintes supplémentaires comme des coûts, distances, vitesses... etc., le problème devient NP-difficile¹⁰.

4.3. Le plus court chemin dans un graphe statique :

Soit $G = (N, A)$ un graphe orienté et pondéré. G est dit graphe statique déterministe si le poids de chaque arc est une valeur constante.

Le problème du plus court chemin pour les graphes statiques déterministes a fait l'objet de nombreuses études. Il est devenu un problème classique de la théorie des graphes. L'objectif est de calculer le plus court chemin entre deux nœuds d'un graphe statique déterministe. Dans cette optique, les notions de fonction de coût d'un chemin et du plus court chemin sont définies.

En 2017, Mansour [126] a défini le problème du plus court chemin dans les graphes statiques déterministes comme suit :

Définition 1 : un graphe $G = (N, A)$ orienté et pondéré (Fig: 2.1) est dit statique déterministe si les poids sur tous les arcs sont des valeurs constantes. N représente l'ensemble des nœuds, et A constitue l'ensemble des arcs. Un chemin entre deux nœuds $(i, j) \in N^2$ du graphe est une suite d'arcs liant i à j , ou bien une suite de nœuds qu'on note $Ch(i, j) = (u_0, u_1, \dots, u_n)$ avec $u_0 = i$ et $u_n = j$.

¹⁰ NP-difficile : en théorie de la complexité de calcul, est la propriété définissant une classe de problèmes qui sont informellement « au moins aussi difficiles que les problèmes les plus difficiles dans NP » [127].

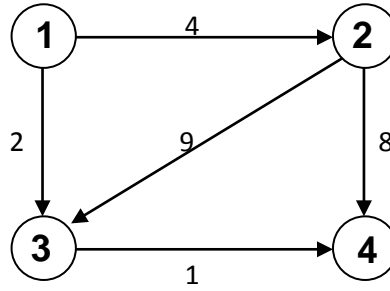


Figure 2.2 : Un graphe statique déterministe.

Définition 2 : Soit $G = (N, A)$ un graphe statique déterministe. On définit la fonction de coût notée C d'un chemin $Ch(i, j)$ dans G de la manière suivante (Equ : 2.4) :

$$C((u_0, u_0, \dots, u_0)) = \begin{cases} \sum_{k=0}^{n-1} FC(u_k, u_{k+1}) & \text{Si } n > 0 \text{ et } \forall k, (u_k, u_{k+1}) \in A \\ 0 & \text{Si } n = 0 \\ \infty & \text{Sinon} \end{cases} \quad (2.4)$$

La fonction $FC(u_k, u_{k+1})$ est la fonction de poids pour les arcs. Alors le plus court chemin entre i et j qu'on note par $PCCh(i, j)$ est donné par la relation suivante (Equ : 2.5) :

$$PCCh(i, j) = Ch(i, j) \text{ avec } C(Ch(i, j)) = \min_{P \in ECh(i, j)} (C(P)) \quad (2.5)$$

Si plusieurs chemins existent avec un coût minimal, alors on choisira un aléatoirement.

De nombreux algorithmes ont été proposés pour résoudre ce problème, en 1957, Prim a publié le premier algorithme qui trouve un arbre d'envergure minimale [128], et en 1959, l'algorithme de Moore du plus court chemin apparut, il est connu souvent par le nom de «Bellman-Ford-Moore». Cet algorithme est basé sur la solution centralisée de Ford [129] et le système d'équations de Bellman [130].

Et dans la même année, le premier algorithme de Dijkstra est publié pour résoudre le problème du plus court chemin dans le cas de la pondération positive [131].

L'Algorithme A^* [132] est autre algorithme déterministe utilisé pour la recherche du plus court chemin dans un graphe, il est beaucoup plus rapide à condition d'avoir une bonne fonction heuristique, et dont l'optimalité n'est garantie que sous certaines conditions. En pratique, l'algorithme A^* est un bon compromis entre coût de calcul et optimalité de la solution.

Plusieurs recherches considèrent hub labelling est le plus rapide algorithme dans ce domaine [133].

4.4. Le plus court chemin dans un graphe dynamique :

Soit $G = (N, A)$ un graphe orienté et pondéré est dit dynamique (Fig : 2.3) si les poids sur tous les arcs sont des valeurs constantes, mais dépendantes du temps. Par conséquent, la fonction de coût pour un lien (i, j) est définie dans ce cas par $FC(i, j, t)$, où t désigne le temps de départ du nœud i . Le temps, pour ce genre de graphe, peut être considéré comme continu ou discret. Mais dans la plupart des cas, et pour simplifier le problème, on considère un temps discret en le divisant en périodes égales de longueurs Δt . La fonction de poids est supposée constante durant chaque période Δt , et le temps n'est pas réinitialisé au niveau de chaque nœud [126].

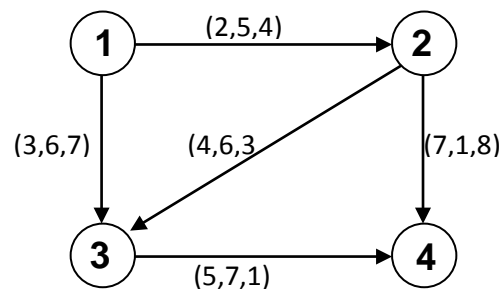


Figure 2.3: Un graphe dynamique déterministe.

Un graphe dynamique est un graphe en évolution dans le temps. Ainsi, il nous semble incontournable de considérer que le processus de l'évolution du graphe est l'un des éléments de sa définition, tout autant que la base temporelle dans laquelle ce processus s'exécute.

Le problème du plus court chemin dynamique a été formulé pour la première fois par Cooke et Halsey [134], où les auteurs ont proposé un algorithme basé sur le principe d'optimalité de Bellman. Depuis, la majorité des travaux qui ont été proposés pour résoudre ce problème en supposant que le temps est discret [135] [136] [137] [138]. La recherche dans le cas continu est effectuée dans [139] [140] [141].

D'après Mansour [126], dans les travaux de [139] il a été montré que pour déterminer le plus court chemin dynamique pour une heure de départ fixée est NP-Difficile, et qu'il est polynomial pour les graphes de type FIFO. [142] et par la suite [143] ont proposé une version dynamique de Dijkstra qui offre une solution optimale dans le cadre des graphes FIFO. En 1998, Pallottino et Scutella [144] ont proposé un paradigme algorithmique appelé Chrono-SPT, pour les graphes non FIFO. L'idée de ce paradigme consiste à visiter les nœuds dans un ordre chronologique, et il est valable que pour le cas où le temps est discret.

Les auteurs A. Orda et R. Rom, 1990 [139] ont été montrés que le problème du plus court chemin dynamique pour une heure de départ fixée est NP-Difficile. Néanmoins Ahn et Y.Shin [142] a été prouvé qu'il est polynomial pour la classe des graphes FIFO¹¹. Dans ce dernier cas, un algorithme de Dijkstra adapté au contexte dynamique peut être utilisé et le résultat qu'il retourne est optimal [142] [143]. En 1998, Pallottino et Scutella [144] ont proposé un paradigme algorithmique appelé chrono-spt, pour les graphes non FIFO. L'idée de ce paradigme consiste à visiter les nœuds dans un ordre chronologique, et il est valable que pour le cas où le temps est discret [126].

4.5. Le problème du plus court chemin stochastique :

Selon Hizem [145], La définition la plus utilisée du plus court chemin stochastique considère comme optimal le chemin qui maximise la valeur espérée d'une fonction d'utilité définie par le décideur [146] [147] [148] [149] [150]. Ou la définition la plus simple dans le cas statique est celle qui décrit l'optimalité d'un chemin par rapport au coût espéré du chemin [151]. Dans ce cas, pour calculer le plus court chemin, il faut calculer l'espérance mathématique du poids de chaque arc. Ensuite, il suffit de dérouler un algorithme de calcul du plus court chemin pour les graphes déterministes comme l'algorithme de Dijkstra en considérant les espérances mathématiques calculées comme étant des poids des arcs [145].

Soit $G = (N, A)$ un graphe orienté et pondéré. G est appelé graphe statique stochastique (Fig : 2.4) si le poids d'un arc $(i, j) \in A$ est défini par une distribution de probabilité discrète. Le poids de l'arc (i, j) est défini de la manière suivante :

$$p(i, j) = \begin{cases} c_1 & \rho_1 \\ c_2 & \rho_2 \\ \vdots & \vdots \\ c_m & \rho_m \end{cases}$$

Avec :

- c_1, c_2, \dots, c_m : sont les poids possibles pour l'arc (i, j) ;
- $\rho_1, \rho_2, \dots, \rho_m$: sont les probabilités associées aux poids, et la somme de ces probabilités égale 1.

¹¹ FIFO (First In, First Out) est une propriété qui modélise l'attente à un nœud [126]. Exemple : une voiture qui emprunte une route la première est certaine d'arriver la première, car aucun véhicule ne peut la dépasser.

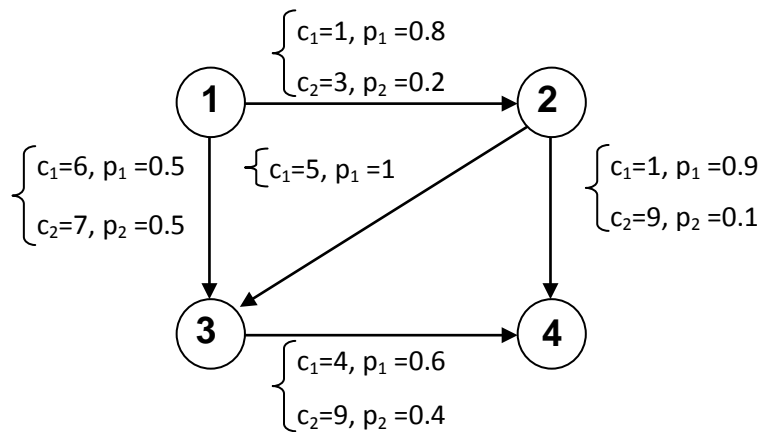


Figure 2.4 : Un graphe statique stochastique [145].

Selon Hizem [145], le problème du plus court chemin dans les graphes statiques stochastiques a été abordé de différentes manières selon la définition de la fonction d'utilité utilisée. On peut les résumer dans ce qui suit :

- La définition qui définit comme étant un chemin optimal qui maximise la valeur espérée d'une fonction d'utilité [146] [147] [148] [149] [150].
- Une définition qui considère qu'un chemin optimal est celui qui maximise la probabilité que la longueur totale du chemin ne dépasse pas une valeur fixe, elle est proposée dans les travaux [152] [153].
- Les auteurs dans [154] [155] [156] ont considéré un chemin optimal comme étant celui qui a la plus grande probabilité d'être le chemin le plus court.

D'autres travaux ont traité le graphe dynamique stochastique, où ils ont introduit la notion du temps dans le graphe. Hall est le premier auteur à avoir traité les graphes dynamiques stochastiques [157]. Il a étudié le problème du calcul d'un chemin entre deux nœuds avec un coût espéré minimal. Le problème est devenu plus compliqué que dans le cas des graphes statiques stochastiques à cause de la nature stochastique et dynamique du graphe [145].

Dans la majorité des travaux, la stratégie de routage est basée uniquement sur les temps d'arrivées au niveau des nœuds de décision et les distributions des poids des arcs sont supposées indépendantes [157] [158]. Deux stratégies de référence existent :

- La première se base sur le concept DOT (Decreasing Order of Time) [159]. Ce concept, développé par Chabini [160], implique l'affectation des labels aux nœuds dans un ordre décroissant par rapport au temps [145].

- La deuxième stratégie se base sur un algorithme à correction de label développé par Miller-Hooks et Mahmassani [161] [162] et qui représente une évolution de l'algorithme qu'ils ont proposé pour le calcul d'un chemin optimal a priori [145].

5. Conclusion :

De nombreuses méthodes sont utilisées pour résoudre les problèmes « multiobjectifs », elles correspondent à des situations et des problématiques différentes. Toutes ces méthodes peuvent être classées selon Delorme [163] en trois grandes catégories suivant le but recherché :

- Celles visant à obtenir une (ou plusieurs) solution(s) représentant un bon compromis entre les différents objectifs, leur but est de trouver une (ou plusieurs) solution(s) qui correspondra (ont) à un bon compromis entre les différents objectifs en fonction des préférences exprimées par le décideur à l'aide des paramètres [163].
- Celles visant à déterminer l'ensemble de la frontière efficace, ces méthodes permettent au décideur de sélectionner à post-priori la (les) solution(s) qui l'intéresse parmi l'ensemble des solutions efficaces [163].
- Et celles visant à calculer une bonne approximation de cette frontière, elle réunit les méthodes ne cherchant qu'une approximation de la frontière efficace. Celles-ci peuvent être utilisées soit dans le cadre d'une méthode interactive, soit pour effectuer une sélection à post-priori [163].

Notons qu'une autre classification, basée sur le type d'interactivité avec le décideur (a priori, interactive ou a post-priori), est parfois utilisée.

Dans le chapitre qui suit, nous expliquerons notre métaheuristique proposée pour résoudre le problème du plus court chemin « multiobjectif ». La logique de cet algorithme est inspirée des comportements des vivants. Ces derniers marquent leurs territoires pour indiquer la possession d'une zone de chasse, de vivre, d'attirer une femelle et de dissuader un autre male de pénétrer chez. L'exploitation de ce phénomène naturel pour résoudre des problèmes d'optimisation combinatoire est le cœur de cette métaheuristique et le centre d'intérêt du troisième chapitre.

Chapitre 3 :

Les animaux marquants et le plus court chemin multiobjectif

1. Introduction :

Lorsqu'un chemin existe entre deux sommets dans un graphe, l'être humain se pose rapidement la question non seulement de trouver un tel chemin, mais bien souvent il est intéressé par le plus court chemin possible entre ces deux sommets. Notre œil est d'ailleurs particulièrement efficace dans cette tâche, tant que le graphe est de taille raisonnable... Mais dès que le graphe comporte plusieurs dizaines de sommets et d'arêtes (comme la ville de Sidi Bel Abbès), trouver le plus court chemin entre deux points devient très vite un casse-tête.

La métaheuristique proposée dans ce chapitre, s'inspire du comportement des vivants, où chaque animal (même l'être humain) indique la possession d'une zone de chasse, de vivre, d'attirer une femelle et de dissuader un autre male de pénétrer chez lui par le marquage de cette zone. Ce dernier représente une protection d'une propriété.

L'exploitation de ce phénomène naturel pour résoudre des problèmes d'optimisation combinatoire est le cœur de cette métaheuristique, des individus circulent et marquent leurs trajets dans ce graphe dans des sens opposés à la recherche de « bons » chemins.

La présente contribution a pour objectif de présenter un algorithme efficace, adéquat, rapide et mis en place pour résoudre le problème de plus court chemin « multiobjectif ».

L'objectif de ce chapitre, est la mise en œuvre d'un système d'information géographique sur le réseau routier urbain avec ses contraintes dites « objectifs » et l'application d'un algorithme inspiré du comportement de marquage naturel des vivants pour résoudre le problème de plus court chemin « multiobjectif » (MOSP) dans ce réseau. Cet algorithme est le centre d'intérêt d'un papier [164] publié le mois du février 2020 (voir l'annexe).

2. Démarches de l'algorithme proposé :

2.1. Principe :

L'idée principale de cet algorithme, est de lancer deux populations sous forme de deux ondes (Fig 3.1), la première est installée au point de départ et l'autre au point d'arrivée. Les deux ondes progressent l'une vers l'autre, les points de rencontre pendant l'affrontement forment des chemins optimaux entre le point de départ et d'arrivée.



Figure 3.1 : Les deux ondes progressent l'une vers l'autre.

Les deux ondes sont représentées dans ce travail par deux populations colorées (noir et rouge). Une population est constituée d'individus de la même couleur et le même nœud de départ. Ces individus se reproduisent, meurent et héritent de leurs parents des informations telles que le point de départ (source), la destination et la couleur. Chaque population est aussi divisée aux deux sous-populations :

- Les individus qui avancent vers la destination et doivent suivre le même sens des arcs ;
- Les individus qui reculent vers la destination et doivent suivre le sens opposé des arcs.

En total, nous avons quatre sous-populations différentes, mais elles coopèrent pour trouver les itinéraires entre le nœud de départ et le nœud d'arrivée dans les deux sens. Ces sous-populations sont (Fig : 3.2) :

- BPA: the black population advancing ;
- BPB: the black population back ;
- RPA: the red population advancing ;
- RPB: the red population back.

Les deux sous-populations (BPA et RPB) explorent des itinéraires qui relient la source — destination, par contre les autres sous-populations cherchent les chemins entre la destination — source (le sens opposé).

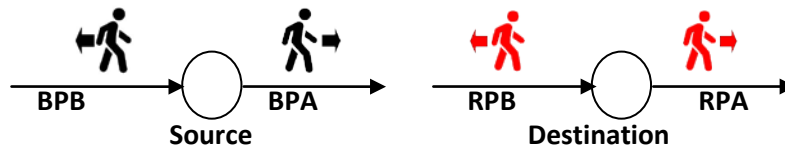


Figure 3.2: Les quatre sous-populations.

Le principe de cette métaheuristique est de placer un individu de chaque population dans le nœud adéquat (le nœud de départ ou d’arrivée), puis ces individus en explorant le graphe se développent en construisant leurs sous-populations. Les rencontres entre les individus des sous-populations provoquent une situation et un événement de l’algorithme. Ces situations sont :

- La rencontre entre les individus de la même sous-population : cette situation peut être provoquée par :
 - Un individu qui trouve son marquage : un circuit est détecté (à éliminer) ;
 - Un individu qui trouve le marquage d’un autre individu de la même sous-population : une autre solution trouvée qu’on doit l’exploiter.
- La rencontre entre les individus (BPA — RPB) ou (RPA — BPB) : une solution est trouvée qu’on doit l’exploiter.

Dans le (Tab : 3.1), les icônes en verts représentent « il y a une ou des solution(s) », par contre les icônes en rouge indiquent « il n’y a pas des solutions » et on doit se débarrasser des individus qui ont provoqué cette situation.

	BPA	BPB	RPA	RPB
BPA	✗	✗	✗	✓
BPB	✗	✗	✓	✗
RPA	✗	✓	✗	✗
RPB	✓	✗	✗	✗

Tableau 3.1 : Les rencontres possibles entre les individus des quatre sous-populations.

La recherche en deux sens à la fois nous a permis de chercher simultanément (en alternance réellement) dans les deux sens (source - destination et destination - source). Et pour ce faire, nous avons réservé pour chaque segment deux rubriques de marquage ou plus. L'avantage de cette logique est de trouver à la fois les chemins possibles entre la source et la destination et l'inverse.

2.2. Représentation de l'algorithme :

Le marquage de territoire est un acte naturel chez la plupart des animaux, il indique la possession d'une zone de chasse, de vivre, d'attirer une femelle et de dissuader un autre male de pénétrer chez lui.

L'homme marque aussi ses propriétés comme les autres animaux territoriaux par les nombreux moyens comme les portes fermées à clé, les grillages, les documents officiels... etc., pour marquer sa propriété personnelle et pour les propriétés collectives il utilise les frontières, les drapeaux, les plaques de signalisation... etc.

