

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

Université Saad Dahlab de Blida

N° D'ordre :



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Foura Faycal Hamida SeifEddine

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Génie des Systèmes Informatiques

Thème : Système de Gestion et de Recommandation de Transport de Marchandises

Soutenu le :

M. Bala Mahfoud
M. Rezoug Nachida
Mme. Aroussi Sana
Mme. Hedjazi Badiâa

Président
Examinatrice
Promotrice
Encadreuse

Promotion
2015 / 2016

Résumé

Une gestion efficace du transport des marchandises est une question vitale pour minimiser les problèmes de congestion dans les villes, du fait que l'économie et la qualité de vie des villes dépendent essentiellement d'un système de transport urbain des marchandises efficace et respectueux de l'environnement. La livraison et la collecte de marchandises dans les villes ont un impact majeur sur la puissance économique, la qualité de vie, l'accessibilité et l'attrait des communes.

Nous proposons dans ce travail un système de gestion et de recommandation de transport de marchandises afin de mettre en relation des demandeurs et des transporteurs de marchandises à travers le web.

Ce système permet de trouver les plus courts chemins selon les critères choisis et recommander les meilleurs transporteurs dans le but de minimiser les coûts, de raccourcir les chemins empruntés et donc gain de temps et de l'argent pour chaque opération de transport.

Il assure ainsi les meilleurs services pour les clients et les transporteurs. Aussi, il permet de réduire les problèmes de congestion que connaissent nos villes.

Pour ce faire, notre système permet la recherche du plus court chemin unicritère en appliquant l'algorithme de Dijkstra et les plus courts chemins multicritères en appliquant l'algorithme d'étiquetage multicritères pour la recherche des plus courts chemins Martins. La recommandation des transporteurs susceptibles à intéresser les clients a été réalisée en utilisant l'approche de recommandation sociale qui représente la meilleure solution dans le cas de notre système.

Mots clés : transport de marchandises, problèmes de congestion, recherche plus courts chemin, unicritère, multicritères, Dijkstra, Martins, recommandation sociale

Remerciements

En préambule à ce mémoire, nous adressons ces quelques mots pour remercier notre Dieu tout puissant de nous avoir donné la volonté, la patience, la santé et la confiance durant toutes nos années d'études.

Nous tenons à remercier sincèrement notre promotrice Mme Aroussi Sana pour l'honneur qu'elle nous a fait en acceptant de diriger ce travail, pour sa grande disponibilité, et ses précieuses directives tout au long de la réalisation de ce projet.

Nos remerciements s'adressent également à notre encadreuse Badiâa Hedjazi, pour le privilège qu'elle nous a fait en proposant ce thème. Sa modestie, sa riche expérience et l'accueil cordial qu'elle nous a toujours réservé.

Nous souhaitons remercier notre ami et collègue Sali Abderrahmane pour son soutien et son aide durant notre cursus universitaire.

Nous tenons aussi à remercier notre collègue Hammouda Inaam d'être la pour nous tout au long de notre parcours universitaire.

Et enfin on remercie tous ceux qui ont contribué de près ou de loin à réaliser ce travail.

Je dédie ce mémoire

- *A mon père, la personne la plus digne de mon estime et de mon respect, sans vous je n'aurais jamais regardé si haut, merci pour tout papa.*
- *A ma mère, aucun dédicace ni remerciement sera à la hauteur des tes sacrifices et de tes prières pour moi, je vous aime maman.*
- *A ma petite sœur Ihcène, vivement le jour où tu me fera une dédicace dans ton mémoire.*
- *A mon frère Amine. Et mes nièces Ichrak, Rahma et Lamia.*
- *A ma sœur Nassima, son époux Maamar. Et à mon neveu Hassan.*
- *A mon oncle Abdenour, ma tante Karima et leurs enfants.*
- *A mon binôme et mon frère intime Sifou. Un ami, c'est à la fois nous-mêmes et l'autre, l'autre en qui nous cherchons le meilleur de nous-mêmes, mais également ce qui est meilleur que nous. Frérot à toutes les nuits blanches qu'on a passé à réviser et à tous nos délires.*
- *A mon ami intime Abderrahmane, sa femme et leur fils Adem.*
- *A Mouloud, Mustapha et à tous mes amis et personnes qui me sont chers.*

Fayçal

Je dédie ce mémoire

A ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

A mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

A mes chers et adorables frères et sœurs, Abdou, Houria, AbdElbasset et Niaama. En témoignage de mon affection fraternelle, de ma profonde tendresse et reconnaissance, je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le tout puissant, vous protège et vous garde.

A ma grand-mère Hnifa, pour sa douceur et sa gentillesse.

A mon autre soi-même, mon frère et mon binôme Fay, que Mille connaissances ne valent pas son amitié. A tous nos moments partagés, pour le meilleur et pour le pire et surtout pour la vie In Shaa Allah.

A tous mes amis, à la famille dont j'ai choisi les membres, à Djaafar, Islam, Houcine, AbdElMounaim, et à tous ceux qui m'ont donné confiance en l'avenir, et qui n'ont pas laissé mon esprit s'abattre et se décourager.

A toutes les personnes qui ont participé à l'élaboration de ce travail, à tous ceux que j'ai omis de citer.

SeifEddine

SOMMAIRE

| | |
|---|-----------|
| Introduction générale | 12 |
| Chapitre I : Problèmes de cheminement dans les réseaux de transport routier..... | 14 |
| Introduction..... | 14 |
| I. Généralités sur le transport routier de marchandises | 15 |
| I.2 Evolution de la logistique du transport routier de marchandises..... | 15 |
| I.2.1 Évolution générale..... | 15 |
| I.2.2 Urbanisation | 15 |
| I.2.3 Développement durable | 15 |
| I.3 Problèmes du transport routier de marchandises | 16 |
| II. Problème du plus court chemin | 17 |
| II.1 Modélisation..... | 17 |
| II.2 Condition nécessaire..... | 17 |
| III. Les algorithmes de résolution | 18 |
| III.1. Algorithmes plus court chemin uni critère..... | 18 |
| III.1.1 Algorithme de Dijkstra | 18 |
| III.1.2 Algorithme de Bellman-Ford..... | 19 |
| III.1.3 Algorithme de Floyd-Warshall..... | 20 |
| III.1.4 Comparaison..... | 20 |
| III.2 Algorithmes plus court chemin multicritères..... | 22 |
| III.2.1 Algorithme de Martins | 23 |
| III.2.2 Algorithme SPEA2..... | 24 |
| III.2.3 Comparaison..... | 24 |
| IV. Conclusion | 25 |
| Chapitre II : Systèmes de recommandation dans le domaine de transport | 26 |
| Introduction..... | 26 |
| I. Les systèmes de recommandation | 27 |
| I.1 Définition..... | 27 |
| I.2 Construction d'un système de recommandation..... | 28 |
| I.2.1 Collecte d'Information sur l'utilisateur | 28 |
| I.2.2 Modèle Utilisateur..... | 29 |
| I.2.3 Liste de recommandations..... | 30 |

| | | |
|--|---|-----------|
| I.3 | Types de système de recommandation | 30 |
| I.3.1 | Recommandation Objet | 30 |
| I.3.1.1 | Exemple..... | 30 |
| I.3.2 | Recommandation Sociale | 31 |
| I.3.3 | Recommandation Hybride | 33 |
| I.3.4 | Comparaison des méthodes de recommandations..... | 34 |
| I.4 | Présentation des méthodes de recommandation sociale | 36 |
| I.4.1 | L'évaluation des articles par les utilisateurs | 36 |
| I.4.1.1 | Évaluation avec investissement de l'utilisateur | 36 |
| I.4.1.2 | Évaluation sans participation de l'utilisateur | 37 |
| I.4.2 | Stockage des données | 37 |
| I.4.3 | Les méthodes de filtrage collaboratif | 38 |
| I.4.3.1 | Méthodes basées sur la mémoire | 38 |
| I.4.3.2 | Méthodes basées sur le modèle | 39 |
| I.4.4 | Formules de notation à étoiles (Rating stars) | 40 |
| I.4.4.1 | La mise en place | 41 |
| II.5 | Travaux antérieurs sur les systèmes de recommandation dans le domaine de transport .. | 42 |
| II.5.1. | Travaux de recherches (Solutions non commerciales) | 42 |
| II.5.2 | Travaux de recherches commerciales | 43 |
| II.5.3 | Synthèse des travaux antérieurs | 44 |
| III. | Conclusion | 46 |
| Chapitre III : Conception du Système de Gestion et de Recommandation de Transport de Marchandises (SGRTM) | | 47 |
| Introduction..... | | 47 |
| I. Description générale | | 47 |
| II. Recherche d'itinéraire | | 48 |
| II.1 | Algorithme de Dijkstra | 49 |
| II.2 | Algorithme de Martins | 51 |
| III. Recommandation transporteurs..... | | 54 |
| IV. Conception du système d'informations pour le SGRTM | | 56 |
| IV.1 | Diagramme des cas d'utilisation | 56 |
| IV.2 | Elaboration du Diagramme de classes | 59 |
| IV.2.1 | Règles de gestion..... | 59 |

| | |
|--|-----------|
| IV.2.2 Le passage vers le modèle relationnel | 61 |
| IV.3 Diagramme d'activité | 61 |
| IV.4 Diagrammes de séquences | 63 |
| V. Conclusion..... | 68 |
| Chapitre IV : Réalisation | 69 |
| Introduction..... | 69 |
| I Présentation des outils de développement | 69 |
| I.1 Choix d'Eclipse J2EE comme environnement de développement JAVA | 69 |
| I.2 Choix d'Apache Tomcat comme serveur | 69 |
| I.3 Choix de MYSQL Server comme SGBD | 69 |
| II.4 Tristramg/osm4routing | 70 |
| II Description détaillée de l'architecture du SGRTM | 71 |
| II.1 Paquetage recherche itinéraire | 71 |
| II.1.1 API «Hipster4j - version 1.0.0-rc2»..... | 71 |
| II.1.2 Justification du choix | 72 |
| II.1.3 Modification apportée | 72 |
| II.2 Paquetage recommandation..... | 73 |
| II.2.1 Collecte des appréciations | 74 |
| II.2.2 Evaluation des notations transporteurs | 74 |
| II.2.3 Classement et affichages Top 3 des transporteurs | 75 |
| III Présentation du prototype (l'application) | 75 |
| III.1 Page d'inscription et Page de connexion | 76 |
| III.2 Recherche itinéraire multicritères | 77 |
| III.3 Choix du moyen de transport selon l'itinéraire choisi | 78 |
| III.4 Compte client | 78 |
| III.5 Compte administrateur | 79 |
| III.6 Compte transporteur | 80 |
| IV. Conclusion | 81 |
| Conclusion générale..... | 82 |
| Références Bibliographiques..... | 83 |

Liste des figures

Chapitre I : Problèmes de cheminement dans les réseaux de transport routier

Figure 1 : Répartition sectorielle de la consommation finale d'énergie en France.....16

Chapitre II : Systèmes de recommandation dans le domaine de transport

Figure 1 : Exemple d'un système de recommandation28

Figure 2 : Système de recommandation AMAZON34

Figure 3 : Echelle de notation sur 540

Chapitre III : Conception du Système de Gestion et de Recommandation de Transport de Marchandises (SGRTM)

Figure 1 : Description générale de l'architecture système47

Figure 2 : Exemple d'Application de l'algorithme de Dijkstra50

Figure 3 : Recherche plus court chemin Martins54

Figure 4 : Diagramme des cas d'utilisation général.....57

Figure 5 : Diagramme des cas d'utilisation gestion des demandes.....58

Figure 6 : Diagramme des cas d'utilisation gestions des missions59

Figure 7 : Diagramme de classes60

Figure 8 : Diagramme d'activité générale62

Figure 9 : Diagramme de séquence d'inscription d'un visiteur63

Figure 10 : Diagramme de séquence de connexion d'un utilisateur64

Figure 11 : Diagramme de séquence de validation d'inscription.....65

Figure 12 : Diagramme de séquence d'ajout d'une demande de transport.....66

Figure 13 : Diagramme de séquence de Recherche, Confirmation/Annulation d'une demande67

Figure 14 : Diagramme de séquence de Recherche, Confirmation d'achèvement d'une mission68

Chapitre IV : Réalisation, tests et résultats

Figure 1 : Description détaillée du SGRTM72

Figure 2 : L'échèle de notation utilisée dans notre application74

Figure 3 : Fiabilité du transporteur75

Figure 4 : Classement des transporteurs les mieux notés.....75

Figure 5a : Page de connexion (authentification)76

Figure 5b : Page d'inscription.....76

Figure 6 : Page de recherche itinéraire multicritères et l'affichage des résultats77

| | |
|--|----|
| Figure 7 : Page d'affichage des moyens de transport et la recommandation des transporteurs | 78 |
| Figure 8 : Page du compte client..... | 79 |
| Figure 9 : Page du compte administrateur | 80 |
| Figure 10 : Page du compte transporteur | 81 |

Liste des tableaux

Chapitre I : Problèmes de cheminement dans les réseaux de transport routier

Tableau 1 : Comparaison des algorithmes plus court chemin uni critère 21

Tableau 2 : Comparaison des algorithmes plus court chemin multicritères 25

Chapitre II : Systèmes de recommandation dans le domaine de transport

Tableau 1 : Formes de collecte d'informations sur l'utilisateur dans un SR 29

Tableau 2 : Profil Livres 31

Tableau 3 : Profil Préférences/Utilisateurs 31

Tableau 4 : Notations films 32

Tableau 5 : Tableau comparatif des méthodes de recommandations 35

Tableau 6 : Une matrice des notes attribuées aux articles par les utilisateurs 37

Tableau 7 : Synthèse des travaux antérieurs sur les systèmes de recommandation dans le transport 45

Chapitre III : Conception du Système de Gestion et de Recommandation de Transport de Marchandises (SGRTM)

Tableau 1 : Déroulement de l'exemple d'application de l'algorithme de Dijkstra 50

Tableau 2 : Notation des utilisateurs 55

Tableau 3 : Notation transporteur 1 selon l'expérience des utilisateurs 55

Tableau 4 : Notes moyennes des transporteurs 55

Tableau 5 : Identification des acteurs 56

Tableau 6 : Description cas d'utilisation général 58

Chapitre IV : Réalisation, tests et résultats

Tableau 1 : Evaluation des notes clients 74

INTRODUCTION GENERALE

Le transport est une question essentielle et complexe des politiques des villes. Cette question est indissociable des mécanismes économiques et sociaux, des politiques d'aménagement du territoire et des choix comme des contraintes de la vie quotidienne. En effet, la mondialisation de l'économie et le développement du commerce international ont intensifié le transport de marchandises.

Le transport est certes nécessaire pour le développement économique et social mais il engendre des externalités négatives qui peuvent affecter les villes, l'économie, la société et l'environnement. Le développement du cas qui met en relation les demandeurs et les offreurs des moyens de transport de marchandises représente une solution efficace qui pourrait résoudre partiellement le problème de transport de marchandises.

La gestion de transport implique plusieurs problèmes de prise de décision, qui sont relatifs aux choix appropriés de la route ainsi que la sélection du transporteur. Ces problèmes ont accru en raison de préférences variées du client et les ressources réduites des transporteurs. La correspondance entre les préférences citées et les offres de services, ne peut pas être faite d'une manière facile manuellement, car dans la plupart des cas de multiples alternatives existent, alors qu'on est limité en termes de temps et d'argent. En général, la gestion d'un système de transport requière des solutions rapides et moins coûteuses pour les demandes clients.

Le développement des systèmes d'aide à la décision et de recommandation apparaît comme une solution efficace de régulation. Ces systèmes permettent de transmettre en temps réel les informations concernant l'offre et la demande sur les réseaux de transport de marchandises.

De ce fait, l'objectif de notre projet est de développer un Système de Gestion et de Recommandation pour le Transport de Marchandises (SGRTM) permettant de fournir une recommandation dans le cas où plusieurs solutions existent. Le client doit être aidé afin d'évaluer proprement les solutions proposées et prendre sa décision finale.

Le présent mémoire est divisé en quatre chapitres :

- le chapitre I discute les problèmes de cheminement dans les réseaux de transport routier
- le chapitre II introduit les systèmes de recommandation dans un cadre général et dans le domaine de transport dans un cadre plus spécifique

- le chapitre III est dédié à la partie conception de notre SGRTM
- et enfin le chapitre IV décrit la réalisation de notre système.

CHAPITRE I :
PROBLEMES DE CHEMINEMENT DANS LES
RESEAUX DE TRANSPORT ROUTIER

Introduction

Le transport de marchandises est une composante fondamentale de la vie urbaine. La mondialisation des activités économiques, l'évolution du comportement des consommateurs et le développement des technologies de pointe ont profondément modifié le transport de marchandises. Ces développements ont conduit à une croissance du transport de marchandises qui se poursuivra inévitablement si aucune mesure nouvelle n'est prise.

Le coût et le temps consacrés au transport d'une marchandise entre les lieux de production et le client final sont souvent vécus par les industriels comme un mal nécessaire. La charge de transport d'un produit fini représente, en effet, une part importante de leurs dépenses. Réduire ces dernières et, surtout, satisfaire aux exigences des clients sont une priorité pour les entreprises souhaitant rester compétitives.

Plusieurs problèmes de prises de décision se posent lors de la gestion de transport, ces problèmes sont dus à la recherche des plus courts chemins ainsi que les transporteurs dans le but de satisfaire les exigences variées du client en prenant en considération les ressources limitées du transporteur, ce qui rend la tâche de correspondance entre l'offre et la demande difficile à réaliser manuellement.

Toutefois, les divers effets négatifs constatés montrent que l'impact de la croissance continue du transport de marchandises n'est pas viable à long terme. Par conséquent, l'organisation efficace du transport des marchandises en ville est devenue cruciale non seulement pour un développement durable.

Il est courant, lorsque l'on cherche à se rendre d'un point à un autre dans un réseau routier, on s'intéresse à la recherche du plus court chemin. Pour ce faire, il existe plusieurs algorithmes qui permettent de calculer ce dernier en se basant sur deux axes théoriques fondamentaux : le plus court chemin uni critère et le plus court chemin multicritères.

Dans ce chapitre, nous définissons dans un premier temps le transport routier de marchandises, son évolution ainsi que les problèmes rencontrés par ce dernier. Ensuite, nous présentons le plus court chemin dans un graphe en nous focalisant sur les algorithmes les plus utilisés — à savoir celui de Dijkstra, Bellman-Ford et Floyd. Enfin, nous nous intéressons au plus court chemin multicritères.

I. Généralités sur le transport routier de marchandises

I.1 Définition du transport routier de marchandises :

Un transporteur routier de marchandises est une entreprise commerciale spécialisée dans le transport de marchandises. Le transporteur routier de marchandise fait partie des transporteurs terrestres, comme les transporteurs ferroviaires, les déménageurs, les transporteurs routiers de personnes, etc. Il exerce dans la plupart des pays une profession réglementée.

I.2 Evolution de la logistique du transport routier de marchandises :

Dans cette partie, nous allons examiner l'évolution générale de la société et de l'élaboration du transport de marchandises.

I.2.1 Évolution générale

Les problèmes du transport de marchandises en ville résultent de toute une série d'évolutions de notre société. Certaines tendances à l'origine de ces problèmes sont liées aux développements sociétaux, technologiques et organisationnels. Ces derniers incluent l'évolution de la politique qui privilégie des processus de consultation plus ouverts et plus larges entre les pouvoirs publics, l'industrie et la société civile [OECD, 04].

I.2.2 Urbanisation

Les villes sont le moteur de la dynamique économique et politique du monde industrialisé. Elles répondent à des besoins qui ne peuvent pas être satisfaits autrement.

L'urbanisation peut conduire à la création de zones métropolitaines, multipolaires ou centralisées. Bien souvent, elle contribue au développement de mégalo-poles comme les grandes villes situées dans le monde entier. Ces développements, qui s'accompagnent d'une croissance du trafic, engendrent divers problèmes de transport [OCDE, 01a].

I.2.3 Développement durable

Le développement durable est devenu le principe directeur dans de nombreux pays industrialisés. La demande de transport de voyageurs et de marchandises devrait augmenter fortement. Toutefois, ses différents effets négatifs, comme la dégradation de l'environnement, les dommages sur la santé de l'homme, les encombrements et les coûts humains des accidents, montrent que les conséquences de l'augmentation continue des services de transport ne sont pas supportables à long terme. Pour parvenir à un développement durable, il

transport ne sont pas supportables à long terme. Pour parvenir à un développement durable, il faudra prendre des mesures à court et à long terme afin que le transport continue de procurer ses avantages économiques et sociaux importants tout en maîtrisant ses effets environnementaux et sociaux néfastes [OCDE, 01b].

I.3 Problèmes du transport routier de marchandises :

Les divers problèmes rencontrés ou causés par le transport de marchandises en ville, qui ont des répercussions politiques importantes au niveau local, national et international, sont [OECD, 04] :

- **Accessibilité et congestion**, qui sont souvent dus :
 - À l'insuffisance de l'infrastructure du transport de marchandises en ville.
 - Aux limitations d'accès.
 - Aux encombrements.
- **Consommation d'énergie** :

Le transport de marchandises contribue grandement à la demande d'énergie des transports. L'étude française menée en 2013 a montré que le secteur du transport de marchandises représente environ 32 % de la répartition sectorielle de la consommation finale d'énergie (figure 1.1).

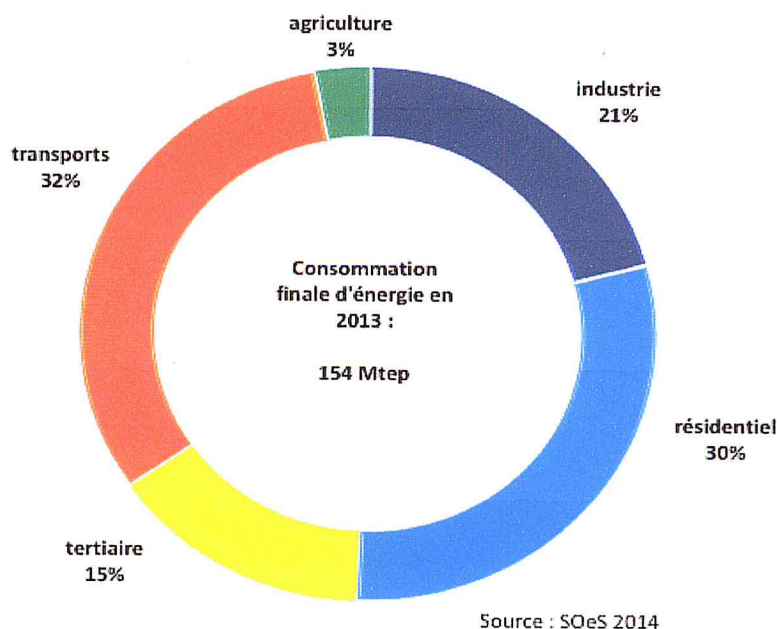


Figure 1.1 Répartition sectorielle de la consommation finale d'énergie en France [OIE, 13]

II. Problème du plus court chemin :

Les problèmes de cheminement dans les graphes (en particulier la recherche d'un plus court chemin) comptent parmi les plus classiques de la théorie des graphes et les plus importants dans leurs applications.

