



École Polytechnique de l'Université de Tours  
64, Avenue Jean Portalis  
37200 TOURS, FRANCE  
Tél. +33 (0)2 47 36 14 14  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)

**Département Informatique**  
**5<sup>e</sup> année**  
**2009 - 2010**

**Projet de Fin d'Etude**



**Nouveaux critères pour du calcul  
d'itinéraire vélo**

**Encadrants**

Gaël Sauvanet  
[gael.sauvanet@univ-tours.fr](mailto:gael.sauvanet@univ-tours.fr)  
Emmanuel Néron  
[emmanuel.neron@univ-tours.fr](mailto:emmanuel.neron@univ-tours.fr)

**Etudiant**

Ke SHANG  
[ke.shang@etu.univ-tours.fr](mailto:ke.shang@etu.univ-tours.fr)

DI5 2009 - 2010

Université François-Rabelais, Tours

Version du 29 mars 2010



# Table des matières

---

<b>1</b>	<b>Présentation du projet</b>	<b>8</b>
1.1	Contexte . . . . .	8
1.2	Description de l'existant . . . . .	8
1.3	Objectifs du projet . . . . .	10
1.4	Description des Besoins . . . . .	10
1.4.1	Besoins fonctionnels . . . . .	10
1.4.2	Besoins non fonctionnels . . . . .	12
1.5	Environnement du projet . . . . .	12
1.6	Technologies utilisées . . . . .	12
1.6.1	Client . . . . .	12
1.6.2	Serveur . . . . .	13
1.6.3	XML-RPC . . . . .	13
1.6.4	Base de données . . . . .	13
<b>2</b>	<b>Méthodes</b>	<b>14</b>
2.1	Présentation du problème . . . . .	14
2.2	Méthodes utilisées . . . . .	15
2.2.1	Algorithme de Dijkstra . . . . .	15
2.2.2	Optimisation multi-objectif . . . . .	17
2.2.3	Algorithme de BCA* . . . . .	19
<b>3</b>	<b>Base de données</b>	<b>21</b>
3.1	Présentation de la base de données . . . . .	21
3.2	Postgresql et Postgis . . . . .	23
3.3	Réalisation de la base de données . . . . .	23
<b>4</b>	<b>Réalisation</b>	<b>24</b>
4.1	Test d'intégration de site Géovélo . . . . .	25

# Table des figures

---

1.1	Le site Géovélo . . . . .	8
1.2	Le diagramme de cas d'utilisation de Géovélo . . . . .	9
1.3	Le modèle de vos préférencea . . . . .	12
2.1	Solutions entre les critères . . . . .	14
2.2	Exemple de l'algorithme de Dijkstra . . . . .	16
2.3	Résultat d'algorithme de Dijkstra . . . . .	17
2.4	Pareto frontier . . . . .	18
2.5	Relation de dominance . . . . .	19
2.6	Un graphe à traiter . . . . .	19
2.7	Algorithme de BCA* . . . . .	20

# Liste des tableaux

---

1.1	Le composant de l'effort . . . . .	11
2.1	Exemple de bi-critère . . . . .	15
2.2	Le résultat de calcul . . . . .	16

# Remerciements

---

Je souhaite tout d'abord remercier beaucoup Gaël Sauvanet pour m'avoir donné toutes les informations nécessaires et tous les conseils. Il m'a aidé tout au long du projet de fin d'étude. Son point de vue m'a été d'une grande utilité pour la compréhension du fonctionnement du système existant. Il m'a expliqué clairement et m'a guidé pour les réflexions concernant les algorithmes.

Un merci à Emmanuel Néron pour ses instructions et ses connaissances. Il m'a proposé des bonnes idées concernant ce projet.

Enfin, merci à tous ceux qui m'ont aidé durant la période du projet de fin d'étude.

# Introduction

---

Lors de la cinquième année du cycle d'ingénieur de l'École Polytechnique de l'Université de Tours Département Informatique. Nous devons effectuer un Projet de Fin d'Étude(PFE) du 10/2009 au 5/2010. Le but est de familiariser les étudiants avec les compétences acquises en cours et d'en acquérir de nouvelles.

Le sujet du projet de fin d'étude est "Nouveaux critères pour du calcul d'itinéraire vélo". Ce projet est basé sur des travaux de mon encadrant Gaël Sauvanet et l'étudiante Juan Zhang dans le cadre du projet Géovélo réalisé par l'association Autour du Train. L'objectif de ce projet de de mettre en place un nouvel algorithme et implémenter des nouveaux critères.

Dans ce rapport, je vais d'abord présenter mon projet. Et puis je vais détailler les méthodes et les algorithmes utilisées, la spécification de la base de données. Enfin ce sera la réalisation de ce projet de chaque étape, je vais aussi parler les difficultés rencontrées.

# Présentation du projet

## 1.1 Contexte

Géovélo est un site qui permet de calculer des itinéraires adaptés au réseau cyclable. Nous pouvons choisir le départ et la destination. Entre les deux points Géovélo nous fournit des itinéraires en tenant compte de différents critères. L'utilisateur peut choisir un itinéraire selon ses préférences de distance et de sécurité. Géovélo est une première en France. L'interface cartographique est basée sur l'API Google. Pour l'instant, il n'est disponible que pour la ville de Tours.

## 1.2 Description de l'existant

Pour l'instant, le site Géovélo fonctionne bien pour trouver l'itinéraire en fonction du point de départ et un point de destination en considérant deux critères (distance et sécurité). Il utilise une version multi-critère de DIJKSTRA ALGORITHM pour calculer l'itinéraire.