L'algorithme proposé est inspiré de ce comportement naturel, pour trouver le chemin le plus court entre deux sommets dans un graphe orienté et pondéré.

Nous avons mis un ensemble des règles qui gèrent l'évolution et le développement de ces sous-populations et l'exploitation des solutions trouvées, ces règles sont comme ce qui suit :

2.2.1. La reproduction des individus :

La reproduction est un acte naturel et un facteur important dans l'évolution de la sous-population, les individus héritent la couleur de la sous-population, le point de départ (source) et la destination (objectif) à ses fils, la reproduction se fait quand et seulement quand l'individu rencontre plusieurs rues à exploiter à la fois (deux rues et plus).

L'individu est identifié par son propre code (empreinte) qui lui distingue du reste des individus de la sous-population. Une table dite « généalogique » est réservée pour les origines de chaque individu. Cette table est mise à jour après chaque reproduction.

L'individu ne reproduit pas dans le cas, où l'individu est bloqué dans une impasse, ou il rencontre une seule rue à exploiter.

2.2.2. L'élimination des individus :

Comme dans la nature, la reproduction incontrôlable est un souci qu'on doit l'éviter en maximum sinon on sera face aux manques des ressources qui peuvent provoquer des problèmes sérieux (dans notre cas, le manque de l'espace mémoire, plantage du système... etc.).

Pour contrôler la croissance massive d'une sous-population, on doit se débarrasser de tout individu jugé inutile, on peut les résumer dans les cas possibles :

- Après chaque reproduction, l'individu « père » doit se disparaître ;
- La rue à exploiter est déjà marquée par un autre individu ;
- L'individu est bloqué dans une impasse.

L'élimination des individus nous a permis d'éviter les saturations de mémoire, et surtout dans les gigantesques graphes (grandes villes).

2.2.3. Le marquage :

Chaque rue du réseau possède deux informations de marquage, une rubrique réservée pour les individus qui parcourent le chemin entre la source et la destination et l'autre pour ceux qui passent en sens contraire. Ces deux rubriques seront utilisées comme un moyen de communication entre les autres individus et comme un moyen de développement des solutions (Fig : 3.3). Dans la recherche par alternance de 4 rubriques de marquage, chaque rue a 04 rubriques où chaque sous-population utilise sa propre rubrique de marquage. Dans ce cas, chaque rue du réseau possède 04 champs de marquage.

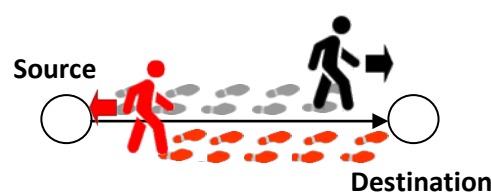


Figure 3.3 : Le marquage des rues par les individus.

La rue est marquée par le premier individu parcouru, son empreinte (identifiant de l'individu) est sauvegardée dans la rubrique de marquage appropriée.

Une fois, la rue (i, j) est parcourue par l'individu P, il enregistre son identifiant dans la rubrique de marquage de cette rue M_{ij} puis il vérifie le nombre des rues adjacentes de la rue courante (N_{AS}) pendant son avancement. S'il est coincé dans une impasse ($N_{AS}=0$), cet individu

est automatiquement éliminé. Sinon s'il y a une seule rue voisine alors cette rue est choisie et sa rubrique de marquage M_{ij} est marquée par P et enfin si le nombre des rues adjacentes est supérieur de 1, l'individu P reproduit pour chaque rue un enfant qui sera installé sur cette rue (Equ : 3.1) :

$$P_{ij}^{M_{ij}}(t) = \begin{cases} \text{Kill them;} & \text{si } N_{AS} = 0 \\ \text{SelectedRoad} = \text{road}(j, k) \text{ and } M_{jk} = P; & \text{si } N_{AS} = 1 \\ \text{SelectedRoad} = \text{road}(j, k) \text{ and } M_{jk} = \text{child}(P); & \text{si } N_{AS} > 1 \end{cases} \quad (3.1)$$

D'où :

P : L'individu P ;

M_{ij} : La rubrique de marquage de la rue courante i, j marquée par l'individu P ;

M_{jk} : La rubrique de marquage de la rue adjacente j, k de la rue courante i, j ;

N_{AS} : Le nombre des rues adjacentes de la rue courante i, j .

La rubrique de marquage joue un rôle très important dans la traçabilité des parcours des individus dans le graphe. Pour cela, chaque sous-population utilise sa propre rubrique de marquage (le cas de la recherche par alternance de 4 rubriques de marquage) (Fig : 3.4), comme on peut réserver une seule rubrique pour deux sous-populations (le cas de la recherche par alternance de 02 rubriques de marquage).

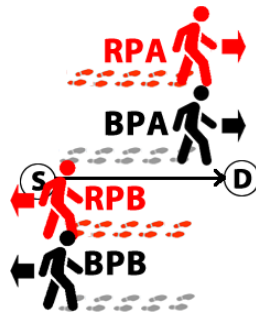


Figure 3.4 : La recherche par alternance de 4 rubriques de marquage.

Le choix de ces deux sous-populations qui partagent la même rubrique dans la recherche par alternance de 02 rubriques de marquage, joue un rôle important dans la qualité de la solution et le temps de l'avoir. Dans notre travail, deux variantes sont proposées :

- Les deux sous-populations qui forment un itinéraire (BPA et RPB partagent la même rubrique, RPA et BPB partagent l'autre) (Fig : 3.5 à gauche) ;

- Les deux sous-populations qui circulent dans le même sens (BPA et RPA partagent la même rubrique, RPB et BPB partagent l'autre) (Fig : 3.5 à droite).

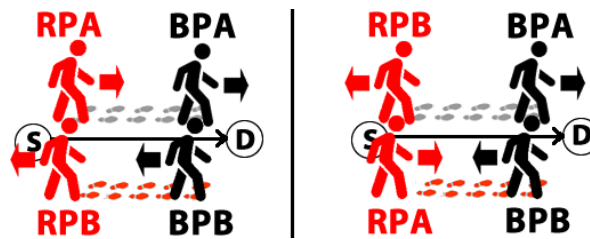


Figure 3.5 : La recherche par alternance de 2 rubriques de marquage (à gauche, les deux sous-populations qui forment un itinéraire : BPA — RPB partagent la même rubrique et RPA — BPB partagent l'autre, à droite, les deux sous-populations qui circulent dans le même sens : BPA — RPA partagent la même rubrique et RPB — BPB partagent l'autre).

2.2.4. La concurrence entre les individus :

La concurrence des individus dans les rues est un problème majeur qu'on doit l'étudier et bien le contrôler. Sinon on sera face au blocage de l'algorithme (saturation) ou des résultats insatisfaisants, surtout dans les recherches de 02 rubriques de marquage.

Pendant son exploration, l'individu peut rencontrer des traces des autres individus dans les rues adjacentes, sinon il continue son parcours sans aucun souci (une rue adjacente libre et non marquée). Dans le cas où la rue rencontrée aurait été marquée, l'individu sera automatiquement bloqué dans cet emplacement et nous sommes devant trois situations possibles :

- Si le marquage trouvé est celui de cet individu, alors ce dernier est bloqué dans un circuit, il sera automatiquement éliminé ;
- Si ce marquage d'un individu de la même sous-population de cet individu, alors il y a un autre chemin trouvé depuis la source ou vers la destination. Cet individu doit être tué ;
- Et en fin, si le marquage de cette rue adjacente est celui d'un individu d'une autre population concurrente (BPA — RPB) ou (RPA — BPB), alors il y a une solution trouvée entre la source et la destination ou l'inverse. Cet individu a réussi d'avoir une solution et il sera automatiquement éliminé.

2.2.5. Les types de recherche :

La façon de trouver des solutions joue un rôle très important, soit dans la qualité de la solution, soit dans la rapidité des réponses, l'algorithme propose deux techniques de recherche et qui sont :

- La recherche séquentielle : l'algorithme cherche d'abord les itinéraires entre la source et la destination, dès qu'il termine et les résultats sont obtenus, il recherche de nouveau les itinéraires entre la destination et la source. Pour contrôler le sens de la recherche (source — destination ou destination — source), il suffit de créer des sous-populations concernées pour chaque direction pendant l'initialisation ;
- La recherche par alternance : toutes les sous-populations participent dans cette recherche. Les individus de deux sens avancent en alternance (une itération pour source - destination et l'autre pour destination - source). On distingue deux types de recherche par alternance selon le nombre des rubriques de marquage :
 - Recherche par alternance sur deux rubriques de marquage : Chaque segment du graphe routier contient deux rubriques de passage des individus : une rubrique est réservée pour les individus de BPA et (RPA ou RPB) et l'autre rubrique est réservée pour les individus de BPB et (RPB ou RPA). La concurrence entre les individus sur les rues peut provoquer des saturations ou blocages. Cette situation critique influence sur le temps et la qualité des solutions et même sur le déroulement de l'algorithme. Nous proposons deux modèles dans ce type de recherches et qui sont :
 - La recherche en même sens : les deux sous-populations qui parcourent le graphe en même sens (la même direction) exploitent la même rubrique, alors elles sont classées comme suit :
 - Rubrique « Avancée » : est exploité par BPA et RPA ;
 - Rubrique « Reculée » : est exploité par BPB et RPB ;
 - La recherche en sens opposés : dans ce cas, les deux sous-populations concurrentes et leurs intersections ramènent une solution, exploitent la même rubrique :
 - Rubrique « Source — destination » : est exploité par BPA et RPB ;
 - Rubrique « Destination — source » : est exploité par BPB et RPA ;

En cas des blocages ou saturations, on doit éliminer complètement les deux sous-populations qui ont provoqué cette situation pour favoriser les autres sous-populations à chercher des solutions.

- Recherche par alternance sur 04 rubriques de marquage : cette méthode exige 04 rubriques de passage des individus, chaque rubrique est réservée pour une sous-population. Cette méthode gère l'évolution des 04 sous-populations en souplesse sans aucune saturation ou concurrence entre individus puisqu'elle alloue à chaque sous-population sa propre rubrique de marquage, mais elle demande plus d'espace mémoire, car elle ajoute pour chaque segment du réseau deux rubriques.

2.2.6. La recherche d'une solution :

Le chemin entre la source et la destination est composé de deux parties ou segments ; ces derniers sont les parcours de deux individus qui ont trouvé cette solution (Fig : 3.6).

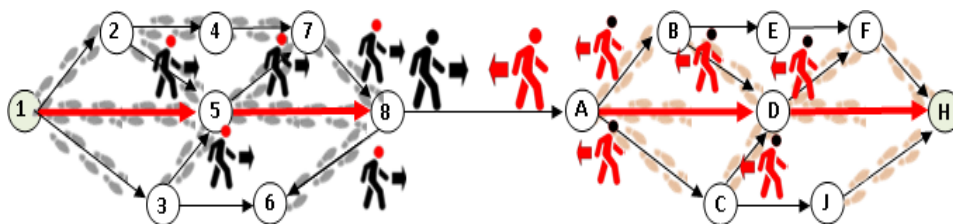


Figure 3.6 : La rencontre de deux individus de deux populations concurrentes forme une solution.

Le chemin d'un individu est un ensemble des rues parcourues, il débute du point de départ et se termine au point de rencontre avec l'autre individu. La fusion entre les deux segments constitue l'itinéraire entre la source (point de départ du premier individu) et la destination (le point de départ du deuxième individu).

Pour composer un parcours d'un individu, l'algorithme utilise une table généalogique qui regroupe tous les identifiants des individus créés avec leurs parents (leurs individus créateurs).

La recherche des racines d'un individu donné utilise une fonction récursive du plus petit enfant au plus grand père. L'exemple d'une table généalogique suivante (Tab : 3.2) explique ce principe :

<i>Child</i>	<i>parent</i>
①	①
2	1
③	①
4	2
⑤	③
6	3
⑦	⑤
8	5
9	⑦

Tableau 3.2: La recherche de la racine dans la table généalogique.

La racine de l'individu 9 est : 9 – 7 – 5 – 3 – 1.

Si l'identifiant de l'individu enfant égale l'identifiant du père, alors la recherche est terminée et la racine de cet individu est trouvée.

La recherche de parcours d'un individu P vérifie d'abord le nœud du segment (i, j) en cours, s'il égale le nœud de départ ($i =$ nœud de départ ou $j =$ nœud de départ), alors la recherche est terminée et le chemin est construit. Sinon il doit vérifier le nombre des rues adjacentes de son emplacement (N_{AS}). Si $N_{AS} = 1$ alors la rue suivante est sélectionnée, par contre si le nombre des rues voisines est supérieur de 1 alors le père de l'individu en cours est choisi et la rue marquée par celui-ci est retenue.

Le modèle mathématique (Equ : 3.2) de cette recherche est représenté comme suit :

$$P_{ij}^{M_{ij}}(t) = \begin{cases} \text{Stop} & \text{si } i = \text{Départ ou } j = \text{Départ} \\ \text{SelectedRoad} = \text{road}(j, k) & \text{si } N_{AS} = 1 \\ \text{SelectedRoad} = \text{road}(j, k) \text{ and } P = \text{father}(P); & \text{si } N_{AS} > 1 \end{cases} \quad (3.2)$$

D'où :

P : identifiant de l'individu ;

M_{ij} : la rue i, j marquée par l'individu.

M_{jk} : la rue j, k est la rue adjacente de la rue M_{ij}

N_{AS} : le nombre des rues adjacentes de la rue i, j

L'organigramme suivant (Fig : 3.7) explique le principe de composition d'un parcours d'un individu donné en se basant sur la table généalogique et les rues adjacentes, dans l'algorithme cet organigramme est représenté par une fonction récursive qu'il permet d'exploiter le parcours d'un individu quelconque.

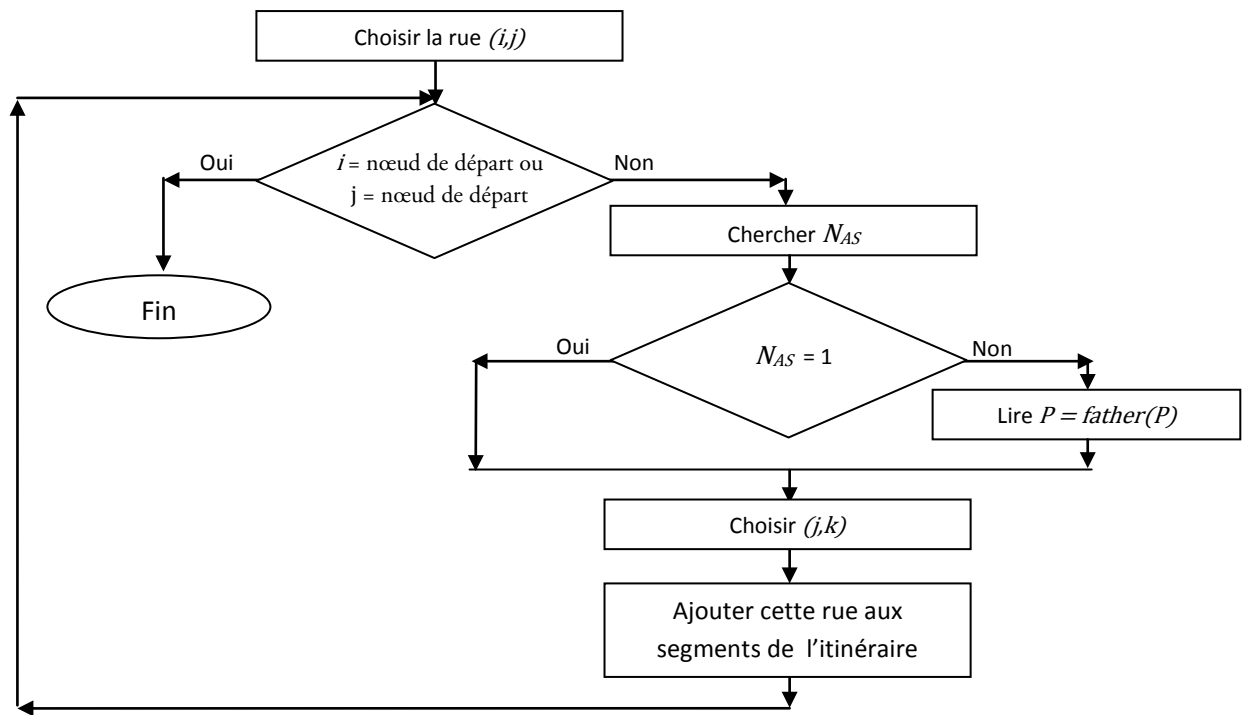


Figure 3.7: L'organigramme de composition d'un parcours d'un individu.

2.2.7. Enrichir la solution :

Les premières rencontres entre deux sous-populations concurrentes (BPA — RPB ou RPB — BPA) donnent des chemins optimaux par rapport aux nombres des arcs entre le point de départ et d'arrivée. Chaque itinéraire trouvé est considéré comme le plus court chemin entre les extrémités dans un graphe orienté sans pondération (le poids de chaque arc égal à 1).

L'exploitation des parcours des individus rencontrés qui ont trouvé les premières solutions nous montre qu'il y a des individus de la même sous-population ; ont été bloquées (puis éliminés) devant des rues déjà marquées par ces deux individus (les individus avec des têtes rouges dans la Fig : 3.8). Les parcours de ces individus bloqués peuvent former d'autres solutions réalisables. L'exploration de ces parcours constitue un ensemble des chemins réalisables entre le nœud de départ et le nœud de rencontre, l'algorithme sauvegarde ces chemins pour l'exploiter dans le « multiobjectif ».

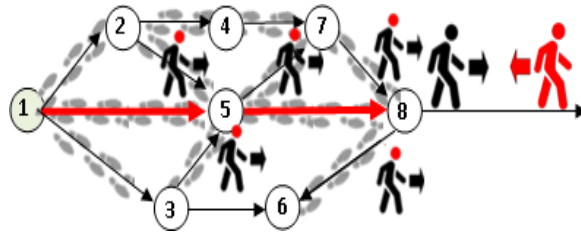


Figure 3.8 : l'itinéraire 1-5-8 est le chemin le plus court par rapport aux nombres d'arcs, et les parcours des individus bloqués (avec des têtes rouges dans la figure) donnent d'autres solutions réalisables.

Les chemins trouvés entre le sommet de départ et le point de rencontre d'après la figure précédente sont recensés dans le tableau suivant (Tab : 3.3) :

N°	chemins	observation
1	1-5-8	optimal
2	1-2-5-8	
3	1-2-5-7-8	
4	1-2-4-7-8	
5	1-5-7-8	
6	1-3-5-8	

Tableau 3.3: Recensement des chemins trouvés.

L'exploitation des parcours des deux individus nous a donné plusieurs chemins réalisables (quelques-uns sont rejetés parce qu'ils sont des circuits) ; la fusion entre les deux chemins de deux individus rencontrés donne un itinéraire entre le sommet de départ et le sommet d'arrivée (Tab : 3.4).