II.1 Modélisation

Le problème du plus court chemin (pcch) peut être posé de la façon suivante : étant donné un graphe $G = (X, U)$, où X : ensemble des sommets et U : ensemble des arcs.

Soit un arc $u \in U$ et deux sommets $i, j \in X$, on associe à chaque arc $u = (i, j)$ un nombre réel noté $l(u)$ ou lij , appelé longueur de l'arc.

Le problème du pcch entre deux sommets i_0 et j_0 du graphe consiste à déterminer, parmi tous les chemins allant de i_0 à j_0 celui dont la longueur totale :

$$l(U) = \sum_{u \in U} l(u) \quad \text{soit minimale [LUC, 10].}$$

Pour qu'un chemin de longueur « optimale » existe entre deux sommets, il faut que ce dernier ne contient pas de circuit absorbant (circuit d'un graphe pondéré, dont la somme des poids des arcs est négative).

On peut s'intéresser à la recherche d'un plus court chemin dans un graphe :

- entre deux sommets donnés
- d'un sommet à tous les autres
- entre tous les couples de sommets

II.2 Condition nécessaire

Le problème du plus court chemin à une solution si et seulement si il n'existe pas dans le graphe des circuits de longueur négative pouvant être atteint à partir de i_0 .

Si cette condition nécessaire est vérifiée, il existe toujours un chemin de longueur minimale qui soit élémentaire. En effet, lorsque tous les circuits du graphe pouvant être atteint à partir de i_0 ont une longueur strictement positive, tout pcch est nécessairement élémentaire [LUC, 10].

III. Les algorithmes de résolution :

Il existe un grand nombre d'algorithmes permettant de résoudre le problème du plus court chemin. Ces algorithmes peuvent être classés selon deux axes théoriques :

- le plus court chemin unicritère qui prend en entrée un graphe orienté pondéré par des réels positifs ou chaque arc possède un seul poids
- le plus court chemin multicritères qui prend en entrée un graphe orienté pondéré par des réels positifs ou chaque arc possède deux ou plusieurs poids au lieu d'un seul

III.1. Algorithmes plus court chemin uni critère :

Dans cette section, nous présentons d'abord le plus court chemin uni critère, en nous focalisant sur les algorithmes les plus utilisés — à savoir celui de Dijkstra, Bellman-Ford et Floyd. Ensuite nous nous intéresserons au plus court chemin multi objectif et ses différents algorithmes. En partant de ces algorithmes, nous allons faire enfin une étude comparative afin de justifier notre choix.

De nombreux algorithmes ont été développés depuis les années 1950 pour calculer le plus court chemin en termes d'un seul critère (distance, temps ou coût). Les plus connus sont : l'algorithme de Dijkstra (1959), de Bellman-Ford (1956 – 1958) et Floyd-Warshall (1959). Ces algorithmes sont décrits par la suite.

III.1.1 Algorithme de Dijkstra :

- Cet algorithme (proposé par E.W. Dijkstra en 1959) permet de calculer le plus court chemin d'un sommet «s» à un sommet «d» ou d'un sommet «s» à tous les autres sommets dans un graphe de longueur positive [LAU, 06].
 - Soit $\pi = l$ la valeur de chemin du sommet «s» vers le sommet «x», ainsi, initialement: $\pi(s) = 0$ et $\pi(x) = \infty$ pour tout sommet $x \neq s$
 - Soit M l'ensemble des sommets marqués, initialement il est vide ($M = \emptyset$)
- Tant qu'il existe un sommet non marqué ($M \neq X$) ou on n'est pas arrivé au sommet destinataire ($x \neq d$) faire:
 1. Choisir un sommet non marqué, soit x ($x \in X-M$), ayant le plus petit π [$\pi(x) = \min \{\pi(y) \text{ tq } x \in X-M\}$]

2. Mettre à jours ses successeurs non encore marqués comme suit:

$$\Pi(y) = \min(\pi(y), \pi(x) + l_{xy}) \text{ tel que } y \in r^+(x) \cap (X - M)$$

3. Marquer le sommet x [$M = M \cup \{x\}$]

III.1.2 Algorithme de Bellman-Ford :

- Cet algorithme (inventé par R. Bellman, S. End et L.R. Ford en 1956), permet de calculer le PCC d'un sommet « s » à tous les autres sommets dans un graphe orienté de longueur quelconque et aussi de détecter la présence d'un circuit absorbant [LUC,10].

- Soit $\lambda^k(i)$ la valeur de chemin du sommet « s » vers le sommet « i » ne contenant pas plus de k arcs, ainsi, initialement : $k = 0$; $\lambda^k(s) = 0$ et $\lambda^k(x) = \infty$ pour $x \neq s$
- Soit M l'ensemble des sommets dont le poids $\lambda^k(s)$ a été modifié à l'itération k , initialement il contient « s » ($M = \{s\}$)

- Tant qu'il existe un sommet marqué ($M \neq \emptyset$) et k est strictement inférieur à n (n présente le nombre des sommets $|X|$) faire:

1. Incrémenter k

2. Initialiser l'ensemble NM à vide (NM contiendra les sommets dont la marque λ^k sera modifiée)

3. Pour tout sommet x de $r^+(M)$ (i.e les successeurs des sommets dont la marque a été modifié au cours de l'itération $k-1$) faire:

a) Mettre à jours sa marque : $\lambda^k(x) = \min(\lambda^{k-1}(x), \lambda^{k-1}(y) + l_{yx})$ tel que $y \in r^+(x) \cap M$

b) Si sa marque a été modifiée ($\lambda^k(x) < \lambda^{k-1}(x)$) alors ajouter x à l'ensemble NM ($NM = NM \cup \{x\}$)

4. Remplacer M par NM

- En absence de circuit absorbant dans le graphe, l'algorithme se termine nécessairement à l'issue de l'itération n ($k = n$) car, au pire des cas, le PCC de s vers tous les autres sommets est un chemin élémentaire possédant $(n-1)$ arcs.
- Si une ou plusieurs marques sont modifiées à l'itération n ($r^+(m) \neq \emptyset$), cela signifie qu'il existe un circuit absorbant.

III.1.3 Algorithme de Floyd-Warshall :

- Cet algorithme permet de calculer le PCC entre tous les couples de sommets dans un graphe orienté sans circuit absorbant de longueur quelconque.
- Numéroté les sommets de 1 à n ($|X| = n$)
- Soit la matrice $A = \{a_{ij}\}$ de taille $n \times n$ définie initialement comme suit:

$$a_{ij} = \begin{cases} 0 & \text{si } i = j \\ lij & \text{si } (i, j) \in U \\ +\infty & \end{cases}$$

- Cet algorithme permet de calculer le PCC de la façon suivante:
- A la première itération, on cherche le PCC entre chaque couple (i, j) passant éventuellement par le sommet 1 ;
- A l'itération k (avec $l > 1$), on cherche le PCC entre chaque couple (i, j) passant par des sommets d'indice inférieur ou égal à k .

III.1.4 Comparaison :

Ces algorithmes se différencient par un certain nombre de caractéristiques (tableau 1):

- **Dijkstra** : sert à résoudre le problème du plus court chemin entre deux sommets d'un graphe connexe dont le poids lié aux arêtes est positif ou nul. Sa complexité est d'ordre $O(n)$ où n est le degré du graphe (nombre des sommets)
- **Bellman-Ford** : calcule des plus courts chemins depuis un sommet source donné dans un graphe orienté pondéré.

Contrairement à l'algorithme de Dijkstra, l'algorithme de Bellman-Ford autorise la présence de certains arcs de poids négatif et permet de détecter l'existence d'un circuit absorbant, (c'est-à-dire de poids total strictement négatif, accessible depuis le sommet source). La complexité de l'algorithme est en $O(|S| |A|)$ où $|S|$ est le nombre de sommets, $|A|$ est le nombre d'arcs [PPCC].

- **Floyd-Warshall** : permet de déterminer les distances des plus courts chemins entre toutes les paires de sommets dans un graphe orienté sans circuit absorbant de longueur quelconque, en temps cubique en le nombre de sommets [BER, 59]. L'algorithme de Floyd-Warshall est de complexité $O(n^3)$.

| Algorithme | Type du PCC | Propriétés du graphe | | Avantages | Inconvénients | Complexité en moyenne |
|---------------------|-----------------------------------|---------------------------------------|-----------------------------------|---|---|---|
| | | Type | Longueur | | | |
| Dijkstra | D'un sommet à tous les autres | Graphe orienté (et non orienté) | Longueur positive | - Recherche toujours le chemin ayant le poids le plus faible - Facile à comprendre et à mettre en place | - Demande une certaine masse de traitements quand le graphe devient grand - Retourne des résultats négatifs lors de l'affectation aux liaisons un poids négatifs | $O(m + n \log n)$ où n est le nombre de sommets et m est le nombre des arcs |
| Bellman-Ford | | Graphe orienté | Longueur quelconque (nombre réel) | -Le cout est minimisé lors de la construction d'un réseau en utilisant cet algorithme -Maximiser les performances du système | - Ne prend pas les pondérations en considération -Les changements dans la topologie du réseau entraînent une réponse lente | $O(n * m)$ |
| Floyd | Entre tous les couples de sommets | Graphe orienté sans circuit absorbant | | Performant quand il s'agit d'un graphe d'une grande densité | Performance réduite lors de son utilisation dans un graphe dense et les arcs ont des poids négatifs | $O(n^3)$ |

Tableau 1.1 Comparaison des algorithmes plus court chemin uni critère

Pour le problème de cheminement dans les réseaux de transport routier, l'algorithme le plus utilisé est celui de Dijkstra pour les raisons suivantes :

- L'algorithme de Dijkstra se repose sur le principe *d'exploration à partir du meilleur*, c'est à dire du meilleur prédécesseur visité. Il fournit donc à chaque étape la valeur définitive du pcc d'un sommet donné. Si on s'intéresse aux chemins de valeur minimale entre le sommet de départ A et un sommet particulier B, on peut arrêter le calcul dès que B est calculé, même si tous les sommets ne sont pas calculés.
- Il peut être utilisé quand tous les arcs ont une valeur non négative, c'est le cas dans un transport routier, la longueur de l'arc peut signifier la distance, le temps ou le coût qui sont des métriques positives.

- Il possède une complexité la plus faible (de l'ordre de n^2 dans le pire cas) au lieu d'être de l'ordre de n^3 comme pour l'algorithme Floyd. Il est donc utile dans des situations où il y a un très grand nombre de sommets

III.2 Algorithmes plus court chemin multicritères :

L'optimisation multicritère consiste à optimiser plusieurs critères ou objectifs. Ainsi vouloir l'itinéraire le plus rapide et le moins cher est généralement incompatible. Afin de gérer les objectifs multiples, un décideur (c'est-à-dire l'utilisateur du système) est obligé d'intervenir. Le moment auquel le décideur intervient définit les trois grandes familles d'algorithmes de résolution [GRA, 10] :

- **Approches à priori** : Consistent à construire une fonction objective qui englobe tous les objectifs afin de se ramener à une optimisation à un seul objectif, permettant ainsi d'utiliser des algorithmes mieux connus. Parmi les techniques les plus utilisées, nous trouvons la linéarisation des objectifs.

La somme pondérée de chaque objectif est l'approche la plus simple. Celle-ci a du sens si chaque objectif peut être ramené à une même unité de mesure (dans une entreprise, un retard d'une heure peut être mesuré en tant que perte d'argent). Cependant ce n'est pas toujours possible.

- **Approches interactives** : Consistent à proposer une solution au décideur qui choisira ensuite sur quels objectifs il souhaiterait améliorer la solution. Le système propose alors une nouvelle solution. Le processus est répété jusqu'à ce que le décideur soit satisfait. Cette approche a été appliquée au problème de plus court chemin multi objectif dans Gabrel, Vanderpooten [GAV, 02] ou encore Galand [GAL, 06].

Il faut cependant noter qu'il ne s'agit pas d'une approche a priori pour laquelle l'utilisateur change à chaque itération le paramétrage. Afin d'améliorer les performances, les approches interactives sont capables d'exploiter les résultats obtenus dans le calcul précédent. Elles sont donc surtout adaptées dans le cas où le temps de calcul est particulièrement important.

- **Approches à posteriori** : Consistent à présenter l'ensemble des solutions au décideur qui choisira celle qu'il considère comme la plus adaptée. Cette approche a l'avantage de n'éliminer aucune solution qui aurait pu intéresser le décideur. Cependant, le décideur souhaite généralement devoir uniquement choisir entre quelques solutions et

non entre plusieurs milliers, d'autant plus que de nombreuses solutions sont très similaires.

Deux techniques peuvent être utilisées pour une approche a posteriori :

- **Méta heuristiques** : l'adaptation de méta heuristique à une approche multi objectif concentre la majorité des efforts de recherche dans le domaine de l'optimisation multi objectif, en particulier en se basant sur l'algorithme génétique « SPEA2 »
- **Approches exactes** : la première approche est proposée par Hansen [HAN, 80 and MAR, 84] pour résoudre le plus court chemin bi critère. L'algorithme de Martins en est l'extension immédiate pour un nombre arbitraire d'objectifs

Pour le problème de cheminement dans les réseaux de transport routier, l'approche posteriori est la plus utilisée. Pour cela, nous allons détailler les deux algorithmes les plus connus dans cette approche, à savoir l'algorithme génétique et l'algorithme de Martins.

III.2.1 Algorithme de Martins

Pour le problème du plus court chemin multi objectif, l'algorithme de *Martins* [MAR, 84] est le plus connu. Il s'agit d'une généralisation de l'algorithme de Dijkstra au cas multi objectif.

Le principe du fonctionnement de l'algorithme de Martins est le suivant :

- Définir une étiquette (label) comme un triplet (nœud, valeurs des objectifs, prédécesseur). À la fin de la résolution, pour chaque nœud il y aura plusieurs labels non dominés.
- Le prédécesseur permet de reconstruire l'itinéraire à suivre pour obtenir une telle valeur d'objectif. Ainsi l'étiquette (u, c, e) signifie qu'au nœud u il existe un chemin de coût c en arrivant par l'étiquette e.
- L'ensemble Q contient donc un ensemble d'étiquettes. À chaque itération, l'étiquette (u, c, e) ayant la plus petite valeur d'objectifs dans l'ordre lexicographique est choisi.
- Cette étiquette est mémorisé dans une archive P car elle correspond à une solution Pareto-optimale. D'ou il ne peut pas exister d'autre étiquette ayant un coût lexicographique plus petit ; il s'agit donc d'une solution non-dominée.
- Pour chaque successeur v, une nouvelle étiquette est créée (v, c + c[u][v], e2). En- fin, on élimine toutes les étiquettes dominées par une autre.

- L'algorithme termine lorsque l'ensemble Q est vide. Étant donné que le nombre d'étiquettes d'un nœud peut devenir exponentiel, le temps d'exécution de l'algorithme est donc également exponentiel.

III.2.2 Algorithme SPEA2

Le principe du fonctionnement de l'algorithme génétique SPEA2 [MPJ, 07] est le suivant :

- Les chromosomes sont représentés par des entiers correspondants aux arcs d'un chemin. Les chromosomes sont donc de longueur variable.
- La population initiale est obtenue en partant de la source, puis en tirant récursivement aléatoirement un nœud parmi les successeurs du dernier nœud jusqu'à atteindre le puits.
- Le croisement est généré en tirant aléatoirement deux individus jusqu'à ce que les deux individus possèdent un nœud en commun pour servir de pivot au croisement.
- La mutation est obtenue en sélectionnant aléatoirement un nœud de l'individu et en supprimant tout le chemin à partir de ce nœud jusqu'au puits. Le reste du chemin est reconstruit de la même manière qu'à l'initialisation.

III.2.3 Comparaison

Il est très difficile d'adapter les algorithmes génétiques au problème du plus court chemin, à cause du grand nombre de contraintes inhérentes à ce problème : une solution générée aléatoirement n'a pratiquement aucune chance d'être réalisable. La grande majorité des problèmes de plus court chemin multi objectifs ont été résolus avec des méthodes exactes notamment l'algorithme de Martins. En effet, cet algorithme est une généralisation de l'algorithme de Dijkstra d'où il est facile à mettre en œuvre.

| Algorithme | Avantages | Inconvénients |
|----------------|---|---|
| SPEA2 | Le seul algorithme génétique qui a pu être appliqué pour résoudre le problème du plus court chemin multi objectif | - Difficile d'adapter ces approches au problème du plus court chemin, à cause du grand nombre de contraintes inhérentes à ce problème - La solution générée aléatoirement a peut de chance d'être réalisable |
| Martins | Facile à mettre en œuvre vu qu'il s'agit d'une généralisation de l'algorithme de Dijkstra au cas multi objectif | Quand le nombre d'étiquettes d'un nœud devient exponentiel, le temps d'exécution de l'algorithme est donc également exponentiel. |

Tableau 1.2 Comparaison des algorithmes plus court chemin multicritère

IV. Conclusion

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 l'utilisation des techniques d'optimisation multi objectif.

Le chapitre suivant de notre mémoire concerne la deuxième partie de notre système qui est les systèmes de recommandations en général et dans le domaine de transport de marchandises en particulier ainsi que les recherches antérieures faites dans ce domaine.

CHAPITRE II :
SYSTEMES DE RECOMMANDATION DANS LE
DOMAINE DU TRANSPORT

Introduction

Notre quotidien nous expose fréquemment à des situations qui nous conduisent à prendre des décisions et à opérer des choix : Nourriture, vêtements, voiture, lycée pour nos enfants, lieu de vacances etc. qu'ils soient cruciaux ou secondaires. Nos choix sont dictés par nos connaissances, préférences, et par les informations dont nous disposons.

Auparavant il y'avait un manque d'informations car même lorsque celles-ci existaient, les moyens d'y accéder étaient peu développés. Aujourd'hui, avec Internet, nous possédons le plus grand entrepôt de données que l'humanité ait pu connaître.

Le Web 1.0 a mis à la disposition du grand public un énorme flux d'informations entraînant une multiplicité des choix qui ne facilite pas toujours l'accès aux données pertinentes pour les utilisateurs. Ainsi, le domaine de la recherche d'information est né pour rendre Internet plus facilement exploitable. Son but principal était d'ordonner les informations disponibles selon leur pertinence relativement à une requête. Le Web 2.0 a fait son apparition, accompagné d'outils plus interactifs comme les forums et les blogs. Le grand public qui était jusque là passif dans le Web 1.0, est devenu actif dans le Web 2.0. Il est désormais devenu aisé pour tout le monde d'ajouter une information ou de partager un avis. Pour s'adapter à ce changement, les techniques de recherche d'informations ont dû intégrer des aspects sociaux en se focalisant sur l'utilisateur et son comportement. Les systèmes de recommandation (SR) en font partie. Contrairement aux systèmes de recherche d'informations qui attendent la requête de l'utilisateur pour agir, les SR tentent d'anticiper les besoins des utilisateurs et leur proposent des items (articles, ressources, produits, pages web, etc.) qu'ils sont susceptibles d'apprécier. En effet, l'objectif d'un SR est de recommander à ses utilisateurs des items sélectionnés parmi un large choix et censés être appréciés par eux. Il tente de prédire si un utilisateur donné appréciera ou non un item. Pour parvenir à un tel but, un SR a besoin d'accumuler des données sur les utilisateurs et/ou les items disponibles. Il sauvegarde les traces de ses utilisateurs, avant d'y appliquer des méthodes statistiques pour prédire leur futur comportement à l'égard d'items. De leur côté, les utilisateurs du SR s'en servent pour des objectifs différents, comme trouver des items convenables, s'exprimer, améliorer son profil, tester la crédibilité du système, ou simplement naviguer. Ayant différents caractères et habitudes, ils se comportent de manières très variées pour réaliser leurs objectifs. Par la suite, le web social a émergé au sein du web 2.0, changeant la manière dont les internautes s'expriment sur le web. Les réseaux sociaux, en pratique, permettent à ses utilisateurs de partager et de propager l'information, d'interagir les uns avec les autres, et d'établir

mutuellement des relations d'amitié et de confiance. Ceci engendre un nouveau type de données exploitables par les SR.

Dans ce chapitre, nous allons définir les systèmes de recommandations et analyser les différentes approches utilisées lors de la mise en place d'un système de recommandation. Ensuite, nous décrivons ainsi les principaux travaux qui ont été élaborés sur les systèmes de recommandation notamment dans le domaine de transport de marchandises.

I. Les systèmes de recommandation

I.1 Définition

Les Systèmes de Recommandation (SR) [REV, 97] permettent de proposer des items pertinents à l'utilisateur. En d'autres termes, ils tentent de prédire pour un utilisateur l'intérêt d'un item. L'item dans ce contexte peut être un produit à acheter, un morceau de musique à écouter, un film à regarder, un livre à lire, une page web à consulter, ou bien autre chose.

La figure 2.1 présente un exemple d'un SR qui recommande des films susceptibles à intéresser l'utilisateur [SRAD, 2010].

Afin de pouvoir fournir des recommandations personnalisées, le SR doit connaître les préférences de chaque utilisateur. Il tente, alors, d'acquérir les informations nécessaires pour construire des profils d'utilisateurs. En particulier, il exploite les traces laissées par les utilisateurs eux mêmes. Il collecte les traces laissées explicitement ou implicitement [Oard and Kim1998]. Les traces explicites sont fournies volontairement par l'utilisateur [RRS, 11]. Les traces implicites sont collectées en traçant les actions spontanées de l'utilisateur pendant la navigation [OAK, 98].

Produits fréquemment achetés ensemble



Prix pour les trois: **EUR 45,94**

[Ajouter ces trois articles au panier](#)

[Afficher la disponibilité du produit et le mode de livraison](#)

- ✓ Cet article : Avatar DVD ~ Sam Worthington
- ✓ [Sherlock Holmes](#) DVD ~ Robert Downey Jr.
- ✓ [Invictus](#) DVD ~ Morgan Freeman

Qu'achètent les clients après avoir consulté cet article ?