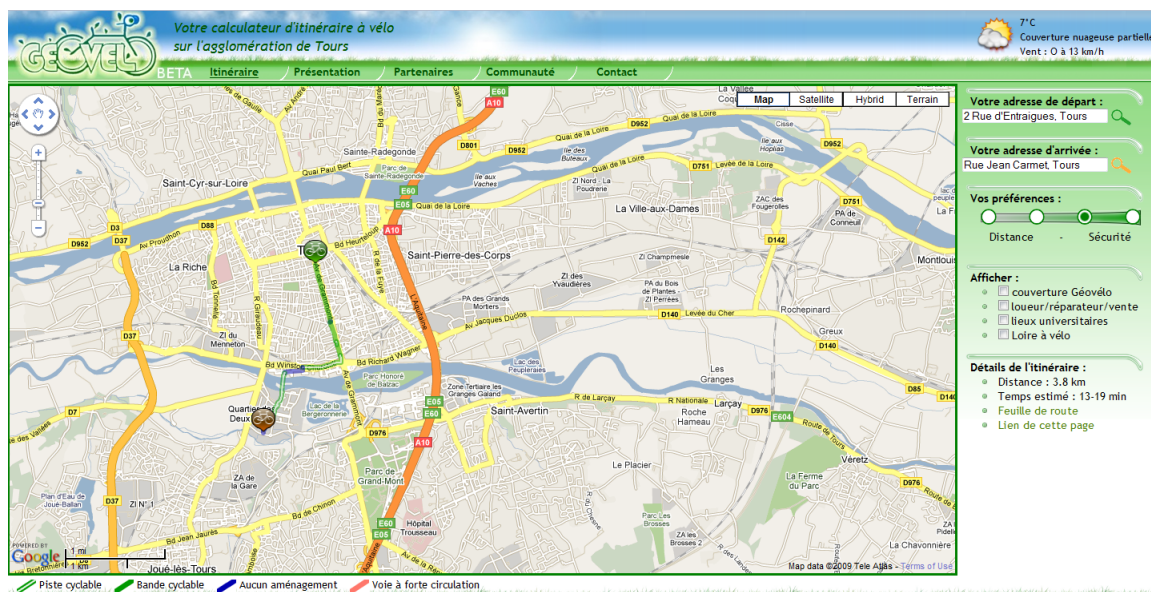


FIGURE 1.1 – Le site Géovélo

L'image ci-dessus est tirée du site web Géovélo actuel. Sur ce site, les fonctionnalités suivantes sont possibles :

### Sélection d'un itinéraire

Pour saisir un itinéraire, il existe 3 méthodes :

- Saisir une adresse de départ ou d'arrivée puis cliquer sur la loupe ;
- Déplacer les curseurs de départ et d'arrivée directement sur la carte à l'aide de la souris ;
- Utiliser le clic droit de la souris sur la carte.



### Sélection d'une préférence

Permet de choisir un itinéraire selon ses préférences, de distance et de sécurité.

### Affichage d'informations géographiques

- Couverture Géovélo : Afficher un cadre qui permet de visualiser la zone géographique couverte par Géovélo ;
- Loueur/réparateur/vente : Afficher l'ensemble des professionnels du vélo ;
- Lieux universitaires : Afficher les principaux lieux universitaires sur l'agglomération. On peut cliquer sur chaque point d'intérêt pour obtenir des informations supplémentaires ;
- Loire à vélo : Afficher le tracé de la Loire à vélo.

### Affichage des détails de l'itinéraire

Après avoir saisi le départ et la destination, Géovélo nous donne la solution avec la distance et le temps estimé. Nous pouvons cliquer sur la feuille de route pour obtenir des informations détaillées. Nous pouvons cliquer sur "lien de cette page" pour obtenir un lien vers un itinéraire de manière à pouvoir le partager par mail ou autre.

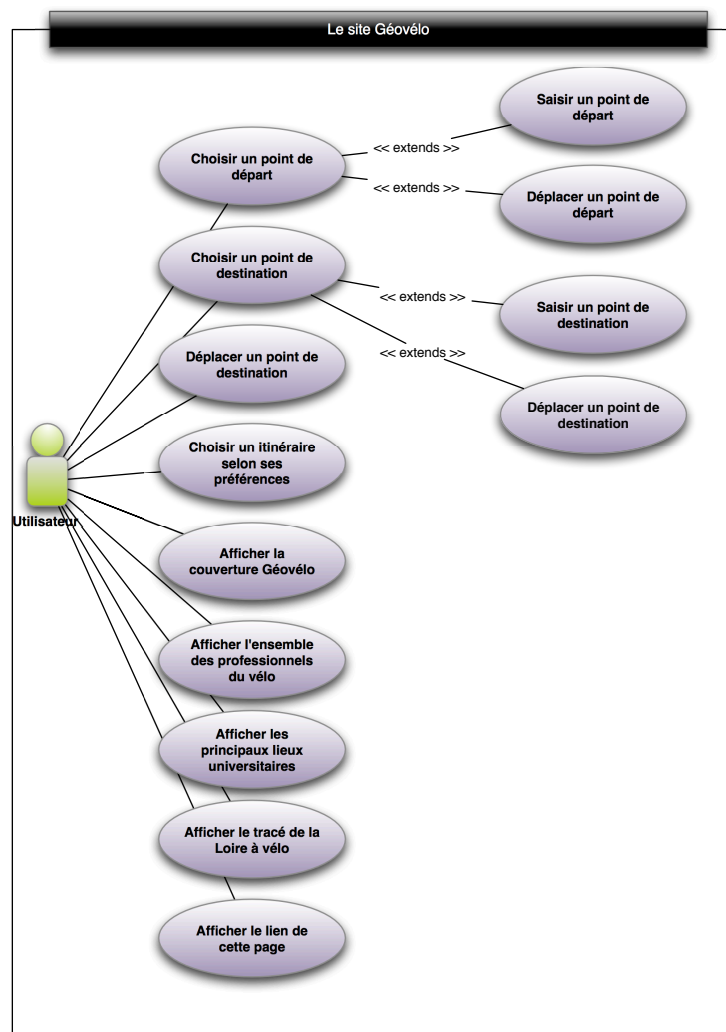


FIGURE 1.2 – Le diagramme de cas d'utilisation de Géovélo

## 1.3 Objectifs du projet

Pour les objectifs de mon projet de fin d'étude, il contient deux missions principalement :

- **Mettre en place un nouvel algorithme BCA\***

L'algorithme de ce projet est développé en utilisant le plus court chemin (une version multi-critère de l'algorithme de Dijkstra) [Dijkstra(1959)]. Évidemment cet algorithme fonctionne bien mais il perd beaucoup de temps du calcul.