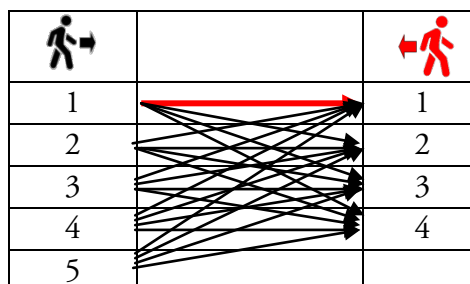


Tableau 3.4: La fusion entre les chemins trouvés forme plusieurs itinéraires réalisables.

Si N est le nombre des chemins trouvés par le premier individu et M est le nombre des chemins trouvés par le deuxième individu ; alors le nombre total des itinéraires trouvés pendant une rencontre égale $N \times M$.

2.2.8. La condition d'arrêt de l'algorithme :

Cette condition est un facteur important et décisif d'un algorithme quelconque, puisqu'il détermine le temps d'exécution et la qualité des résultats obtenus. Cet algorithme en sujet doit être arrêté dans les cas suivants :

- Si toutes les sous-populations sont nulles, alors l'algorithme sera automatiquement terminé ;
- Si une seule sous-population est nulle, la sous-population concurrente est automatiquement supprimée et la recherche de l'itinéraire approprié est rejetée ;
- Si toutes les solutions sont trouvées selon la profondeur demandée ;
- Si une sous-population est bloquée ou saturée (elle ne se reproduit plus pendant un nombre d'itérations donné) à cause des concurrences entre les individus. Cette sous-population et sa concurrente sont simplement éliminées pour favoriser les autres sous-populations à trouver des solutions.

2.3. L'algorithme proposé et le plus court chemin « multiobjectif » :

2.3.1. Présentation :

Il existe de nombreux problèmes dans le monde réel pouvant être modélisé par des problèmes combinatoires, par exemple la recherche d'un chemin dans un réseau routier. Pour ces problèmes, on va en général chercher à optimiser un objectif comme le coût de ce chemin. Parfois, il est nécessaire de prendre en compte plusieurs objectifs. Cependant, il n'existe pas en général une solution qui soit optimale sur chacun des objectifs. La notion de solution optimale unique à un problème d'optimisation combinatoire « multiobjectif » quand elle existe, n'a plus de sens. Il est possible de calculer un ensemble de solutions de compromis pour lesquelles il n'existe aucune autre solution qui soit meilleure sur chacun des objectifs. Ces solutions sont dites efficaces. L'ensemble des solutions efficaces pouvant être important, il se pose alors la question de cerner la meilleure solution efficace d'un problème « multiobjectif ». L'aide à la décision « multiobjectif » propose des réponses à cette question. Il faut pour y répondre collecter des informations sur les préférences d'un expert du domaine capable d'exprimer ses préférences sur les solutions du

problème. Cet expert est appelé le décideur. Les informations préférentielles exprimées par le décideur permettent alors de construire un modèle de préférence, qui permet par exemple de choisir la meilleure solution.

L'approche de la recherche de solutions de compromis consiste à construire des solutions dont le vecteur de coûts représente sur chaque objectif un compromis par rapport aux coûts des autres solutions efficaces. Supposons que chaque objectif soit très important et qu'une solution mauvaise sur au moins un objectif ne puisse être considérée comme une solution optimale s'il existe une solution qui n'est mauvaise sur aucun objectif, alors cette approche doit être préférée à l'approche précédente consistante à optimiser une somme pondérée [165].

Deux approches ont été abordées pour résoudre ce problème [166].

- L'approche a posteriori consiste à calculer l'ensemble des solutions de compromis. Une méthode classique de la littérature est présentée et améliorée en utilisant des prétraitements. Les améliorations proposées reposent sur les recherches de plus court chemin « monoobjectif » et qui permettent de calculer sur chacun des nœuds du graphe, des bornes inférieures et supérieures sur les coûts des chemins (du nœud de départ vers le nœud de destination) [166];
- La deuxième approche a priori prend en compte les préférences de l'utilisateur pour se concentrer sur le calcul d'une solution de meilleur compromis. La méthode utilisée permet d'orienter la recherche des chemins, de façon à privilégier les sous-chemins les plus prometteurs du point de vue des préférences de l'utilisateur [166].

2.3.2. Définitions :

On suppose que la fonction objective prioritaire est la fonction $Min(Sum_S_{Obj})$. Et S est l'ensemble des segments trouvés par les individus rencontrés $S = (S_1, S_2, S_3, \dots, S_n)$, et Obj est l'ensemble des critères environnementaux, organisationnels pris en compte $Obj = (Obj_1, Obj_2, Obj_3, \dots, Obj_n)$. Notre problème de plus court chemin « multiobjectif » est modélisé comme suit (Equ : 3.3):

$$Min(Sum_S_{Obj}) = \sum_{i=1}^n S_{index_Obj_i} \quad (3.3)$$

Où :

$Sum_S_{index_Obj}$: La somme des indexes des objectifs d'un segment S ;

n : Le nombre des objectifs à calculer ;

S_{index_Obj} : Index ou la position de cet objectif après le trié d'un segment S .

2.3.3. Application :

Avant d'expliquer la solution « multiobjectif », nous présentons d'abord notre modèle logique des données de notre programme, ce modèle logique est utilisé pour la construction d'une base de données utilisant un système de gestion de base de données relationnel (SGBDR). Au niveau logique, la relation est de type mathématique et correspond à une entité ou une relation. De façon plus pragmatique, une relation logique peut être vue comme un tableau à n colonnes (appelées « attributs ») qui jouent un rôle sémantique précis. Une ou plusieurs colonnes jouent le rôle de « clé » (c'est l'identifiant). Dans le formalisme couramment utilisé, la relation est représentée par un schéma qui est défini par le nom de la relation, suivi de la liste des attributs dans laquelle on souligne le ou les attributs qui jouent le rôle de « clé » [167].

Les solutions sont organisées dans 03 tables différentes (Fig : 3.9), la table « les solutions » contient des solutions trouvées par l'algorithme, où chaque solution représente une rencontre entre deux individus de deux sous-populations concurrentes, cette table est composée des attributs suivants :

- Identifiant de la solution ;
- La sous-population du premier individu ;
- Le vecteur des chemins réalistes de la première sous-population ;
- La sous-population du deuxième individu ;
- Le vecteur des chemins réalistes de la deuxième sous-population.

La deuxième table sauvegarde la somme des coûts (objectifs) de ces chemins, chaque chemin est connu par son identifiant et un vecteur qui regroupe la somme des poids des arcs constituant ce chemin. Et enfin, la troisième table garde les identifiants des chemins avec leurs arcs.

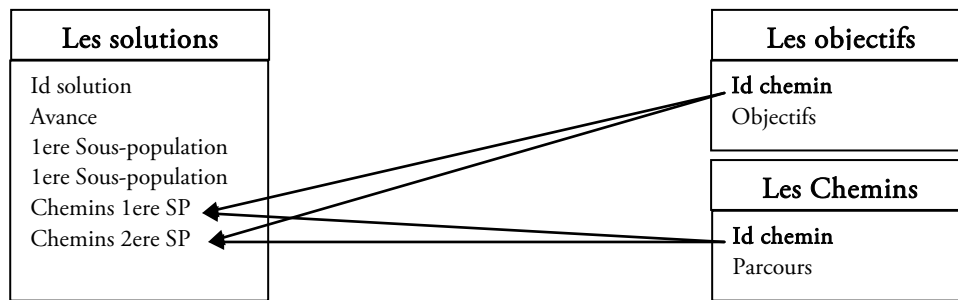


Figure 3.9 : Le modèle logique des données.

Le meilleur compromis entre des objectifs d'une solution est calculé dans la table des objectifs.

Les tableaux suivants (Tab : 3.5, Tab : 3.6, Tab : 3.7) expliquent ce principe :

N°	Chemins	C1	C2	C3	...	Cn	$Sum_S_{index_C}$
1	1-5-8	1	3	2		6	1
2	1-2-5-8	3	1	2		3	2
3	1-2-5-7-8	4	2	4		8	3
4	1-2-4-7-8	4	3	3		9	3
5	1-5-7-8	5	4	5		7	4
6	1-3-5-8	6	4	2		2	5

Tableau 3.5 : On commence par le tri de la première objective, puis la rubrique

$Sum_S_{index_C}$ calcule la somme des index.

La table « objectifs » ne contient pas l'attribut « chemins » comme il est mentionné dans (Tab:3.5), mais, elle est ajoutée dans ce tableau juste pour comprendre le principe.

N°	Chemins	C1	C2	C3	...	Cn	$Sum_S_{index_C}$
2	1-2-5-8	3	1	2		3	2+1
3	1-2-5-7-8	4	2	4		8	3+2
1	1-5-8	1	3	2		6	1+3
4	1-2-4-7-8	4	3	3		9	3+3
5	1-5-7-8	5	4	5		7	4+4
6	1-3-5-8	6	4	2		2	5+4

Tableau 3.6 : Le tri de la deuxième objective, et puis la rubrique $Sum_S_{index_C}$ calcule la somme des index.

Après le calcul de tous les index des chemins, l'algorithme doit trier l'attribut $Sum_S_{index_C}$ pour connaître le meilleur compromis entre les chemins d'une sous-population, le même processus sera appliqué pour les chemins de la deuxième sous-population. Et la fusion entre les deux meilleurs chemins des deux sous-populations nous donne sûrement l'itinéraire le plus court « multiobjectif » d'une solution.

N°	chemins	C1	C2	C3	...	Cn	$Sum_P_{index_C}$
2	1-2-5-8	3	1	2		3	9
1	1-5-8	1	3	2		6	12
6	1-3-5-8	6	4	2		2	14
3	1-2-5-7-8	4	2	4		8	18
4	1-2-4-7-8	4	3	3		9	19
5	1-5-7-8	5	4	5		7	21

Tableau 3.7 : Le tri de la rubrique $Sum_S_{index_C}$ qui contient la somme des index.

Jusqu'à maintenant, nous avons trouvé seulement le meilleur parcours dans une seule solution, il reste de calculer le chemin « multiobjectif » parmi toutes les solutions disponibles, l'algorithme propose alors de refaire le même principe cité ci-dessous sur toutes les solutions disponibles. Le meilleur chemin parmi les meilleurs itinéraires de chaque solution existante donne le chemin optimal recherché (le meilleur entre les meilleurs).

3. Conclusion :

Le problème de plus court chemin « multiobjectif » a pour objectif le choix d'un chemin dans un graphe en tenant compte simultanément de plusieurs objectifs. Dans notre cas, nous avons sept objectifs à savoir le temps, la distance, le trafic, le nombre des ralentisseurs, le nombre des feux tricolores, la hauteur autorisée et le poids autorisé.

Nous avons mis en place un modèle du plus court chemin « multiobjectif » ayant pour objectif d'aider au choix d'un « plus court chemin » dans un réseau routier urbain. Dans notre cas, « plus court chemin » signifie un chemin avec moins d'impacts environnementaux (feux tricolores) et sécuritaires (hauteur autorisée, poids autorisé, ralentisseurs) et garantissant des critères économiques (distance, temps) satisfaisants.

L'exploitation du marquage naturel des vivants (pour indiquer la possession d'une zone) pour résoudre des problèmes d'optimisation combinatoire est le cœur de cette métaheuristique, des individus circulent et marquent leurs trajets dans ce graphe dans des sens opposés à la recherche de « bons » chemins.

Nous avons ainsi fourni un système d'aide à la décision pour les décideurs qui souhaitent réduire leurs impacts sur l'environnement lors du choix d'itinéraire et pour la planification de leurs expéditions. Le système permet de se baser sur un certain nombre de données d'entrée pour fournir un itinéraire répondant aux exigences du décideur.

Dans le chapitre suivant, nous concrétiserons la solution proposée dans ce chapitre et nous analyserons les résultats obtenus par l'application de cet algorithme sur notre réseau routier urbain de la ville de Sidi Bel Abbès.

Chapitre 4 :

Implémentation et résultats

1. Introduction :

La base de données sur laquelle repose le SIG est dite spatiale lorsqu'elle permet la définition de tables spatiales présentant un ou plusieurs attributs complexes : les attributs spatiaux. Ces attributs particuliers contiennent des informations sur la forme et la localisation des entités géographiques modélisées par la table spatiale.

Grâce aux attributs spatiaux, les enregistrements d'une table présentent des propriétés particulières permettant leur visualisation cartographique, mais aussi leur traitement par des opérations relevant du domaine de l'analyse spatiale. Actuellement, la plupart des SGBD proposent une extension pour la gestion de l'information géographique : PostGIS, Oracle Spatial, MySQL, etc.

La plupart des systèmes de gestion de bases de données fonctionnant sur les modèles présentés proposent des extensions spécifiques pour la gestion de données géographiques. L'extension la plus connue est une des plus complètes et des plus documentées, est l'extension PostGIS fonctionnant sous le SGBD (Système de gestion de bases de données) relationnel PostgreSQL. Cette extension permet par exemple d'utiliser des attributs géographiques (points, lignes et polygones) et d'effectuer des requêtes spatiales. Le modèle orienté « graphe » est toutefois considéré comme étant le plus prometteur et le mieux adapté pour représenter des données spatiales ayant de multiples relations entre elles, bien qu'il ne puisse pas encore répondre à tous les besoins d'analyse auxquels répondent les extensions spatiales associées aux SGBD relationnels. En ce sens, il demeure pour l'instant incomplet [168].

La concrétisation de la solution choisie dans le chapitre précédent fait appel aux questions suivantes :

- Quels sont les outils (en matériels et en logiciels) nécessaires pour concrétiser la méthode théorique de cet algorithme ?
- Quelle est la méthodologie la plus efficace pour simplifier l'application proposée ?
- Quel est le système de gestion de base de données (SGBD) approprié qui sauvegarde nos données (géographiques, images, numériques... etc.) ?
- Quel est le langage de programmation adéquat pour programmer et appliquer les algorithmes de plus court chemin simple et « multiobjectifs » ?

Les réponses à ces questions et autres forment nos buts opérationnels dans ce chapitre.

2. Présentation de l'algorithme :

2. 1. Principe et objectifs :

L'objectif de notre étude est de préparer un environnement adéquat pour réaliser et évaluer notre algorithme, pour ce faire on doit :

- Extraire les données géographiques réelles et correctes de la ville d'expérimentation ;
- Simplifier l'interface du site web, pour pouvoir l'exploiter facilement ;
- Prendre en considération un ensemble des coûts de déplacements (la distance, le nombre de ralentisseurs, le nombre des feux tricolores, la hauteur autorisée, le poids autorisé... etc.) des véhicules dans un réseau routier urbain, pour calculer les chemins « multiobjectifs » ;
- Évaluer l'optimalité et la rapidité de notre produit et comparer ses résultats obtenus avec ceux d'un autre algorithme connu dans le domaine du plus court chemin, cette comparaison nous a permis de vérifier son efficacité ;
- Exploiter les algorithmes proposés par le projet PgRouting pour évaluer notre algorithme ;
- Appliquer notre algorithme proposé — dans le chapitre précédent — et qui permet de trouver le plus court chemin « multiobjectif » (MOSP) dans une ville algérienne (Sidi Bel Abbes précisément).

2.2. Aspect physique et logiciel :

2.2.1. Aspect physique

Les tests ont été effectués sur un micro-ordinateur : Intel(R) Core(TM) i7-6700T CPU @ 2.80 GHz, 2801 MHz, 4 cœur(s), 8 processeur(s) logique(s), une mémoire vive de 8,00 Go octet, un disque dur de 500 Géga octet, Microsoft Windows 7 Professionnel et la configuration recommandée de la carte VGA est 1366 x 768 (la résolution d'écran) et 32 bits (qualité de couleur).

2.2.2. Aspect logiciels :

a. Présentation du projet PgRouting :

Actuellement, il y a un nombre important des logiciels qui permettent de modéliser les réseaux routiers en différentes dimensions (2D, 3D), et l'accès à l'information spéciale est devenu très

facile et gratuit. Ceci a donné naissance aux plusieurs projets de la recherche d'apparaître et même de diffuser leurs résultats et code source gratuitement.

PgRouting est une extension de PostGIS et ajoute les fonctionnalités de Routing au couple PostGIS/PostgreSQL. PgRouting est un développement antérieur à pgDijkstra (par Camptocamp SA). Il a été étendu par l'entreprise Orkney Inc. et est actuellement développé et maintenu par Georepublic. PgRouting fournit les fonctions pour :

- Plus court chemin Dijkstra : algorithme de routage sans heuristique ;
- Plus court chemin A-Étoile : routage pour grand un ensemble de données (avec heuristiques);
- Plus court chemin Shooting-Star : routage prenant en compte le sens giratoire (avec heuristiques) ;
- Problème du voyageur de commerce (TSP) ;
- Distance de pilotage (Isolines).

De nombreux nouveaux algorithmes vont être ajoutés dans un futur proche :

- Dial-a-Ride-Problem solver (DARP) ;
- Support du routage multimodal ;
- Algorithme de recherche de plus court chemin dépendant du temps/dynamique ;
- A-Étoile utilisant deux directions ;
- All-Pair shortest path algorithm.

Les avantages de l'approche base de données pour les algorithmes de routage sont :

- L'accessibilité par nombreux clients au travers de JDBC, ODBC, ou directement en utilisant Pl/pgSQL. Les clients peuvent être des ordinateurs personnels ou des périphériques mobiles ;
- Utilise PostGIS pour le format des données géographiques, qui utilise le Format Text Bien Connu (WKT) de l'OGC et le Format Binaire Bien Connu (WKB) ;
- Les logiciels OpenSource comme qGIS et uDig peuvent modifier les données/les attributs ;
- Les modifications apportées aux données peuvent être intégrées au moteur de routage instantanément. Il n'y pas de besoin de pré-calcul ;
- Le paramètre de « coût » peut être calculé dynamiquement à l'aide du SQL et ces valeurs peuvent provenir de différents attributs ou tables ;
- PgRouting est disponible sous licence GPLv2.

Les fonctions PgRouting sont basées sur les coûts, qui sont dynamiques par opposition aux coûts pré-calculés et peuvent donc s'appliquer à différents types de scénarios routiers réels et fonctionner sans reconstruire ou recharger l'ensemble du réseau routier.

Intégration de notre algorithme dans le PgRouting, nous a permis de bien évaluer notre programme avec les algorithmes de routage disponibles.

b. Présentation des données spatiales OpenStreetMap (OSM) et osm2PgRouting :

OpenStreetMap (OSM) est un ensemble de données ouvertes, disponibles sous licence libre ODbL. Ces données mondiales sont alimentées par la communauté d'utilisateurs ainsi que par de grands organismes. Par exemple, l'entreprise néerlandaise Automotive Navigation Data (AND) a importé ses données dans OSM, tout comme les États unis avec la base de données TIGER. La donnée routière peut être de très bonne qualité dans certains pays et quasi inexistante dans d'autres. Les grandes villes sont bien renseignées et on y retrouve les attributs nécessaires à la simulation de trafic urbain. Certaines villes proposent même l'emplacement des feux tricolores, des passages pour piétons, des arrêts de bus. Quand la donnée est bien complète, les largeurs des tronçons, la vitesse maximale y étant autorisée, leur nombre de voies sont indiqués. De plus, OpenStreetMap est en constante évolution et le nombre d'entrées ne fait qu'augmenter depuis sa création [169] [170] [171].