-  **95%** achètent l'article présenté sur cette page :
[Avatar](#) ★★★★★ (339)
EUR 13,98
-  **2%** achètent
[Sherlock Holmes](#) ★★★★★ (21)
EUR 15,98
-  **1%** achètent
[Shutter Island](#) ★★★★★ (21)
EUR 14,98
-  **1%** achètent
[La Princesse et la Grenouille](#) ★★★★★ (17)
EUR 15,98

[Découvrez des articles similaires](#)

Figure 2.1. Exemple d'un système de recommandation [SRAD, 2010]

I.2 Construction d'un système de recommandation

La construction d'un système de recommandation passe généralement par 3 étapes:

1. La première consiste à recueillir de l'information sur l'utilisateur.
2. La deuxième consiste à bâtir un modèle utilisateur contenant l'information recueillie.
3. La troisième consiste à extraire une liste de recommandations.

I.2.1 Collecte d'Information sur l'utilisateur :

Pour être pertinent, un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur. On distingue deux formes de collecte de données, explicite (filtrage actif) ou implicite (filtrage passif). Ces deux formes sont comparées dans le tableau suivant :

| Forme | Description | Exemple | Avantages | Inconvénients |
|--|--|---|--|---|
| Explicite (Filtrage actif) | repose sur le fait que l'utilisateur indique explicitement intérêts au système. | demander à un utilisateur de commenter, noter, aimer des contenus qui l'intéressent. On utilise souvent une échelle de <i>ratings</i> | capacité à reconstruire l'historique d'un individu et capacité à éviter d'agréger une information qui ne correspond pas à ce dernier | les informations recueillies peuvent contenir un biais dit de déclaration. |
| Implicite (Filtrage passif) | repose sur une observation et une analyse des comportements de l'utilisateur effectué de façon implicite dans l'application qui embarque le SR | analyser la fréquence de consultation d'un contenu par un utilisateur | aucune information n'est demandée aux utilisateurs, toutes les informations sont collectées automatiquement | les données récupérées sont plus difficilement attribuables à un utilisateur et peuvent donc contenir des biais d'attribution |

Tableau 2.1. Formes de collecte d'informations sur l'utilisateur dans un SR

I.2.2 Modèle Utilisateur :

Le modèle utilisateur se présente généralement sous forme de matrice. On peut le représenter comme un tableau qui contient des données recueillies sur l'utilisateur associées aux produits disponibles sur le site web.

Un autre point important est comment le temps influence le profil de l'utilisateur. Les intérêts des utilisateurs, généralement, évoluent au cours du temps. Les données du modèle utilisateurs devraient donc constamment être réajustées pour rester conformes aux nouveaux centres d'intérêts de l'utilisateur [RS].

I.2.3 Liste de recommandations :

Pour extraire une liste de suggestions à partir d'un modèle utilisateur, les algorithmes utilisent la notion de mesure de similarité entre objets ou personnes décrits par le modèle utilisateur. La similarité a pour but de donner une valeur ou un nombre (au sens mathématique du terme) à la ressemblance entre deux choses. Plus la ressemblance est forte, plus la valeur de la similarité sera grande. A l'inverse, plus la ressemblance est faible, et plus la valeur de la similarité sera petite.

I.3 Types de système de recommandation :

Il existe trois approches possibles pour un système de recommandation:

- **Recommandation Objet** (Filtrage basé contenu)
- **Recommandation Sociale** (Filtrage collaboratif)
- **Recommandation Hybride**

I.3.1 Recommandation Objet :

Il s'agit de recommander des objets (ou contenus) en se basant sur les qualités et les propriétés intrinsèques de l'objet lui-même et en les corrélant avec les préférences et intérêts de l'utilisateur. Ce type de système permet d'extraire un certain nombre de caractéristiques et attributs propres à un contenu, afin de pouvoir recommander à l'utilisateur des contenus additionnels possédant des propriétés similaires. Cette méthode crée un profil pour chaque objet ou contenu, c'est-à-dire un ensemble d'attributs/propriétés qui caractérisent l'objet [MAT, 12].

I.3.1.1 Exemple :

Dans le cas d'un site de vente de livre en ligne, on se base sur les caractéristiques du livre pour effectuer des recommandations, comme par exemple le sujet que traite l'ouvrage, son genre, son auteur, l'éditeur, etc. On pourrait ainsi recommander le livre Harry Potter à un utilisateur, si on sait d'une part que ce livre est un roman fantastique et d'autre part que l'utilisateur aime les romans fantastiques. Un système de recommandation pourra donc accomplir cette tâche seulement s'il a à disposition deux types d'information :

- 1) la description des caractéristiques du livre (Tableau 2.2 [Profile livres])

Chapitre II : SYSTEMES DE RECOMMANDATION

2) un profil utilisateur qui décrit les intérêts (passés) de celui-ci en termes de préférence de type de livre (Tableau 2.3 [Profile utilisateur])

La tâche de recommandation consiste donc à déterminer les livres qui correspondent le mieux aux préférences de l'utilisateur.

| Titre | Genre | Auteur | Type | Prix | Mots clés |
|----------------------|-------------------|------------------|-----------|-------|---|
| The Night of the Gun | Memoir | David Carr | Paperback | 29.90 | Press and journalism. Drug addiction. Personal memoirs. New York |
| The Lace Reader | Fiction, Mystery | Brunonia Barry | Hardcover | 49.90 | American contemporary Fiction, detective. historical |
| Into the Fire | Romance, Suspense | Suzanne Brickman | Hardcover | 45.90 | American fiction, murder, neo-Nazism |

Tableau 2.2. Profil Livres [Profile livres]

| Titre | Genre | Auteur | Type | Prix | Mots clés |
|-------|----------------------|----------------|-----------|-------|--------------------------------|
| ... | Fiction. Suspense | Brunonia Barry | Paperback | 25.65 | Detective, murder, New York |

Tableau 2.3. Profil Préférences/Utilisateurs [Profile utilisateur]

I.3.2 Recommandation Sociale :

La recommandation sociale permet de recommander des choses sur la base du comportement passé des utilisateurs similaires, en effectuant une corrélation entre des utilisateurs ayant des

préférences et intérêts similaires. On utilise des méthodes qui collectent et analysent des données sur le comportement, les activités, les préférences des utilisateurs et des algorithmes tentent de prédire ce que l'utilisateur aimera en cherchant des utilisateurs qui ont les mêmes comportements que l'utilisateur à qui l'on souhaite faire des recommandations. L'idée sous-jacente est de dire que si une personne A a la même opinion (ou les mêmes goûts) qu'une personne B sur un objet x , alors la personne A a plus de chance d'avoir la même opinion que B sur un autre objet y , plutôt que d'avoir la même opinion que quelqu'un choisi au hasard pour l'objet y . L'idée de base est donc de dire que si des utilisateurs ont partagés des mêmes intérêts dans le passé, il y a de fortes chances qu'ils partagent aussi les mêmes goûts dans le futur [MAT, 12].

Exemple :

Comme illustré dans le Tableau 2.4, un tableau ou matrice de *notation* (rating) de films, avec sur un axe les utilisateurs et sur un autre les films. Chaque cellule de la matrice contient le rating donné par un utilisateur pour un film. Un signe “+” indique que l'utilisateur a aimé le film, un signe “-” qu'il ne l'a pas aimé, pas de signe signifie qu'il n'a pas d'avis particulier sur ce film. Pour pouvoir prédire si Ken apprécierait le film “*Fargo*” et éventuellement lui recommander ce film, on compare les ratings de Ken à ceux des autres utilisateurs sélectionnés. On peut alors voir que Ken et Mike ont des ratings identiques, et que Mike a aimé le film *Fargo*, on pourrait alors prédire que Ken aimera aussi ce film et lui faire cette suggestion.

| | Amy | Jef | Mike | Chris | Ken |
|--------------|-----|-----|------|-------|-----|
| The Piano | - | - | + | | + |
| Pulp Fiction | - | + | + | - | + |
| Clueless | + | | + | + | - |
| Cliffhanger | - | - | - | - | |
| Fargo | - | + | + | - | ? |

Tableau 2.4. Notations films

I.3.3 Recommandation Hybride :

Cette recommandation est une combinaison des deux approches décrites ci-dessus. Elle permet de résoudre des problèmes comme le *démarrage à froid* et la *rareté* qu'on retrouve dans une approche de recommandation uniquement sociale. D'autre part, si par exemple on considère deux utilisateurs avec les mêmes goûts mais qui n'ont pas évalué ou *noté* des objets en commun, un filtrage collaboratif pur ne les considérera pas comme similaires ou voisins. Rappelons que la mesure de similarité standard ne prend en compte que les éléments pour lesquels l'utilisateur actif et l'utilisateur à comparer ont effectué une notation. L'idée est alors de pouvoir assigner une valeur par défaut aux éléments qui ont été noté seulement par un des deux utilisateurs, afin d'améliorer la qualité de prédiction en cas de rareté. En appliquant préalablement un algorithme de recommandation objet sur les contenus pour en exploiter leurs descriptions et caractéristiques, accompagné ensuite d'un algorithme de recommandation sociale pour effectuer les recommandations peut aider à résoudre ces limitations. Autrement dit, pour les cas de rareté, lorsque peu d'items ont été évalués par les utilisateurs et qu'un filtrage collaboratif n'est pas possible, ce qu'on fait, c'est qu'on assigne en premier lieu un pseudo-rating ou vote artificiel par défaut à l'utilisateur sur les contenus disponibles en utilisant préalablement un algorithme objet, puis on applique ensuite sur la matrice (contenant peu de vrais rating et beaucoup de pseudo-ratings) un filtrage collaboratif [MAT, 12].

Exemple :

Amazon¹ : utilise les deux approches (sociale et objet). Amazon possède un système très sophistiqué, les recommandations sont d'une part personnalisées en se basant sur le comportement individuel passé de l'utilisateur (historique de navigation et historique d'achat), et d'autre part Amazon utilise aussi les caractéristiques de l'article lui-même (recommandation objet) et les comportements d'autres personnes (recommandation sociale). Tous ceux qui ont déjà fait un achat sur Amazon ont probablement dû lire le message de la part d'Amazon qui nous dit "les gens qui ont acheté x ont aussi acheté y". Cette approche est tout simplement l'approche basée contenu qu'on a décrite auparavant. Il semblerait que le système de recommandation d'Amazon représente plus du 30% du CA global de la société (en 2009)!

¹ <https://www.amazon.fr/>

What Do Customers Buy After Viewing This Item?

52% buy the item you viewed

19% buy this alternative

5% buy this alternative



[Geoffrey Beene Wrinkle Free Fitted...](#)



[Geoffrey Beene Fitted Wrinkle Free...](#)



[Arrow Wrinkle Free Sateen Solid Dress...](#)

| Amazon Personalized Recommendations System | |
|--|--|
| Your Browsing History | Your Purchase History |
| Actual Items | New releases (Item recommendation) |
| Related Items (Item recommendation) | Related Items (Item recommendation) |
| Others Purchased (Social Recommendation) | Others Purchased (Social recommendation) |

Figure 2.2. Système de recommandation AMAZON [Amazon RE]

I.3.4 Comparaison des méthodes de recommandations :

Le tableau suivant permet de comparer les trois méthodes de recommandations décrites ci-dessus en citant les avantages et les inconvénients de chaque méthode.

| Type de recommandation | Avantages | Inconvénients |
|-------------------------------|---|---|
| Recommandation Objet | <ul style="list-style-type: none"> - Pas besoin d'une large communauté d'utilisateurs pour pouvoir effectuer des recommandations. - Une liste de recommandations peut être générée même s'il n'y a qu'un seul utilisateur. | <p>Les caractéristiques et propriétés, sont fournis pas les créateurs du livre ou du film. La problématique, c'est l'acquisition de caractéristiques subjectives et qualitative des propriétés, comme le style et le design, qui peuvent difficilement être acquises automatiquement, et devront plutôt être introduites manuellement avec tout ce que ça implique, comme le coût, les éventuelles erreurs.</p> |
| Recommandation Sociale | <ul style="list-style-type: none"> - Cette approche n'exploite pas ou ne demande aucune connaissance sur les contenus eux-mêmes. - Pas besoin de se baser sur l'analyse des propriétés intrinsèques d'un livre ou d'un contenu, la recommandation sociale est capable de recommander des contenus sans avoir besoin de comprendre le sens ou la sémantique du contenu lui-même. | <p>-Démarrage à froid : Le lancement d'un service de recommandation peut souffrir au début du manque d'utilisateurs et d'informations sur ceux-ci [ZLT, 01].</p> <p>-Rareté : le nombre de produits ou contenus est énormes sur certaines plates-formes, et même les utilisateurs les plus actifs auront noté ou valorisé qu'un tout petit sous-ensemble de toute la base de données. Donc, même l'article le plus populaire n'aura que très peu de bonnes notes.</p> |
| Recommandation Hybride | <p>Le but de cette approche est de réunir le meilleur des deux approches objet et sociale explicités ci-dessus. L'idée est de compenser le <i>démarrage à froid</i> et la <i>rareté</i>.</p> | <p>Une nouvelle approche, qui se montre difficile à mettre en œuvre</p> |

Tableau 2.5. Tableau comparatif des méthodes de recommandations

Dans notre travail, nous nous intéressons à la recommandation sociale qui est largement utilisée avec les réseaux sociaux, cette méthode désormais compte des relations de confiance entre les utilisateurs, ce qui nous a conduit à l'appliquer dans le domaine de transport, nous proposons un moteur de recommandation sociale qui prend en compte les avis des clients (en attribuant un poids à chaque client selon le nombre de sa participation) sur les différents transporteurs existants sur la plateforme. Ainsi, quand on choisit un itinéraire, on voit apparaître les transporteurs les mieux notés par les clients, afin d'offrir un support d'aide à la décision aux clients lors de la demande de transport.

I.4 Présentation des méthodes de recommandation sociale

On va présenter dans cette partie les différentes techniques utilisées dans la recommandation sociale ainsi que les formules de calcul du système de notation (ratings).

La motivation du *filtrage collaboratif* permet d'étendre la notion de *bouche à oreille entre amis* à des milliers de personnes sur Internet, qui sont susceptibles de vous donner leur avis. Les objets pour lesquels on veut évaluer l'intérêt des internautes peuvent être de toute sorte: films, restaurants, jeux, articles, etc. Typiquement, les étapes du système de recommandation sont les suivantes [CAN, 01]:

1. collecter les appréciations des utilisateurs sur les articles qu'ils parcourent,
2. intégrer ces informations dans les profils d'utilisateur,
3. et utiliser ceux-ci ensuite pour aider les utilisateurs dans leurs prochaines recherches.

I.4.1 L'évaluation des articles par les utilisateurs

Tout d'abord, la première notion à prendre en compte est celle du taux de satisfaction des utilisateurs pour les articles qu'ils choisissent. En effet, pour caractériser un profil d'utilisateur, la première chose à comprendre, c'est si ce qu'il a lu lui a plus ou moins plu, ou pas du tout. Pour cela, deux choix sont possibles: demander à l'utilisateur de noter lui-même les articles à partir d'une échelle de notes fixée (par exemple de 0-Nul à 5-génial), ou bien évaluer automatiquement cette note, grâce aux informations que l'on peut récolter à partir des données.

I.4.1.1 Évaluation avec investissement de l'utilisateur

L'idéal pour apprendre les goûts, et donc le profil d'un utilisateur, est que celui-ci nous donne son avis lui-même sur le plus des articles possible de la base de données. Pour cela, la plupart

des systèmes de recommandation demandent à leurs nouveaux utilisateurs de participer à la définition de leur profil initial, en passant d'abord par une phase de notation, selon une échelle de notes fixée, d'articles qui leurs sont proposés. Pour cette méthode, un problème est alors de trouver l'échelle de notes la plus adaptée pour permettre une notation aussi précise que possible des articles par les utilisateurs. Remarquons à ce niveau que la façon de noter des articles varie fortement selon l'utilisateur. Par exemple, certains ne notent que les articles qui leur plaisent, alors que d'autres utilisent l'entièreté de l'échelle proposée. Il faut donc bien comprendre ce que chaque note signifie pour chaque utilisateur.

Il existe plusieurs systèmes de notations qu'on va décrire par la suite.

I.4.1.2 Évaluation sans participation de l'utilisateur

Si l'on veut éviter à l'utilisateur de s'investir dans la définition de son profil, au lieu de lui demander si une page lui a plu ou non, on peut le deviner, en se servant des informations que l'on peut obtenir lors de son passage sur le site. En effet, chaque fois qu'un internaute visite une page d'un site web, il laisse derrière lui des traces de son passage. Pour traiter ce problème, P.K. Chan [PHI, 99] propose de développer une *Page Estimateur d'intérêt*, pour prédire si la page proposée a été appréciée ou non. Pour cela, quatre sources générales d'informations ont été identifiées: *l'historique* et *le marque-page* du côté utilisateur; *l'accès log* et *le contenu des pages* du côté serveur.

I.4.2 Stockage des données

Les notes obtenues sont alors stockées dans la base de données utilisateur. Typiquement, on représente ces données par une matrice de notes des utilisateurs sur les pages qu'ils ont parcourues. La base de données utilisateur se présente alors sous la forme du tableau 2.6 suivant :

| | Article 1 | Article 2 | Article 3 | Article 4 | Article 5 |
|---------------|-----------|-----------|-----------|-----------|-----------|
| Utilisateur 1 | | | 5 | 4 | |
| Utilisateur 2 | | | 3 | 4 | 5 |
| Utilisateur 3 | | | 4 | 4 | 5 |
| Utilisateur 4 | 5 | 3 | | | 5 |
| Utilisateur 5 | 5 | 4 | | | 5 |

Tableau 2.6 : Une matrice des notes attribuées aux articles par les utilisateurs [CAN, 01]

Formellement, cette matrice correspond à un ensemble de votes $v_{i,j}$ des utilisateurs (i) sur les articles (j).

Par exemple ici, on a $v_{3,3}=4$. v_i correspond au vecteur décrivant l'utilisateur i (l'ensemble de ses votes, $\{v_{i,j} | i\}$). Dans notre exemple, on a $v_3=(0,0,4,4,5)$. On utilise la notation I_i pour désigner l'ensemble des articles pour lesquelles l'utilisateur i a voté. On a alors $I_3= \{3, 4,5\}$.

Et nous noterons $p_{i,j}$ l'estimation du vote de l'utilisateur i sur l'article j , le but étant que $p_{i,j}$ soit le plus proche possible de $v_{i,j}$.

I.4.3 Les méthodes de filtrage collaboratif

Deux classes d'algorithmes de filtrage collaboratif, exploitant ensuite ces données pour aider les utilisateurs dans leurs futures recherches, ont été distinguées [JDK, 98] :

- les *algorithmes basés sur la mémoire* utilisent l'entièreté de la base de données utilisateur pour faire des prédictions,
- alors que les *algorithmes basés sur les modèles* utilisent la base de données utilisateur pour estimer ou apprendre un modèle, qui sera ensuite utilisé pour les prédictions.

I.4.3.1 Méthodes basées sur la mémoire

Les algorithmes de recommandation sociale utilisent généralement différentes variantes d'un mécanisme basé sur le voisinage proche. Dans cette approche, un nombre d'utilisateurs (plus proches voisins) est identifié et sélectionné sur la base de la similarité de leurs intérêts et préférences avec l'utilisateur actif. On utilise alors principalement la notation (par ex: films) de ces utilisateurs "voisins" pour calculer des similarités avec l'utilisateur actif. Pour chaque produit p que l'utilisateur n'a pas encore vu, une prédiction est faite en se basant sur les ratings de p assignés par le panel d'utilisateurs voisins. Cette méthode suppose deux conditions initiales :

1. On assume que si des utilisateurs ont eu des goûts similaires dans le passé, ils auront aussi des goûts similaires dans le futur.
2. Les préférences des utilisateurs restent stables et cohérentes dans le temps.

Au lieu de se baser uniquement sur l'utilisateur le plus semblable, la prédiction est normalement calculée à partir de la moyenne pondérée des ratings de plusieurs utilisateurs. Le poids donné au rating de chaque utilisateur est déterminé par le degré de corrélation entre cet utilisateur et l'utilisateur pour qui on désire faire la recommandation. Pour mesurer le degré de

corrélation entre deux utilisateurs, on fait appel généralement au coefficient de corrélation de Pearson r .

Le coefficient de corrélation de Pearson se calcule à partir des valeurs présentes dans la matrice et peut prendre une valeur allant de +1 (forte corrélation positive) à -1 (forte corrélation négative). Par exemple, un coefficient de corrélation 0.70 entre 2 utilisateurs indiquerait une bonne similarité de leurs intérêts.

On n'est pas non plus limité à la *notation*, on peut aussi se baser sur des données plus implicites en observant le comportement de l'utilisateur sur le site, par opposition à la récolte de données explicite comme l'est la *notation*. On peut observer par exemple quelle musique il a écouté, quel article il a lu, et on croise ses infos avec celles du reste des utilisateurs afin de lui proposer de nouvelles suggestions.

I.4.3.2 Méthodes basées sur le modèle

Cette autre approche, propose une inversion de l'approche basé mémoire. Au lieu de mesurer la corrélation entre des utilisateurs, les *notations* sont utilisées pour mesurer la corrélation entre les contenus (films), en s'aidant toujours du coefficient de corrélation de Pearson, mais cette fois-ci appliqué au contenu. Si par exemple les *notations* des deux films "film1" et "film2" ont une parfaite corrélation, c'est-à-dire qu'ils ont reçu les mêmes notations (positif, négatif ou nul) de la part des utilisateurs, on peut ainsi prédire qu'un nouveau utilisateur aimera le film "film1" car il a aimé "film2" (les notations des utilisateurs sur ces deux films sont parfaitement corrélés).

Pour le dire autrement, l'approche basée modèle propose de rechercher en premier lieu des contenus similaires et ensuite de faire une recommandation à l'utilisateur. Cette approche permet de faire un traitement préalable sur la matrice pour déterminer les contenus similaires et ainsi pouvoir effectuer des prédictions en temps réel, contrairement à l'approche basé mémoire très gourmande en mémoire.

Autant le coefficient de corrélation de Pearson est très utilisé pour déterminer des utilisateurs similaires (cas de la recommandation sociale basé mémoire), dans l'approche basée modèle, on utilise plutôt comme indicateur de mesure de similarité entre items un autre indicateur qu'on appelle de similarité cosinus ajustée. Les valeurs possibles pour cet indicateur vont comme dans le cas de mesure de Pearson de +1 (forte similarité positive) à -1 (forte similarité négative). Une fois que la similarité entre les items a été établie à l'aide de la similarité

cosinus ajustée, on peut alors prédire une notation pour un article grâce aux notations effectuées par l'utilisateur sur les articles similaires.

Dans l'approche basée modèle, l'idée est donc de construire à l'avance la matrice de similarité entre articles. Et en temps réel, on peut facilement ensuite déduire la prédiction de recommandation (la notation) d'un produit pour un utilisateur actif en déterminant depuis la matrice déjà construite quels sont les produits les plus similaires, et en calculant la valeur moyenne de notation sur ces produits effectués par les utilisateurs voisins.