Pour l'améliorer mon encadrant propose un nouveau algorithme : BEST COMPROMISE SOLUTION A\* (BCA\* - une généralisation d'A\* pour la recherche de solutions de compromis dans des problème d'optimisation multi-objectifs) [Futtersack and Perny(2000)]. C'est un algorithme qui est spécialement conçu pour la recherche de solutions de compromis. L'intérêt est que nous n'avons pas besoin de calculer tous les résultats, le mécanisme de BCA\* permet d'enlever des points qui ne nous intéressent pas pour trouver une solution plus rapidement en réduisant l'espace de recherche.

- **Implémenter des nouveaux critères**

Pour l'instant, le site Géovélo fonctionne bien en considérant seulement deux critères (distance et sécurité). L'objectif de ce projet est d'implémenter les nouveaux critères qui contiennent trois critères principaux (effort, sécurité, tourisme) :

**Le critère d'effort** Il existe de nombreuses définitions de l'effort, l'idée à retenir est que nous cherchons à évaluer la difficulté d'un trajet. Nous pouvons prendre en compte plusieurs critères comme la pente, la distance, la météo... Il y a aussi tout l'aspect psychologique : la difficulté n'est pas la même suivant les personnes. Par exemple, si quelqu'un est motivé, il pourra surmonter plus facilement une côte. Il est important de prendre en compte l'effort car c'est un élément de l'information que nous le devons fournir à un cycliste. nous voulons favoriser ce mode de transport, il faut donc lui apporter de nombreux services. Connaître le parcours le plus simple ou "mesurer" la difficulté de son parcours est donc utile pour un cycliste.

**Le critère de sécurité** Le critère de sécurité consiste à prendre en compte l'itinéraire en différents sécurités. Il permet utilisateur de choisir l'itinéraires conseillé pour le vélo.

**Le critère de tourisme** Le critère de tourisme consiste à trouver un itinéraire. Dans cet itinéraire, L'ensemble de points contiennent le point de départ, les point d'intérêts tels que les musées, les jardins, les monuments... et le point de destination. Un touriste devra commencer par le point de départ jusqu'au point de destination en passant obligatoirement par tous les points d'intérêts sélectionnés.

Selon le temps restant, nous devons aussi modifier l'interface du site Géovélo.

## 1.4 Description des Besoins

### 1.4.1 Besoins fonctionnels

Nous avons identifié les besoins fonctionnels suivants. Ces besoins identifient le travail à réaliser dans le cadre du projet.

- **Besoin 1** : Mettre en place BCA\* en C++ sous la forme d'un web service ;

Il consiste à lire d'abord des sources et comprendre la fonctionnalité d'algorithme existant, et puis implémenter BCA\* pour deux critères (distance et sécurité), enfin tester la fonctionnalité et la compatibilité ;

BCA\* est une méthode multi-objective. Cet algorithme prend en paramètre les préférences de l'utilisateur (poids sur chaque critère), après il retourne l'itinéraire le plus adapté aux préférences de l'utilisateur. Cet algorithme explore en priorité les labels les plus "prometteurs" et éliminent ceux qui ne sont pas intéressants.

- Besoin 2 : Implémenter des nouveaux critères ;

Cette tâche consiste à modéliser de nouveaux critères (effort, sécurité, tourisme) en collaboration avec 3 étudiants du DA dont leur sujet porte sur l'effort, la sécurité et le tourisme :

### Le critère d'effort

Nous pouvons voir la figure ci-dessous fournie par Simon Laporte (un étudiant du DA), le critère d'effort contient beaucoup de paramètres que nous devons considérer, cette tâche sera divisée en deux parties. Une partie pour Simon, il devra modéliser ce critère et essayer de résumer d'une formule permettant de considérer tous les paramètres. Et pour moi, je devrai implémenter cette formule et afficher la valeur d'effort calculé par cette formule. Nous nous répéterons jusqu'à ce que nous trouverons la formule optimale.

Sujet	Thèmes	Critères	Sous-critères
Efforts	Tâche	Motif	Domicile-Travail
			Loisir
			Sport
		Objectifs	Vitesse
		Distance	
		Dénivelé	
	Environnement	Climat	Température
			Hygrométrie
			Vent
		Revêtement	
		Intersections	Nombre
		Matériel	Poids
			Roues
		Participants	
	Individus	Physique	Age
			Sexe
			Aptitudes
		Psychologique	Difficulté perçue
			Effort consenti

TABLE 1.1 – Le composant de l'effort

### Le critère de sécurité

Le site Géovélo existe déjà le critère de sécurité, il suffit de récupérer d'abord les données améliorées par l'étudiant du DA, et puis nous devons mettre à jour la base de données.

Quand nous allons finir implémenter des nouveaux critères. Nous pourrons utiliser BCA\* pour tester la fonctionnalité et la compatibilité.

- Besoin 3 : Modifier l'interface du site web.

Nous allons implémenter un nouvel algorithme et ajouter les nouveaux critères pour le site Géovélo, donc il faudra aussi utiliser le langage PHP/XAJAX/CSS pour modifier la zone de vos préférences.

Le modèle est comme la figure ci-dessous :

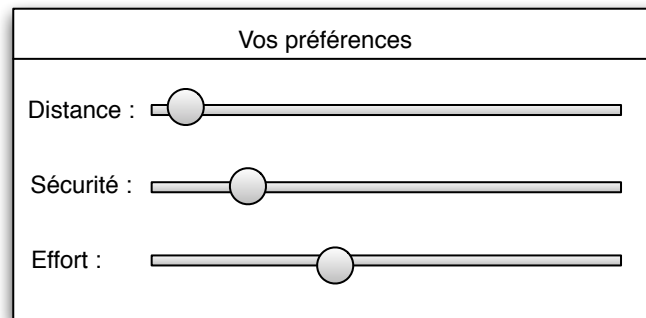


FIGURE 1.3 – Le modèle de vos préférences

Parce que nous utilisons nouvel algorithme pour calculer l'itinéraire, donc il faut modifier un peu sur le site Géovélo. Nous devons d'abord ajouter notre nouveau critère pour la zone de vos préférences. Ensuite le modèle de sélection n'est pas très évident pour l'utilisateur, mon encadrant propose une idée en utilisant Slider à la place.

