Master Informatique spécialité Informatique Avancée et Applications

Option Algorithmique et Recherche Opérationnelle

Faculté des sciences de Luminy

Projet de fin d'année :

Choix du meilleur itinéraire par un automobiliste

Projet encadré par :

Yann Vaxès

Réalisé par l'équipe composé de :

Nabil	**Bernine**
Kiril	**Gashteovski**
Maximilien	**Perrad**

Aix-Marseille Université *2012*.

Table des matières

Table des figures

Liste des tableaux

Introduction

Le projet de fin d'année que nous réalisons ici fait partie de notre formation de Master Informatique Avancée et Applications, option Algorithmique et Recherche Opérationnelle. Il vise à mettre en application les connaissances théoriques acquises sur un problème concret.

La problématique traitée est celle proposée pour le *Jeu Concours Mathématique 2011-2012*, sous le thème de *Choix du meilleur itinéraire par un automobiliste*, organisé conjointement par *la Fédération Française des Jeux Mathématiques* et *la Société de Calcul Mathématique SA* en partenariat avec le journal *Auto Plus*. Ils concernent la résolution d'un problème "de société", mais simplifié pour les besoins de la résolution.

Il s'agit d'un graphe sous forme d'un quadrillage de dix ligne horizontales et dix ligne verticales, sur laquelle se déplace un automobiliste. Il souhaite trouver les trajets les plus profitables pour lui selon différents critères (coût, durée, contribution des deux).
Une première étape consiste à calculer en respectant les limitations de vitesse, les trajets le plus rapides puis le moins coûteux. Dans une seconde étape, le conducteur souhaite faire le trajet en un temps inférieur au temps du trajet le plus rapide Il devra donc faire quelques dépassement de vitesses et par conséquent payer éventuellement des amandes si ses dépassements sont détectés par des radars fixes ou mobiles. L'objectif sera alors de minimiser l'espérance du coût du trajet.

La première étape a été résolue avec l'algorithme de Dijkstra en pondérant les arcs par des durées puis par des coûts. Dans la seconde partie un algorithme de marquage est utilisé pour résoudre une version à deux critères du problème. Une marque est une liste de labels non-dominés, chaque label est un couple représentant les deux critères coût et temps. Une discrétisation de la vitesse a permis de ne considérer que les valeurs les plus pertinentes.

1

Position du problème

Un automobiliste se déplace sur une carte sous forme d'un quadrillage de dix (10) ligne horizontales et dix (10) lignes verticales (FIG.1.1). Pour atteindre le point $B = (9, 9)$ en partant d'un point $A = (0, 0)$, l'automobiliste parcourt à chaque intersection un des segments horizontaux ou verticaux.