I.4.4 Formules de notation à étoiles (Rating stars)

A la recherche d'améliorations et d'innovations dans l'optique de rendre les sites plus intéressants et plus attractifs, les systèmes de notation sont un excellent moyen de permettre aux utilisateurs d'interagir sur les pages en donnant leur avis.

Les étoiles sont des symboles souvent utilisés dans les systèmes de notation. On retrouve ce système de classement dans les critiques de cinéma, de télévision, de gastronomie ou dans certaines pages web pour permettre aux internautes de les évaluer.

Une échelle d'évaluation est définie avec un nombre maximal d'étoiles donné ; ainsi, un élément jugé médiocre n'a aucune étoile tandis qu'un excellent aura le maximum. On trouve parfois des demi-étoiles pour définir une note intermédiaire.



www.shutterstock.com · 298117844

Figure 2.3 Echelle de notation sur 5 [Rating stars]

I.4.4.1 La mise en place

Plusieurs formules existent permettant le calcul de la note moyenne d'un article noté par un ou plusieurs utilisateurs, on va décrire deux d'entre elles :

Formule1 : Calcul de la moyenne de notations

L'évaluation des articles avec échelle de notation est calculée selon la moyenne pondérée basée sur note assigné à chaque vote. La moyenne de notations est calculée comme le suit [Rating average] :

$$\frac{X1W1 + X2W2 + X3W3 \dots XnWn}{N}$$

Ou:

X = la valeur choisie sur l'échelle de notation

W= le nombre fois ou l'article a été noté

Formule2 : Estimation Bayésienne (Top 250 de l'Internet Movie Database)

Le top 250 de l'Internet Movie Database est un classement des deux cents cinquante meilleurs films de l'histoire du septième art selon les utilisateurs du site internet Internet Movie Database. Ce classement est reconnu comme l'un des plus fiables et révélateur sur les préférences des spectateurs grâce à la formule d'estimation Bayésienne utilisée pour l'établir et grâce au nombre de votants : certains films ont reçu plus d'un million de votes.

Présentation du classement :

Le classement comprend 250 entrées comportant une note sur 10 points. Cette note est basée sur les votes des utilisateurs du site et est calculée selon la formule d'estimation Bayésienne suivante :

$$W = \frac{Rv + Cm}{v + m}$$

Avec:

W = Note pondérée

R = Note du film donnée par les utilisateurs

v = Nombre de votes

m = Nombre minimum de votes pour figurer dans le classement

C = vote moyen (actuellement 7.0)

II.5 Travaux antérieurs sur les systèmes de recommandation dans le domaine de transport

Plusieurs SR ont été élaborés ces dernières années notamment dans le domaine de transport. On distingue les travaux de recherches des solutions commerciales. Dans ce qui suit, nous allons citer les plus importants d'entre eux.

II.5.1. Travaux de recherches (Solutions non commerciales)

- **Système FTMarket [LAZ, 09]**

Le système FTMarket (Freight Transportation Market) permet de FTMarket de gérer plusieurs types d'opérations de transport [LAZ, 09] [KLM, 06].

Lors de la recherche du plus court chemin cette approche propose à l'utilisateur cinq critères à choisir afin d'élaborer son plan de transport, les critères sont :

Express / Economique / Sécurité / Fiabilité / Définition utilisateur

Dans le cadre de ce travail [KLM, 06] [LAM, 06], un algorithme de construction de solutions optimales a été présenté pour la demande de transport désirée. Cet algorithme prend en considération le coût et la durée pour chaque sous-chemin.

La procédure de recommandation commence immédiatement après la construction susmentionnée des solutions alternatives. C'est un processus complexe qui est effectué dans trois phases majeures, qui sont comme le suit :

- évaluation de marchandises et les données de transport
- exploitation des données de transport à travers un processus d'exploration de données
- la sélection d'une méthode de recommandation ou bien la synthèse

Après la phase de classement, l'évaluation de chaque chemin est récupérée. Ce système récupère tout les chemins possibles qui peuvent être construits d'après une demande de transport. Ces chemins sont présentés à l'utilisateur par une interface appropriée.

- **Système [ROSE] [LZS, 09]**

ROSE (Service de routage) est une application pour téléphones mobiles, qui suggère des emplacements pour l'utilisateur et les guide via un transport public. Il réagit en temps réel à des retards dans le système de transport public et calcule des itinéraires de remplacement lorsque nécessaire. Pour la planification d'itinéraire, les auteurs ont proposé l'algorithme optimal h_{eu} afin d'incorporer les multiples préférences non monotones de utilisateur dans un

algorithme de type A*. Pour faciliter l'ensemble du processus de la planification d'un voyage, ROSE combine la recommandation des événements et lieux avec la navigation.

II.5.2 Solutions commerciales

Les Systèmes de recommandation sont maintenant utilisés de plus en plus par les sites de commerce électronique. On va citer dans ce qui suit quelques sites utilisant ces systèmes.

- **Transport for London (TfL)¹**

Transport for London (TfL) est l'organisme public local responsable des transports en commun de la ville de Londres et du Grand Londres, au Royaume-Uni¹. Cet organisme assure l'implantation d'un plan des transports et l'établissement d'un service régulier à travers Londres. Les TfL ont développé un planificateur électronique d'itinéraire qui permet aux utilisateurs de se retrouver entre les différents moyens de transports de la capitale anglaise.

- **Travelocity²**

Travelocity est une agence de voyages qui permet aux entreprises de planifier, d'acheter et de gérer leur voyage en fonction de leurs capacités en ligne et hors ligne. Travelocity, fournit des services de voyage en ligne pour les vacanciers et les voyageurs d'affaires aux États-Unis et au Canada. Il permet aux clients de trouver diverses offres de voyage, tels que des forfaits vacances, vols, hôtels, voitures, croisières, et billets d'avion; planifier et réserver des excursions vers diverses destinations en Asie, en Afrique, les États-Unis, le Canada, l'Amérique du Sud, en Europe, en Australie et dans le Pacifique Sud, au Mexique et en Amérique centrale et dans les Caraïbes.

- **Tripadvisor³**

TripAdvisor est un site web international d'origine américaine qui offre des avis et des conseils touristiques émanant de consommateurs (hôtels, restaurants, villes et régions, lieux de loisirs, etc.) et qui fournit également des outils de réservation de logements et de billets d'avion comparant des centaines de sites web afin de trouver les meilleurs prix.

¹ <https://www.tfl.gov.uk>

² <https://www.travelocity.com>

³ <https://www.tripadvisor.in>

II.5.3 Synthèse des travaux antérieurs :

Les travaux de recherches antérieures qui ont été élaborées auparavant se diffèrent selon :

- le type de la recherche : commerciale et non commerciale
- les algorithmes utilisés lors de la résolution du problème du plus court chemin en terme unicritère ou multicritères
- les critères à satisfaire
- les types de systèmes de recommandation utilisés

Les propriétés de chaque recherche sont résumées dans le tableau suivant :

Remarque : les points d'interrogations dans le tableau indiquent qu'on ne connaît pas l'algorithme utilisé lors de la résolution du problème du plus court chemin, et le type de système de recommandation utilisé surtout pour les solutions commerciales.

| | Type de transport | Cheminement | | Méthode de recommandation |
|--|----------------------------|--|------------|---|
| | | Critères | Algorithme | |
| FTMarket System's Recommendation Module | Transport de marchandises | Multicritères (Express- Economique -Sécurité- Fiabilité- Définition utilisateur) | Martins | Hybride (recommandation sociale la recommandation objet) |
| ROSE | Transport public | Unicritère | <i>heu</i> | ? |
| Transport for London (TfL) | Transport public | -chemin rapide -chemin avec moins de changement de stations | ? | ? |
| Travelocity | Transport public et aérien | -ligne directe -destinations multiples | ? | ? |
| TripAdvisor | Transport aérien | -meilleurs prix -vol sans escales | ? | ? |

Tableau 2.7 : Synthèse des travaux antérieurs sur les systèmes de recommandation dans le transport

III. Conclusion

Les systèmes de recommandation sont un moyen de recherche d'information aidant les utilisateurs de service en ligne à trouver des produits ou documents qui les intéressent.

Ils figurent en bonne place sur les sites web les plus fréquentés (Youtube, Facebook, IMDb, Amazon,...) couvrant les contenus culturels, réseaux sociaux, la presse et le commerce en ligne, les réservations de voyage,...

Dans cette partie, nous avons analysé les différentes approches utilisées lors de la mise en place d'un système de recommandation, ainsi que les objectifs divers et variés fixés par un système de recommandation.

CHAPITRE III :

**CONCEPTION DU SYSTEME DE GESTION ET DE
RECOMMANDATION DE TRANSPORT DE
MARCHANDISES (SGRTM)**

Introduction

Dans ce chapitre, nous exposons l'analyse de notre système en s'approfondissant dans la description détaillée de ses différentes fonctionnalités et leurs techniques d'implémentation.

I. Description générale

Notre projet consiste à la réalisation d'un Système de Gestion et de Recommandations de Transport de Marchandises (SGRTM) qui met en relation les demandeurs (clients) et les offreurs des moyens de transport (transporteurs). L'architecture de notre système est illustrée dans la figure suivante :

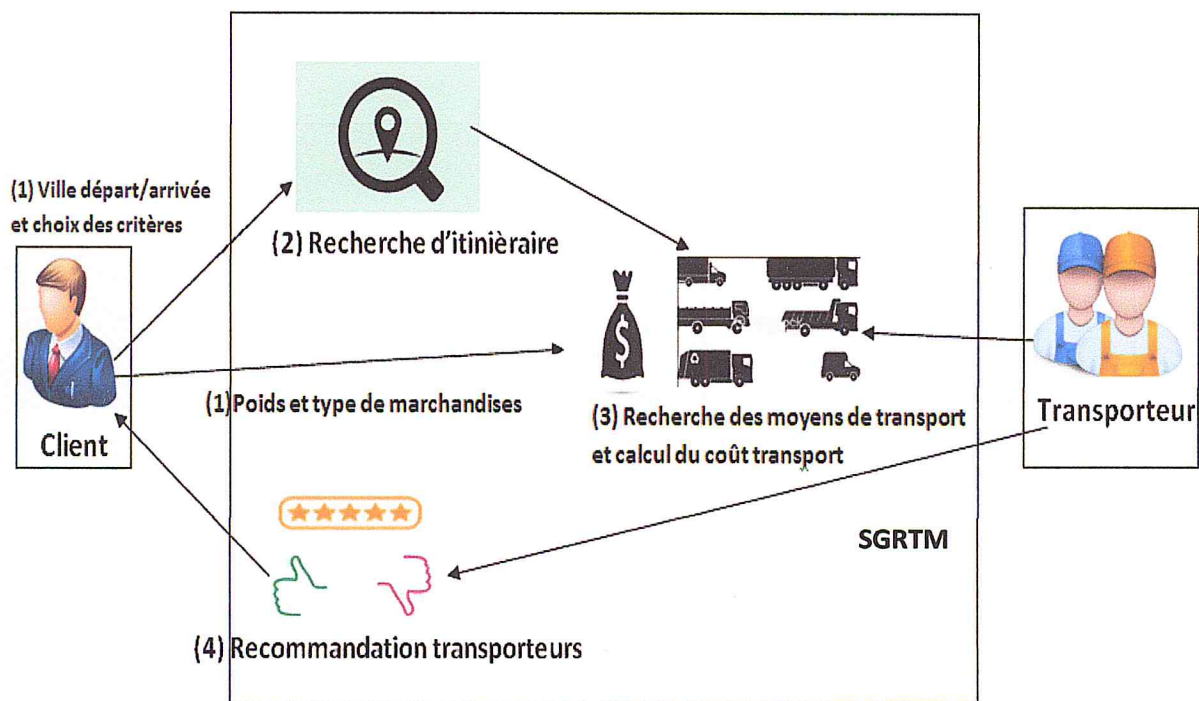


Figure 3.1 Description générale de l'architecture système

Notre SGRTM nécessite que le client introduit les informations essentielles afin d'accomplir une mission de transport. Ces informations se résument en :

- Itinéraire : ville de départ et ville d'arrivée.
- Choix de critères d'évaluation: unicitère (distance ou temps) et bicritères (distance et temps).
- Informations concernant la marchandise transportée : poids (kg/tonne) et type.

En se basant sur les informations introduites par l'utilisateur, nous cherchons d'abord les plus courts chemins (ou itinéraires) en termes de distance et/ou de temps. Ensuite, nous cherchons les moyens de transports adéquats selon le type de marchandise ainsi que les transporteurs propriétaires de ces moyens. Cette étape nous permet de calculer les coûts du transport selon les différentes tarifications fixées par les transporteurs (qui dépend du temps ou/et de distance). Enfin, nous indiquons les meilleurs transporteurs en se basant sur notre système de recommandation. Dans ce qui suit, nous allons détailler la recherche d'itinéraire ainsi que la recommandation de transporteurs.

II. Recherche d'itinéraire

La recherche d'un itinéraire permet de résoudre le problème de cheminement c'est-à-dire trouver les plus courts chemins d'une ville à une autre selon les critères choisis par l'utilisateur. Ces critères sont :

- Temps: donc on recherche à minimiser la durée du trajet. Notons ici que le temps de traverse d'un tronçon dépend du trafic, pour cela, on considère le temps moyen calculé comme suit :

$$\text{Temps moyen} = \frac{\text{Temps sans trafic} + \text{Temps avec trafic}}{2}$$

- Distance : donc on recherche à minimiser la distance du trajet. Nous tenons à préciser que le coût du transport dépend du temps et de distance de manière contradictoire ; un trajet plus court en termes de distance (kilométrage) engendre généralement un coût moins élevé qu'un trajet plus rapide (temps minimal).

Par ailleurs, le choix de critère engendre deux cas possibles :

- **Recherche du plus court chemin unicritère :**

Quand l'utilisateur introduit un seul critère (distance ou temps) comme un choix, le système fait appel à l'algorithme de Dijkstra afin de résoudre le problème de cheminement selon le critère choisi.

- **Recherche des plus courts chemins bicritères :**

Quand les deux critères sont choisis au même temps, les plus courts chemins sont calculés en utilisant l'algorithme de Martins (algorithme de construction de solutions optimales). Cette

approche prend en considération la distance et la durée pour chaque sous-chemin, dans le but de trouver les meilleures solutions.

Une fois le processus de recherche terminé, l'utilisateur recevra une description analytique de chaque itinéraire c'est-à-dire un compromis entre la durée et la distance.

Dans les sous sections suivantes, nous donnons le pseudo code des algorithmes de recherche de pcch utilisés avec un exemple d'illustration.

II.1 Algorithme de Dijkstra

Voici le pseudo code de l'algorithme de Dijkstra que nous avons utilisé : [CRCT, 01]

Entrée:

1. Un graphe orienté $G = (S, A)$
2. Poids w sur les arcs A (i.e., $w : A \rightarrow \mathbb{R}^+$). Le poids peut présenter soit la distance soit la durée.
3. Un sommet source $s \in S$.

Sortie: Un tableau ds indexé par S tel que $ds[t]$ est la longueur du plus court chemin de s à t dans G .

```
1: fonction Dijkstra ( $G, w, s$ )
2:   Soit  $P$  une file de priorité vide.
3:   Soit  $ds$  un tableau de taille  $|S|$  initialisé à «non définie»
4:   Soit  $Vus$  un ensemble vide de sommets.
5:    $ds[s] \leftarrow 0$ 
6:   Ajouter  $v$  à  $Vus$ 
7:   pour tout arc  $a$  sortant de  $s$  faire
8:     longueur  $\leftarrow ds[s] + wa$ 
9:     insérer ( $P, a, longueur$ )
10:  fin pour
11:  tant que  $P$  est non-vide. faire
12:     $(u, v), poids \leftarrow \text{extraire-min}(P)$ 
13:    si  $v \notin Vus$  alors
14:       $ds(v) \leftarrow poids$ 
15:      Ajouter  $v$  à  $Vus$ 
16:    pour arc  $a$  sortant de  $v$  faire
17:      longueur  $\leftarrow ds[u] + wa$ 
18:      insérer ( $P, a, longueur$ )
19:    fin pour
20:  fin si
21:  fin tant que
22: fin fonction
```

Chapitre III : CONCEPTION DU SGRTM

En appliquant cet algorithme sur l'exemple de la figure suivante, nous avons obtenu les résultats du tableau 3.1.

Dans la figure suivante les sommets symbolisent les villes, les arcs symbolisent les tronçons et les coûts indiquent la distance (en Km) entre ces villes.

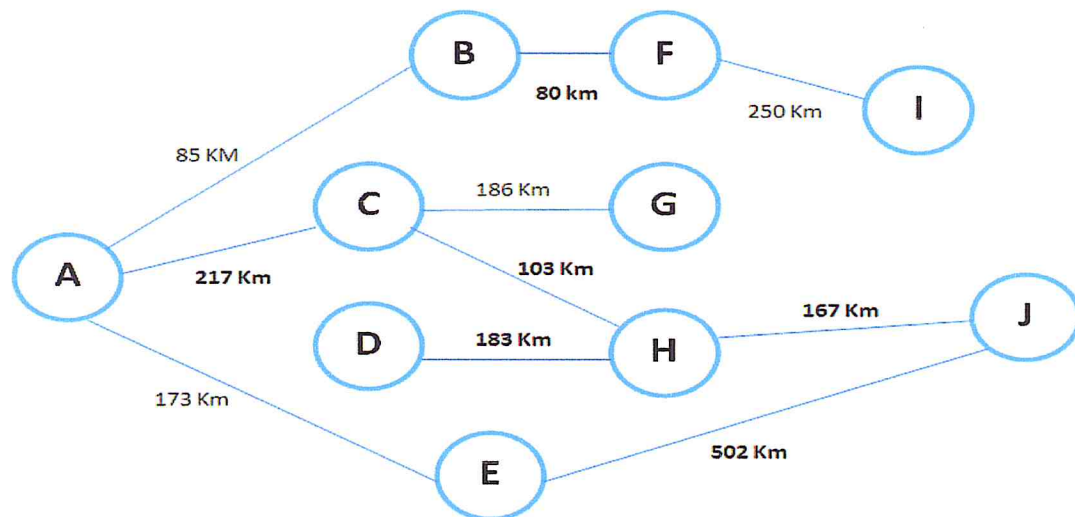


Figure 3.2 Exemple d'Application de l'algorithme de Dijkstra

| Etapes sommets | init | Itération 1 | Itération 2 | Itération 3 | Itération 4 | Itération 5 | Itération 6 | Itération 7 | Itération 8 | Itération 9 |
|----------------|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| A | 0(*) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | | 85 (*) | 85 | 85 | 85 | 85 | 85 | 85 | 85 | 85 |
| C | | 217 | 217 | 217 | 217(*) | 217 | 217 | 217 | 217 | 217 |
| D | | | | | | | 503 | 503 | 503 | 503 |
| E | | 173 | 173 | 173(*) | 173 | 173 | 173 | 173 | 173 | 173 |
| F | | | 165(*) | 165 | 165 | 165 | 165 | 165 | 165 | 165 |
| G | | | | | | 403 | 403 (*) | 403 | 403 | 403 |
| H | | | | | | 320 (*) | 320 | 320 | 320 | 320 |
| I | | | | 415 | 415 | 415 | 415 | 415 (*) | 415 | 415 |
| J | | | | | 675 | 675 | 487 | 487 | 487(*) | 487 |

Tableau 3.1 Déroulement de l'exemple d'application de l'algorithme de Dijkstra

A partir de ce tableau, on peut déduire par exemple que :

Étape 1 : on choisit la ville B. En effet, c'est la ville qui a la distance minimale (85) retournée par la fonction *extraire-min* et indiquée dans le tableau par *. Une fois ce sommet est ajouté à l'ensemble des sommets *Vus*, on met sa valeur en gras et on met à jour le seul voisin (F). Sa distance devient $85+80 = 165$.

Étape 3 : on choisit la ville E, on met sa valeur en gras et on met à jour le voisin J (675) ($173+502$).

II.2 Algorithme de Martins

Dans un problème d'optimisation bicritères, au lieu d'optimiser une seule fonction objectif, il existe de nombreuses fonctions d'objectif qui peuvent être en conflit les unes les autres. L'objectif est de trouver toutes les solutions possibles qui sont non dominiées. L'algorithme de Martins [MAR, 84] est une extension multicritère de l'algorithme de Dijkstra [DIJ, 59] dans lequel l'opérateur "min" est remplacé par le test de dominance¹. L'idée de l'algorithme de Martins est assez intuitive. À chaque itération, et pour chaque sommet, deux types de marquages sont utilisés: marquage permanent et marquage temporaire. L'algorithme sélectionne l'étiquette lexicographiquement petite à partir des étiquettes temporaires, la convertit en une étiquette permanente, et propage l'information contenue dans cette étiquette à toutes les étiquettes temporaires de ses successeurs. La procédure s'arrête lorsqu'il n'existe aucune étiquette temporaire. L'algorithme de Martins calcule l'ensemble maximal complet des chemins efficaces et utilise plusieurs critères.

L'algorithme de Martins est divisé en deux parties majeures qui sont décrites comme le suit :

¹ Lorsqu'on obtient une multitude de solutions. Seul un nombre restreint de ces solutions va nous intéresser. Pour qu'une solution soit intéressante, il faut qu'il existe une relation de dominance entre la solution considérée et les autres solutions, dans le sens suivant :

On dit que x_1 domine x_2 si :

- x_1 est au moins aussi bon que x_2 dans tous les objectifs, et,
- x_1 est strictement meilleur que x_2 dans au moins un objectif.