#### 1.4.2 Besoins non fonctionnels

- Besoin 4 : En terme de fiabilité, le site Géovélo devra être stable et devra supporter une charge d'environ 5000 visiteurs/mois ;
- Besoin 5 : La documentation d'installation et de configuration doit être rédigée ;
- Besoin 6 : Les commentaires du code source doivent être lisibles et compréhensibles ;
- Besoin 7 : Le temps de calcul d'un itinéraire doit être inférieur à 3 secondes ;
- Besoin 8 : L'interface du site Géovélo doit être conviviale et ergonomique.

### 1.5 Environnement du projet

Ce PFE est effectué sous le système d'exploitation Ubuntu8.

Le codage doit être réalisé en C++. Sous Ubuntu8, nous installons l'outil codeblocks comme environnement de développement et les paquets concernant la librairie de communication XMLRPC.

Nous utilisons Postgresql comme SGBD et son module géographique Postgis. Nous utilisons l'outil pgadmin3 pour gérer la base de données et l'outil qgis pour visualiser et traiter les données géographiques.

Nous utilisons le serveur apache2 et son module PHP5 comme serveur web.

### 1.6 Technologies utilisées

Pendant mon projet de fin d'étude, il est nécessaire d'utiliser différents langages et outils. Pour la partie de calcul d'itinéraires, nous utilisons C++ et LEDA. Pour la partie de site Géovélo, nous utilisons HTML, CSS, Javascript, PHP, Xajax, API de Google Maps. Pour la partie de base de données, nous utilisons PostgreSQL et PostGIS. Je vais les diviser en quatre parties (Client, Serveur, XMLRPC, Base de données) pour expliquer leur utilisations et fonctionnalités.

#### 1.6.1 Client

**AJAX** Une technique de développement web. Il permet de créer des applications web interactives. Le plus gros avantage est qu'avec l'utilisation de AJAX, il est possible de maintenir des données sans besoin

de rafraîchir la page toute entière. Les applications web peuvent rendre une réponse à l'utilisateur plus rapidement et éviter d'envoyer les informations qui n'ont pas besoin d'être modifiées.

**XAJAX** Un framework open source utilisant le langage PHP, il permet d'utiliser le langage PHP pour développer des fonctionnalités AJAX sans connaître le langage JavaScript.

**API Google Maps** Nous savons que Google Maps est un outil utilisé pour suivre l'emplacement actuel, les conditions en temps réel du trafic, mais il offre aussi un outil de recherche, et l'API Google Maps permet d'intégrer Google Maps dans son propre site en utilisant JavaScript. Ensuite il est possible de développer avec JavaScript pour manipuler la carte.

### 1.6.2 Serveur

**LEDA** Une librairie C++ qui offre une grande variété d'algorithmes et de structures adaptés aux graphes et aux calculs géométriques. Dans mon projet il est nécessaire de calculer un itinéraire sur la carte, donc c'est plus facile de calculer l'itinéraire en utilisant cette librairie existante.

### 1.6.3 XML-RPC

Un protocole RPC (Remote procedure call), un ensemble de spécifications qui permettent à des applications tournant dans des environnements différents de communiquer facile. Cette technique qui permet à deux machines de communiquer ou d'échanger des données au format XML. Il est utilisé pour échanger des informations entre des pages du site et le serveur de calcul d'itinéraire. Nous pouvons l'utiliser pour transmettre des structures de données complexes. Il existe huit types de données : array, base64, boolean, date/time, double, integer, string, nil.

### 1.6.4 Base de données

**PostgreSQL** Un système de gestion de base de données relationnelles. Actuellement, il est un des systèmes les plus puissants, le plus riche en fonctionnalités et le plus complexe des bases de données du logiciel libre. Il fonctionne principalement sous le système UNIX (Linux, FreeBSD, Ubuntu...).

**PostGIS** Une extension qui ajoute la capacité de gestion des données spatiales sur le système PostgreSQL. Il offre des fonctionnalités comme les objets spatiaux, l'indexation spatial, les fonctions de manipulation spatiales et les opérateurs. En résumé il permet de manipuler des données géographiques facilement.

# Méthodes

---

Pour calculer l'itinéraire, nous devons utiliser plusieurs méthodes et algorithmes. Dans ce chapitre, nous allons détailler les méthodes et algorithmes que nous utilisons.

## 2.1 Présentation du problème

Il existe déjà beaucoup d'études concernant des problèmes de plus court chemin avec l'objectif unique en utilisant DIJKSTRA ALGORITHME. L'objectif est de trouver un chemin en minimisant la somme du coût de ses arcs entre le départ et la destination. Mais l'optimisation d'un objectif unique n'est pas suffisante, car il existe plusieurs objectifs que nous devons considérer dans le problème véritable.

Par exemple, l'objectif de site Géovélo ne consiste pas seulement à trouver un itinéraire plus court, il faut considérer plusieurs critères (distance, sécurité, effort, tourisme) en même temps. Les différents critères sont toujours indépendants sur chaque itinéraire, il n'existe pas de relation entre eux, donc nous ne pouvons pas considérer qu'une critère. Géovélo permet de fournir la solution bien équilibrée en considérant le poids de chaque critère que l'utilisateur choisit.