Osm2PgRouting est un outil en ligne de commande qui rend simple l'importation de données OpenStreetMap dans une base de données PgRouting. Il construit le réseau routier automatiquement et crée les tables pour les types de données et les classes de routes. osm2PgRouting a été écrit initialement par Daniel Wendt et est maintenant disponible sur le site du projet PgRouting [172]. L'osm2PgRouting est un outil pratique, mais c'est aussi une boîte noire. Il y a de nombreux cas où osm2PgRouting ne peut pas être utilisé. Certaines données de réseau sont fournies avec la topologie du réseau qui peut être utilisé par PgRouting tel quel. Certaines données de réseau sont stockées au format Shape file (.shp) et nous pouvons les charger dans une base de données PostgreSQL à l'aide de l'outil de conversion de PostGIS » « shape2postgresql ».

Après l'importation des données dans le PostGis, nous devons nous assurer que nos données fournissent une topologie correcte du réseau, ce qui correspond aux informations par rapport au début et à la fin d'un tronçon.

Une fonction « **pgr_createTopology** » est mise à la disposition des utilisateurs, elle permet l'assignation des valeurs pour les colonnes « source » et « target » pour chaque tronçon et il peut prendre en compte le fait qu'un sommet puisse être éloigné d'un autre suivant une certaine tolérance. Cette fonction exige comme paramètre : la table des rues, la tolérance, la colonne de géométrie et l'identifiant de la rue.

L'application de cette fonction sur les trois réseaux urbains d'expérimentation (Sidi Bel Abbes, Oran et Alger) nous a donné les résultats suivants (Tab 4.1) :

Villes	Temps nécessaire de création	Nombre des routes	Nombre des nœuds
Alger	136490ms	38715	29921
Oran	69041ms	20442	13577
Sidi bel abbes	23695ms	7581	5113

Tableau 4.1 : La création de la topologie des trois villes d'expérimentation.

c. Présentation de l'application cartographique :

c.1. L'application cartographique « serveur » :

D'après la figure (Fig : 4.1), l'interface de cartographie en ligne permettant l'intégration et l'orchestration des fonctions de deux serveurs et un système de gestion de base de données (SGBD) [173].

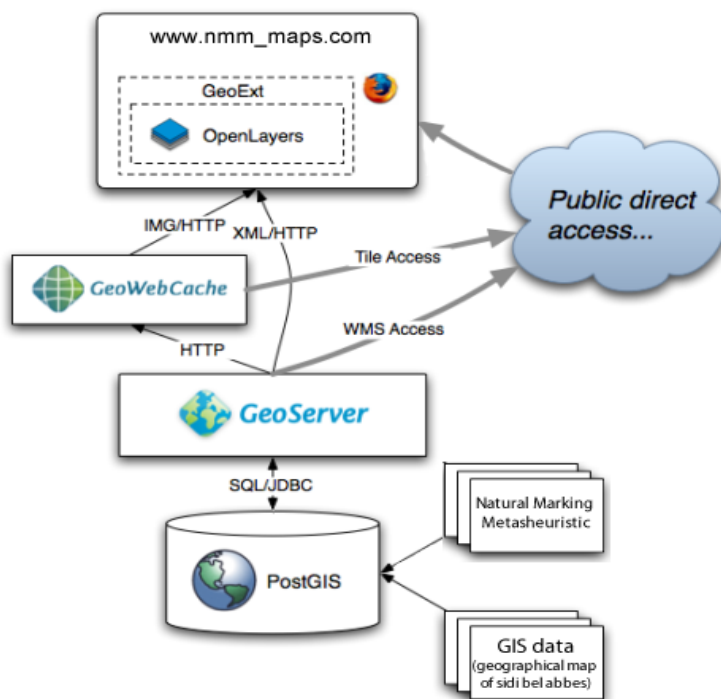


Figure 4.1: Les architectures utilisées dans le projet PgRouting [174] (Image modifiée).

Il est donc possible de distinguer :

- La base des données géographiques Postgres/Postgis : PostgreSQL™ est un système de gestion de bases de données relationnelles objet (ORDBMS) fondé sur POSTGRES¹², Version 4.2™. Ce dernier a été développé à l'université de Californie au département des sciences informatiques de Berkeley. POSTGRES est à l'origine de nombreux concepts qui ne seront rendus disponibles au sein de systèmes de gestion de bases de données commerciaux que bien plus tard.

Il a plus de 15 ans de phase de développement actif et une architecture éprouvée qui lui a valu une solide réputation de fiabilité, d'intégrité des données et d'exactitude. Et grâce à sa licence libérale, PostgreSQL™ a été développé par une équipe mondiale de bénévoles¹³.

PostgreSQL n'est contrôlé par aucune société ou autre entité privée [176]. Son code source

¹² Le projet POSTGRES™, mené par le professeur Michael Stonebraker,. Le développement de POSTGRES™ a débuté en 1986. Plusieurs versions majeures de POSTGRES™ ont vu le jour. La plupart des versions qui suivirent, jusqu'à Postgres95™, portèrent sur la portabilité et la fiabilité. POSTGRES™ fut utilisé dans plusieurs applications, en recherche et en production. le projet POSTGRES™ de Berkeley se termina officiellement avec la version 4.2 [175].

¹³ PostgreSQL a été créé à UCB par un professeur d'informatique nommé Michael Stonebraker. Stonebraker a lancé Postgres en 1986 en tant que projet de suivi de son prédécesseur, Ingres, qui appartient désormais à Computer Associates [176].

peut être utilisé, modifié et distribué librement, quel que soit le but visé, qu'il soit privé, commercial ou académique [177].

PostgreSQL fonctionne sur tous les principaux systèmes d'exploitation, y compris Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) et Windows. Il prend en charge le texte, les images, les sons et la vidéo, et comprend des interfaces de programmation pour C/C++, Java, Perl, Python, Ruby, Tcl et Open Database Connectivity (ODBC) [176].

PostgreSQL prend en charge quatre langages de procédure standard, ce qui permet aux utilisateurs d'écrire leur propre code dans n'importe lequel des langages et il peut être exécuté par le serveur de base de données PostgreSQL. Ces langages procéduraux sont - PL/pgSQL, PL/Tcl, PL/Perl et PL/Python. En outre, d'autres langages procéduraux non standard tels que PL/PHP, PL/V8, PL/Ruby, PL/Java, etc., sont également pris en charge [176].

Les utilisateurs de PostgreSQL peuvent définir des fonctions internes écrites en C (ou dans un langage pouvant être rendu compatible avec C, comme le C++). Ces fonctions sont compilées en objets dynamiques chargeables (encore appelés bibliothèques partagées) et sont chargées par le serveur à la demande [178]. Ou encore utiliser la bibliothèque libpqxx, qui est l'API client C ++ officielle pour PostgreSQL. Le code source de libpqxx est disponible sous la licence BSD [176].

PostGis permet la manipulation d'informations spatiales sous forme de géométries avec le SGBD PostgreSQL. L'extension a été proposée pour la première fois en 2001 dans sa version 8.1 sous une licence GPL. Elle permet le stockage des objets géographiques conformément aux spécifications OGC (Point, Multipoint, Linestring, Multilinestring, Polygon, Multipolygon et GeometryCollections) sous les formats WKT/WKB. PostGis propose des fonctions spécifiques pour faire des requêtes SQL spatiales. Il permet de stocker les géométries en dimension 2D, 3D, ou 4D [179]. PostgreSQL/PostGis est considéré le leader des SGBD libre pour les données géographiques est supporté par des organismes institutionnels comme IGN, IRSN¹⁴, et JRC¹⁵ [180].

Le SGBD PostgreSQL associé à son extension PostGIS qui permet la centralisation des données géographiques diffusées et produites vers et depuis la plateforme. Il permet la gestion des commentaires ainsi que les opérations d'administration du site Web ;

¹⁴ IRSN : Institut de Radioprotection et de Sûreté Nucléaire.

¹⁵ JRC: Joint Research Center–Union Européenne.

- Apache est le logiciel de serveur Web le plus utilisé actuellement. En effet, ce dernier est installé sur plus de 46 % des serveurs Web actifs. Il consiste à servir des sites Web sur internet. Pour atteindre cet objectif, il agit comme un intermédiaire entre le serveur et les machines des clients. Il extrait le contenu du serveur sur chaque requête d'utilisateur et le transmet au Web [181]. Apache est conçu pour prendre en charge de nombreux modules : interprétation du langage (Perl, PHP, Python et Ruby, serveur proxy, Common Gateway Interface, Server Side Includes), réécriture d'URL, négociation de contenu, protocoles de communication additionnels, etc. [182].
- Le serveur cartographique Geoserver, un open source qui permet la gestion de différents formats de données SIG et il est très accessible, vu l'abondante littérature qui existe sur le logiciel. Geoserver est retenu pour sa compatibilité avec les normes de l'OGC (supporte les standards de l'OGC que sont : WMS (Web Map Service), WFS (Web Feature Service) et WCS (Web Coverage Service)), il offre toutes les opérations obligatoires associées aux normes WMS, WFS et WCS et il permet d'utiliser des bibliothèques préconçues comme GeoTools ou GeoAPI [183]. Ce serveur centralise l'information géographique afin de permettre sa diffusion et sa représentation graphique et il récupère les objets géographiques contextuels du SGBD PostgreSQL/PostGis [173].

Le Web Map Service (WMS) permet de produire des cartes sous la forme d'images. Ces images sont principalement diffusées sous la forme des tuiles Rasters (PNG, GIF ou JPG). Les images produites sont géoréférencées, ce qui permet leur superposition et de construire ainsi une carte. Certains serveurs de WMS permettent également la reprojection des données spatiales, c'est-à-dire la création d'images dans une projection autre que celle des données d'origine. La plupart des serveurs WMS acceptent en entrée des données aux formats standards SIG (SHP, TAB) ou encore s'appuient sur des systèmes de gestion de base de données (SGBD) tels que PostGIS (<http://postgis.refractions.net/>) ou Oracle Spatial [184]. Il existe de nombreux serveurs de WMS [185]. Les plus célèbres et aussi les plus couramment utilisés sont MapServer et GeoServer.

c.2. L'application cartographique « cliente » :

L'affichage des données géographiques — mises à disposition par le serveur cartographique — sur l'interface se fait au travers de la librairie orientée objet OpenLayers basée sur JavaScript et AJAX. Elle offre un canevas graphique simplifiant l'implémentation de l'interface graphique et la sémiologie graphique des données SIG [184].

OpenLayers est un logiciel libre constitué d'une bibliothèque de fonctions JavaScript. Il est basé sur la technologie AJAX et permet la mise en place d'applications cartographiques. OpenLayers offre la possibilité d'afficher des fonds cartographiques tuilés ainsi que des marqueurs provenant d'une grande variété de sources de données. Aussi, il permet d'inter-opérer avec de nombreux standards OGC et formats de données normalisés [183]. Notre site utilise OpenLayers avec les bibliothèques JavaScript ExtJS¹⁶ et GeoExt¹⁷. En ce qui concerne ExtJS, elle intègre des nombreux composants de qualité qui permettent de concevoir des formulaires avancés, des menus et barres d'outils, des panels et boîtes de dialogues, la gestion d'onglets ou encore la création de graphiques. GeoExt quant à lui est chargé de jouer le lien entre OpenLayers et ExtJS.

Dans l'interface du client (Fig 4.2), le fond cartographique de la ville de Sidi Bel Abbes est fourni au choix par Google Map, Openstreetmap et Yahoo Maps.

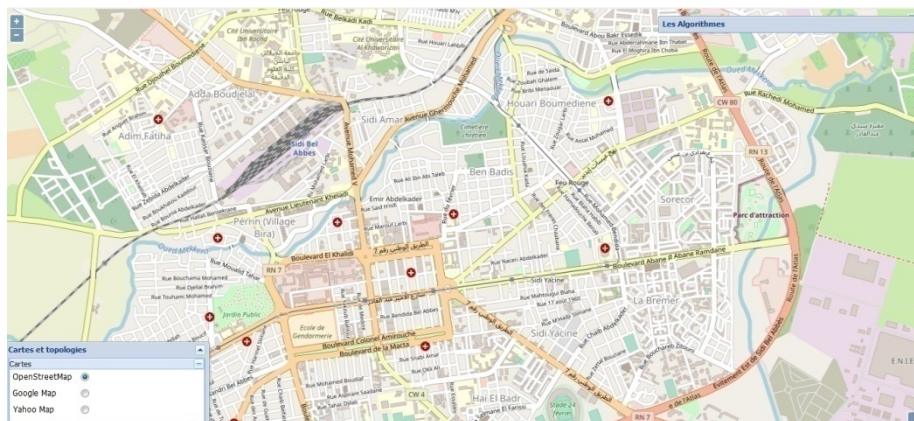


Figure 4.2 : L'interface client « carte de sidi bel abbes » (la carte Raster de openstreetmap.org).

Dans l'interface suivante (Fig : 4.3), l'utilisateur peut consulter la topologie de cette ville fournie par le serveur « Geoserver » et le SGBDG PostGis.

¹⁶ ExtJS est un framework JavaScript de Sencha permettant de construire des applications web interactives. Il possède l'une des plus grandes bibliothèques de composants d'interface modulaire pré-construits.

¹⁷ GeoExt est une librairie JavaScript destinée aux applications web avancées. GeoExt rassemble les connaissances géospatiales de OpenLayers avec une interface utilisateur issue de Ext JS pour aider à construire une application Web ressemblant à une application bureautique [186].



Figure 4.3: La topologie de la ville de sidi bel abbes.

Dans la (Fig : 4.4), nous avons appliqué l'algorithme proposé pour trouver le chemin le plus court « multiobjectif » entre deux sommets (emplacements) dans la carte. L'itinéraire trouvé est en bleu.

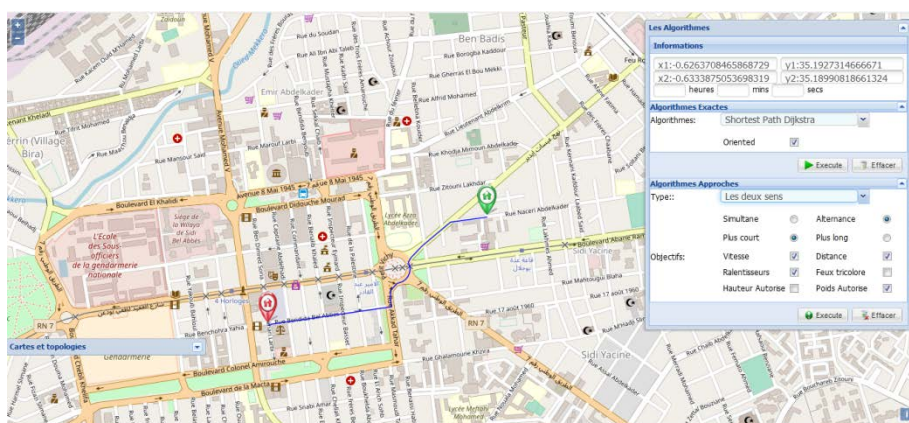


Figure 4.4: Le chemin le plus court « multiobjectif » dans la ville de Sidi Bel Abbès.

c.3. La programmation de la métaheuristique proposée :

Les algorithmes de Pgrouting sont programmés sous l'environnement Visual C++ de Microsoft et pour gérer les graphes, ils ont utilisé la bibliothèque de C++ Boost Graph Library.

Notre produit est développé dans le même environnement, d'un côté pour exploiter les algorithmes de Pgrouting pour l'évaluation et d'autres côtés pour mettre tous les algorithmes dans les mêmes conditions :

- Microsoft Visual C+++ est un environnement de développement qui est utilisé pour créer des applications Windows en langages de programmation C, C++ et C++/CLI et intégrant différents outils pour développer, compiler, déboguer un programme en C++ s'exécutant sur

Windows, ainsi que des bibliothèques comme les MFC¹⁸. Il a par la suite été intégré au Framework Visual Studio, qui constitue ainsi un cadre unique aux divers environnements de développements de Microsoft. Le terme de Visual C++ est toutefois toujours employé pour désigner l'ensemble constitué par Visual Studio configuré pour C et C++ [187]. L'environnement de programmation inclut l'accès à de nombreuses bibliothèques de code partagées qui permettent aux développeurs d'utiliser du code déjà sous Windows au lieu d'avoir à l'écrire à partir de zéro. Microsoft Visual C++ est disponible, en fait, en différentes versions, reconnaissables à l'année de sortie et au fait qu'elles sont destinées aux processeurs x86 (32 bits) et x64 (64 bits) [188]. Dans notre cas nous avons choisi la version 10 Professionnel puisqu'il est disponible dans le laboratoire de recherche ;

- Boost Graph (BGL) est une bibliothèque permettant de créer et manipuler des graphes propose une interface standard pour manipuler des graphes. Cette interface est similaire aux conteneurs de la bibliothèque standard et permet l'accès aux différents éléments à l'aide d'itérateurs. Il est donc possible d'utiliser les algorithmes de la bibliothèque standard pour travailler sur les graphes ou d'utiliser les algorithmes spécifiques pour les graphes fournis par Boost Graph. On peut aussi manipuler les éléments du graphe, accessibles via la classe de traits : les sommets (ou nœuds, vertex dans BGL), les arcs (pour un graphe orienté ou arêtes pour un graphe non orienté, edge dans BGL) et le graphe lui-même (graph dans BGL) [189]. Boost Graph (BGL) permet de résoudre les problèmes de routage dans les graphes et propose plusieurs algorithmes (Disjksta , A star, ...ect), cette bibliothèque de C ++ permet de manipuler les éléments du graphe, accessibles via la classe de traits : les sommets (ou nœuds, vertex dans BGL), les arcs (edge dans BGL) et le graphe lui-même (graph dans BGL). Boost Graph (BGL) permet l'accès aux différents éléments à l'aide d'itérateurs [190]. Dans notre solution proposée, le chargement des données dans les graphes est un facteur important dans l'évaluation de l'algorithme, tant qu'il est réduit, l'algorithme sera meilleur. Les expériences (Tab.1) sur un graphe important (le réseau urbain de la ville de Sidi Bel Abbes) ont montré que la programmation avec le BGL est beaucoup meilleure que la programmation avec les classes et les pointeurs (Tab 4.2).

¹⁸ Les MFC sont une bibliothèque de classes C++ qui encapsulent les différentes fonctions, notamment de gestion des fenêtres ou de messages, propres à Windows. On dit aussi que les classes MFC encapsulent l'API (Application Programming Interface) Windows [191].