Algorithme de recherche exhaustive

compte \leftarrow 1

h(compte) \leftarrow *s*

X \leftarrow { *compte* }

tant_que *X* \neq \emptyset **faire**

x \leftarrow un élément de *X*

X \leftarrow *X* - {*x*}

i \leftarrow *h(x)*

pour tout (*i,j*) \in *A* **faire**

compte \leftarrow *compte* + 1

h(compte) \leftarrow *j*

X \leftarrow *X* \cup { *compte* }

ajouter (*x*, *compte*) à l'arbre

fin_pour

fin_tant_que

(Cet algorithme détermine les chemins non dominés)

Algorithme général d'étiquetage

$\{chemin(x) : \text{chemin dans } G \text{ est représenté par un arc de } 1 \text{ à } x\}$

$\{D_j : \text{un ensemble de candidats pour les chemins non dominés de } s \text{ à } j\}$

$compte \leftarrow 1$

$h(compte) \leftarrow s$

$D_s \leftarrow \{chemin(1)\}$

$X \leftarrow \{compte\}$

tant que $X \neq \emptyset$ **faire**

$x \leftarrow$ un élément de X

$X \leftarrow X - \{x\}$

$i \leftarrow h(x)$

pour tout $(i,j) \in A$ **faire**

si $chemin(x) \diamond \langle i, (i,j), j \rangle$ **est non dominé dans** D_j

alors

$COMPTE \leftarrow compte + 1$

$\{chemin(compte) \text{ sera le chemin } chemin(x) \diamond \langle i, (i,j), j \rangle\}$

ajouter $(x, compte)$ à l'arbre

$h(compte) \leftarrow j$

$X \leftarrow X \cup \{compte\}$

$D_j \leftarrow D_j \cup \{chemin(compte)\}$

Supprimer tous les chemins dominés dans D_j

Supprimer tous les nœuds correspondant à l'arbre L'arbre de X

Fin_si

Fin_pour

Fin_tant_que

$D \leftarrow D_t$

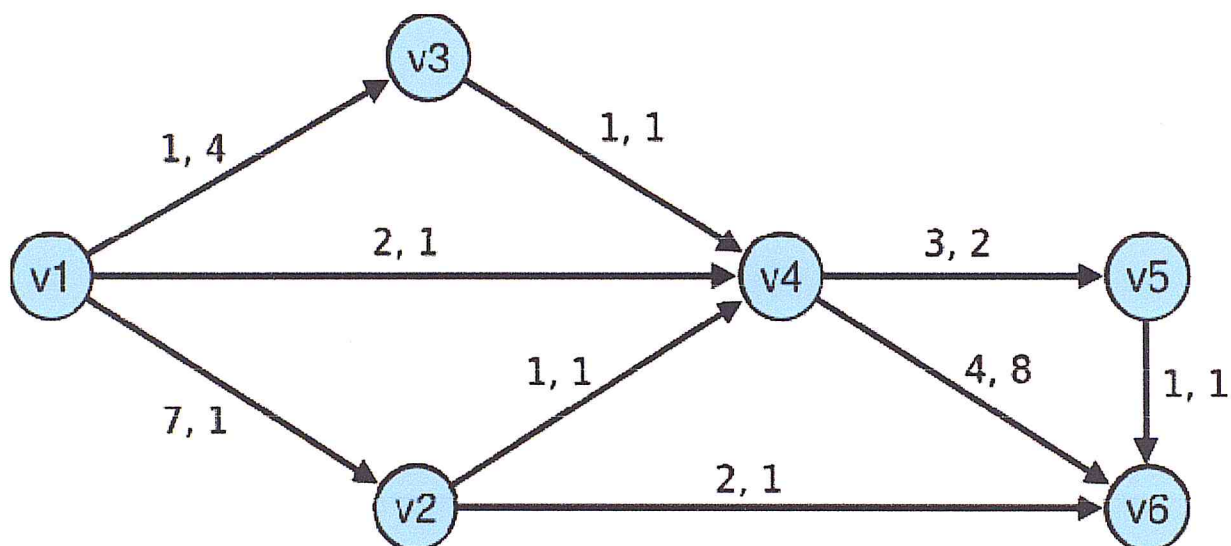


Figure 3.3 Recherche plus court chemin Martins

Pour expliquer comment effectuer une recherche bicritères générale en appliquant l’algorithme de martins sur un graphe, nous allons prendre l’exemple représenté ci-dessus où chaque arc est étiqueté avec deux valeurs (C1=temps, C2=distance) au lieu d'une seule. Ainsi, le coût d'un chemin est défini comme la somme de la première et la seconde valeur de chaque arc dans le chemin, l'objectif est de trouver tous les chemins non dominés de V1 à V6, minimisant les deux valeurs.

Par exemple, le chemin $V1 \rightarrow V4 \rightarrow V6$ a un coût de $(2,1) + (4,8) = (6,9)$, alors que le coût du chemin $V1 \rightarrow V2 \rightarrow V6$ $(7,1) + (2,1) = (9,2)$. Quel est le meilleur chemin?

Le premier chemin a un coût de 6 pour la première valeur, qui est meilleur que le deuxième chemin, mais il a un cout de 9 pour la seconde valeur, ce qui est pire que la valeur du deuxième chemin. Ainsi, chaque chemin est non dominé par rapport à l'autre.

Toutefois, le chemin $v1 \rightarrow v4 \rightarrow v5 \rightarrow v6$ a un coût total de $(6,4)$, ce qui est mieux que le chemin avec un coût de $(6,9)$. Par conséquent, ce nouveau chemin domine le précédent.

III. Recommandation transporteurs:

Une fois les solutions construites (itinéraires et moyens de transport), la procédure de recommandation sociale commence. Cette dernière passe par les étapes suivantes :

- La collection des appréciations des utilisateurs sur les transporteurs. Pour cela, nous demandons à l'utilisateur de noter lui même les transporteurs à partir d'une échèle de note fixée de 1 à 5.

- Pour prédire la pertinence d'un transporteur pour un utilisateur, nous calculons donc la moyenne des notes données aux transporteurs par les utilisateurs en prenant en considération le type de marchandise. Pour ce faire, nous avons choisi la formule de calcul de la moyenne de notation toute en attribuant un poids à la note de l'utilisateur selon son nombre de participations sur la plateforme. Ce poids est appelé poids d'expérience.
- Classer et afficher le top trois des transporteurs susceptibles d'intéressé l'utilisateur.

L'exemple suivant illustre notre processus de recommandation ou l'expérience indique la fiabilité du client selon le nombre de missions qu'il a demandé à effectuer :

| | Expérience | Transporteur1 | Transporteur2 | Transporteur 3 |
|--------|------------|---------------|---------------|----------------|
| User 1 | 25% | 1 | 3 | 4 |
| User 2 | 50% | 2 | 1 | 2 |
| User 3 | 75% | 2 | 4 | 4 |
| User 4 | 100% | 3 | 5 | 5 |

Tableau 3.2 Notation des utilisateurs

A partir de ce tableau, on va calculer la note transporteur selon l'expérience de l'utilisateur avec la formule suivante :

$$\text{note}_{\text{fiable}} = \text{note}_{\text{client}} * \text{experience}_{\text{client}}$$

| | User 1 | User 2 | User 3 | User 4 |
|-------------|--------|--------|--------|--------|
| Note fiable | 0.25 | 1 | 1.5 | 3 |

Tableau 3.3 Notation transporteur 1 selon l'expérience des utilisateurs

Ainsi, la note moyenne pour chaque transporteur sera comme suit :

| | Transporteur 1 | Transporteur 2 | Transporteur 3 |
|--------------|----------------|----------------|----------------|
| Note moyenne | 1.43 | 2.31 | 2.5 |

Tableau 3.4 Notes moyennes des transporteurs

En conclusion, le meilleur transporteur est Transporteur 3.

IV. Conception du système d'informations du SGRTM

Dans cette section, nous allons apporter plus de détails à notre solution en clarifiant les aspects techniques. Pour ce faire, nous avons utilisé UML (Unified Modeling Language) qui représente un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes. Ces diagrammes donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. Dans notre travail, nous avons utilisé les diagrammes de cas d'utilisation, de classe, d'activités et de séquences.

IV.1 Diagramme des cas d'utilisation

Notre système possède quatre acteurs qui sont : L'administrateur, le client, le visiteur et le transporteur. Ces derniers sont décrits dans le tableau suivant :

| Acteur | Rôle |
|----------------|---|
| Administrateur | Gère les comptes utilisateurs notamment leurs inscriptions (Confirmation/Refus) |
| Client | Une fois connecté, le client effectue des recherches d'itinéraire afin de valider sa demande transport, comme il peut gérer son espace personnel (informations/demandes) |
| Visiteur | Effectue des recherches d'itinéraire, sans avoir le droit de valider une demande |
| Transporteur | Gère son espace personnel qui se résume à la modification des ses informations, la gestion des demandes clients (accepter/refuser/signaler l'achèvement) et la gestion de ses moyens de transport |

Tableau 3.5 Identification des acteurs

Chaque acteur déclenche les fonctionnalités représentées par les cas d'utilisation sur la figure (3.4)

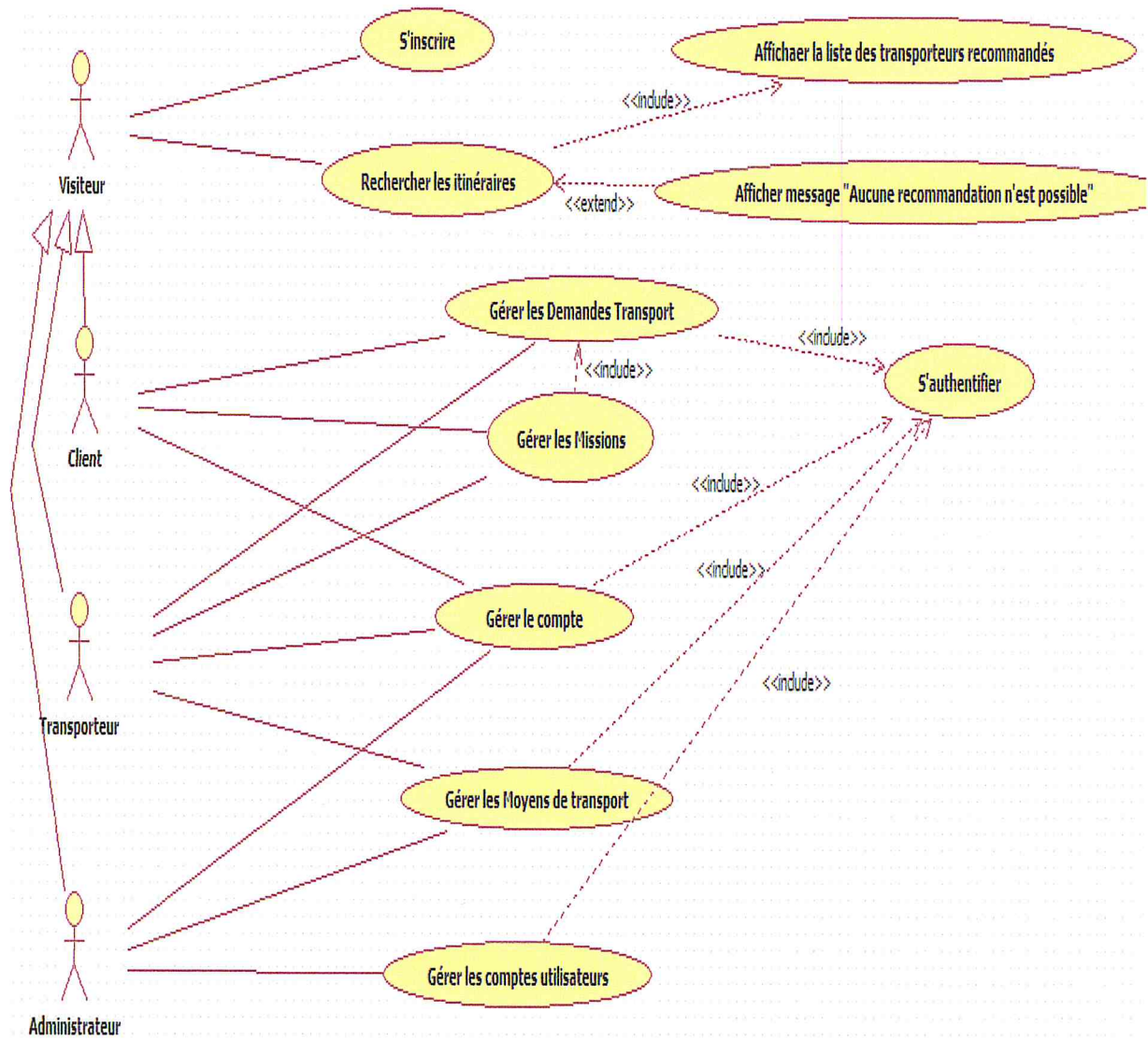


Figure 3.4 Diagramme des cas d'utilisation général

| Cas d'utilisation | Acteur | Description |
|---------------------------------|---|---|
| Rechercher itinéraire | Visiteur / Client / Transporteur / Administrateur | L'acteur introduit les informations de recherche et les critères, selon ces derniers les itinéraires et les moyens de transport seront affichés tout en recommandant le top trois des transporteurs |
| Gérer les demandes de transport | Client | Le client choisi un itinéraire et un moyen de transport et enfin il confirme la demande |
| | Transporteur | Le transporteur recherche les demandes pour qu'il puisse les confirmer ou les refuser |
| Gérer les missions | Client | Le client confirme la réception de la marchandise |
| | Transporteur | Le transporteur signale l'achèvement de la mission |
| Gérer le compte | Administrateur / Client / Transporteur | L'acteur pourra modifier ses informations personnelles |
| Gérer les moyens de transport | Administrateur / Transporteur | L'acteur pourra ajouter ou supprimer un moyen de transport |
| Gérer les comptes utilisateurs | Administrateur | L'administrateur valide ou refuse une demande d'inscription |

Tableau 3.6 Description cas d'utilisation général

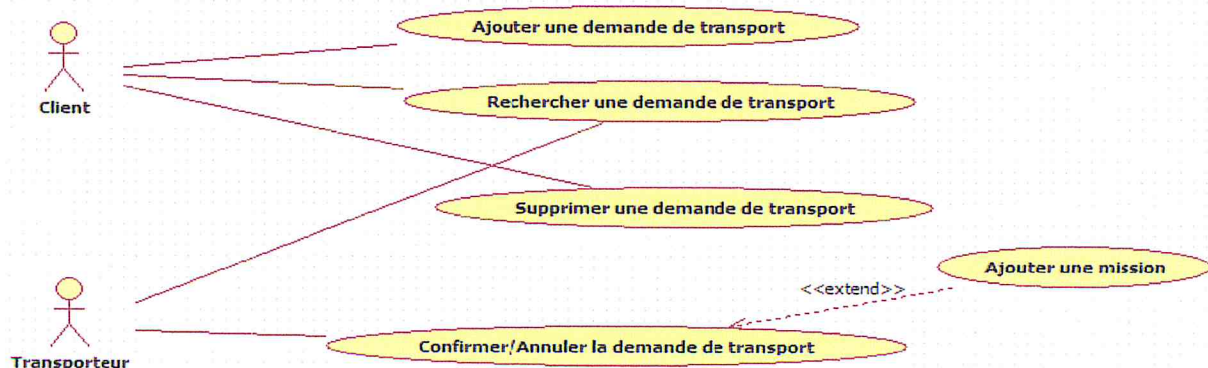


Figure 3.5 Diagramme des cas d'utilisation gestion des demandes

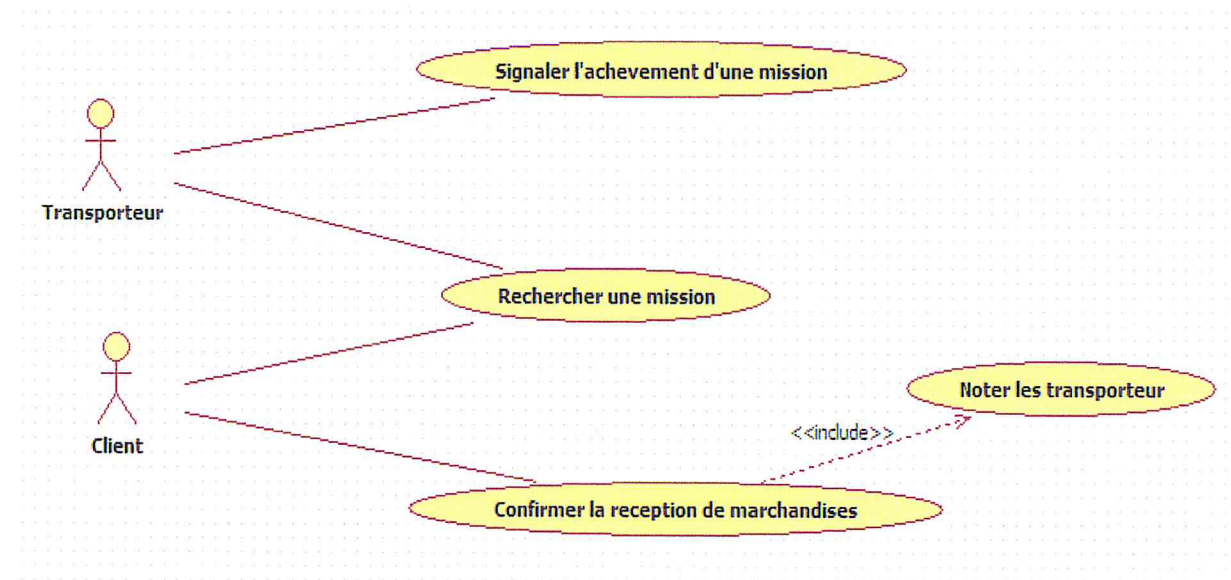


Figure 3.6 Diagramme des cas d'utilisation gestion des missions

IV.2 Elaboration du Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée Objet. Il s'agit d'une vue statique qui on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application

IV.2.1 Règles de gestion :

- un client peut faire une ou plusieurs demandes de transport. Ces demandes doivent être confirmées par le transporteur pour qu'elles deviennent des missions ;
- une mission concerne un et un seul transporteur, et nécessite un seul moyen de transport ;
- le transporteur peut accomplir plusieurs missions ;
- un transporteur possède un ou plusieurs moyens de transport. Chaque moyen peut transporter un seul type de marchandises ;
- une mission concerne un itinéraire bien défini. Cet itinéraire est un ensemble des tronçons spécifiés par ville de départ et ville d'arrivée.

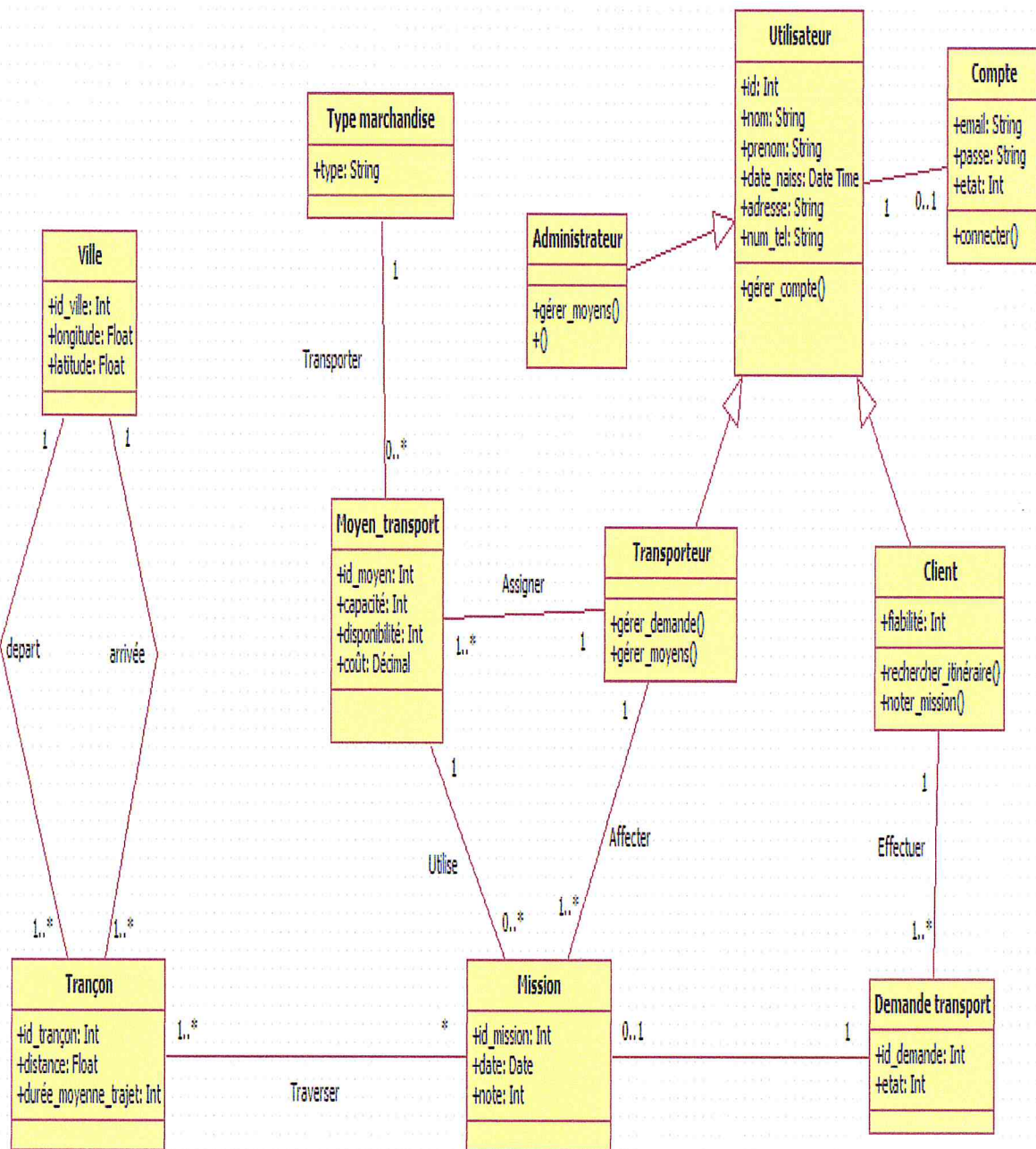


Figure 3.7 Diagramme de classes

IV.2.2 Le passage vers le modèle relationnel :

Le modèle relationnel représente depuis plusieurs années la tendance principale du marché pour l'implémentation des bases des données, la raison est que ce modèle est fondé sur les bases mathématiques qui facilitent sa manipulation. Lors d'une conception d'une base de données relationnelle l'étape principale consiste à définir un bon schéma relationnel (dont les relations sont en troisième forme normale). C'est à partir du diagramme de classe qui décrit le modèle objet des données de notre système et à l'aide de certaines règles de passage, que nous déduisons le schéma relationnel suivant :

Tronçon (id_tronçon, id_ville_depart*, id_ville_arrivee*, distance, durée_moyenne)

ville (id_ville, longitude, latitude)

Mission (id_mission, id_demande*, id_transporteur*, id_moyen*, date, note)

Moyen_transport (id_moyen, matricule, capacité, disponibilité, cout)

Transporteur (id_transporteur, nom, prénom, date-naiss, email, adresse, passe, num_tel)

Client (id_client, nom, prénom, date-naiss, email, adresse, passe, num_tel, fiabilité)

Compte (email, passe, etat)

Type marchandise (type)

Demande transport (id_demande, etat, id_client*)

Assigner (id_moyen, id_transporteur)

IV.3 Diagramme d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Dans ce qui suit, nous allons décrire le processus de gestion et recommandation du transport de notre système en utilisant un diagramme d'activité générale.

ou le déroulement d'un cas d'utilisation. Dans ce qui suit, nous allons décrire le processus de gestion et recommandation du transport de notre système en utilisant un diagramme d'activité générale.