Par exemple, nous pouvons considérer que deux critères (distance et sécurité) :

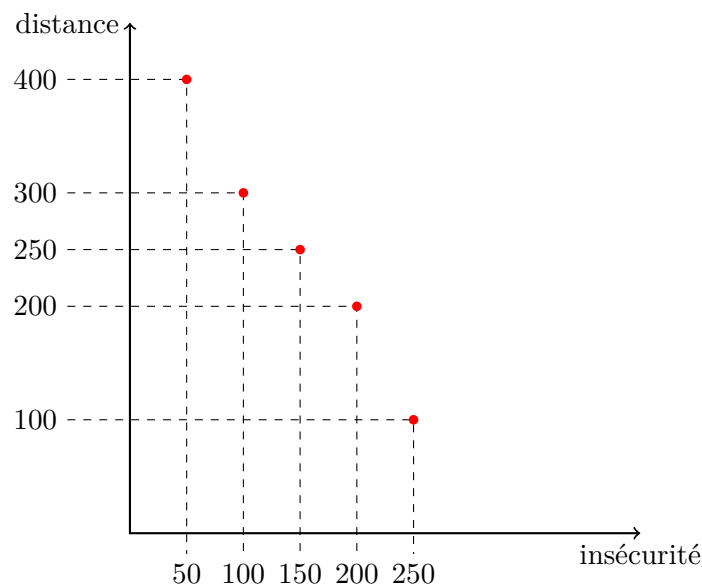


FIGURE 2.1 – Solutions entre les critères

Solution	Distance	Insécurité
1	400	50
2	300	100
3	250	150
4	200	200
5	100	250

TABLE 2.1 – Exemple de bi-critère

Nous pouvons voir l'exemple ci-dessus, évidemment nous ne pouvons pas déterminer laquelle est la meilleure. Ca dépend la sélection de l'utilisateur. Ce type de problème est appelé **MULTIOBJECTIVE PROBLEM** [[Wikipedia](#)].

Pour l'instant, nous pouvons utiliser le calcul d'itinéraire bi-critère existant qui se base sur **DIJKSTRA ALGORITHM** pour calculer un itinéraire intermédiaire entre deux points adjacents. Cet algorithme est différent de l'algorithme classique **DIJKSTRA ALGORITHM**. Nous obtenons qu'une solution concernant le plus court chemin en utilisant l'algorithme classique. Mais cet algorithme existant nous fournit plusieurs solutions intermédiaires entre deux points adjacents. Chaque solution est bien représentée par un couple distance, insécurité correspondant aux deux critères.

Pour calculer un itinéraire entre le départ et la destination selon différents critères, la mécanique de l'algorithme existant est de trouver toutes les solutions au premier, et puis parmi les résultats il retient quelques itinéraires satisfaisants les besoins de l'utilisateur. Evidemment, cet algorithme fonctionne bien, mais toutes les solutions de calcul pourra prendre beaucoup de temps, de plus elles ne sont pas forcément utile si l'utilisateur a besoin d'obtenir qu'une solution de compromis.

## 2.2 Méthodes utilisées

Dans cette partie, je vais détailler les différentes méthodes utilisées comprenant **DIJKSTRA ALGORITHM**, **BCA\***, l'optimisation multi-objectif.

### 2.2.1 Algorithme de Dijkstra

**DIJKSTRA ALGORITHM** a été publié par l'informaticien néerlandais Edsger Dijkstra en 1959. Il sert à résoudre le problème du plus court chemin. Cet algorithme permet de trouver une chaîne de longueur minimale entre deux sommets d'un graphe orienté ou non. Dans ce projet, nous pouvons utiliser cet algorithme pour trouver le plus court chemin de chaque critère.

#### Principe de l'algorithme de Dijkstra

Nous considérons un graphe  $G = [x, U]$  qui contient éventuellement des circuits et où toutes les longueurs des arcs sont positives :  $\forall u \in U, l(u) \geq 0$  ;

Nous cherchons le plus court chemin du sommet 1 à tous les autres ;

Nous notons  $\Pi(i)$  : un potentiel associé au sommet  $i$ , qui représente la longueur du plus court chemin de 1 à  $i$  actuelle ;

Nous notons  $\bar{\Pi}(i)$  : le potentiel final associé à  $i$ , donc la longueur du plus court chemin de 1 à  $i$  ;

Nous notons  $S$  : l'ensemble des sommets pour lesquels nous connaissons le potentiel final.

---

**Algorithm 1** DIJKSTRA

---

1.  $\bar{\Pi}(1) = 0$   
 $\forall u/(1, j) \in U, \Pi(j) = l(1, j)$   
 $\forall u/(1, j) \notin U, \Pi(j) = \infty$   
 $S = 1$
  2. TQ  $S \neq X$  Faire  
     Sélectionner  $k \in X \setminus S / \min_{i \in X \setminus S} \Pi(i)$   
      $\bar{\Pi}(k) = \Pi(k)$   
     Pour tout  $l \in X \setminus S, (k, l) \in U$  Faire  
          $\Pi(l) = \min(\pi(l), \Pi(k) + l(k, l))$   
     FinPour  
      $S \leftarrow S \cup k$
- FTQ
- 

**Exemple de l'algorithme de Dijkstra**

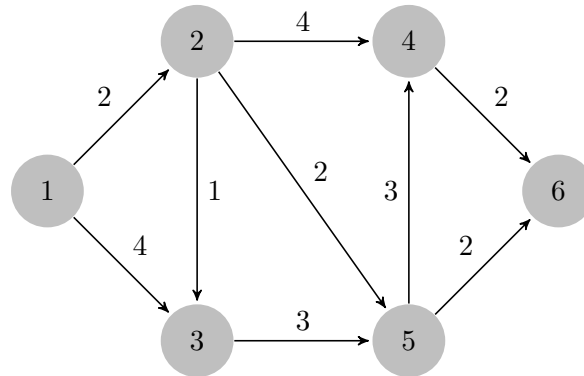


FIGURE 2.2 – Exemple de l'algorithme de Dijkstra

Nous pouvons utiliser la table ci-dessous pour montrer le résultat du calcul.