Edges	Création d'un graphe avec "classe/pointeur" en Ms	Création d'un graphe avec "BGL" en Ms
50	62	16
100	78	16
500	359	32
1000	811	62
2000	2668	140
4000	7535	281
6000	26177	421
7000	23666	514

Tableau 4.2 : Le temps nécessaire pour créer un graphe avec les techniques de programmation.

c.4. Installer l'algorithme dans le noyau de Postgresql :

Dans Windows, une bibliothèque de liens dynamiques (DLL) est un type de fichier exécutable qui agit comme une bibliothèque partagée de fonctions et de ressources. La liaison dynamique est une fonctionnalité du système d'exploitation. Il permet à un exécutable d'appeler des fonctions ou d'utiliser des ressources stockées dans un fichier séparé. Ces fonctions et ressources peuvent être compilées et déployées séparément des exécutables qui les utilisent [192]. Une DLL est une sorte de programme indépendant de toute application, non exécutable, mais contenant de nombreuses fonctions accessibles depuis un programme exécutable (ou une autre DLL). On les associe généralement à des librairies de fonctions, un peu comme les librairies statiques (LIB), à la seule différence que celles-ci n'ont pas besoin d'être liées lors de la compilation. La DLL est installée une fois pour toutes en mémoire, lors de son premier chargement par une application. En utilisant Visual C++ (Microsoft Visual Studio .Net 2010), notre algorithme a été converti en dll et installé dans le dossier : \PostgreSQL\9.1\lib et les requêtes Sql de configuration sont copiées dans : \PostgreSQL\9.1\share\extension.

PostgreSQL supporte de nombreux langages de programmation pour l'écriture des fonctions utilisateur, mais SQL et PL/pgSQL sont les plus courants. Nous avons rédigé nos propres fonctions dites « Wrapper¹⁹ » en PL/pgSQL, adaptée à notre structure de données diminue la nécessité de passer par des requêtes SQL encombrantes.

¹⁹ Un Wrapper est une fonction de PostgreSQL comme une autre, qui contient des fonctionnalités de pgRouting et PostGIS sous une forme plus facile à utiliser que lors de l'appel des fonctions pgRouting, PostGIS ou utilisateurs directement [193].

3. Résultats et interprétations :

3.1. Présentation du lieu d'implémentation:

Les expérimentations ont été appliquées sur des données réelles de la ville de SIDI BEL ABBES, extraites de la base de données géographique de www.openstreetmap.com.

Cette ville est représentée par un graphe orienté de 7581 arcs (segments) et 5113 sommets (nœuds). Chaque arc est pondéré par plusieurs poids (la distance, le nombre de ralentisseurs, le nombre de feux tricolores, la hauteur autorisée, le poids autorisé... etc.) entre ses deux extrémités.

Inspiré des travaux de PgRouting, l'algorithme est programmé en C++ de Microsoft Visual Studio 2010, puis il est converti en DLL et ajouté au noyau de Postgresql.

Nous avons développé une application permettant de proposer un système d'aide à la décision. Pour ce faire, nous avons construit une base de données (Postgresql) composée d'un réseau routier basée sur une cartographie (www.openstreetmap.com) utilisant des données géographiques réelles de la commune de SIDI BEL ABBES (Fig : 4.5).

Pour évaluer cet algorithme, nous avons créé une liste de 1000 routes (seulement les nœuds de départ et les nœuds d'arrivée choisis en aléatoire — on a utilisé la fonction `rand()` de c++ —), puis nous avons étudié les 2000 itinéraires obtenus après la recherche. Notons que l'algorithme cherche les itinéraires qui relient le nœud de départ au nœud d'arrivée ou/et le nœud d'arrivée au nœud de départ (dans les deux sens).

Dans le rapport de la recherche, l'algorithme enregistre le temps consommé — la fonction `GetTickCount()` de c++ est utilisée —, la longueur de la solution optimale, le meilleur compromis entre les objectifs et d'autres informations de déroulement.

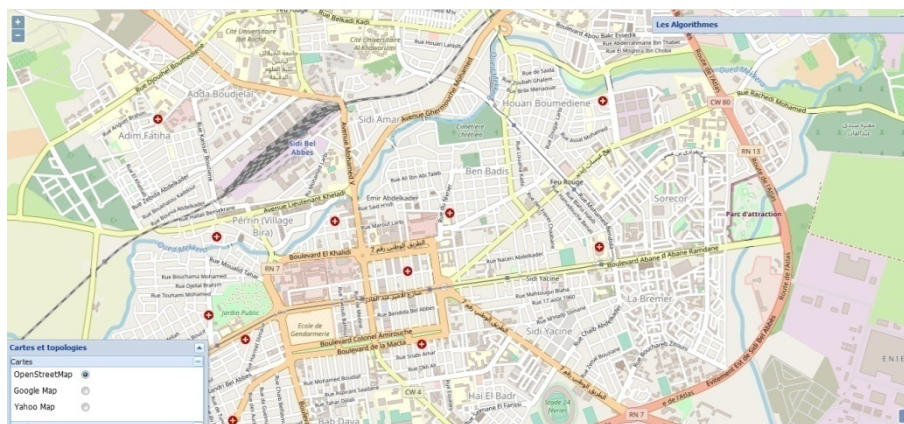


Figure 4.5 : La ville de Sidi Bel Abbès.

La topologie de la ville de SIDI BEL ABBES est créée par la requête SQL :

```
SELECT pgr_createTopology('ways', 0.00001, 'the_geom', 'gid');
```

- « Ways » : est la table des rues de la ville ;
- « 0.00001 » : la tolérance ;
- « the_geom » et « gid » : deux champs dans la table « Ways », le premier champ est réservé pour les données géométriques et le deuxième garde les identifiants des rues.

La topologie crée une table nommée « ways_vertices_pgr », qui groupe tous des nœuds de la carte, cette table est représentée dans le serveur cartographique Geoserver sous forme des points « en rouge dans la Fig : 4.6 », ces derniers se sont les intersections des segments et des rues.



Figure 4.6 : Les nœuds de la ville de Sidi Bel Abbès.

Dès que la table « Ways » est corrigée après l'application de la fonction de la topologie `pgr_createTopology`, les rues sont représentées sous Geoserver sous la forme d'une grille en « bleu » (Fig : 4.7):

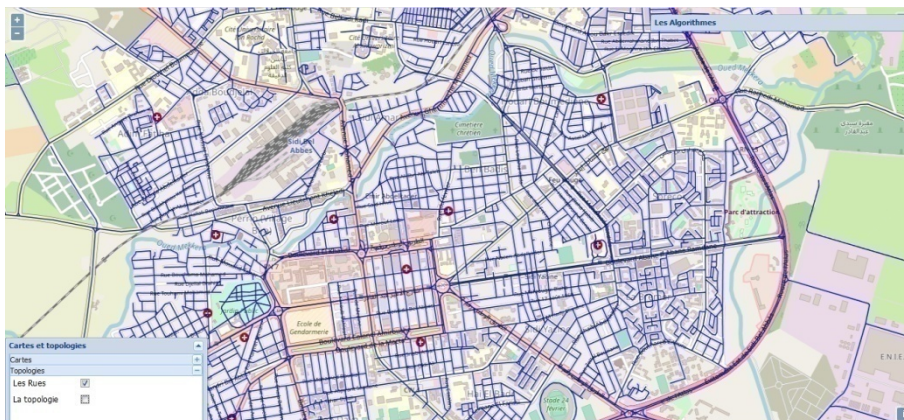


Figure 4.7 : Les segments de la ville de Sidi Bel Abbès.

3.2. Estimation de la base des données :

Le volume de la base de données vide du réseau routier urbain national algérien (seulement les structures des fichiers) est de 1950 octets, et son volume plein avec des données du réseau national est d'environ 20.075.350.681 octets \simeq 18.9 Géga octet.

3.3. Interprétations des résultats :

L'évaluation dépend de deux facteurs importants et qui sont :

- L'optimalité : est le chemin le plus court entre la source et la destination et le contraire (Fig : 4.8) ;
- La rapidité de l'algorithme : le temps nécessaire pour trouver une solution optimale.

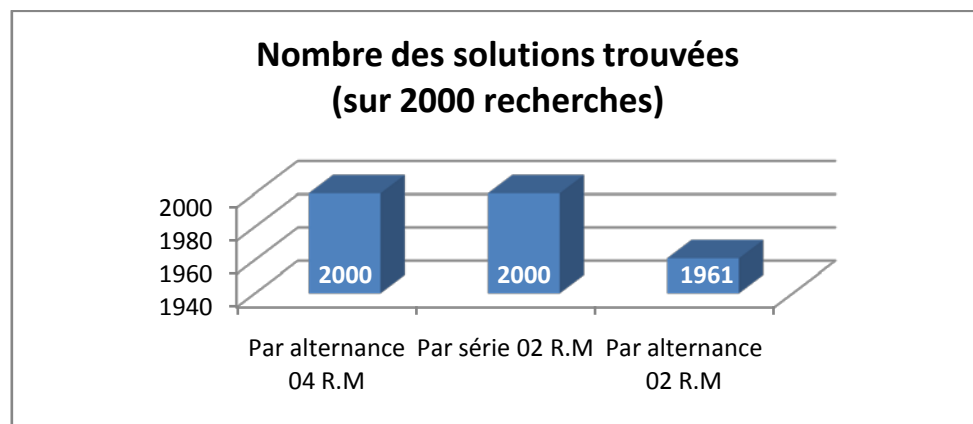


Figure 4.8 : Nombre des solutions trouvées par les trois techniques.

Les résultats de l'optimalité entre la technique séquentielle et la technique par alternance, nous a montré (Fig : 4.8) :

- Les deux techniques (la technique séquentielle de 02 rubriques de marquage et la technique de 04 rubriques de marquage) ont devancé la technique par alternance de 02 rubriques de marquage dans 2000 recherches, cette dernière a échoué dans 69 recherches.

Par contre, dans les 2000 recherches, la méthode par alternance de 04 rubriques de marquage a devancé les autres dans 1324 recherches, la technique par alternance est plus rapide dans 579 recherches, mais ils existent comme même des itinéraires où la technique séquentielle était plus

rapide (23 recherches). Il y a des cas où les trois techniques ont consommé le même temps pour donner des résultats (Fig : 4.9).

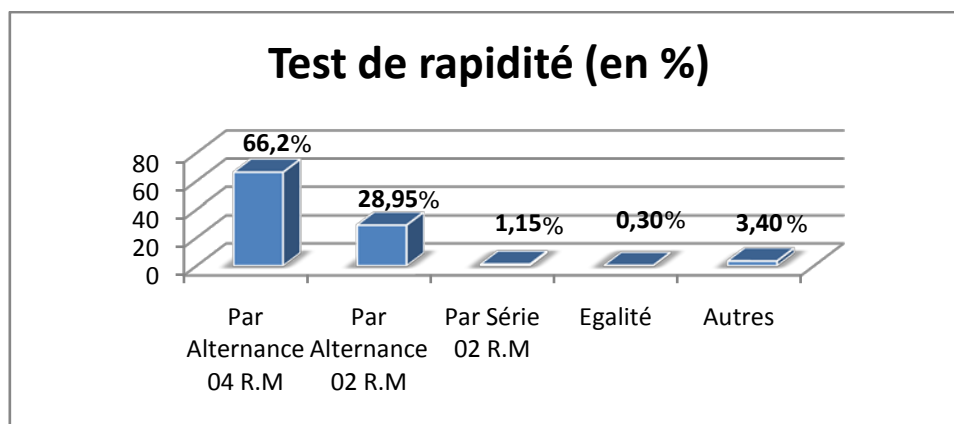


Figure 4.9 : La rapidité entre les deux techniques.

Bien que la technique par alternance de 02 rubriques de marquage soit plus rapide par rapport à la technique séquentielle, mais elle provoque des saturations et blocages du réseau. Cette situation est due par la concurrence des individus dans les rues, les expérimentations nous ont montré que le réseau est saturé 39 fois dans 2000 itinéraires (Fig : 4.10). Cette saturation a paralysé complètement l'algorithme.

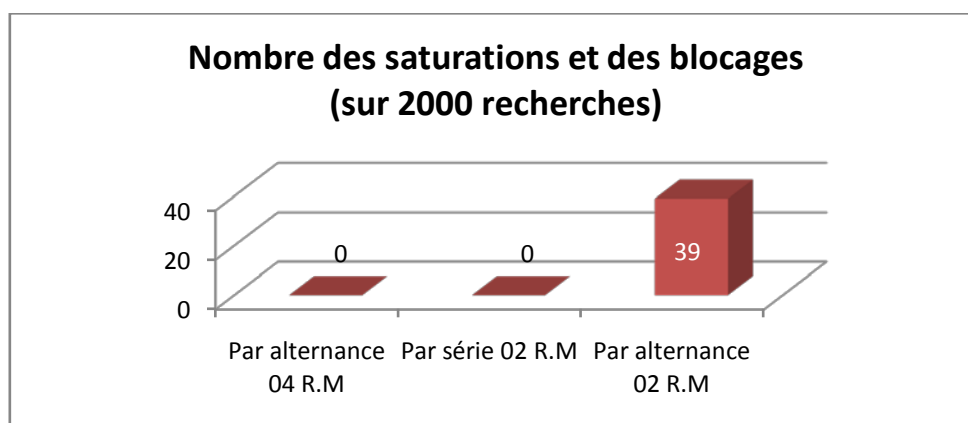


Figure 4.10 : Les nombres de saturation du réseau dans la technique par alternance.

Pour bien évaluer les solutions trouvées par cet algorithme, nous avons comparé nos résultats avec les résultats de l'algorithme de dijkstra qui est très connu dans l'optimisation des itinéraires.

Toutes les solutions de la technique séquentielle ou la technique par alternance de 04 rubriques de marquage sont correctes, optimales et égalent les résultats obtenus par l'algorithme de dijkstra. Par contre, la technique par alternance de 02 rubriques de marquage a échoué dans la recherche des itinéraires optimaux dans plusieurs cas (1.95 % des recherches) (Fig : 4.11).

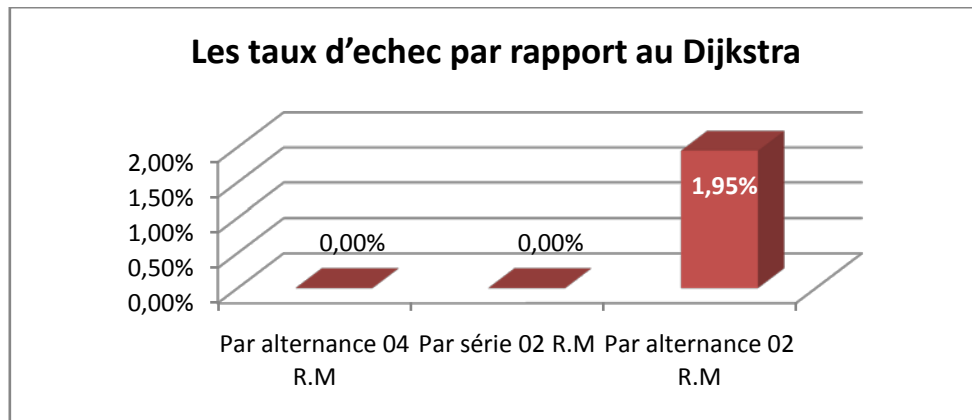


Figure 4.11 : Les taux d'échec de la méthode par alternance 02 rubriques de marquage.

Dans la figure suivante (Fig : 4.12), nous avons testé le temps d'exécution des programme avec des nombres différents des objectifs, nous avons commencé par 1, 2, 3..., jusqu'à 10 objectifs, les résultats montrent que la courbe du temps d'exécution est en croissance tant qu'on a plus des objectifs.

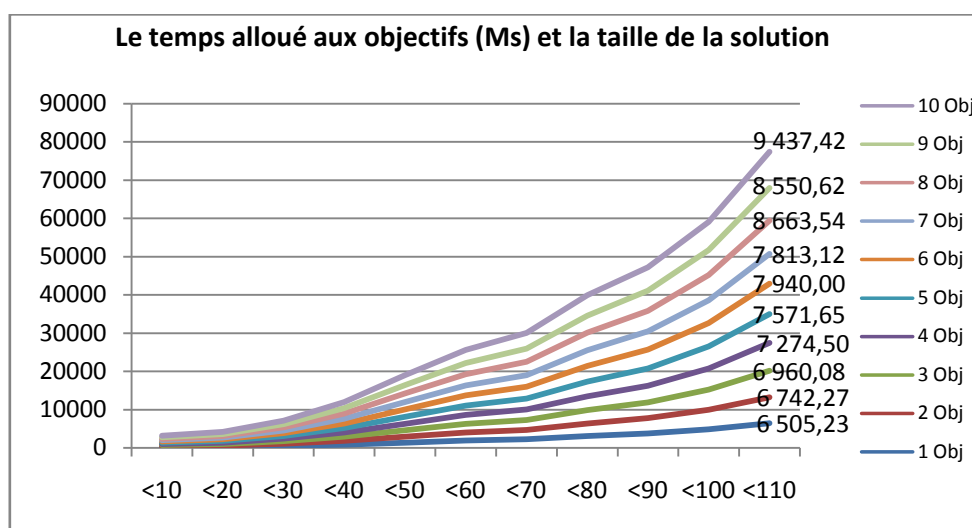


Figure 4.12 : Influence du nombre des objectifs sur le temps d'exécution.

D'après les résultats obtenus, le temps d'exécution est influencé par la taille de la solution et par le nombre des objectifs à traiter. Dans la figure, l'algorithme a donné une solution optimale (sa taille est moins de 10 arcs) avec un seul objectif dans environ 6503 ms, par contre, il a consommé 9437 ms pour donner une solution (sa taille est moins de 110 arcs) avec un compromis de 10 objectifs.

Après les résultats obtenus ci-dessous, la technique par alternance (04 rubriques de marquage) a prouvé son efficacité (l'optimalité et la rapidité des réponses) pour résoudre le problème du plus court chemin dans un graphe orienté et pondéré par rapport aux autres techniques proposées.

4. Conclusion :

L'optimisation des chemins est basée sur le schéma du réseau routier urbain, planifiant le meilleur processus de chemin. C'est le cœur de la recherche de chemin le plus court. Le sac d'algorithmes de PgRouting basé sur PostgreSQL et PostGIS fournit une fonction d'analyse de chemin qui comprend les algorithmes classiques Dijkstra, Shooting A * et Shooting *. Il fournit également la solution du problème du voyageur de commerce, le calcul de la distance parcourue et des problèmes urbains ciblés tels que le trafic à sens unique, les feux de signalisation dans un support puissant.

Afin de valider le modèle théorique de l'algorithme proposé et présenté au troisième chapitre, ce chapitre est donc dédié à l'implémentation de cet algorithme permettant de proposer un système d'aide à la décision. Pour ce faire, nous avons construit une base de données d'un réseau routier urbain réel d'une ville algérienne (Sidi Bel Abbès). Inspirés des applications du projet PgRouting, nous avons également construit notre base de données sous le système de gestion des bases de données Postgresql et son extension géographique PostGis, sur les paramètres d'entrée du modèle notamment les coûts (la distance, le nombre de ralentisseurs, le nombre de feux tricolores, la hauteur, le poids autorisé sur les ponts... etc.).

La façon de trouver des solutions joue un rôle très important, soit dans la qualité de la solution, soit dans la rapidité des réponses, l'algorithme a proposé deux techniques de recherche (séquentielle et par alternance) avec 02 rubriques de marquage, mais la méthode de recherche par alternance avec 04 rubriques de recherche a des données des résultats meilleurs et rapides.