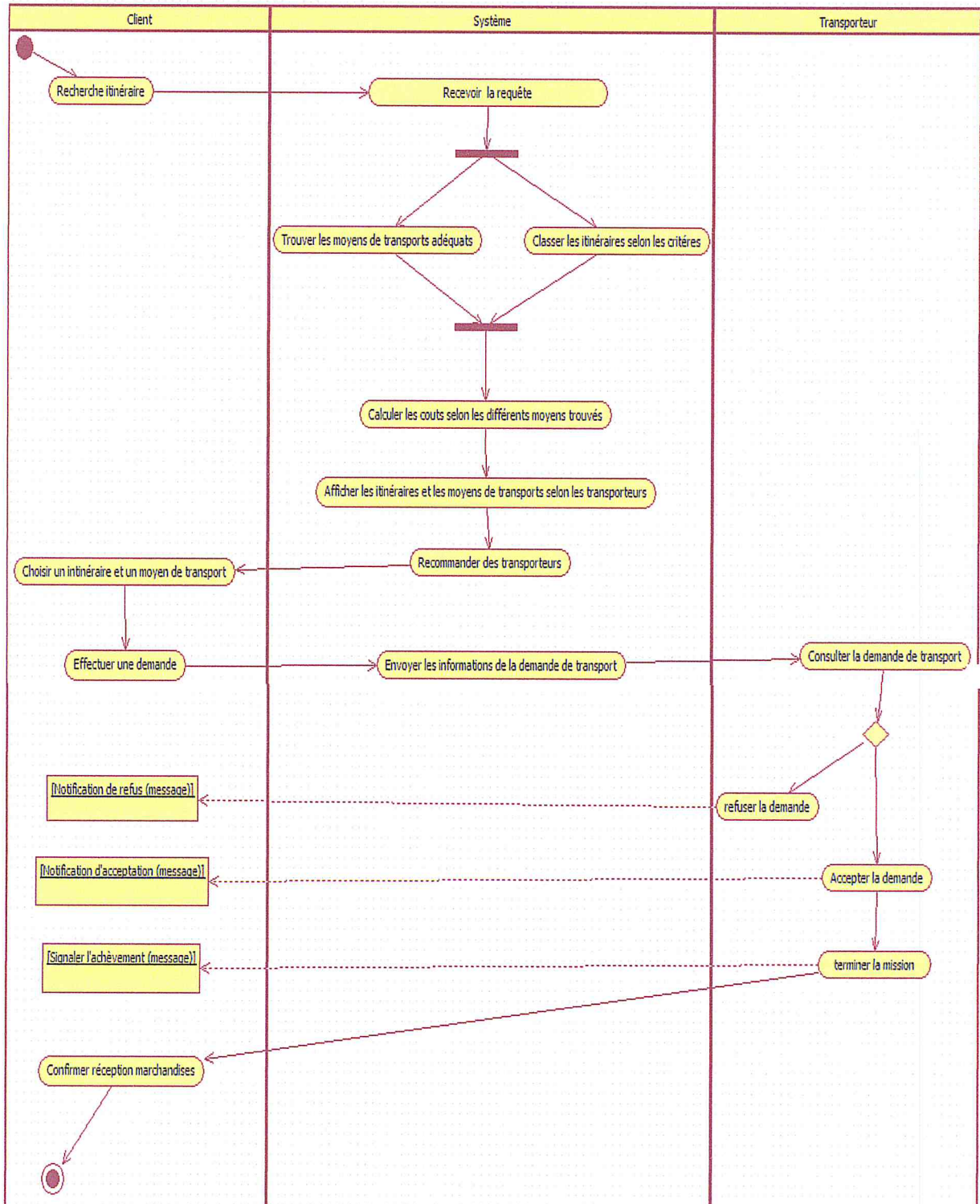


Figure 3.8 Diagramme d'activité générale

IV.4 Diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Dans notre cas, l'environnement se constitue d'une base de données, qui contient les informations sur les clients et les transporteurs ainsi que des informations sur les opérations de transport et les marchandises, etc. Les principaux diagrammes de séquences sont décrits comme suit :

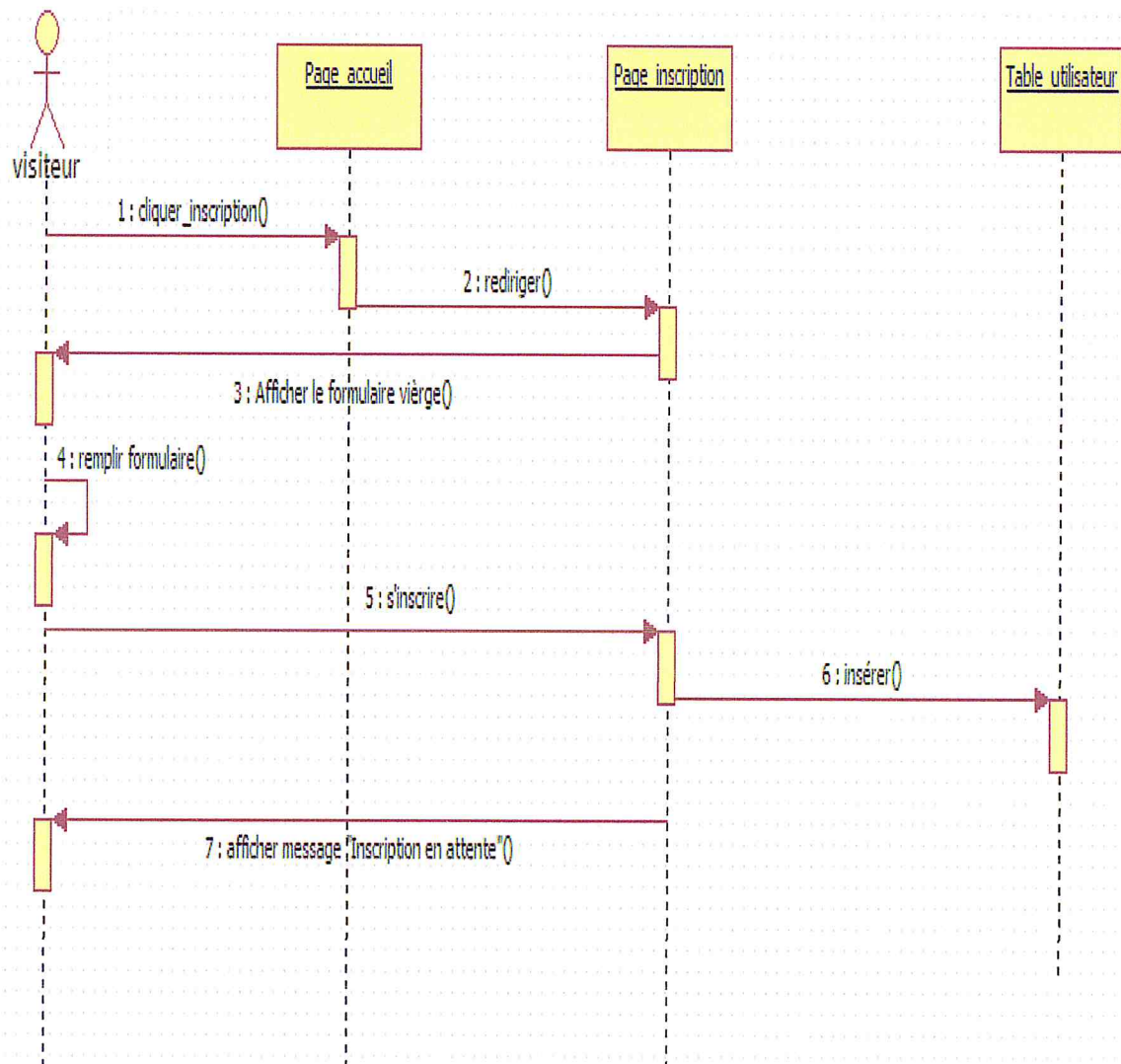


Figure 3.9 Diagramme de séquence d'inscription d'un visiteur

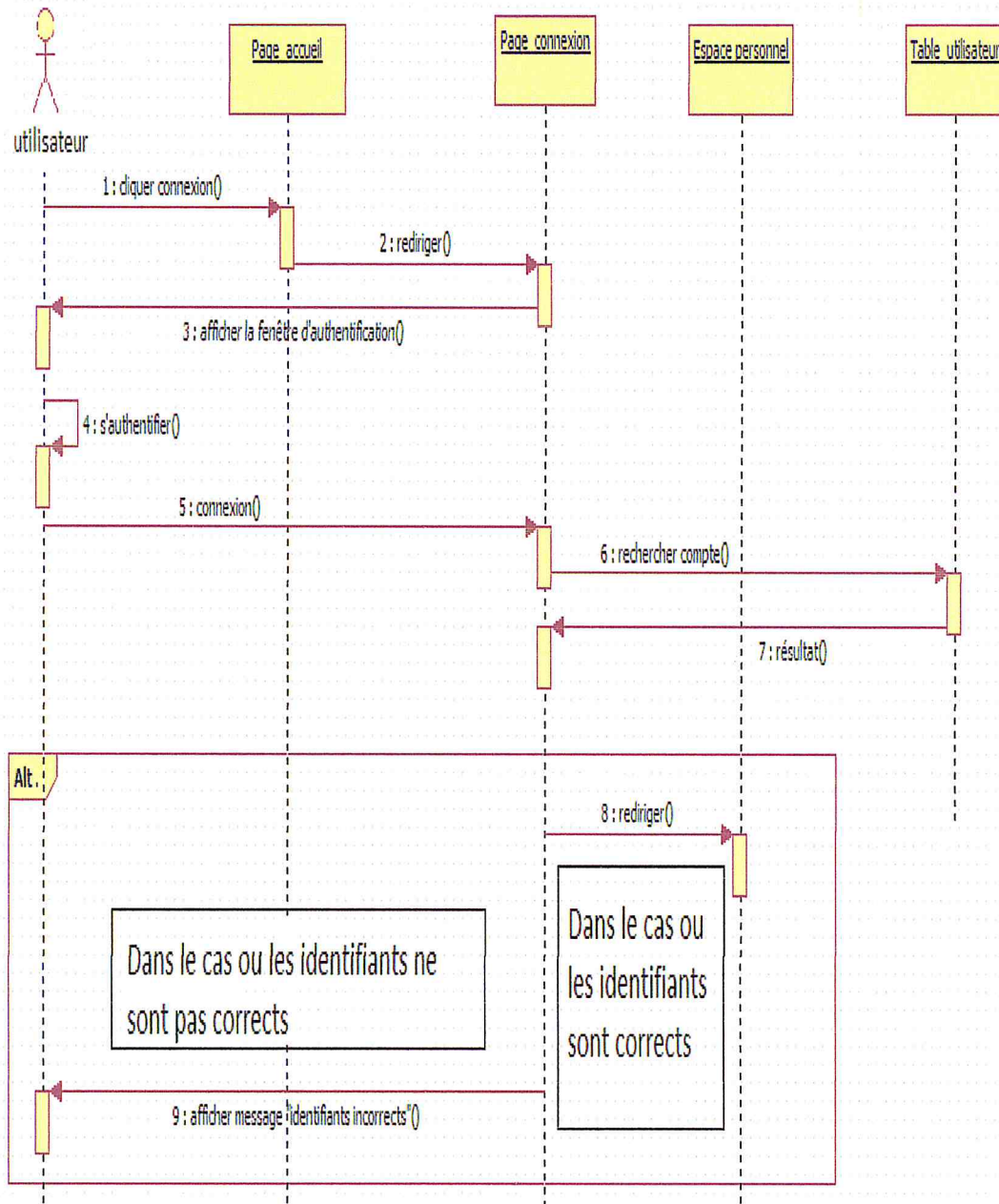


Figure 3.10 Diagramme de séquence de connexion d'un utilisateur

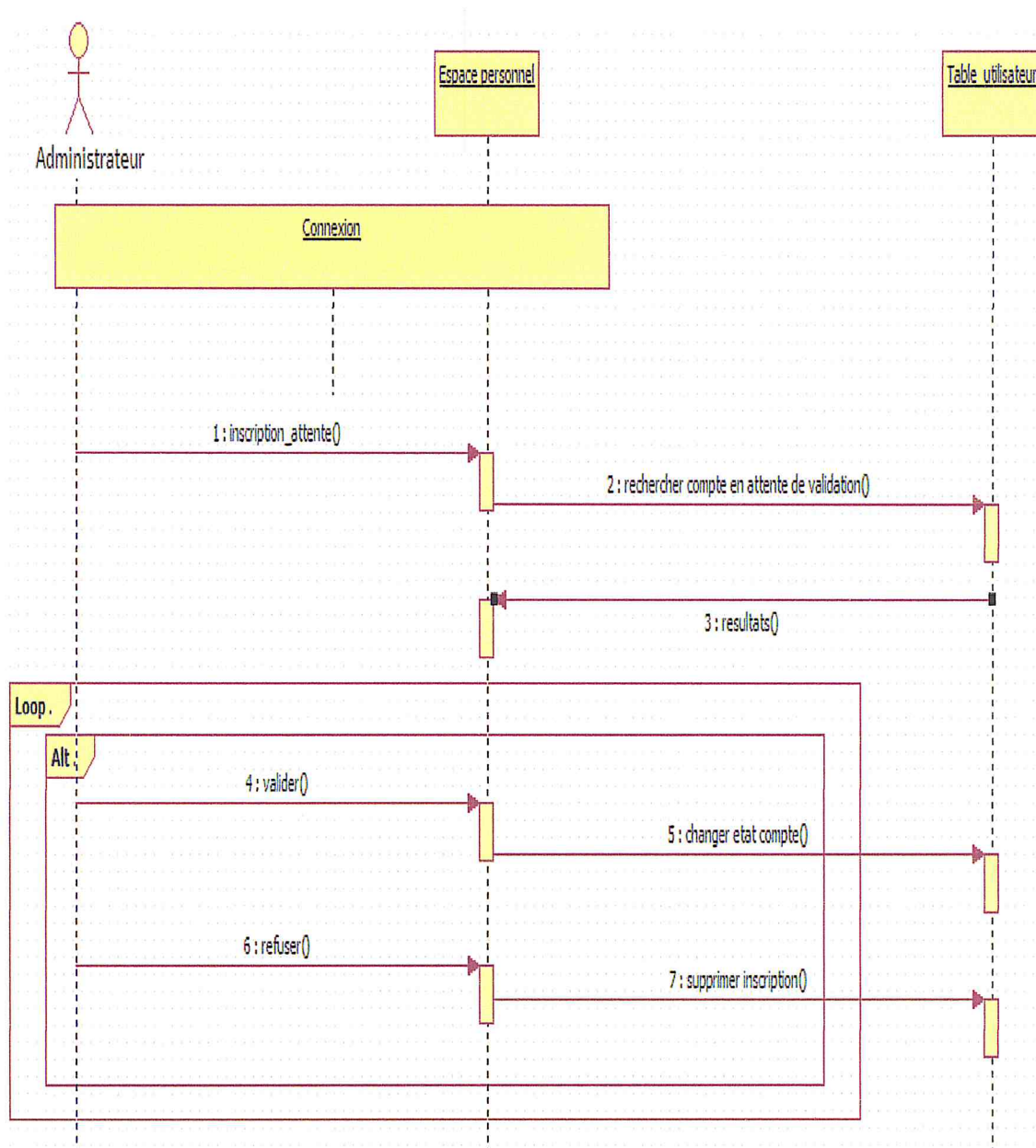


Figure 3.11 Diagramme de séquence de validation d'inscription

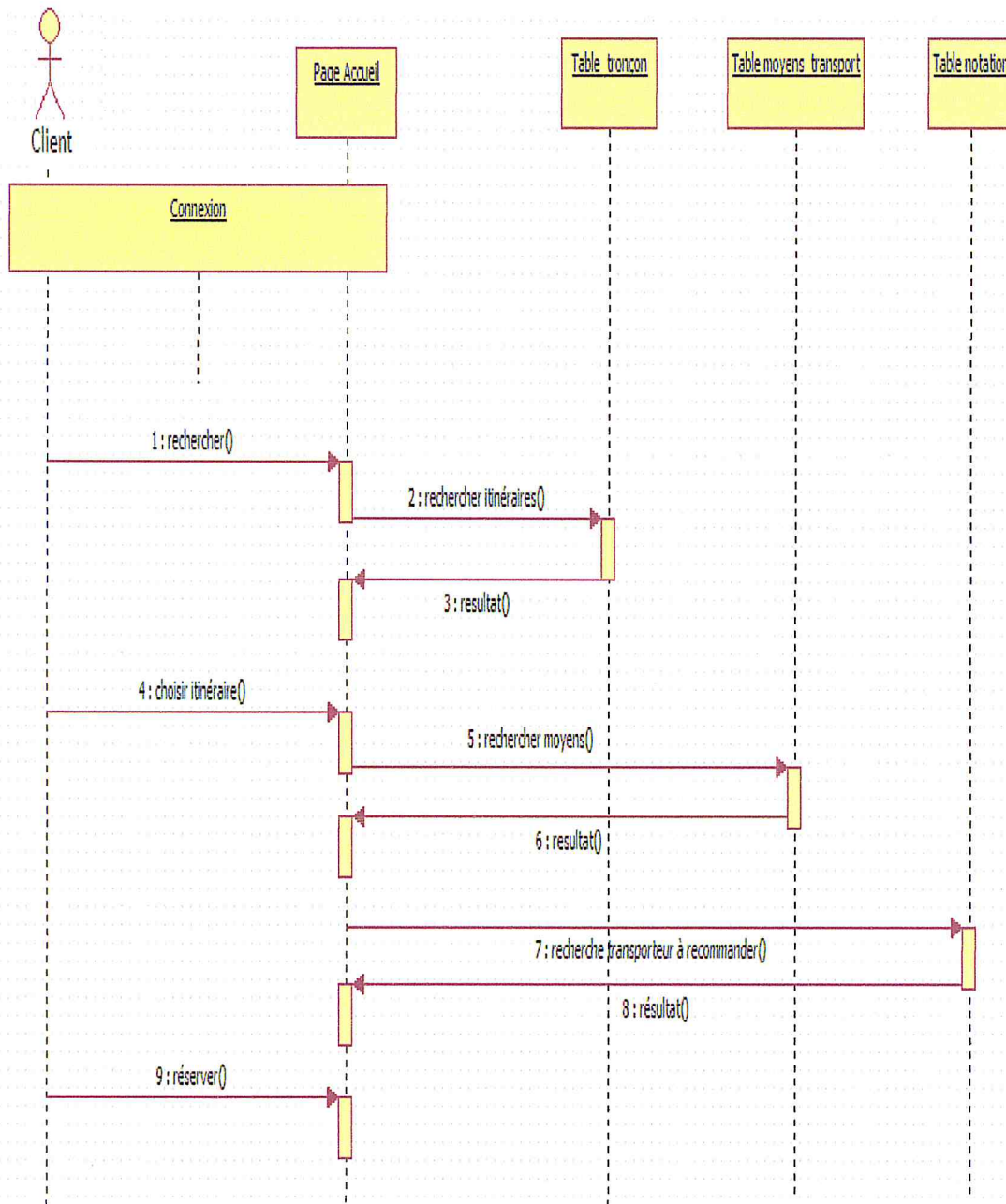


Figure 3.12 Diagramme de séquence d'ajout d'une demande de transport

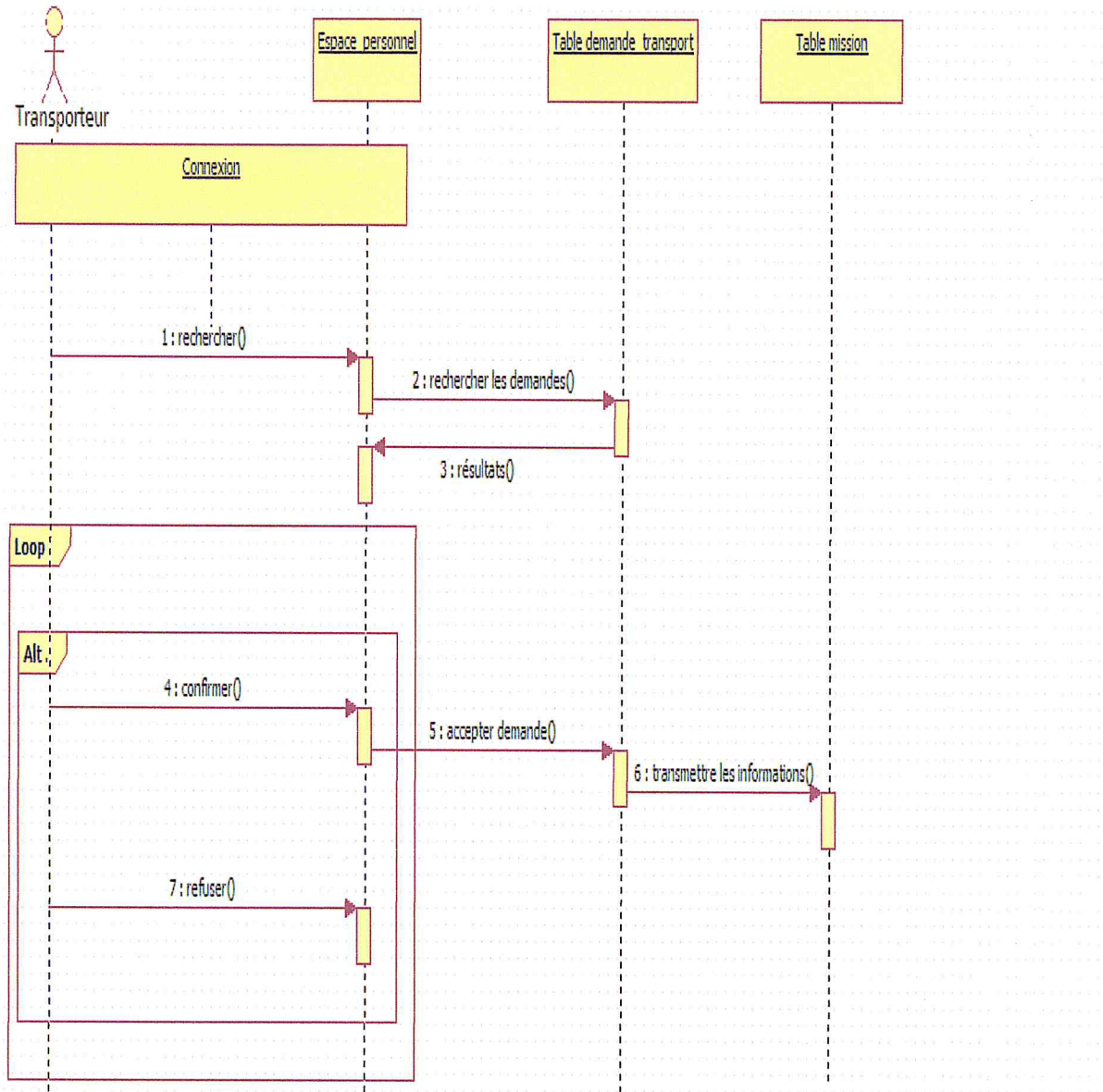


Figure 3.13 Diagramme de séquence de Recherche, Confirmation/Annulation d'une demande