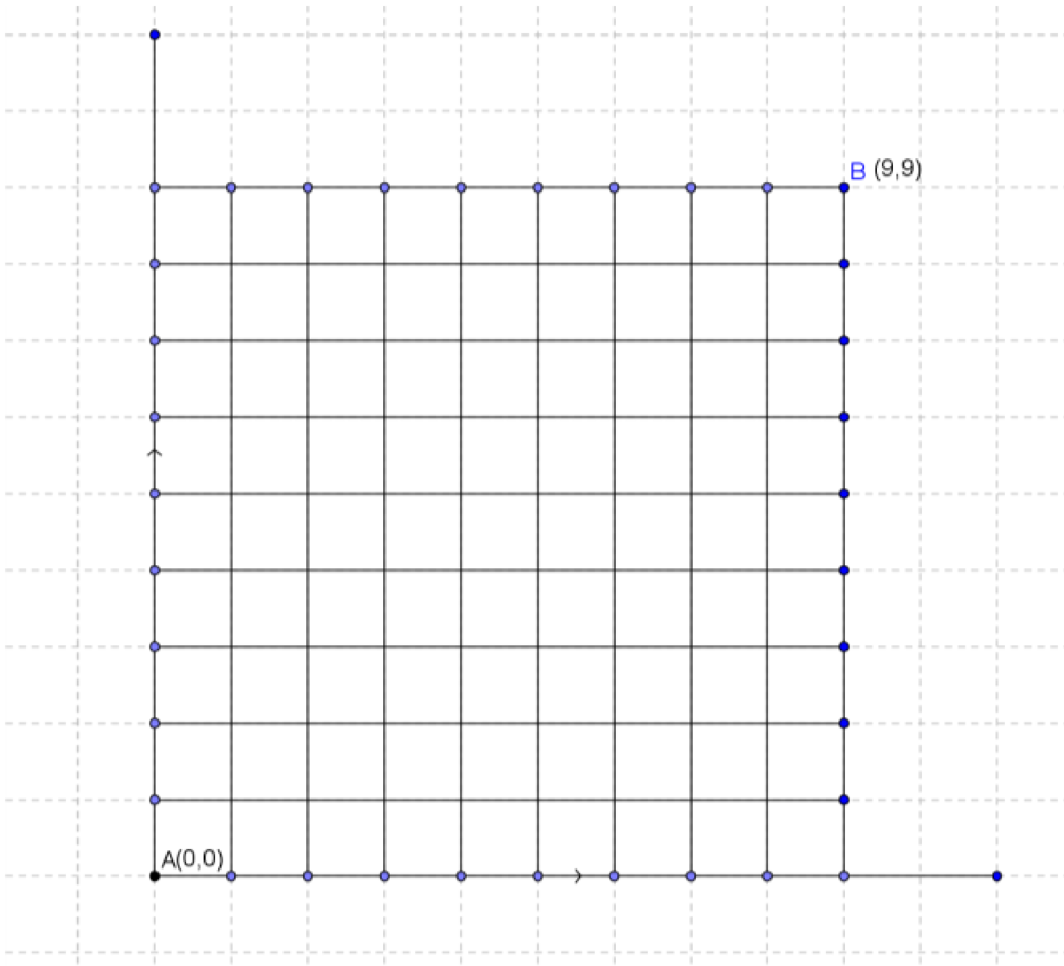


FIG. 1.1 – Grille du déplacement de l'automobiliste.

La longueur de chaque segment est de $d = 100Km$ et un segment est noté par un triplet composé des coordonnées du point de bas-gauche et une lettre V pour les segments verticaux et H pour les horizontaux. Ainsi, le segment reliant un point (x, y) et $(x+1, y)$ est noté (x, y, H) ; et le segment reliant (x, y) et $(x, y+1)$ est noté (x, y, V).

1.1 Limitations de vitesse

Le déplacement sur chaque segment est limité par une vitesse maximale autorisée, cette vitesse peut être $50,\ 90,\ 110$ ou $130[Km/h]$.
La limitation réglementaire de vitesse sur chaque segment de la grille concernée par cette étude est donnée par les deux tableaux (TAB1.1) et (TAB1.2). Les segments horizontaux sont caractérisés par leur extrémité gauche et les verticaux par leur extrémité basse.

	0	1	2	3	4	5	6	7	8
0	110	130	130	130	130	130	130	130	130
1	90	110	110	110	110	90	90	50	50
2	50	90	90	90	90	90	50	50	50
3	50	90	110	110	110	90	90	90	90
4	90	90	90	90	90	90	90	90	50
5	90	90	110	110	110	110	110	110	90
6	50	50	90	90	50	50	50	90	50
7	90	90	50	50	90	90	110	110	110
8	90	90	90	90	50	50	110	110	110
9	90	90	90	90	50	90	90	90	90

Tab. 1.1 – Limitation de vitesse sur les segments horizontaux

	0	1	2	3	4	5	6	7	8	9
0	90	110	90	90	50	50	50	90	90	130
1	50	90	90	90	50	110	90	90	90	130
2	50	90	90	90	90	110	90	50	50	130
3	50	90	110	90	50	90	90	90	50	130
4	90	90	110	90	90	90	90	90	50	50
5	90	50	110	90	50	90	50	50	50	50
6	50	50	110	50	50	50	50	50	110	50
7	90	90	110	50	50	90	90	90	110	90
8	90	90	110	90	90	90	110	90	90	110

Tab. 1.2 – Limitation de vitesse sur les segments verticaux

1.2 Les coûts

Le déplacement sur les segments engendre différents coûts. On considère pour cette étude les coûts suivants :

1. Les trajets où la vitesse maximale autorisée est de $130[Km/h]$ sont munis de *péages*, le coût de péage est fixé a 3 €/segment ;
2. Un coût C^1 lié au temps passé sur le segment (hôtel, restaurant,etc), calculé par la formule $C^1 = \alpha\ t$[€], où $\alpha = 5$[€/h] est un coût horaire ;

3. Un coût C^2 lié à la consommation du véhicule : $C^2 = \beta\ Conso$, où $\beta = 1.3$[€/l] est le coût du carburant au litre. La consommation $Conso$ est fonction de la vitesse et est donné par la formule : $Conso = a\ v + b$, où $a = 0.0625[lh/Km]$ et $b = 1.875[l]$.