*Conclusion générale et
perspectives*

L'objectif de cette thèse a été l'optimisation d'itinéraires pour des véhicules dans les réseaux routiers urbains d'une ville algérienne. Pour le calcul de ces itinéraires, la recherche bibliographique a montré qu'il existe de nombreux algorithmes de calcul de plus court chemin, mais il reste encore une certaine marge de progression en ce qui concerne la modélisation d'un réseau routier urbain réel et de l'utilisation des techniques d'optimisation « multiobjectifs ».

La modélisation des systèmes d'information géographique (S.I.G), est une tâche rigoureuse qui demande beaucoup de précision quant à l'organisation des données, leur nature et leur forme de représentation. Cette modélisation est considérée comme le cœur et le centre de masse d'un SIG, ainsi que l'analyse spatiale qui représente la raison d'être de la SIG, dans laquelle réside sa puissance. La divergence qui existe, entre le milieu extérieur qui alimente les SIG par les données et le milieu intérieur d'un SIG, qui intègre, stocke et gère l'information géographique, est reflétée dans la modélisation par l'existence de deux modes différents de représentations des données spatiales, dont le mode Raster est dicté par un milieu externe, qui est la plus grande source d'acquisition des données et le mode vectoriel le plus répandu et plus avantageux, car il représente l'information géographique de façon concise et appropriée aux besoins des SIG. D'un autre côté, l'importance de l'analyse spatiale qui représente une spécificité des SIG, réside dans sa capacité à satisfaire les différents besoins de l'utilisateur en termes de production de l'information géographique utile à partir des données acquises [42]. Tous concepts que nous avons expliqués dans le premier chapitre.

Dans le deuxième chapitre, nous avons présenté les nombreuses méthodes de résolution qui existent pour les problèmes « multiobjectifs ». Elles correspondent à des situations et des problématiques différentes. Plusieurs classements de ces méthodes sont proposés dans la littérature, la première classe les méthodes suivant leur façon de traiter les fonctions « objectifs ». Par contre, la deuxième adopte le point de vue « décideur ». Les métaheuristiques et en particulier les algorithmes évolutionnaires, travaillant sur des populations de solutions, sont bien adaptées à l'optimisation « multiobjectifs ».

Le marquage de territoire est un acte naturel chez la plupart des animaux, il indique la possession d'une zone de chasse, de vivre, d'attirer une femelle et de dissuader un autre male de pénétrer chez lui. L'homme marque aussi ses propriétés comme les autres animaux territoriaux par

les nombreux moyens comme les portes fermées à clé, grillages, documents notariaux... etc., pour marquer sa propriété personnelle et pour les propriétés collectives il utilise les frontières, les drapeaux, les plaques de signalisation... etc. Inspiré de ce comportement naturel, notre algorithme utilise des individus qui circulent et marquent leurs trajets dans le graphe d'une ville et dans des sens opposés à la recherche de « bons » chemins. L'exploitation de ce marquage des individus pour résoudre des problèmes d'itinéraires « multiobjectifs » est le cœur de cette métaheuristique et le centre d'intérêt du troisième chapitre.

Et dans le dernier chapitre, nous avons mis en place un modèle du plus court chemin « multiobjectifs » ayant pour objectif d'aider au choix d'un « plus court chemin multiobjectifs » dans un réseau routier urbain. Dans notre cas, « plus court chemin » signifie un chemin avec moins d'impacts environnementaux (feux tricolores), sécuritaires (Hauteur autorisée, poids autorisé, ralentisseurs) et garantissant des critères économiques (distance, temps) satisfaisants. Nous avons ainsi fourni un système d'aide à la décision pour les décideurs qui souhaitent trouver le meilleur compromis entre ces impacts lors du choix d'itinéraire. Le système permet de se baser sur un certain nombre de données d'entrée pour fournir un itinéraire répondant aux exigences du décideur.

La perspective la plus immédiate est l'amélioration des performances de cette solution dans le temps d'exécution. En effet, en routage mono-objectif, pour minimiser le temps de parcours, il est possible de calculer un itinéraire à travers la planète entière en environ une milliseconde. Nous sommes encore très loin d'avoir de telles performances dans le routage « multiobjectifs ».

Bibliographie

- [1] I. Jguirim, "Modélisation et génération d'itinéraires contextuels d'activités urbaines dans la ville", Université de Bretagne occidentale Brest, 2016.
- [2] S. FEZZAI, *L'usage des espaces urbains: Théories et notions de base*: E-Kutub Ltd, 2018.
- [3] M. Bresso and A. Haurie, *Gestion de l'environnement et entreprise* vol. 12: PPUR presses polytechniques, 1996.
- [4] CNRTL, *URBAIN : Définition de URBAIN*. <https://cnrtl.fr/definition/urbain>, Consulté le: 11/08/2020.
- [5] INSEE, Unité urbaine selon l'INSEE dans la Somme - data.gouv.fr. <https://www.data.gouv.fr/fr/datasets/unite-urbaine-selon-linsee-dans-la-somme>, Consulté le: 20/12/2020
- [6] S. Le Feon, "Evaluation environnementale des besoins de mobilité des grandes aires urbaines en France-Approche par Analyse de Cycle de Vie", École Nationale Supérieure des Mines de Saint Étienne, 2014.
- [7] M. C. BIGNOLI Claude A. (2020). *Savoirs de base en sécurité routière FICHE n°08 août 2006*. <https://www.cerema.fr/fr/centre-ressources/boutique/savoirs-base-securite-routiere>, Consulté le: 11/08/2020.
- [8] A. N. Française, *Question n°70084*, . <http://questions.assemblee-nationale.fr/q14/14-70084QE.htm> , Consulté le: 11/08/2020
- [9] R. Alain and R.-D. Josette, *Le nouveau petit Robert de la langue française 2017, Dictionnaire alphabétique et analogique de la langue française* Dictionnaires Le Robert, 2017.
- [10] Wikipedia.org, *Réseau routier — Wikipédia*. https://fr.wikipedia.org/w/index.php?title=R%C3%A9seau_routier&oldid=173695663, Consulté le: 10/08/2020
- [11] Thesaurus.gouv.qc.ca, *Fiche du terme : Réseau routier - Thésaurus de l'activité gouvernementale*. <http://www.thesaurus.gouv.qc.ca/tag/terme.do?id=11037>, Consulté le: 11/08/2020
- [12] R. Brunet, H. THERY, and R. FERRAS, *Les mots de la géographie, Dictionnaire critique: Collection Dynamiques du territoire*, RECLUS-La documentation Française, Paris, 1992.
- [13] M. J. De Smith, M. F. Goodchild, and P. Longley, *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*: Troubador publishing ltd, 2007.
- [14] G. Nicolas, *Quatrièmes rencontres de Théo Quant: Besançon, 11 et 12 février 1999*: Presses Univ. Franche-Comté, 2001.
- [15] C. MORENCY, "Étude de méthodes d'analyse spatiale et illustration à l'aide de microdonnées urbaines de la Grande Région de Montréal", *Les Cahiers scientifiques du transport*, 2006.
- [16] T. C. Bailey and A. C. Gatrell, *Interactive spatial data analysis* vol. 413: Longman Scientific & Technical Essex, 1995.
- [17] N. Cressie, "Statistics for Spatial Data, revised edition, Wiley Series in Probability and Statistics", in *Applied Probability and Statistics*, ed: Wiley & Sons New York, 1993.
- [18] H. J. Miller, "What about people in geographic information science", *Re-presenting geographic information systems*, pp. 215-242, 2005.
- [19] Wikipedia.org, *Méridien — Wikipédia*. <https://fr.wikipedia.org/w/index.php?title=M%C3%A9ridien&oldid=176753420>, Consulté le: 19/11/2020
- [20] Wikiwand.com, *Méridien de Greenwich | Wikiwand*. https://www.wikiwand.com/fr/M%C3%A9ridien_de_Greenwich, Consulté le: 09/12/2020.
- [21] Survie-et-survivalisme.com, *S'orienter à l'aide d'une carte : les coordonnées géographiques - Guide de survie*. <http://www.survie-et-survivalisme.com/sorienter-a-laide-dune-carte-les-coordonnees-geographiques>, Consulté le: 20/11/2016.
- [22] Wikipedia.org, *Projection de Mercator*. http://zims-fr.kiwix.campusafrika.gos.orange.com/wikipedia_fr_all_maxi/A/Projection_de_Mercator, Consulté le: 09/12/2020.
- [23] Techno-Science.net, *Transverse Universelle de Mercator : définition et explications*. <https://www.techno-science.net/definition/4947.html>, Consulté le: 09/12/2020.
- [24] Bel-horizon.eu, *bel-horizon.eu - Les projections de types UTM*. <https://www.bel-horizon.eu/nos-randonnees/notre-bibliotheque-de-traces-gps/liste-des-traces/files/file/63.html>, Consulté le: 31/12/2020.

- [25] F. Varenne, "Epistémologie des modèles et des simulations", in colloque « Les modèles : possibilités et limites » organisé par la Société Française de Physique à la Bibliothèque Nationale de France, 2008.
- [26] J.-P. Antoni, C. Morency, A. Cambien, G. Vuidel, D. Nguyen-Luong, J.-P. Nicolas, V. Piron, J. Delons, A. Bonnafous, and Y. Crozet, *Modéliser la ville. Formes urbaines et politiques de transport*: Economica, 2011.
- [27] K. Lynch, "The image of the city", *MIT press, Etats Unis*, vol. 11, 1960.
- [28] Wikipedia.org, *Modèle cristallérien — Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Mod%C3%A8le_christall%C3%A9rien&oldid=155175009, Consulté le: 27/12/2018.
- [29] A. R. Bouzgarrou, "Analyse des formes morpho-fonctionnelles urbaines : mise en place d'un indicateur de mutations paysagères de la ville de Monastir entre 1956 et 2013", Université de BRETAGNE occidentale – Université de SOUSSE, 2019.
- [30] J. Laterrasse, "Ville et économie: une histoire qui ne fait que commencer?", in *Annales des Mines-Realites industrielles*, 2008, pp. 12-17.
- [31] P.-H. Derycke, *Penser la ville: théories et modèles*: FeniXX, 1995.
- [32] E. Ducom, "Le modèle des ceintures limitrophes (fringe belts): une application aux villes françaises", Université Rennes 2, 2005.
- [33] F. Bouzahzah, "Dynamique urbaine et nouvelle centralité", Université des frères MENTOURI - CONSTANTINE, 2017.
- [34] G. Morel, "Stabilité et coloration des graphes sans P5", Université de GRENOBLE, 2011.
- [35] J.-G. Fages, "Exploitation de structures de graphe en programmation par contraintes", Université de Nantes Angers Le Mans, 2014.
- [36] F. Viger, "Contributions à la mesure et à la modélisation de la topologie de l'internet", Université PIERRE ET MARIE CURIE - PARIS 6, 2007.
- [37] H. Hajji, "Gestion des risques naturels: une approche fondée sur l'intégration des données", Institut National des Sciences Appliquées de Lyon, INSA-Lyon, 2005.
- [38] C. Schrenk, "Vector and Raster Data Model", *Pinterest*, 2017.
- [39] M. N. DeMers, *"Fundamentals of geographic information systems"*: John Wiley & Sons, 2008.
- [40] A. Bouvier, M. George, F. Le Lionnais, and I. Bros, *Dictionnaire des mathématiques*: Presses universitaires de France, 1979.
- [41] K. Devlin, "Mathematics: The Science of Patterns: The Search for Order in Life", *Mind*, 1994.
- [42] M. Betka, "Etude et mise en oeuvre des passerelles entre les modèles vectoriels topologiques 2D et 3D", Université des sciences et de la Technologie Houari Boumediène, 2015.
- [43] H. Beguin, J. Deconninck, and D. Peeters, "Optimiser la localisation des écoles primaires: le cas de Mouscron, Belgique", *Revue d'économie régionale et urbaine*, pp. 795-806, 1989.
- [44] F. Gandon, "Synthèse sur les SIG et perspectives d'utilisation en hydrologie", ORSTOM Montpellier, 1991.
- [45] C. Regis and S. Dominique, *La géométrie et ses propriétés*. http://www.gitta.info/DigitModel/fr/html/Vector_learningObject1.html, Consulté le: 06/01/2015.
- [46] Q. M. GmbH, *Extension de fichier DXF*. <https://www.online-convert.com/fr/format-fichier/dxf>, Consulté le: 06/09/2020.
- [47] T. K. Peucker and N. Chrisman, "Cartographic data structures", *The American Cartographer*, vol. 2, pp. 55-69, 1975.
- [48] A. De La Losa, "Modélisation de la troisième dimension dans les bases de données géographiques", Université de Marne-la-Vallée, 2000.
- [49] J. L. De Oliveira, F. Pires, and C. B. Medeiros, "An environment for modeling and design of geographic applications", *Geoinformatica*, vol. 1, pp. 29-58, 1997.
- [50] B. Bucher, "L'aide à l'accès à l'information géographique: un environnement de conception coopérative d'utilisations de données géographiques", Université Paris VI-Pierre et Marie Curie, 2002.
- [51] P. Bordin, *SIG: concepts, outils et données*: Hermès Science Publications, 2002.

- [52] A. Da Cunha, *Enjeux du développement urbain durable: transformations urbaines, gestion des ressources et gouvernance*: PPUR presses polytechniques, 2005.
- [53] J. Denègre and F. Salgé, "Les Systèmes d'information géographique, Collection: Que sais-je", *Presses Universitaires de France–PUF*, 1996.
- [54] G. Touya, "Le Modèle CollaGen: collaboration de processus automatiques pour la généralisation cartographique de paysages hétérogènes", 2011.
- [55] J.-F. Guégan and M. Choisy, *Introduction à l'épidémiologie intégrative*: De Boeck Supérieur (De Boeck université), Belgique, 2008.
- [56] N. Nova, *Les médias géolocalisés: comprendre les nouveaux espaces numériques*: Fyp Éditions, France, 2009.
- [57] A. Puricelli, "Réingénierie et Contrôle Qualité des Données en vue d'une Migration Technologique", Lyon, INSA, 2000.
- [58] H. Gumuchian, C. Marois, and V. Fèvre, *Initiation à la recherche en géographie: aménagement, développement territorial, environnement*: PUM, 2000.
- [59] Ubaciis, *Cartographie: les cartes*. <http://ubaciis.free.fr/cartographie-les-cartes.html>, Consulté le: 23/10/2019.
- [60] Wikipedia.org, *SPOT (satellite) — Wikipédia*. [https://fr.wikipedia.org/w/index.php?title=SPOT_\(satellite\)&oldid=176550759](https://fr.wikipedia.org/w/index.php?title=SPOT_(satellite)&oldid=176550759), Consulté le: 13/11/2020.
- [61] C. MOUILAH, "Les méta-heuristiques et le Système d'Information Géographique (SIG)", Université des Sciences et de la Technologie d'Oran Mohamed-Boudiaf (USTOMB), 2013.
- [62] Educalingo.com, *MINIMISATION - Définition et synonymes de minimisation dans le dictionnaire français*. <https://educalingo.com/fr/dic-fr/minimisation>, Consulté le: 30/12/2020.
- [63] B. Bidégaray and L. Moisan, *Petits problèmes de mathématiques appliquées et de modélisation: issus des concours d'entrée à l'Ecole normale supérieure de Cachan* vol. 9: Springer Science & Business Media, 2000.
- [64] Y. Rossier, *Systèmes de gestion industrielle: une étude de cas*: PPUR presses polytechniques, 1991.
- [65] M. Serfati, *De la méthode: recherches en histoire et philosophie des mathématiques*: Presses Univ. Franche-Comté, France, 2002.
- [66] H. C. Camille, "Optimisation pour la resolution des problèmes difficiles par les méthodes classico-métaheuristiques", Université d'ANTANANARIVO, 2017.
- [67] T. AMEUR, "Optimisation du processus d'usinage en utilisant la méthode des essaims de particules", Université Mohamed Khider-Biskra, 2016.
- [68] M. Lachhab, "Proposition d'un outil d'aide à la décision multicritère sous incertitudes à base de colonies de fourmis: une approche intégrée appliquée à la gestion des risques dans les projets d'ingénierie système", Université de TOULOUSE, 2018.
- [69] V. Gardeux, "Conception d'heuristiques d'optimisation pour les problèmes de grande dimension: application à l'analyse de données de puces à ADN", Université de PARIS-ESTCRÉTEIL, 2011.
- [70] H. Zidani, "Représentation de solution en optimisation continue, multi-objectif et applications", Ecole MOHAMMADIA d'ingénieurs et institut national des sciences appliquées de ROUEN, 2013.
- [71] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?", in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 2001, pp. 1042-1049.
- [72] I. Y. Kim and O. De Weck, "Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation", *Structural and multidisciplinary optimization*, vol. 31, pp. 105-116, 2006.
- [73] T. Pinville, "Robotique évolutionniste: influence des pressions de sélection sur l'émergence d'une forme de mémoire interne", Université Pierre et Marie Curie, France, 2013.

- [74] H. Chouaib, "Sélection de caractéristiques: méthodes et applications", Paris Descartes University: Paris, France, 2011.
- [75] C. C. Coello, A. D. Christiansen, and A. H. Aguirre, "Multiobjective design optimization of counterweight balancing of a robot arm using genetic algorithms", in *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, 1995, pp. 20-23.
- [76] A. Charnes and W. W. Cooper, "Management models and industrial applications of linear programming", *Management science*, vol. 4, pp. 38-91, 1957.
- [77] Y. Ijiri, *Management goals and accounting for control* vol. 3: North Holland Pub. Co., 1965.
- [78] L. Duckstein, "Multiobjective optimization in structural design: The model choice problem", ARIZONA university TUCSON dept of systems and industrial engineering, 1981.
- [79] K. L. HADJ BENALI, "Approches de résolution du problème d'ordonnement d'atelier de production", Université des sciences et de la technologie HOUARI BOUMEDIENE, Alger, 2016.
- [80] J.-F. Bérubé, M. Gendreau, and J.-Y. Potvin, "An exact ϵ -constraint method for bi-objective combinatorial optimization problems: Application to the Traveling Salesman Problem with Profits", *European Journal of Operational Research*, vol. 194, pp. 39-50 2009.
- [81] J. D. Schaffer, "Some experiments in machine learning using vector evaluated genetic algorithms", *Vanderbilt University, Nashville, USA*, 1984.
- [82] N. Smairi, "Optimisation par essaim particulière: adaptation de tribes à l'optimisation multiobjectif", Université de Paris Est, 2013.
- [83] M. P. Fourman, "Compaction of symbolic layout using genetic algorithms", in *Proceedings of the 1st international conference on genetic algorithms*, 1985, pp. 141-153 (cité en page 20).
- [84] Z. Wang, "Optimisation avancée pour la recherche et la composition des itinéraires comodaux au profit des clients de transport", Ecole Centrale de Lille, 2015.
- [85] V. Pareto, "Cours d'Economie politique. Tome second", *Rouge editeur, Lausanne*, 1897.
- [86] D. E. Goldberg, "Genetic Algorithms in Search Optimization and Machine Learning", *Addison-Wesley*, p. 126, 1989.
- [87] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization", in *Icga*, 1993, pp. 416-423.
- [88] I. Alaya, "Optimisation multi-objectif par colonies de fourmis: cas des problèmes de sac à dos", université de la MANOUBA - Université CLAUDE BERNARD LYON 1, 2009.
- [89] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary computation*, vol. 2, pp. 221-248 1994.
- [90] D. Goldberg, "Genetic Algorithms in Search, Optimization & Machine Learning (p. 126) Addison-Wesley", *Reading, Mass*, 1989.
- [91] D. S. Weile, E. Michielssen, and D. E. Goldberg, "Multiobjective synthesis of electromagnetic devices using nondominated sorting genetic algorithms", in *IEEE Antennas and Propagation Society International Symposium. 1996 Digest*, 1996, pp. 592-595.
- [92] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations", Air force inst of tech wright-pattersonafb oh school of engineering, 1999.
- [93] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II", in *International conference on parallel problem solving from nature*, 2000, pp. 849-858.
- [94] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE transactions on Evolutionary Computation*, vol. 6, pp. 182-197, 2002.
- [95] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization", in *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*, 1994, pp. 82-87
- [96] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm", *TIK-report*, vol. 103, 2001.