	1	2	3	4	5	6
$\Pi(0)$	0	2	4	$+\infty$	$+\infty$	$+\infty$
$\Pi(1)$		2	3	6	4	$+\infty$
$\Pi(2)$			3	6	4	$+\infty$
$\Pi(3)$				6	4	6
$\Pi(4)$				6		6
$\Pi(5)$						6

TABLE 2.2 – Le résultat de calcul



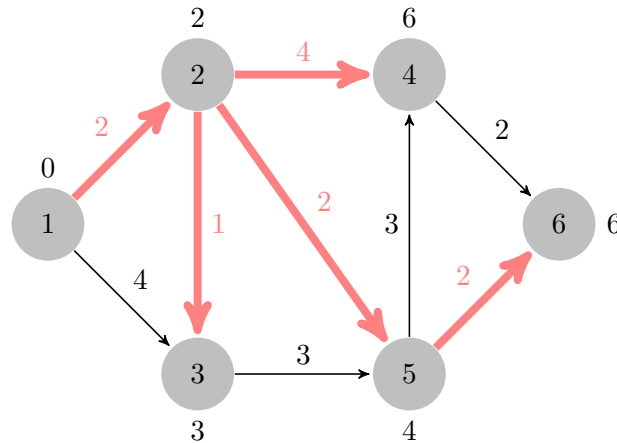


FIGURE 2.3 – Résultat d'algorithme de Dijkstra

Nous pouvons trouver les plus courts chemins entre le sommet de départ (sommet 0) et des autres sommets.

### 2.2.2 Optimisation multi-objectif

L'optimisation multi-objectif est une branche de l'optimisation combinatoire, elle est utilisée fréquemment dans nombreux secteurs de l'industrie concernés (Télécommunications, Transport, Environnement, Mécanique, Aéronautique, ...). Comme le suggère le nom, un problème d'optimisation multi-critère consiste à optimiser plusieurs fonctions objectif simultanément. C'est difficile de résumer des objectif en un objectif unique, car il existe des conflits entre ces objectif.

En général, Les objectifs de l'optimisation sont contradictoires, c'est-à-dire que l'objectif est de trouver des bons compromis parmi toutes les solutions. Ces problèmes peuvent être NP-complets.

Dans la partie suivante, je vais présenter la définition mathématique d'un problème d'optimisation multi-objectif.

#### Définition : Problème multi-objectif

Etant donné  $X = (x_1, x_2, \dots, x_n)$  est le vecteur des variables de décision. Il satisfait les contraintes ci-dessous :

$$g_i(X) \geq 0, i = 1, 2, \dots, k \quad (1)$$

$$h_i(X) = 0, i = 1, 2, \dots, l \quad (2)$$

Supposons  $r$  objectifs, et tous ces objectif sont conflictuels, la fonction optimale est :

$$f(X) = (f_1(X), f_2(X), \dots, f_r(X))$$

Trouver  $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ , optimiser  $f(X^*)$  sous les contraintes (1) et (2). Les solutions qu'on obtient sont les solutions PARETO EFFICIENCY. [\[Wikipedia\]](#)

#### Définition : Pareto efficiency

Pareto efficiency est un concept en économie. Ce terme est nommé d'après Vilfredo Pareto, un économiste italien qui a utilisé le concept dans ses études d'efficacité économique et la répartition des revenus. Un optimum de Pareto est un état dans lequel nous ne pouvons pas améliorer le bien-être d'un individu sans détériorer d'un autre.

Nous pouvons dire que le vecteur  $X^* \in F$  est un optimum de Pareto, si pour chaque  $X \in F$ , soit  $\forall i \in I (f_i(X) = f_i(X^*))$ ,  $I = \{1, 2, \dots, r\}$ , soit  $\exists j \in I, f_j(X) > f_j(X^*)$ .

Sachant que  $F = \{X \in R^n | g_i(X) \geq 0, i = 1, 2, \dots, k; h_j(X) = 0, j = 1, 2, \dots, l\}$

En général, l'optimisation multi-objectif donne un ensemble de solutions, plus d'une solution. Dans un ensemble de solutions P, on sélectionne un ensemble de solutions non dominées. Dans l'espace de recherche, les solutions non dominées sont les solutions Pareto optimal. L'ensemble de solutions non dominées forme PARETO FRONTIER.

L'objectif d'optimisation multi-objectif est de trouver un ensemble de solutions non dominées et proche du PARETO FRONTIER.

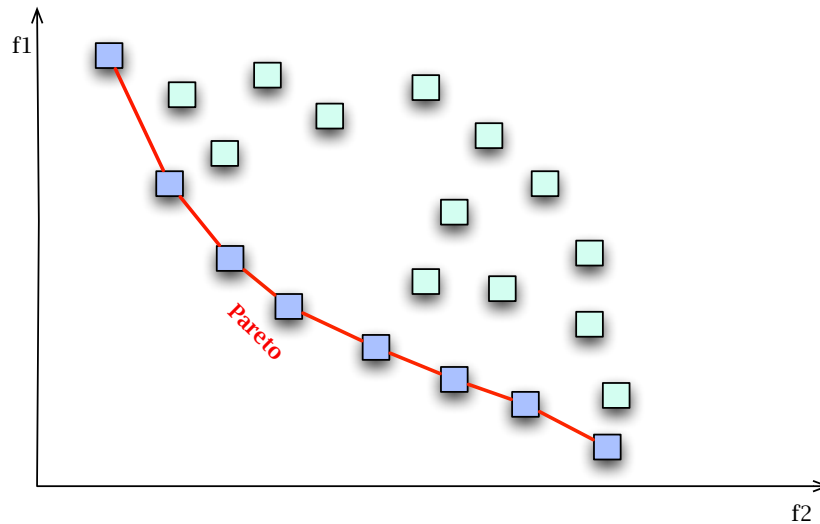


FIGURE 2.4 – Pareto frontier

### Définition : Relation de dominance

P est un ensemble de solutions, il existe k objectifs :  $f_1(), f_2(), \dots, f_k()$ . Pour chaque individu dans l'ensemble P. Il y a trois relations entre les deux solutions x et y. Pour x et y, soit x domine y, soit y domine x, soit x et y ne se domine pas leur même. Nous pouvons donc définir la relation ci-dessous :

#### – La dominance :

Une solution x et une solution y appartiennent à l'ensemble P, x domine y si et seulement si :  $\forall i \in \{1, 2, \dots, k\} : f_i(x) \leq f_i(y) \text{ et } \exists j \in \{1, 2, \dots, k\} : f_j(x) < f_j(y)$ . Nous notons  $y \prec x$ .

Si la solution x domine la solution y, nous disons que x est la solution non dominée et y est la solution dominée par x. Le symbole  $\prec$  représente la relation dominée.