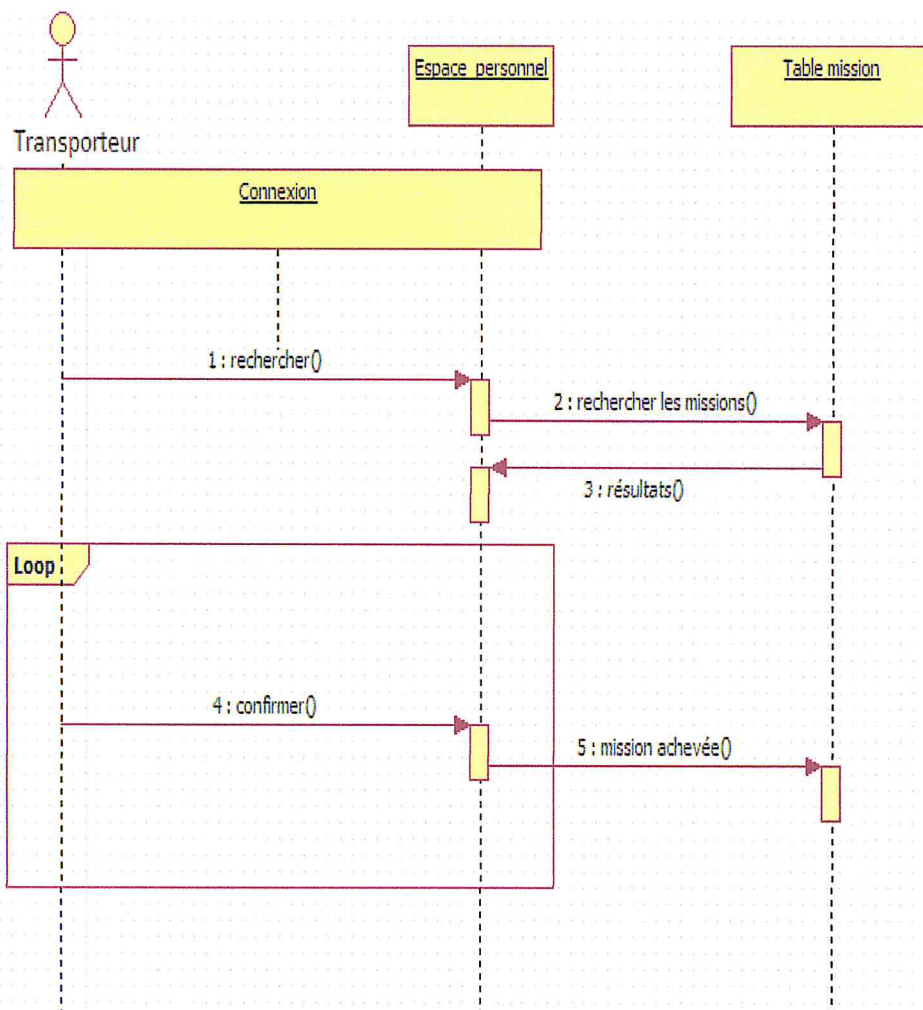


Figure 3.14 Diagramme de séquence de Recherche, Confirmation d’achèvement d’une mission

V. Conclusion

Dans ce chapitre, nous avons effectués l’étude conceptuelle de notre système, ainsi que les grandes fonctionnalités et leurs techniques d’implémentation. Grace à cette étude nous pouvons maintenant passer à la réalisation de notre système.

CHAPITRE IV :

REALISATION

Introduction

Nous arrivons maintenant à la phase de réalisation qui permet l'implémentation de la solution retenue au niveau des phases précédentes. Dans ce chapitre, nous commençons par donner une brève description de l'environnement. Ensuite, nous présentons l'architecture de l'application et illustrons son implémentation par quelques captures d'écran représentant les fonctionnalités offertes.

I. Présentation des outils de développement:

Pour implémenter notre système nous avons utilisé les outils suivants :

I.1 Choix d'Eclipse J2EE comme environnement de développement JAVA :

Eclipse est un environnement de développement (IDE) historiquement destiné au langage Java, même si grâce à un système de plugins il peut également être utilisé avec d'autres langages de programmation, dont le C/C++ et le PHP. C'est un outil pour les développeurs Java qui sert à la création d'applications Web et Java EE, constitué d'un Java IDE, outils pour Java EE, JPA, JSF Mylyn, EGit et autres. Eclipse couvre tout le cycle de vie du développement d'une application, allant de la conception à la programmation, le débogage, l'optimisation et enfin le déploiement.

I.2 Choix d'Apache Tomcat comme serveur :

Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'*Apache Software Foundation*. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

I.3 Choix de MYSQL Server comme SGBD :

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

Nous l'avons choisi pour notre application car il compte de nombreux avantages :

- Il gère de manière fiable une grande quantité de données dans un environnement multiutilisateurs
- Plusieurs utilisateurs peuvent accéder simultanément aux mêmes données.
- Empêche tout accès non autorisé et fournit des solutions efficaces pour la récupération des données après incident.

Cet outil fournit un analyseur de données OpenStreetMap pour les transformer en un nœud-arc adapté pour les applications de routage.

I.4 Tristramg/osm4routing :

L'entrée est un fichier XML OpenStreetMap. Le fichier peut être lu :

- à partir d'un fichier .osm
- à partir d'un fichier bzip2
- à partir d'un fichier gzip

La sortie peut être :

- un fichier CSV
- base de données (postgres, mysql, sqlite, postgis)

À la fois en production on obtient deux fichiers/tables :

- Les nœuds qui symbolisent les villes contenant :
 - Id_ville (64 bit entier)
 - Longitude (décimal réel)
 - Latitude (décimal)
- Les arcs qui symbolisent les tronçons reliant les villes contenant :
 - Id_tronçon (64 bit entier)
 - ID de nœud source (64 bit entier)
 - ID de nœud cible (64 bit entier)
 - Longueur (réel en mètres)

II. Description détaillée de l'architecture du SGRTM

La figure 4.1 donne une description détaillée de notre SGRTM en décrivant les différents modules utilisés dans ce dernier.

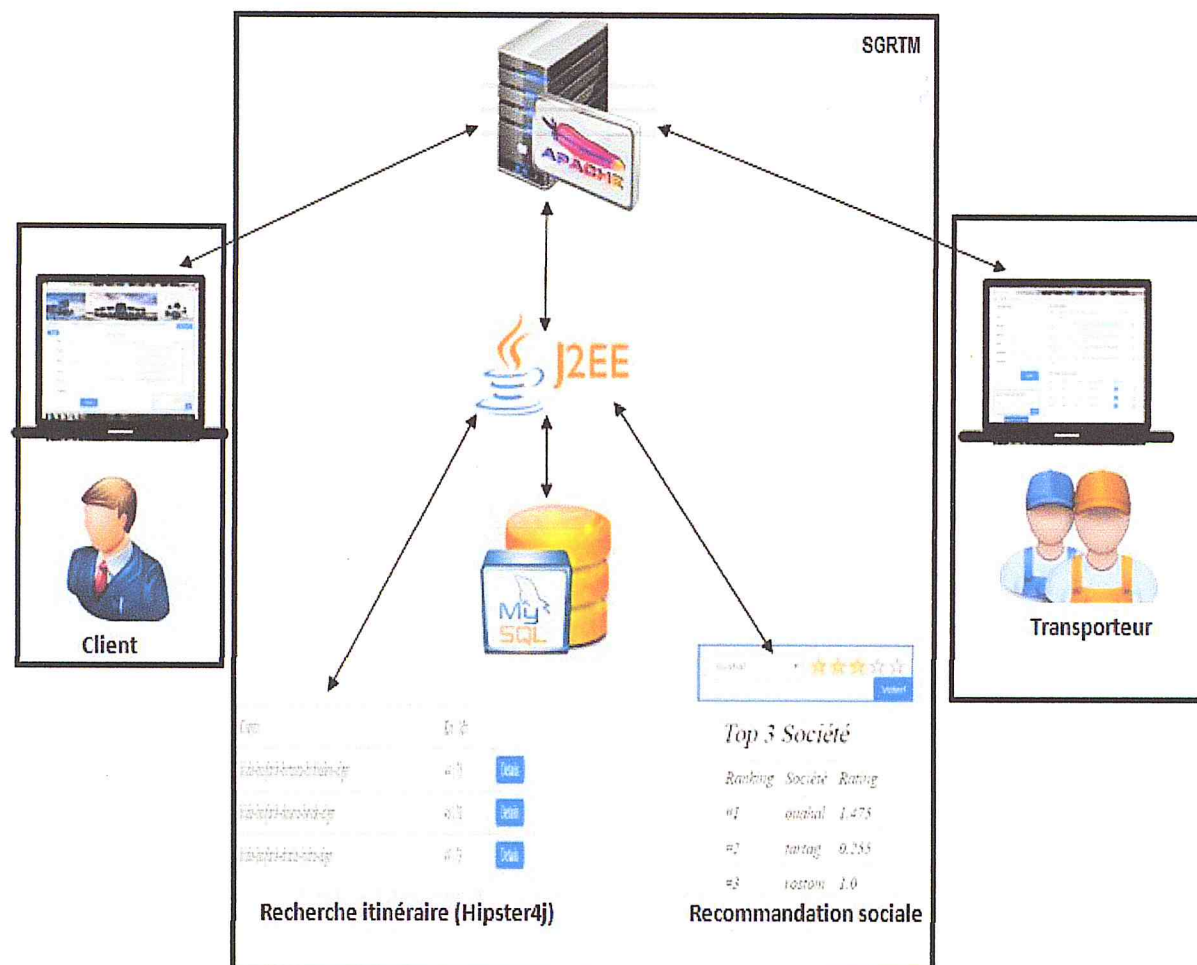


Figure 4.1 Description détaillée du SGRTM

II.1 Module recherche itinéraire

Le module recherche itinéraire est utilisé dans le but de résoudre le problème de cheminement c'est-à-dire rechercher les plus court chemins dans les deux cas unicritère et bicritères. Pour cela, nous avons utilisé l'API «Hipster4j - version 1.0.0-rc2».

II.1.1 API «Hipster4j - version 1.0.0-rc2»

L'objectif de Hipster4j est de fournir une librairie flexible et facile à utiliser, de recherche heuristique pour Java et Android. Hipster4j s'appuie sur un modèle flexible avec des opérateurs génériques qui permettent de réutiliser et de modifier le comportement des algorithmes d'une manière facile.

La version actuelle de la bibliothèque est livrée avec les algorithmes de recherche les plus utilisés : L'algorithme de Dijkstra, Bellman-Ford, A étoile (A*), Martins (L'algorithme d'étiquetage pour le problème du plus court chemin multi objectifs).

Comme la recherche des plus courts chemins comprend deux cas différents selon les critères, on a choisi deux algorithmes parmi l'ensemble des algorithmes proposée par cette Api :

- Dijkstra pour la recherche du plus court chemin unicritère.
- Martins pour la recherche du plus court chemin bicritères.

II.1.2 Justification du choix :

Il existe des bibliothèques Java implémentant les algorithmes de recherche (tél que Dijkstra ou A*) avec un modèle extensible, flexible et facile à utiliser. En fait, la plupart des bibliothèques se fient à des structures de graphes ou bien des méthodes d'implémentation récursives ce qui ne permette pas d'offrir un contrôle total sur l'algorithme. Il est très répandu de trouver plusieurs projets (comme les bibliothèques de graphes, les bases de données NoSQL, etc.) qui réinventent constamment la roue à chaque fois qu'ils ont besoin d'utiliser un algorithme commun de recherches afin de résoudre un problème concret.

Hipster4j vise à résoudre ce problème en fournissant les composantes communes d'un algorithme de recherche et un ensemble d'algorithmes de recherches classiques prêts à être utilisés dans n'importe quel type de projet.

II.1.3 Modification apportée :

L'application de l'algorithme de Dijkstra pour la recherche unicritère, inclus dans l'Api de Hipster4j a été utilisée sans introduire de modifications.

Par contre, pour offrir plusieurs choix d'itinéraire à l'utilisateur nous avons modifié l'application de l'algorithme de Martins (L'algorithme d'étiquetage pour le problème du plus court chemin multi critères) de la façon suivante :

- Suppression de la fonction *CompareTo (Cost o)* :

Nous avons supprimé la fonction *CompareTo (Cost o)* dans le but d'afficher tout les chemins, au lieu d'afficher les chemins non dominés seulement, le code de cette fonction est le suivant :

```
public int compareTo(Cost o) {  
    if (c1 <= o.c1 && c2 <= o.c2){  
        if (c1 < o.c1 || c2 < o.c2){  
            return -1;  
        }  
    } else if (o.c1 <= c1 && o.c2 <= o.c2){  
        if (o.c1 < c1 || o.c2 < c2){  
            return 1;  
        }  
    }  
    return 0;  
}
```

- L'ajout d'une fonction de classement d'itinéraire :

Pour classer les itinéraires dans un ordre ascendant nous avons implémenté une fonction de tri, le code de cette fonction est le suivant :

```
private ArrayList<String> tri_itineraire (ArrayList<String> resultat) {  
    int k = 1,m=0,n=0,f=0; String x,y;  
    while (k<resultat.size()-2){  
        m = Integer.parseInt (resultat.get(k));  
        f=k+2;  
        while (f<resultat.size ()){  
            n = Integer.parseInt (resultat.get (f));  
            if (m>n) {  
                x=resultat.get (k-1); y=resultat.get (k);  
                resultat.set (k-1, resultat.get (f-1));  
                resultat.set (k, resultat.get (f));  
                resultat.set (f-1, x);  
                resultat.set (f, y);  
            }  
            f=f+2;  
        }  
        k=k+2;  
    }  
    return resultat;  
}
```

II.2 Module recommandation

Le module de recommandation contient les différentes étapes élaborées par l'approche de recommandation sociale que nous avons appliquée à notre système. Les fonctionnalités de cette approche ont été réalisées à travers des requêtes SQL. Par la suite, nous allons détailler les grandes fonctionnalités de ce module :

II.2.1 Collecte des appréciations :

A la fin de chaque mission de transport, le client pourra donner son avis sur cette mission. Pour ce faire, nous proposons au client une échelle de notation fixée de 1 à 5 afin d'exprimer son avis sur le transporteur (figure 4.1). Une fois la notation faite, elle est sauvegardée dans notre base de données pour une prochaine évaluation.

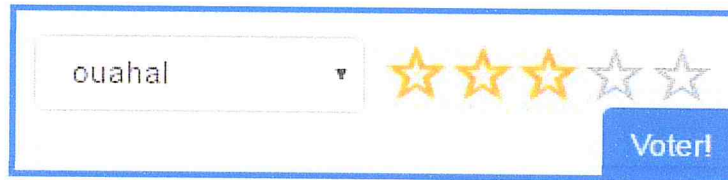


Figure 4.2 L'échèle de notation utilisée dans notre application

II.2.2 Evaluation des notations transporteurs:

La deuxième étape dans le processus de cette approche est d'évaluer les transporteurs en calculant la moyenne des notes données aux transporteurs par les clients.

Cette étape nécessite l'attribution d'un poids à la note de chaque client exprimant son expérience afin de donner plus de fiabilité à la recommandation. Ce poids dépend du nombre de demandes de transport effectué par le client :

| Nombre de missions | Poids de Fiabilité |
|--------------------|--------------------|
| 0.. 5 | 25% |
| 5.. 10 | 50% |
| 10.. 15 | 75% |
| 15.. ou plus | 100% |

Tableau 4.1 : Evaluation des notes clients

Ainsi, la note attribuée par le client au transporteur est plus fiable lorsqu'il effectue plusieurs demandes de transports (missions).

A la fin de cette étape, nous calculons la note moyenne attribuée au transporteur par plusieurs clients en utilisant la formule suivante :

$$\frac{X1W1 + X2W2 + X3W3 \dots XnWn}{\text{Total}}$$

Ou:

X = la note choisie sur l'échelle de notation

W= la fiabilité du client

Total= le nombre des clients

Notons ici que le poids de la fiabilité est affiché au client dans son espace personnel comme illustré dans la figure suivante :



Figure 4.3 Fiabilité du transporteur

II.2.3 Classement et affichages Top 3 des transporteurs:

Une fois l'étape d'évaluation de chaque transporteur terminée, nous classifions ces transporteurs selon la moyenne de leurs notes et nous affichons le top trois des transporteurs susceptibles d'intéresser l'utilisateur comme suit :

Top 3 Société

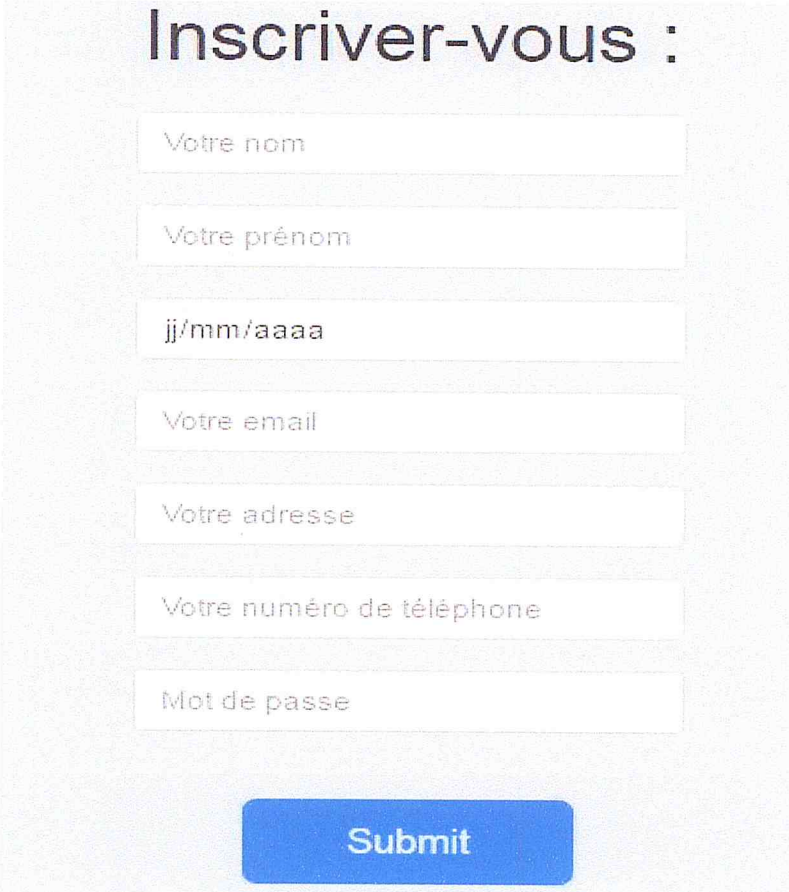
| <i>Ranking</i> | <i>Société</i> | <i>Rating</i> |
|----------------|----------------|---------------|
| #1 | <i>ouahal</i> | <i>1,475</i> |
| #2 | <i>tartag</i> | <i>1.0</i> |
| #3 | <i>rostom</i> | <i>0.255</i> |

Figure 4.4 Classement des transporteurs les mieux notés

III Présentation du prototype (l'application) :

Dans cette section, nous allons présenter le fonctionnement de notre système en montrant ses différentes pages qui sont conçues de manière à être simples, naturelles pour aboutir à une utilisation facile.

III.1 Page d'inscription et page de connexion :



Inscriver-vous :

Votre nom

Votre prénom

jj/mm/aaaa

Votre email

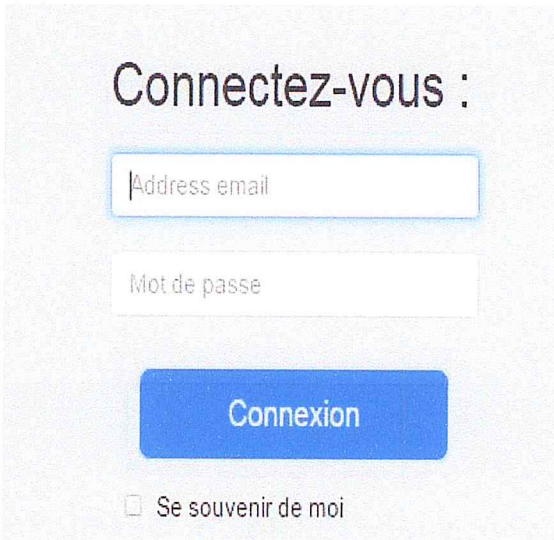
Votre adresse

Votre numéro de téléphone

Mot de passe

Submit

Figure 4.5b Page d'inscription



Connectez-vous :

Address email

Mot de passe

Connexion

Se souvenir de moi

Figure 4.5a Page de connexion (authentification)

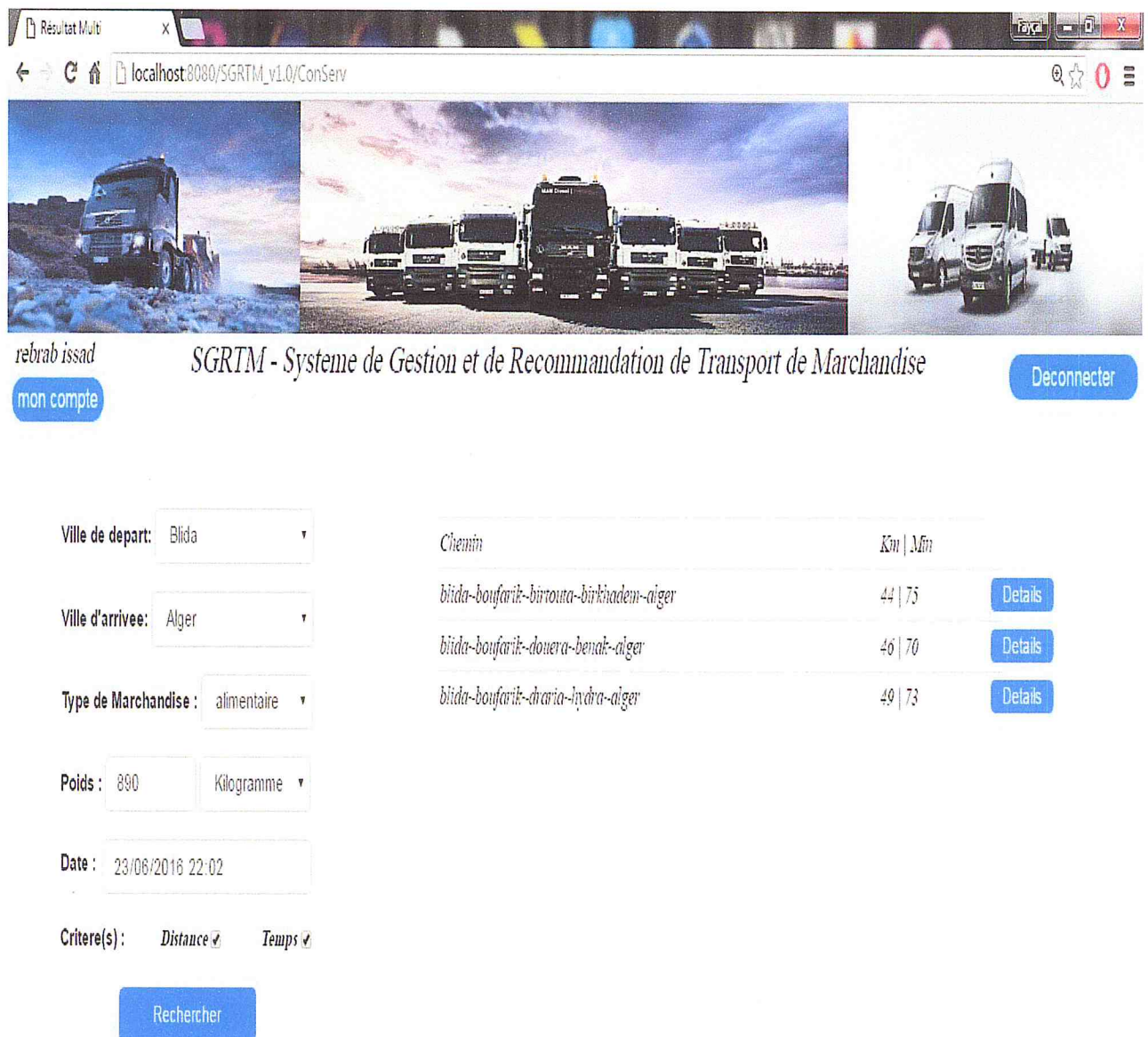
Chapitre IV : REALISATION, TESTS ET RESULTATS

Chaque visiteur désirent s'inscrire (nouveau client, nouveau transporteur), remplit la demande d'inscription. La figure 4.5b suivante représente le formulaire d'inscription.