- [97] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach", *IEEE transactions on Evolutionary Computation*, vol. 3, pp. 257-271, 1999.
- [98] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy", *Evolutionary computation*, vol. 8, pp. 149-172, 2000.
- [99] A. Zinflou, "Algorithmes évolutionnaires pour l'ordonnancement industriel: application à l'industrie automobile", Université du Québec à Chicoutimi, 2008.
- [100] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization", in *International conference on parallel problem solving from nature*, 2000, pp. 839-848.
- [101] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization", in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 2001, pp. 283-290.
- [102] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization", *Evolutionary computation*, vol. 10, pp. 263-282, 2002.
- [103] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions", *Evolutionary computation*, vol. 13, pp. 501-525, 2005.
- [104] C. Grosan, "Improving the performance of evolutionary algorithms for the multiobjective 0/1 knapsack problem using/spl epsiv/-dominance", in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, 2004, pp. 1958-1963.
- [105] J. D. Knowles, "Local-search and hybrid evolutionary algorithms for Pareto optimization", University of Reading, UK., 2002.
- [106] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization", in *International conference on evolutionary multi-criterion optimization*, 2007, pp. 742-756.
- [107] Y. Collette and P. Siarry, *Optimisation multiobjectif*: Editions Eyrolles, 2002.
- [108] C. Dhaenens, "Optimisation Combinatoire Multi-Objectif: Apport des méthodes coopératives et contribution à l'extraction de connaissances", Université Lille 1, 2005.
- [109] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization: A review of the state-of-the-art", *IEEE Access*, vol. 6, pp. 41256-41279, 2018.
- [110] Y. Collette and P. Siarry, *Optimisation multiobjectif: Algorithmes*: Editions Eyrolles, 2011.
- [111] F. Glover, "Future paths for integer programming and links to artificial intelligence", *Computers operations research*, vol. 13, pp. 533-549, 1986.
- [112] I. H. Osman and G. Laporte, "Metaheuristics: A bibliography", Springer, 1996.
- [113] S. Voß, S. Martello, I. H. Osman, and C. Roucairol, *Meta-heuristics: "Advances and trends in local search paradigms for optimization"*, Springer Science & Business Media, 2012.
- [114] T. Stützle, "Local search algorithms for combinatorial problems-analysis, algorithms and new applications", *DISKI-Dissertationen zur Künstliken Intelligenz. In x, Sankt Augustin, Germany*, 1999.
- [115] H. Hachimi, "Hybridations d'algorithmes métaheuristiques en optimisation globale et leurs applications", Ecole MOHAMMADIA d'ingénieurs, université MOHAMMED V AGDAL, 2013.
- [116] S. Jacquin, "Hybridation des métaheuristiques et de la programmation dynamique pour les problèmes d'optimisation mono et multi-objectif: application à la production d'énergie", Université de Lille 1, 2015.
- [117] E.-G. Talbi, *"Metaheuristics: from design to implementation"*, vol. 74, John Wiley & Sons, 2009.
- [118] E. Aarts, E. H. Aarts, and J. K. Lenstra, *"Local search in combinatorial optimization"*, Princeton University Press, 2003.
- [119] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search", in *Handbook of metaheuristics*, ed: Springer, 2003, pp. 320-353.

- [120] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies", *Journal of statistical physics*, vol. 34, pp. 975-986, 1984.
- [121] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, pp. 29-41, 1996.
- [122] J. Kennedy and R. Eberhart, "Particle swarm optimization. Encycl", *Mach. Learn*, vol. 4, pp. 760-766, 2010.
- [123] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonsel self discrimination in a computer", in *Proceedings of 1994 IEEE computer society symposium on research in security and privacy*, 1994, pp. 202-212.
- [124] J. O. Kephart, "A biologically inspired immune system for computers", in *In proc. Of the fourth international workshop on synthesis and simulation of living systems, artificial life IV*, 1994.
- [125] J. Holland, "Adaptation in natural and artificial systems: an introductory analysis with application to biology", *Control and artificial intelligence*, 1975.
- [126] F. Manseur, "Algorithmes pour un guidage optimal des usagers dans les réseaux de transport", Paris Est, 2017.
- [127] R. Richard. (2020). NP-difficile . <https://www.24pm.com/117-definitions/447-np-difficile>, Consulté le:23/09/2020
- [128] H. Sullivan, T. R. Bashkow, and D. Klappholz, "A large scale, homogenous, fully distributed parallel machine, II", in *ACM SIGARCH computer architecture news*, 1977, pp. 118-124.
- [129] T. Hain, "An IPv6 provider-independent global unicast address format", *Internet Draft*, 2002.
- [130] L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP)--Version 1 functional specification", *Resource*, 1997.
- [131] S. Floyd and E. Kohler, "RFC 4341-Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control", *Internet Engineering Task Force, Request For Comments*, 2006.
- [132] P. J. Schwartz, *Recherche de chemin par l'algorithme A**, www.developpeur.com, Consulté le: 14/02/2008
- [133] I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck, "A hub-based labeling algorithm for shortest paths in road networks", in *International Symposium on Experimental Algorithms*, 2011, pp. 230-241.
- [134] K. L. Cooke and E. Halsey, "The shortest route through a network with time-dependent internodal transit times", *Journal of mathematical analysis and applications*, vol. 14, pp. 493-498, 1966.
- [135] I. Chabini, "A new shortest path algorithm for discrete dynamic networks", in *Proceedings of the 8th IFAC Symposium on Transport Systems*, 1997, pp. 551-556.
- [136] I. Chabini, "Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time", *Transportation research record*, vol. 1645, pp. 170-175, 1998.
- [137] S. E. Dreyfus, "An appraisal of some shortest-path algorithms", *Operations research*, vol. 17, pp. 395-412, 1969.
- [138] E. Klafszky, "Determination of shortest path in a network with time-dependent edge-lengths", *Statistics: A Journal of Theoretical and Applied Statistics*, vol. 3, pp. 255-257, 1972.
- [139] A. Orda and R. Rom, "Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length", *Journal of the ACM (JACM)*, vol. 37, pp. 607-625, 1990.
- [140] A. B. Philpott, "Continuous-time shortest path problems and linear programming", *SIAM journal on control and optimization*, vol. 32, pp. 538-552, 1994.
- [141] A. B. Philpott and A. Mees, "Continuous-time shortest path problems with stopping and starting costs", *Applied mathematics letters*, vol. 5, pp. 63-66, 1992.
- [142] B.-H. Ahn and J.-Y. Shin, "Vehicle-routeing with time windows and time-varying congestion", *Journal of the Operational Research Society*, vol. 42, pp. 393-400, 1991.

- [143] D. E. Kaufman and R. L. Smith, "Fastest paths in time-dependent networks for intelligent vehicle-highway systems application", *Journal of Intelligent Transportation Systems*, vol. 1, pp. 1-11, 1993.
- [144] S. Pallottino and M. G. Scutella, "Shortest path algorithms in transportation models: classical and innovative aspects", in *Equilibrium and advanced transportation modelling*, ed: Springer, 1998, pp. 245-281.
- [145] M. M. Hizem, "Recherche de chemins dans un graphe à pondération dynamique: application à l'optimisation d'itinéraires dans les réseaux routiers", École centrale de LILLE, 2008.
- [146] R. P. Loui, "Optimal paths in graphs with stochastic or multidimensional weights", *Communications of the ACM*, vol. 26, pp. 670-676, 1983.
- [147] J. F. Bard and J. E. Bennett, "Arc reduction and path preference in stochastic acyclic networks", *Management science*, vol. 37, pp. 198-215, 1991.
- [148] P. B. Mirchandani and H. Soroush, "Optimal paths in probabilistic networks: A case with temporary preferences", *Computers & operations research*, vol. 12, pp. 365-381, 1985.
- [149] I. Murthy and S. Sarkar, "A relaxation-based pruning technique for a class of stochastic shortest path problems", *Transportation science*, vol. 30, pp. 220-236, 1996.
- [150] D. Rasteiro and A. B. Anjo, "Metaheuristics for stochastic shortest path problem", in *Proceedings of 4th MetaHeuristics International Conference (MIC), Porto, Portugal*, 2001.
- [151] M. Jamali, "Learning to Solve Stochastic Shortest Path Problems", *Rapport Technique, Sharif University of Technology*, 2006.
- [152] H. Frank, "Shortest paths in probabilistic graphs", *Operations research*, vol. 17, pp. 583-599, 1969.
- [153] G. Andreatta, "F. Ricaldone, and L. Romeo, Exploring stochastic shortest path problems", *ATTI, Giornale di Lavoro* 1985.
- [154] V. G. Adlakha, "Note—An improved conditional Monte Carlo technique for the stochastic shortest path problem", *Management science*, vol. 32, pp. 1360-1367, 1986.
- [155] J. Kamburowski, "A note on the stochastic shortest route problem", *Operations research*, vol. 33, pp. 696-698, 1985.
- [156] C. E. Sigal, A. A. B. Pritsker, and J. J. Solberg, "The stochastic shortest route problem", *Operations research*, vol. 28, pp. 1122-1129, 1980.
- [157] R. W. Hall, "The fastest path through a network with random time-dependent travel times", *Transportation science*, vol. 20, pp. 182-188, 1986.
- [158] J. L. Bander and C. C. White, "A heuristic search approach for a nonstationary stochastic shortest path problem with terminal cost", *Transportation science*, vol. 36, pp. 218-230, 2002.
- [159] S. Gao and I. Chabini, "Optimal routing policy problems in stochastic time-dependent networks", *Transportation Research Part B: Methodological*, vol. 40, pp. 93-122, 2006.
- [160] I. Chabini, "A new algorithm for shortest paths in discrete dynamic networks", *IFAC Proceedings Volumes*, vol. 30, pp. 537-542, 1997.
- [161] E. D. Miller-Hooks and H. S. Mahmassani, "Least possible time paths in stochastic, time-varying networks", *Computers & operations research*, vol. 25, pp. 1107-1125, 1998.
- [162] E. D. Miller-Hooks and H. S. Mahmassani, "Least expected time paths in stochastic, time-varying transportation networks", *Transportation science*, vol. 34, pp. 198-215, 2000.
- [163] X. Delorme, "Modélisation et résolution de problèmes liés à l'exploitation d'infrastructures ferroviaires", Université de Valenciennes et du Hainaut-Cambrésis, 2003.
- [164] C. Mouilah and A. Rahmoun, "A Balanced Traffic Routing Using the Bio-inspired Traversing and Marking Metaheuristics", *Revue d'Intelligence Artificielle*, vol. 34, pp. 39-44, 2020.
- [165] H. Fouchal, "Optimisation de l'intégrale de Choquet pour le calcul de plus courts chemins multi-objectifs préférés", Nantes, 2011.
- [166] G. Sauvanet, "Recherche de chemins multiobjectifs pour la conception et la réalisation d'une centrale de mobilité destinée aux cyclistes", Université FRANÇOIS RABELAIS DE TOURS, 2011.

- [167] R. PRELAZ-DROUX, " *Système d'information et gestion du territoire*", Presses polytechniques et universitaires romandes, 1995.
- [168] P.-L. Mongeon-Bourbonnais, "Outil intégré de collecte, d'analyse et de visualisation de données de mobilité", École Polytechnique de Montréal, 2016.
- [169] P. Neis, D. Zielstra, and A. Zipf, "The street network evolution of crowdsourced maps: OpenStreetMap in Germany 2007–2011", *Future Internet*, vol. 4, pp. 1-21, 2012.
- [170] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps", *IEEE Pervasive Computing*, vol. 7, pp. 12-18, 2008.
- [171] J. Richard, "Apport des SIG et de la réalité virtuelle à la modélisation et la simulation du trafic urbain", Université Paris Est, 2018.
- [172] D. Kastl and F. Junod. (2020). *Travaux pratiques - Routage FOSS4G avec pgRouting, le réseau routier d'OpenStreetMap et GeoExt* <http://www.postgis.fr/chrome/site/docs/workshop-routing-foss4g/docs/pgRoutingWorkshop.pdf>, Consulté le: 30 September 2020
- [173] V. Thierion, "Contribution à l'amélioration de l'expertise en situation de crise par l'utilisation de l'informatique distribuée: application aux crues à cinétique rapide", École nationale supérieure des mines de Paris, 2010.
- [174] F. Quadro, *Arquitetura de Sistemas GIS*. <https://www.fernandoquadro.com.br/html/2009/05/11/arquitetura-de-sistemas-gis>, Consulté le: 23/06/2016.
- [175] P. G. D. Group. (2008). PostgreSQL 8.3. 7 Documentation. <https://www.postgresql.org/docs/8.3/release-8-3-7.html>, , Consulté le: 17/09/2020.
- [176] Tutorialspoint.com, *PostgreSQL Tutorial - Tutorialspoint*. <https://www.tutorialspoint.com/postgresql/index.htm>, Consulté le: 17/09/2020.
- [177] Postgresql.org, *PostgreSQL: Documentation: 12: PostgreSQL 12.4 Documentation*. <https://www.postgresql.org/docs/12>, Consulté le: 07/10/2020.
- [178] U. CNRS, *35.9. Fonctions en langage C*. http://www.i3s.unice.fr/~rueher/Cours/BD/DocPostgreSQL9-5_HTML/xfunc-c.html, Consulté le: 08/10/2020
- [179] R. Obe and L. Hsu, " *PostGIS in action*", Manning Publications, 2015.
- [180] B. Khalfi, "Modélisation et construction des bases de données géographiques floues et maintien de la cohérence de modèles pour les SGBD SQL et NoSQL", Université PARIS 8 – Université DE TUNIS, 2017.
- [181] Hostinger.fr, *Qu'est ce qu'Apache ? Une description complète du Serveur Web Apache*. <https://www.hostinger.fr/tutoriels/quest-ce-quapache-serveur-web-apache/#Quest-ce-quun-serveur-Web>, Consulté le: 23/01/2020.
- [182] Wikipedia.org, *Apache HTTP Server — Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Apache_HTTP_Server&oldid=174246280, Consulté le: 29/08/2020
- [183] M. Massala, "Recherche intelligente d'informations géographiques à partir des toponymes, des métadonnées et d'une ontologie: application aux forêts du Bassin congolais", Université de TOULOUSE, 2013.
- [184] M. Beaufiles, "Fusion de données géoréférencées et développement de services interopérables pour l'estimation des besoins en eau à l'échelle des bassins versants", Ecole doctorale Arts et Métiers, 2012.
- [185] A. Aime and J. McKenna, *WMS Performance Shootout 2011*. <https://www.slideshare.net/gatewaygeomatics.com/wms-performance-shootout-2011>, Consulté le: 07/07/2010.
- [186] Postgis.fr, 2. À propos — *Travaux pratiques - Routage FOSS4G avec les outils de pgRouting, le réseau routier d'OpenStreetMap et GeoExt*. <http://postgis.fr/chrome/site/docs/workshop-routing-foss4g/docs/build/html/chapters/about.html>, Consulté le: 02/04/2012.
- [187] Wikipedia.org, *Visual C++ — Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Visual_C%2B%2B&oldid=161415379, Consulté le: 30/07/2019.
- [188] Paul, *Qu'est-ce que Microsoft Visual C++ et les versions à installer ?*. <https://www.clicouest.fr/quest-microsoft-visual-c-versions-installer>, Consulté le: 18/06/2018.
- [189] G. Belz. (2012)). Commencer facilement avec Boost Graph. www.developpez.com, Consulté le: 18/06/2020.

- [190] J. Siek, A. Lumsdaine, and L.-Q. Lee, "*The boost graph library: user guide and reference manual*", Addison-Wesley, 2002.
- [191] Cours-gratuit.com, [PDF] *Programmes Windows sous Visual C++*. <https://www.cours-gratuit.com/cours-c/programmes-windows-sous-visual-c>, Consulté le: 13/12/2020
- [192] Corob-msft, *Création de DLL C/C++ dans Visual Studio*. <https://docs.microsoft.com/fr-fr/cpp/build/dlls-in-visual-cpp?view=vs-2019>, Consulté le: 11/10/2020.
- [193] Sigterritoires.fr, *Développer une application avec pgrouting sous Windows (5):écrire un Wrapper | Blog SIG & Territoires*. <https://www.sigterritoires.fr/index.php/developper-une-application-avec-pgrouting-sous-windows-5ecrire-un-wrapper>, Consulté le: 07/04/2019

Annexe

A Balanced Traffic Routing Using the Bio-inspired Traversing and Marking Metaheuristics

Cheikh Mouilah^{1*}, Abdellatif Rahmoun²

¹ Biomathematics Laboratory, Univ. Sidi Bel-Abbes, Sidi Bel Abbes P.B. 89, 22000, Algeria

² LabRI-SBA, ESI SBA, Higher School of Computer Science, Sidi Bel Abbes P.B. 73, 22016, Algeria

Corresponding Author Email: cheikh.mouilah@univ-sba.dz

<https://doi.org/10.18280/ria.340105>

Received: 5 October 2019

Accepted: 13 December 2019

Keywords:

nature-inspired metaheuristics, balanced traffic routing, marking algorithm, geographic information system

ABSTRACT

Recent research focuses on crossing and marking the nodes of a directed graph when it comes to shortest path algorithms. Such behavior is common to animals. Each animal marks its territory using different types of techniques such as smell and sound. In this work, the main idea is to create four populations of marking individuals, two of them looking for the “source—destination” route, and the other two looking for the opposite route. The simultaneous search of these two itineraries (alternating) or one after the other (sequential) are the two variants of this algorithm. To find a route in the urban road network, two populations cooperate, one is installed in the source node and the other is placed in the destination node. As long as there are no meetings between individuals from these two populations, they progress alternately. The first meeting points between individuals form feasible and optimal paths between the starting point and the ending point. Experimental results show that work-linked research is faster, but its risks being blocked, due to competition between individuals on the same path. However, the sequential search is more efficient and it has managed to have all the optimal solutions.

1. INTRODUCTION

The spatial data of a road network can be represented using a huge graph where its vertices are the crossroads and its edges are the segments of the street. The problem of the shortest path in a network is a big concern for several economic, military and even social areas.

Methods dealing with the shortest path go back to the 1950's, and precisely in 1957. Prim [1] published the first algorithm that performs a tree of minimal size. In 1959, Moore [2] proposes an algorithm of the shortest path, often known as “Bellman-Ford-Moore”. This algorithm is based on Ford's centralized solution and Bellman's equation [3].