#### – Non relation :

Une solution x et une solution y appartiennent à l'ensemble P, nous disons que la solution x et la solution y sont non relation si et seulement s'il n'existe pas la relation dominée entre elles.

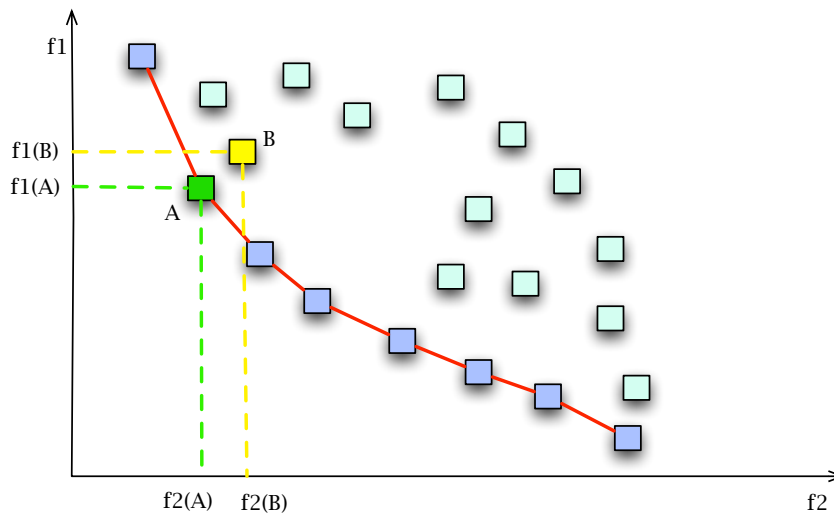


FIGURE 2.5 – Relation de dominance

Dans la figure ci-dessus, on peut savoir  $f1(B) > f1(A)$  et  $f2(B) > f2(A)$ . En fonction de la définition de dominance, le point A domine le point B. Nous notons  $B \prec A$ . Donc le point A est une solution non dominée et le point B est une solution dominée. Le point B n'appartient pas à l'ensemble de PARETO FRONTIER, parce qu'il est dominé par le point A.

### 2.2.3 Algorithme de BCA\*

Pour résoudre le problème de l'algorithme existant, mon encadrant propose un nouvel algorithme BCA\* qui permet de rechercher un meilleur compromis. Cet algorithme se base sur une énumération implicite de solutions non-dominées, à la manière de MOA\* [Stewart and White(1991)], en se focalisant, durant la recherche, sur l'obtention d'un compromis adapté aux préférences de l'utilisateur.

#### Principe de l'algorithme de BCA\*

#### Exemple de l'algorithme de BCA\*

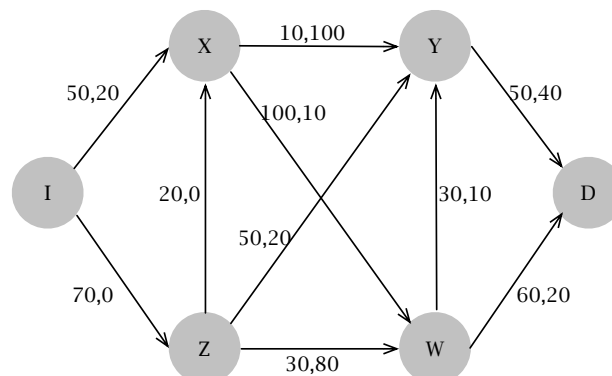


FIGURE 2.6 – Un graphe à traiter

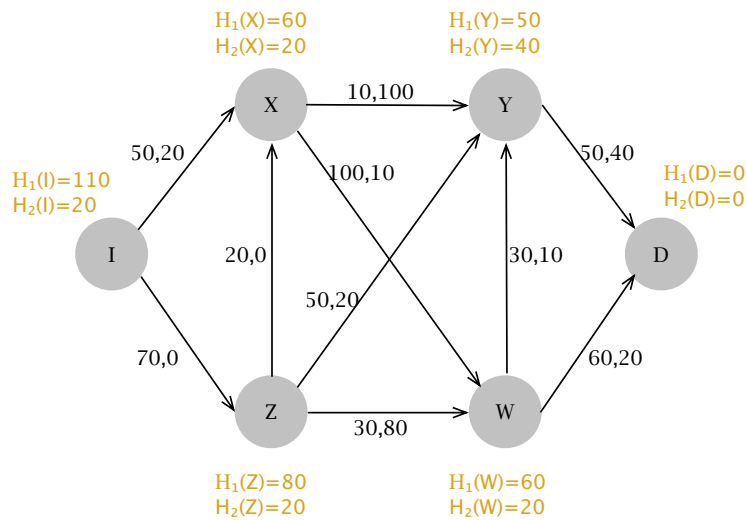


FIGURE 2.7 – Algorithme de BCA\*

Nous pouvons voir la figure ci-dessus, c'est un graphe contenant 6 sommets. Le noeud I est un sommet du départ et le noeud D est un sommet de la destination. Cet exemple considère que deux critères (distance et sécurité).

Nous pouvons d'abord utiliser DIJKSTRA ALGORITHM qui permet d'obtenir le plus court chemin en considérant respectivement le critère de la distance et le critère de la sécurité.