Pour pouvoir accéder à son espace personnel, chaque utilisateur (client, transporteur) doit s'authentifier (Adresse email/Mot de passe), la figure 4.5a représente la page de connexion.

III.2 Recherche itinéraire multicritères :

La figure suivante montre les informations à remplir par un client lors de la recherche itinéraire multicritères, ainsi que les résultats des itinéraires (avec le bouton détails).



The screenshot shows a web browser window with the URL `localhost:8080/SGRTM_v1.0/ConServ`. The page header includes the user name `rebrab issad`, a `mon compte` button, the title `SGRTM - Systeme de Gestion et de Recommandation de Transport de Marchandise`, and a `Deconnecter` button. The search form contains the following fields:

- Ville de depart: Blida
- Ville d'arrivee: Alger
- Type de Marchandise: alimentaire
- Poids: 890 Kilogramme
- Date: 23/06/2016 22:02
- Critere(s): Distance, Temps

A `Rechercher` button is located below the form. The results table is as follows:

| Chemin | Km Min | Details |
|--|----------|---------|
| <i>bida-boufarik-birroua-birkhadem-alger</i> | 44 75 | Details |
| <i>bida-boufarik-douera-benak-alger</i> | 46 70 | Details |
| <i>bida-boufarik-avaria-ayaha-alger</i> | 49 73 | Details |

Figure 4.6 Page de recherche itinéraire multicritères et l'affichage des résultats

III.3 Choix du moyen de transport selon l'itinéraire choisi :

La figure suivante montre le détail d'un itinéraire une fois choisi par l'utilisateur, ainsi que la recommandation du top trois transporteurs

The screenshot shows a web browser window with the URL: localhost:8080/SGRTM_v1.0/ResultatMulti2Connecte.jsp?km=44&min=75&itineraire=blida~boufarik~birtouta~birkhadem~alger&poids1=890&type_marche=Q. The page title is "SGRTM - Systeme de Gestion et de Recommandation de Transport de Marchandise". The user is logged in as "rebrab issad" with a "mon compte" button and a "Deconnecter" button.

The main heading is "blida~boufarik~birtouta~birkhadem~alger (44/75)".

On the left, there are search filters:

- Ville de depart: Blida
- Ville d'arrivee: Alger
- Type de Marchandise: alimentaire
- Poids: Poids (selected), Kilogramme
- Date: jj/mm/aaaa --
- Critere(s): Distance Temps
- Rechercher button

In the center, there is a table of transport options:

| Transporteur | Moyen de transport | Coût (KM) | Coût (Min) |
|------------------------------|--------------------|-----------|------------|
| <input type="radio"/> ouahai | gonow | 8800 | 11250 |
| <input type="radio"/> ouahai | jack-530 | 12320 | 13500 |
| <input type="radio"/> ouahai | gonow | 3036 | 5100 |

Below the table are buttons: "Envoyer demande" and "Revenez en arrière".

At the bottom, there is a section "Top 3 Societe":

| Ranking | Societe | Rating |
|---------|---------|--------|
| #1 | rostom | 1.0 |
| #2 | tantag | 0.255 |

Figure 4.7 Page d'affichage des moyens de transport et la recommandation des transporteurs

III.4 Compte client :

La figure suivante montre l'espace personnel du client où il pourra modifier ses coordonnées, ainsi que l'historique des demandes qu'il a effectué et leurs état actuel (refusée/en cours/terminée). De plus, le client pourra noter une mission une fois achevée.

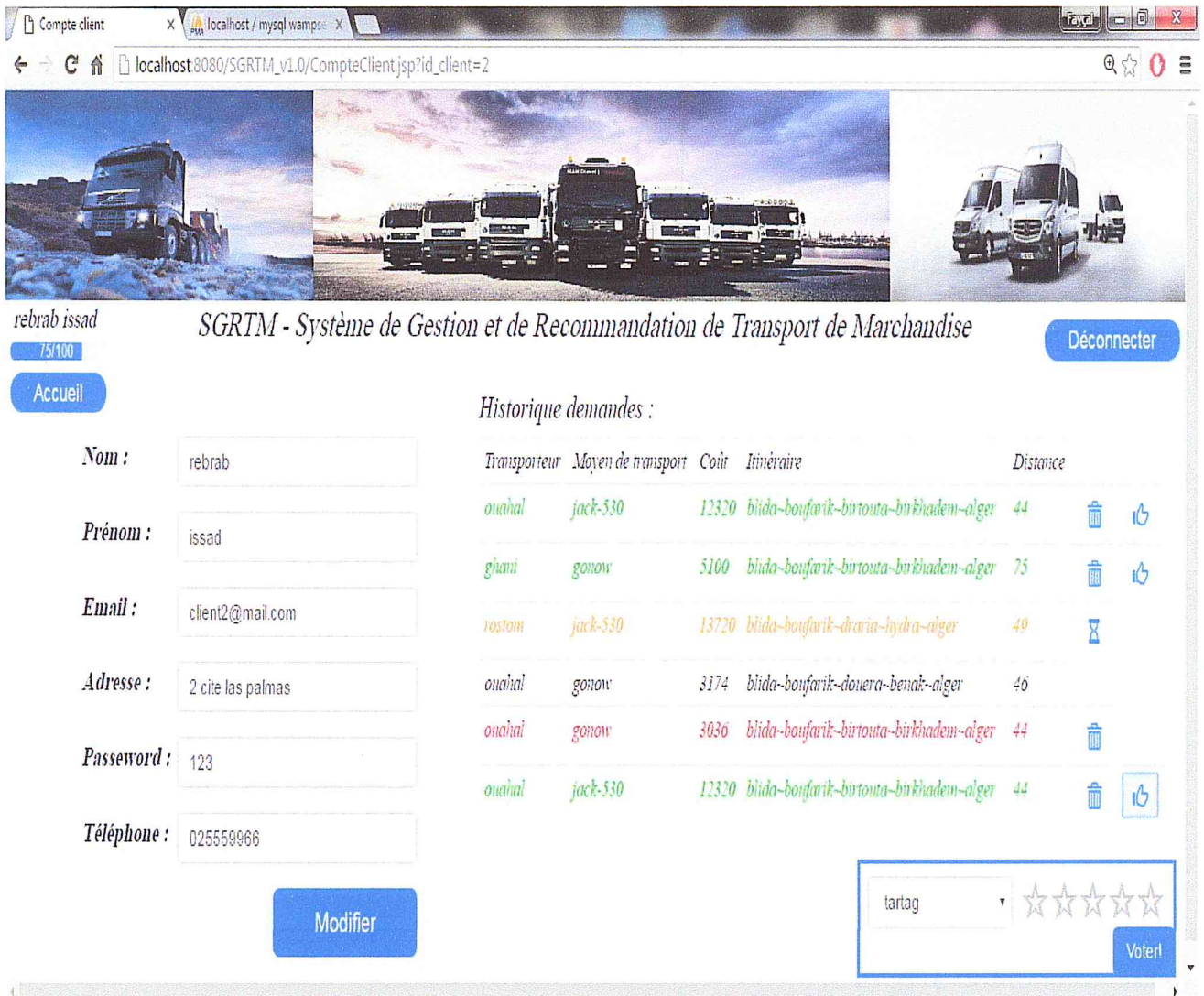


Figure 4.8 Page du compte client

III.5 Compte administrateur :

La figure suivante montre l'espace personnel de l'administrateur où il effectuera la gestion des comptes client/transporteur, ainsi que la gestion des moyens de transport.

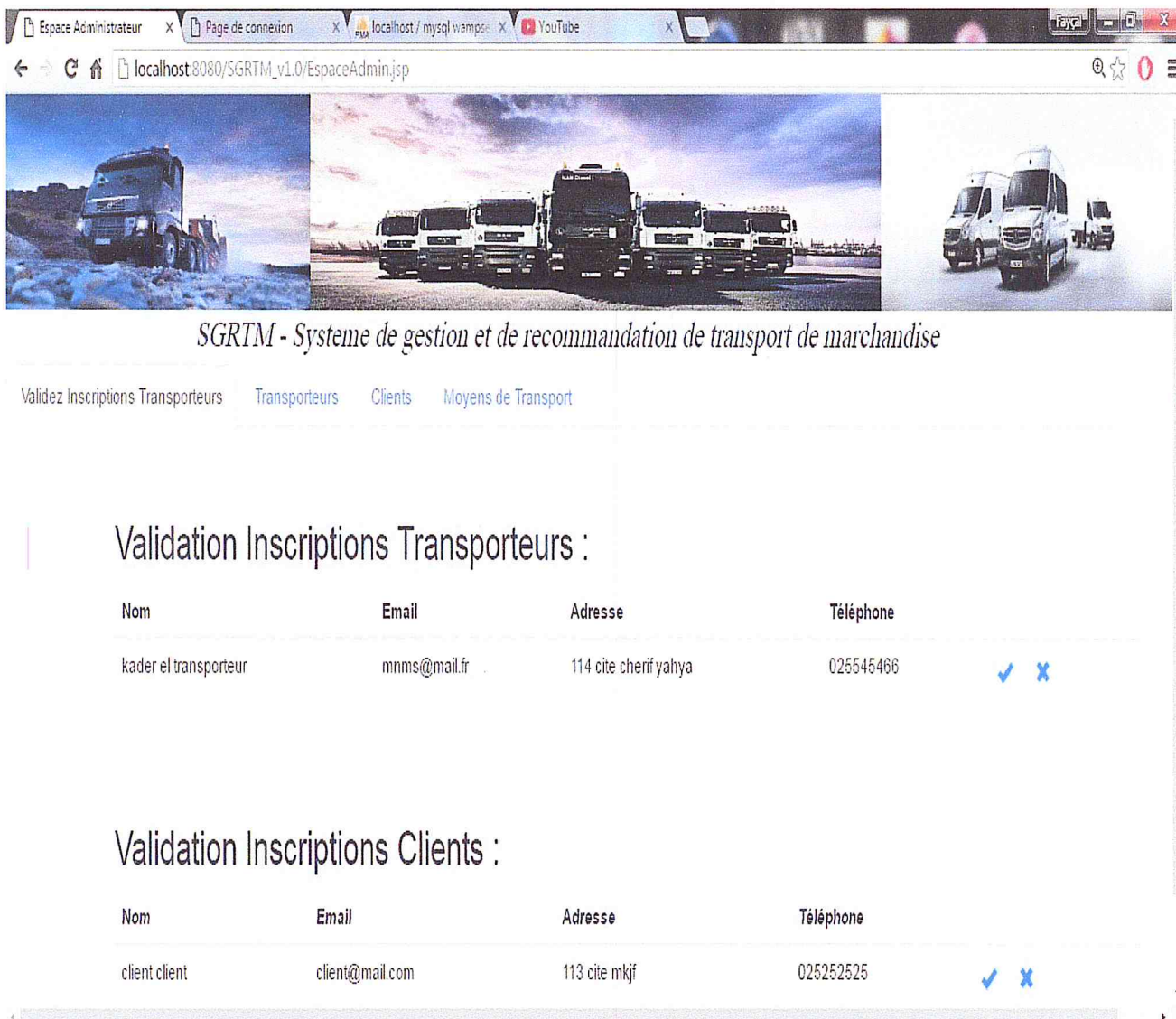


Figure 4.9 Page du compte administrateur

III.6 Compte transporteur :

La figure suivante montre l'espace personnel du transporteur où il pourra modifier ses informations, ajouter un moyen de transports, gérer les demandes clients et voir la disponibilité de ses moyens.

The screenshot shows a web browser window with the URL `localhost:8080/SGRTM_v1.0/CompteClientDeleteDemande?id_transporteur=2&id=1&str=transporteur`. The page content is divided into several sections:

- Coordonnées :** A form with fields for Nom (ouahal), Prenom (mohamed), Adresse (6 cite las palmas), Mail (trans2@mail.com), Telephone (025566666), and Password (123). A blue 'Modifier' button is located below the form.
- Les demandes :** A table listing transport requests.

| Cliant | Moyen de transport | Poids | Critere | Date | Itineraire | Distance |
|--------------------|--------------------|-------|----------|------------------|--------------------------------------|----------|
| redrab issad | jack-530 | 09 | distance | 2016-06-17 22:02 | blida-boufarik-bououa-boukhoum-alger | 44 |
| zeghar messaoud | jack-530 | 877 | temps | 2016-06-25 00:00 | blida-boufarik-douera-benak-alger | 70 |
| zeghar messaoud | gonow | 0 | temps | 2016-06-22 22:02 | blida-boufarik-douera-benak-alger | 70 |
| zeghar messaoud | jack-530 | 0 | temps | 2016-06-22 22:02 | blida-boufarik-douera-benak-alger | 70 |
- Vos moyens de transport :** A table listing transport means.

| Moyen | Matricule | Capacité | Type marchandise | Disponibilité |
|----------|-----------|----------|------------------|--------------------------|
| gonow | mmms1 | 930 | alimentaire | <input type="checkbox"/> |
| jack-530 | mmms5 | 5300 | alimentaire | <input type="checkbox"/> |
| gonow | mmms9 | 930 | alimentaire | <input type="checkbox"/> |
- Ajout moyen de transport :** A form with a dropdown menu (gonow), input fields for matricul, prix(km), and prix(min), and an 'Ajouter' button. Below it is a blue button labeled 'Ajouter un nouveau moyen'.

Figure 4.10 Page du compte transporteur

I.V Conclusion :

Dans ce chapitre nous avons présenté l'implémentation de notre système. Nous avons introduit les plateformes et les différents outils utilisés, puis nous avons donné une description du système à travers des fenêtres de capture d'écran qui représentent les interfaces de notre système. Nos interfaces sont conçues de manière à être simples et faciles à utiliser.

CONCLUSION GENERALE

Dans ce mémoire, nous avons conçu et réalisé un système de gestion et de recommandation de transport de marchandises qui met en relation les demandeurs et les offreurs des moyens de transport en utilisant les modules de recherche des plus courts chemins ainsi que le module de recommandation de transporteurs.

Pour ce faire, nous avons présenté dans un premier temps les problèmes de cheminement dans les réseaux de transport routiers ainsi que la résolution de ces derniers qui nécessite la recherche des plus courts chemins selon les critères choisis.

Ensuite, nous avons introduit dans le deuxième chapitre les systèmes de recommandation ce qui nous a permis de voir les différents travaux de recherche sur la recommandation dans le domaine de transport.

Notre système est conçu comme une réponse aux problèmes posés dans le but de trouver les plus courts chemins satisfaisants les critères choisis par les clients et de trouver les moyens de transport qui correspondent à leurs demande à moindre cout tout en recommandant une liste des meilleurs transporteurs.

Les modules utilisés notamment l'algorithme de Dijkstra et l'algorithme de Martins ont bien servi à résoudre le problème du plus court chemin unicritère et multicritères, ainsi que le module de recommandation implémenté suivant l'approche de la recommandation sociale, permettant d'enrichir notre système en offrant un support d'aide à la décision aux utilisateurs du système.

Notre système permet de rechercher les plus courts chemins dans les deux cas unicritère et multicritères tout en recommandant les transporteurs susceptibles à intéresser le client lors de la demande transport, ce qui permet de satisfaire les différentes exigences de ce dernier. En revanche, notre système ne prends pas en compte la notion du temps réel c'est-à-dire on ne peut pas introduire le trafic et les travaux en considération lors de la recherche des plus courts chemins.

Nous pourrons par la suite améliorer notre système en rajoutant des modules qui permettront la mise à jour automatique des informations requises lorsqu'on calcule la durée d'un chemin, en prenant en compte le trafic et les travaux se trouvant en temps réel sur les différents tronçons de ce chemin. Nous pourrons aussi rajouter quelques critères lors de la recherche du plus court chemin, tel que : le critère de la sécurité ou bien la fiabilité du transporteur.

Références Bibliographiques

- [BER, 59] Bernard Roy, « Transitivité et connexité. », C. R. Acad. Sci. Paris, vol. 249, 1959, p. 216–218
- [CAN, 01] Candillier, L. (2001). *Apprentissage automatique de profils de lecteurs*. Lille: Laboratoire d'Informatique Fondamentale de Lille.
- [DIJ, 59] Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische, Mathematik*, 1: 269-271, 1959.
- [GAL, 06] Galand, L. (2006). Interactive search for compromise solutions in multicriteria graph problems. , pp. 22–25.
- [GAV, 02] Gabrel, V. and Vanderpooten, D. (2002). Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. *European Journal of Operational Research*, 139(3), 533–542.
- [GRA, 10] Gräbener, T. (2010). *Calcul d'itinéraire multimodal et multiobjectif en milieu urbain* (Doctoral dissertation, Université des Sciences Sociales-Toulouse I) 27, 32 p.
- [HAN, 80] Hansen, P. (1980). Bicriterion path problems. , pp. 109.
- [JDK, 98] John Breese, David Heckerman, and Karl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *uncertainty in Artificial Intelligence*, 1998.
- [KLM, 06] N. Karacapilidis, A. Lazanas, G. Megalokonomos, P. Moraitis, On the Development of a Web-based System for Transportation Services, *Information Sciences*, 176(13) (2006), pp. 1801-1828.
- [LAM, 06] A. Lazanas, G. Megalokonomos, Optimizing Alternative Routes Retrieval in an Agent-based Transportation Management System. *Proceedings of the International Conference on Service Systems and Service Management (ICSSSM 2006)*, Troyes, France, pp. 1525-1530.
- [LAU, 06] Laurent Canet, Algorithmique, graphes et programmation dynamique, *Notes de Cours & Rapport de Travaux Pratiques*, 2003
- [LAZ, 09] Lazanas, A. (2009). Performing Hybrid Recommendation in Intermodal Transportation-the FTMarket System's Recommendation Module. *arXiv preprint arXiv:0909.2376*.
- [LUC, 10] Lucas Letocart, Cours d'Algorithmique de graphes, Institut Galilée, Université Paris 13.
- [LZS, 09] Ludwig, B., Zenker, B., & Schrader, J. (2009). Recommendation of

personalized routes with public transport connections. In *Intelligent interactive assistance and mobile multimedia computing* (pp. 97-107). Springer Berlin Heidelberg.

- [MAR, 84] Martins, E. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2), 236–245.
- [MAR, 84] Martins, E.Q.V. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16: 236-245, 1984.
- [MAT, 12] Mathieu. (2012, Avril 25). *Les algorithmes de recommandation*. Consulté le Mars 22, 2016, sur Podcast Science: <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>
- [MPJ, 07] Maria, J., Pangiliana, A. and Janssens, G. (2007). Evolutionary algorithms for the multiobjective shortest path planning problem. *International journal of computer and information science and engineering*, 1(1), 54–59.
- [OAK, 98] Douglas Oard and Jinmook Kim. Implicit feedback for recommender systems. Faculty Publications, pages 81–83, January 1998.
- [OCDE, 01a] OCDE (2001a), Perspectives de l'environnement de l'OCDE, Paris.
- [OCDE, 01b] OCDE (2001b), Développement durable: les grandes questions, Paris.
- [OECD, 04] OECD. (2004), *Transport urbain de marchandises : Les défis du XXI^e siècle*, OECD Publishing, Paris.
- [OIE, 13] *Notes de conjuncture & articles*. (23, Septembre 2014). Consulté le Février 18, 2016, sur Observatoire de l'industrie électrique: <http://www.observatoire-electricite.fr/La-consommation-d-energie-en-320>
- [PHI, 99] Philip Chan. A non-invasive learning approach to building user profiles. *Web Usage Analysis and User Profiling*, 1999.
- [PPCC] Messaoudi, C. (s.d.). *THÉORIE DES GRAPHERS*. Consulté le Février 2016, 20, sur <https://sites.google.com/site/cuatmessaoudi/home/cours-4>
- [Rating average] *Rating & Ranking Average Calculations*. (s.d.). Consulté le Mars 06, 2016, sur Survey Monkey: http://help.surveymonkey.com/articles/en_US/kb/What-is-the-Rating-Average-and-how-is-it-calculated#part1
- [Rating stars] <http://www.shutterstock.com/s/rating+stars/search-vectors.html?page=1&inline=298117844>

- [REV, 97] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3) :56–58, 1997.
- [RRS, 11] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 1–35. Springer US, January 2011
- [RS] *Recommender Systems*. (2012). Consulté le Mars 02, 2016, sur <http://recommender-systems.org/>
- [SRAD, 2010] Systèmes de recommandation. (2010). *Journées thématiques AAFD'10* (p. 6). NumSight.
- [ZLT, 01] Zitzler, E., Laumanns, M., Thiele, L. and others (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. , pp. 95–100.