In the same year, the Dijkstra algorithm is proposed to solve the problem of the shortest path in the case of positive weighting [4].

Several works in the current literature pointed that the “hub-labeling” is considered as being the fastest algorithm in this field [5].

Metaheuristics have emerged to solve all kinds of discrete problems and may also be adapted to continuous problems [6]. Hence, several metaheuristics are proposed to solve the problem of the shortest path like Tabu Search (TS) [7, 8]. Other methods based on evolutionary algorithms are also proposed such as genetic algorithms (GA) [9-12], as well as swarm intelligence techniques such as colony optimization (ACO) [13-18] and particle swarms optimization (PSO) [19, 20].

In order to improve the performance in terms of runtime; many researchers propose a hybridization scheme of some metaheuristics [21-24].

In this article, we propose a novel algorithm inspired by the behavior of striking animals. Territory marking is a natural behavior of most animals so to conquer territories for hunting, living, attracting females and forbidding to other males to trespassing such area.

2. ALGORITHM

2.1 Presentation

The main idea of this algorithm is to launch two populations in two waves; the first one is set at the starting point and the other at the end point. As long as there is no crossroad, the two waves will progress alternately. The points of crossroad are set between them (two waves) form optimal paths between the starting and ending point (Figure 1).

Two colored populations (black, red) represent the two waves in this situation.

The population is composed of a group of individuals which reproduce and kill. Each population is divided into two subpopulations (Figure 2):

- Individuals that are advancing to the destination must use the same direction of the edges.
- Other Individuals are moving in the opposite direction of the edges.

Finally, we get four (4) classes of subpopulations:

- BPA: the black population advancing (forward);
- BPB: the black population in a backward direction;
- RPA: the red population advancing (forward);
- RPB: the red population in a backward direction;





Figure 1. The evolution of both populations

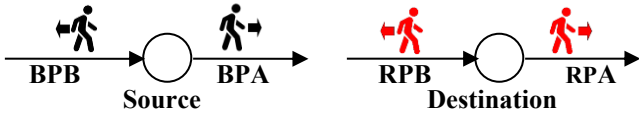


Figure 2. The four subpopulations classes

To find a path between the source and the destination, it is required to have a meeting between an individual from the BPA subpopulation and an individual from the RPB subpopulation, as well as a meeting point of one RPA individual with another from BPB, that is a path between the destination and the source. (Table 1) summarizes all of the possible situations.

Table 1. Possible meetings between individuals from the four subpopulations

	BPA	BPB	RPA	RPB
BPA	✗	✗	✗	✓
BPB	✗	✗	✓	✗
RPA	✗	✓	✗	✗
RPB	✓	✗	✗	✗

The green sign represents "a solution is found" and the red sign represents "a meeting situation to be ignored".

Any individual from the BPA or RPB subpopulation may meet other individual from the RPB or BPA respectively, thus a source-destination route is found. To have a destination-source route, a crossroad must be found between a BPB or RPA individual and another RPA or BPB individual respectively.

The advantage of this approach is to find paths between source and destination and the opposite direction.

2.2 The algorithm

The principles that manage the evolution and control of the two populations are as follows:

2.2.1 Reproduction of the individuals

An individual must reproduce if it meets more than one unmarked adjacent street; it creates new individuals in these adjacent streets. As of the moment, when these individuals (offspring) are born, the genealogical table is updated, and the "offspring" individual inherits the population characteristics from its parent (color, destination ... etc.).

However, if the individual encounters only one adjacent street, it continues its way without any reproduction.

2.2.2 The process of killing individuals

To limit massive population reproduction, any individual considered as unnecessary must be removed. This may be summarized as follows:

- After any reproduction, the "parents" must disappear;
- Another individual from the same population has already marked the adjacent street.
- Any individual in a dead end is blocked.

2.2.3 The marking

Each street in the network has two marking information, one regarding individuals that move from source to destination and the others that move in the opposite direction (Figure 3). These two marking methods act as a communication media between individuals, thus as a means of developing new solutions.



Figure 3. Marking streets scheme

The first individual marks the street, its fingerprint (individual ID) is saved in an appropriate table.

2.2.4 Competition of individuals

During exploration, individuals may encounter footprints of other individuals in the adjacent streets, this individual is considered blocked if and only if all the adjacent streets are marked; hence we come up with three possible situations:

- If the marking belongs to this individual, then it will be automatically removed;
- If its marking belongs to an individual of the same subpopulation, then this individual must be killed;
- Finally, if the marking belongs to the adjacent street of an individual from another competing population, then this consists of a new solution that should be saved.

2.2.5 Composing a path

An individual's path is a group of crossed from the starting point to the meeting point of another individual. The fusion of the two segments is the route between the source (starting point of the first individual) and the destination (starting point of the second individual).

To compose an itinerary of an individual, the algorithm uses a genealogical table (Figure 4) that groups all identifiers of the individuals created with their parents.

The search for the roots of a given individual using a recursive function from the smallest child to the largest father.

Child	Parent
①	①
2	1
③	①
4	2
⑤	③
6	3
⑦	⑤
8	5
9	⑦

Figure 4. The search for the root in the genealogical table

The root of individual 9 is: 9 - 7 - 5 - 3 - 1

If the identifier of the child individual equals the parent's identifier, then the search is completed and the root of that individual is then found.

An individual's route vertices check the adjacent streets of its location, if there is a street that is already marked by that individual or its parent; that street is retained and becomes the new position; and the process continues up to the starting point.

The following organizational chart (Figure 5) explains the principle of composition of a given individual's path based on the genealogical table and adjacent streets:

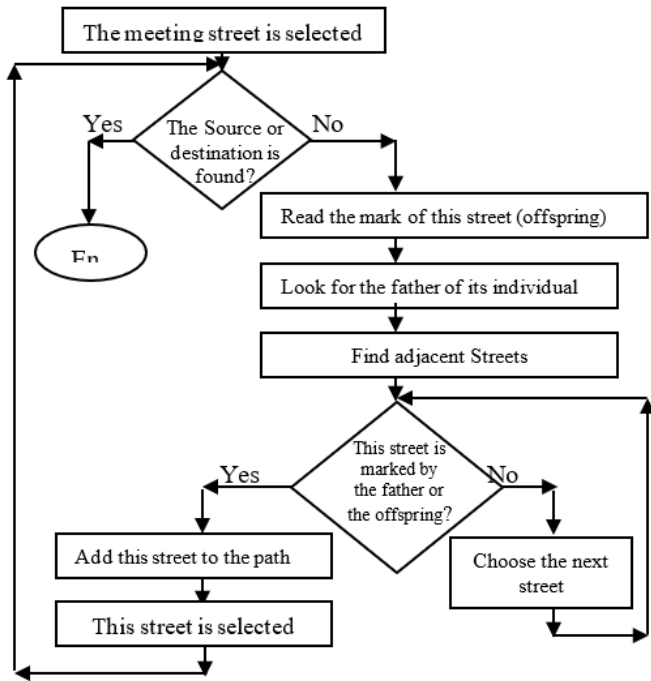


Figure 5. Organization chart of an individual's path search

This Organization chart represents a recursive function, which makes it possible to create a path for any individual, and the fusion between the two paths of two individuals makes up the itinerary between the starting point and the ending point.

2.2.6 Enhancement of the solution

The first meetings between two concurrent subpopulations (BPA—RPB or RPB—BPA) give optimal paths in relation to the number of edges between the starting and ending point. Each route that is found is seen as the shortest path in the non-weighted graph (the point of all edges equals 1).

The exploitation of the routes of the individuals encountered (that provided the solution) reveals that there are individuals (from the same subpopulation) that are blocked and then eliminated. The paths of these individuals constitute other potential solutions (as shown in Figure 6).

The paths between the starting vertex and the meeting point from the previous figure are listed in Table 2.

The exploitation of the paths of the two individuals gave us several possible paths (some of them are rejected because they are circuits); the fusion between two paths of the two individuals we met gives us a route between the starting and the ending vertex (as shown in Table 3).

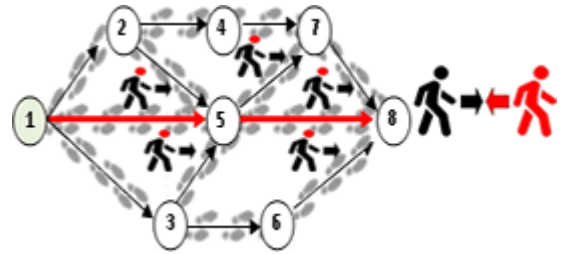
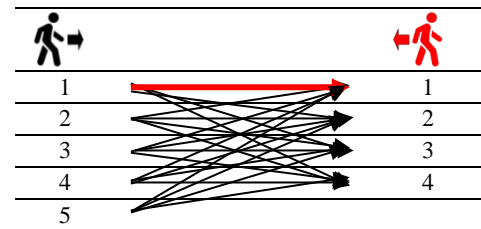


Figure 6. Itinerary “1–5–8” is the shortest path compared to the number of edges, and the paths of blocked individuals (with red heads in the figure) may lead to other feasible solutions

Table 2. List of roads found

N°	Paths	observation
1	1-5-8	Minimal path
2	1-2-5-8	
3	1-2-5-7-8	
4	1-2-4-7-8	
5	1-5-7-8	
6	1-3-5-8	
7	1-3-6-8	

Table 3. The fusion between the paths found constitutes several realizable routes



Let:

- N: is the number of paths found by the 1st individual.
- M: is the number of paths found by 2nd individual.

The total number of routes is equal to N x M.

2.2.7 Sequential or alternating research

The method of finding solutions is very important, either in the quality of the solution or in the speed of the responses. Therefore, the algorithm proposes two techniques of search and which are:

- Sequential search: the algorithm searches first for routes between the source and destination, as it ends and the results are obtained. It searches for routes between the destination and the source. To determine the direction of the search (source—destination or destination—source), it is sufficient to create appropriate subpopulations for each direction during initialization.
- Alternate research: all subpopulations participate in this research. Individuals in two directions move alternately (one iteration for source—destination and the other for destination—source). The competition between individuals on the streets may cause saturation or blockages. This critical situation is affected by time and quality of the solutions and also the progress of the algorithm. To unblock this situation; if the algorithm which determines a subpopulation is stable (there are no new births or individuals eliminated) for a specified

sequential technique are much better in the remaining 02%.

In (Table 4), we resume the results obtained from the 03 experiments of the comparison.

Table 4. The summary table of the comparison between the two techniques

Methods	Optimality	The best time	Saturation
Alternately	00 %	84,625%	0,75 %
Sequential	02 %	13,75%	0%
Equal	98%	1,625	---

To properly evaluate the solutions found by this algorithm, we compared results to other well-known algorithm in route optimization (Dijkstra).

Dijkstra is an optimization algorithm to find the optimal path (the shortest path) between two vertices in an oriented and weighted graph with a single positive cost.

The proposed algorithm allows to find several feasible solutions between the two vertices of the extremity in an oriented graph, and then it chooses the optimal route between them instead; the disjkstra algorithm gives a single path, that is the shortest between two vertices.

We compare the solutions obtained by the two algorithms on the same routes.

All solutions of the sequential technique are as good as Dijkstra’s algorithm. On the contrary, the alternately technique failed to find the optimal routes in several cases (set_1 =1.5%, set_2 =1%, set_3 =1.5% and set_4 = 4%) (Figure 10).

Error rate in optimality compared to the dijkstra algorithm

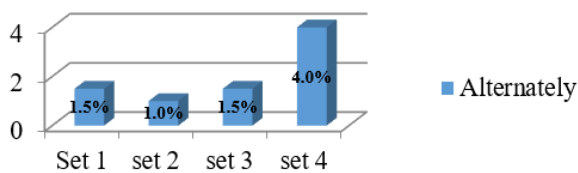


Figure 10. The failure rates of the alternately method

We then use the sequential method to solve the problem of the shortest path in an oriented and weighted graph despite the fact that is slower than the alternately method.

This metaheuristic has found all the optimal paths required in 400 different (randomly selected) routes, depending on the results obtained from the experiments. Additionally, it gave us other realizable solutions near the optimal, which is an advantage that can be exploited to solve multi-objective problems.

Another advantage of this approach is that the complete path of the graph is not necessary, to find a shorter path. Just launch two waves in the two vertices (source and destination) of a route and the intersection points of these waves can give optimal paths.

4. CONCLUSIONS

The goal of this article is to optimize routes for vehicles in the urban road networks of an Algerian city. Shortest path algorithms in the literature rely on mathematical modeling such road networks which come to be a difficult task. We therefore attempted in this work to use a metaheuristic that is inspired by the behaviors of landmark animals where each animal (an individual) marks the street it traverses with its

footprint, so that this individual reproduces, die and inherit from their parent in a given population from source to destination.

The meeting of two individuals from two different populations; constitutes an optimal route between the source and the destination and the exploitation of these optimal routes may lead to other possible paths.

The algorithm uses two vertices techniques:

- The sequential method: it has succeeded in finding all the solutions without any failure, but it is slow compared to the alternately method.
- The alternately method is faster, sensitive to saturation and less optimal compared to the sequential method.

This algorithm comes up with an optimal (“shortest”) route as well as some possible paths between source and destination in an oriented graph. Further work may emphasize on solving problems of “multi-objective” shortest path.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments and suggestions that greatly improved the presentation of this work.

This work was partially supported by the MESRS (Ministry of Higher Education and Scientific Research, Algeria) (DGRSDT, PRFU: C00L03UN220120200004 and PRFU: C00L07ES220120180003).

REFERENCES

- [1] Sullivan, H., Bashkow, T.R., Klappholz, D. (1977). A large scale, homogenous, fully distributed parallel machine, II. ACM SIGARCH Computer Architecture News, pp. 118-124. <https://doi.org/10.1145/633615.810660>
- [2] Hain, T. (2002). An IPv6 provider-independent global unicast address format. Internet Draft.
- [3] Zhang, L., Berson, S., Herzog, S., Jamin, S. (1997). Resource reservation protocol (RSVP)--Version 1 functional specification. Resource. <https://doi.org/10.17487/rfc2205>
- [4] Floyd, S., Kohler, E. (2006). RFC 4341-profile for datagram congestion control protocol (DCCP) congestion control ID 2: TCP-like congestion control. Internet Engineering Task Force, Request for Comments. <https://doi.org/10.17487/rfc4341>
- [5] Abraham, I., Delling, D., Goldberg, A.V., Werneck, R.F. (2011). A hub-based labeling algorithm for shortest paths in road networks. International Symposium on Experimental Algorithms, pp. 230-241. https://doi.org/10.1007/978-3-642-20662-7_20
- [6] Ayari, N. (2010). Métaheuristiques parallèles hybrides pour l’optimisation combinatoire: Problème de règles de Golomb. Université de Jendouba TUNISIE: 15.
- [7] Hertz, A., Laporte, G., Mittaz, M. (2000). A tabu search heuristic for the capacitated arc routing problem. Operations Research, 48(1): 129-135. <https://doi.org/10.1287/opre.48.1.129.12455>
- [8] Brandão, J., Eglese, R. (2008). A deterministic tabu search algorithm for the capacitated arc routing problem. Computers & Operations Research, 35(4): 1112-1126. <https://doi.org/10.1016/j.cor.2006.07.007>
- [9] Ahn, C.W., Ramakrishna, R.S. (2002). A genetic

- algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, 6(6): 566-579. <https://doi.org/10.1109/TEVC.2002.804323>
- [10] Dey, A., Pradhan, R., Pal, A., Pal, T. (2018). A genetic algorithm for solving fuzzy shortest path problems with interval type-2 fuzzy arc lengths. *Malaysian Journal of Computer Science*, 31(4): 255-270. <https://doi.org/10.22452/mjcs.vol31no4.2>
- [11] Tsujimura, Y., Gen, M. (1998). Entropy-based genetic algorithm for solving TSP. 1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat..No.98EX111), pp. 285-290. <https://doi.org/10.1109/KES.1998.725924>
- [12] Nagata, Y., Soler, D. (2012). A new genetic algorithm for the asymmetric traveling salesman problem. *Expert Systems with Applications*, 39(10): 8947-8953. <https://doi.org/10.1016/j.eswa.2012.02.029>
- [13] Głałowski, M., Musznicki, B., Nowak, P., Zwierzykowski, P. (2012). Shortest path problem solving based on ant colony optimization metaheuristic. *Image Processing & Communications*, 17(1-2): 7-17. <https://doi.org/10.2478/v10248-012-0011-5>
- [14] Ok, S.H., Seo, W.J., Ahn, J.H., Kang, S., Moon, B. (2011). An ant colony optimization approach for the preference-based shortest path search. *Journal of the Chinese Institute of Engineers*, 34(2): 181-196. <https://doi.org/10.1080/02533839.2011.565574>
- [15] Bezerra, L.C., Goldberg, E.F., Buriol, L.S., Goldberg, M.C. (2011). Grace: A generational randomized ACO for the multi-objective shortest path problem. *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 535-549. https://doi.org/10.1007/978-3-642-19893-9_37
- [16] Wang, C.X., Cui, D.W., Wang, Z.R., Chen, D. (2005). A novel ant colony system based on minimum 1-tree and hybrid mutation for TSP. *International Conference on Natural Computation*, pp. 1269-1278. https://doi.org/10.1007/11539117_168
- [17] Skinderowicz, R. (2013). Ant colony system with selective pheromone memory for SOP. *International Conference on Computational Collective Intelligence*, pp. 711-720. https://doi.org/10.1007/978-3-642-40495-5_71
- [18] Mouilah, C., Belkadi, K. (2012). Application of "AntNet" in an urban road network: Case of Algerian cities. *International Conference on Education and e-Learning Innovations*, pp. 1-7. <https://doi.org/10.1109/ICEELI.2012.6360648>
- [19] Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., Wang, Q. (2007). Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, 103(5): 169-176. <https://doi.org/10.1016/j.ipl.2007.03.010>
- [20] Liao, Y.F., Yau, D.H., Chen, C.L. (2012). Evolutionary algorithm to traveling salesman problems. *Computers & Mathematics with Applications*, 64(5): 788-797. <https://doi.org/10.1016/j.camwa.2011.12.018>
- [21] Mohemmed, A.W., Sahoo, N.C. (2007). Efficient computation of shortest paths in networks using particle swarm optimization and noising metaheuristics. *Discrete Dynamics in Nature and Society*, 2007(4): 1-25. <https://doi.org/10.1155/2007/27383>
- [22] Davoodi, M., Golsefidi, M.M., Mesgari, M. (2019). A hybrid optimization method for vehicle routing problem using artificial bee colony and genetic algorithm. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42: 293-297. <https://doi.org/10.5194/isprs-archives-XLII-4-W18-293-2019>
- [23] Dudeja, C. (2019). Fuzzy-based modified particle swarm optimization algorithm for shortest path problems. *Soft Computing*, 23: 8321-8331. <https://doi.org/10.1007/s00500-019-04112-1>
- [24] Maleki, F., Yousefikhoshbakht, M. (2019). A hybrid algorithm for the open vehicle routing problem. *Iran University of Science & Technology*, 9(2): 355-371.