# Base de données

---

## 3.1 Présentation de la base de données

Pour trouver un itinéraire, il faut récupérer des données comme les coordonnées de départ, de destination, et des tronçons, etc. Pour l'instant, il existe déjà les 7 tables :

```
TABLE agglo_teleatlas
(
  gid serial NOT NULL,
  id bigint DEFAULT (-1),
  "name" text,
  oneway character varying(2),
  categorie1 smallint DEFAULT 0,
  categorie2 smallint DEFAULT 0,
  tranquillite1 smallint DEFAULT 0,
  tranquillite2 smallint DEFAULT 0,
  desserte smallint DEFAULT 0,
  paysage smallint DEFAULT 0,
  x double precision DEFAULT 0,
  y double precision DEFAULT 0,
  angle double precision DEFAULT 0,
  the_geom geometry,
  the_geom2 geometry,
  revetement smallint DEFAULT 0,
  f_elev smallint DEFAULT 0,
  t_elev smallint DEFAULT 0,
  loire_a_velo smallint DEFAULT 0,
  the_geom3 geometry,
  type_route smallint DEFAULT 0,
)
```

```
CREATE TABLE edge_ta
(
  gid serial NOT NULL,
  source integer NOT NULL,
  destination integer NOT NULL,
  id_tr integer NOT NULL,
  difficulte double precision,
  longueur double precision,
  categorie1 smallint DEFAULT 0,
  tranquillite smallint DEFAULT 0,
  desserte integer DEFAULT 0,
  sens_creation smallint DEFAULT 0,
)
```

```
CREATE TABLE edge_ta
(
  gid serial NOT NULL,
  source integer NOT NULL,
  destination integer NOT NULL,
  id_tr integer NOT NULL,
```

```
difficulte double precision,  
longueur double precision,  
categorie1 smallint DEFAULT 0,  
tranquillite smallint DEFAULT 0,  
desserte integer DEFAULT 0,  
sens_creation smallint DEFAULT 0,  
)
```

```
CREATE TABLE geometry_columns  
(  
  f_table_catalog character varying(256) NOT NULL,  
  f_table_schema character varying(256) NOT NULL,  
  f_table_name character varying(256) NOT NULL,  
  f_geometry_column character varying(256) NOT NULL,  
  coord_dimension integer NOT NULL,  
  srid integer NOT NULL,  
  "type" character varying(30) NOT NULL,  
)
```

```
CREATE TABLE loueur_reparateur  
(  
  gid serial NOT NULL,  
  nom character varying(100),  
  loueur smallint DEFAULT 0,  
  reparableur smallint DEFAULT 0,  
  vente smallint DEFAULT 0,  
  adresse character varying(250),  
  code_postal integer DEFAULT 0,  
  ville character varying(200),  
  site_internet character varying(250),  
  horaires character varying(250),  
  photo character varying(250),  
  the_geom geometry,  
  the_geom2 geometry,  
)
```

```
CREATE TABLE node_ta  
(  
  gid serial NOT NULL,  
  the_geom geometry,  
  the_geom2 geometry,  
  elev smallint DEFAULT 0,  
  altitude integer,  
)
```

```
CREATE TABLE poi  
(  
  gid serial NOT NULL,  
  categorie smallint DEFAULT 0,  
  souscategorie smallint DEFAULT 0,  
  nom character varying(100),  
  adresse character varying(250),  
  code_postal integer DEFAULT 0,  
  ville character varying(200),  
  site_internet character varying(250),  
  horaires character varying(250),  
  photo character varying(250),  
  the_geom geometry,  
  the_geom2 geometry,  
  description character varying(300),
```

)

## 3.2 Postgresql et Postgis

## 3.3 Réalisation de la base de données

# Réalisation

---



## 4.1 Test d'intégration de site Géovélo

Quand on a fini de modifier la zone de vos préférences, il faut tester son intégration. Ca signifie que les utilisateurs ne possèdent pas tous le même navigateur, certains possèdent Internet Explorer, d'autres Firefox. Le problème n'est pas non plus la multitude de navigateurs, mais également les différentes versions de chacun. Il est souvent fastidieux de tester un site sur les différents navigateurs existants. Selon la recherche, on a décidé d'utiliser des outils suivants :

- **BrowserShots** (<http://browsershots.org>)

C'est un service en ligne open-source et gratuit, ce site permet de tester automatiquement votre site Internet sur un peu plus de 50 navigateurs.

- **BrowserLab** (<http://browserlab.adobe.com>)

C'est une application développée par Adobe produisant des rendus selon différents navigateurs web pour un site donné. En comparant avec BrowserShots, pour l'instant elle est compatible seulement avec Internet Explorer, Firefox, Chrome et Safari. Mais BrowserLab permet d'analyser les différences entre deux navigateurs. Une petite subtilité permet de superposer les fenêtres entre 2 navigateurs.

# Bibliographie

---

- [Dijkstra(1959)] E.W. Dijkstra. A note on two problems in connection with graphs». Numerische Mathematik, (1) : 269-271, 1959.
- [Futtersack and Perny(2000)] M. Futtersack, P. Perny. BCA\*, une généralisation d'A\* pour la recherche de solutions de compromis dans des problèmes d'optimisation multi-objectifs. In Actes de la conférence RFIA'2000, vol. 3, pages 377-386, 2000.
- [Wikipedia] Wikipedia, Multiobjective optimization, [http://en.wikipedia.org/wiki/Multiobjective\\_optimization](http://en.wikipedia.org/wiki/Multiobjective_optimization).
- [Stewart and White(1991)] B. S. Stewart et C.C. White, Multiobjective A\*, Journal of the Association for Computing Machinery , Vol 38, pp. 775-814, 1991.
- [Sauvanet and Néron] G. Sauvanet, E. Néron. «Search for the best compromise solution on Multiobjective shortest path problem»
- [Zhang(2009)] Juan Zhang. «Calcul d'itinéraires touristiques pour vélo», Rapport de fin d'étude, École Polytechnique de l'Université de Tours, 64, avenue Jean Portalis, 37200 Tours, d2009.
- [Wikipedia] Wikipedia, Pareto efficiency, [http://en.wikipedia.org/wiki/Pareto\\_efficiency](http://en.wikipedia.org/wiki/Pareto_efficiency).



# Nouveaux critères pour du calcul d'itinéraire vélo

---

Département Informatique  
5<sup>e</sup> année  
2009 - 2010

Projet de Fin d'Etude

**Résumé** : Description en français

**Mots clefs** : Mots clés français

**Abstract** : Description en anglais

**Keywords** : Mots clés en anglais

---

## Encadrants

Gaël Sauvanet

[gael.sauvanet@univ-tours.fr](mailto:gael.sauvanet@univ-tours.fr)

Emmanuel Néron

[emmanuel.neron@univ-tours.fr](mailto:emmanuel.neron@univ-tours.fr)

## Etudiant

Ke SHANG

[ke.shang@etu.univ-tours.fr](mailto:ke.shang@etu.univ-tours.fr)

DI5 2009 - 2010

Université François-Rabelais, Tours