1.3 Les radars

La mesure de dissuasion pour les automobilistes peu soucieux des limitations de vitesse est l'installation de radars. Un radar couvre la totalité du segment et l'automobiliste se déplace à vitesse constante sur le segment.

On trouve deux types de radars :

1. Les radars fixes installés sur des segments connus :

$$(0,8,V);(1,2,V);(1,4,V);(2,0,V);(3,5,V);(5,3,V);(5,8,V);(7,2,V);(9,2,V);(9,5,V);$$

$$(3,0,H);(8,1,H);(2,2,H);(5,3,H);(3,4,H);(0,5,H);(4,6,H);(4,7,H);(0,8,H);(7,9,H)\ ;$$

2. Les radars mobiles, au nombre de vingt 20. Par convention, chaque radar peut se trouver sur n'importe quel segment avec équiprobabilité et indépendamment des autres, ie. plusieurs radars mobiles peuvent être placé sur un même segment y compris un segment contenant un radar fixe.

Les règles de détection et de pénalisation des infractions sont définies par :

- Si le véhicule dépasse la vitesse limite de plus de 10%, il sera détecté avec la probabilité 0.7 et encourt une amende de 100[€] ;
- Si il la dépasse de plus de 50%, il sera détecté avec une probabilité 0.9 et encourt une amende de 200[€].

1.4 Problématique

1. Quels serait le trajet de coût minimal, si l'automobiliste respecte les limitations de vitesse ?
2. Déterminer le trajet de temps minimal (T^o), avec un déplacement respectant les limitations de vitesse.

3. Si un conducteur souhaite faire le trajet en un temps ne dépassant pas les **80%** du temps optimal T^o, quel trajet minimise l'espérance du coût total du trajet, et quelle est sa valeur ?

2

Le problème déterministe

2.1 Introduction

Dans un premier temps le problème traité concernera le cas où le véhicule respecte les limitations de vitesses. Dans ce cas les radars n'interviennent pas dans la modélisation du problème.

Il s'agit donc, de trouver les trajets optimaux pour le véhicule. Deux problèmes sont dérivés alors : le premier concerne la mise en évidence des trajets minimisant le coût total et le second se donne de trouver le trajet dont le temps total est minimal.

2.2 Modélisation du problème

Le quadrillage de déplacement du véhicule est vue comme un graphe où les segments de route de distance d ($d = 100Km$ dans notre cas) représentent les arêtes du graphe (non-orienté).

Étant sur un sommet de coordonnées (x, y) on peux se déplacer comme suit :

$$\begin{array}{ccccc} & & (x,y+1)_{y<M} & & \\ & & \uparrow & & \\ (x-1,y)_{x>0} & \leftarrow & (x,y) & \rightarrow & (x+1,y)_{x<N} \\ & & \downarrow & & \\ & & (x,y-1)_{y>0} & & \end{array} ;\text{Où} \begin{cases} N = & \textit{"Nombre de segments horizontaux"} ; \\ M = & \textit{"Nombre de segments verticaux".} \end{cases}$$

Ainsi, chaque sommet et relié au plus à quatre autres sommets, la notation des arcs sera donnée par : du point (x, y) à $(x+1, y)$ sera noté (x, y, H) ; (x, y) à $(x-1, y)$ par $(x-1, y, H)$; (x, y) à $(x, y+1)$ par (x, y, V) et l'arc (x, y) à $(x, y-1)$ par $(x, y-1, V)$.

Sur chaque arête du graphe, on définit un poids p selon le problème traité.

2.2.1 Trajets à temps minimal

Il s'agit de trouver un ou plusieurs trajets allants du point $A = (0, 0)$ au point $B = (9, 9)$ les plus rapides (ie. avec un temps total de parcours minimal) en respectant les limitations de vitesses sur chaque segment.

On considère alors le graphe donné par le quadrillage et le poids sur chaque arc est donné en calculant le temps passé sur chaque segment sachant que le véhicule se déplace à une vitesse égale à la limitation de vitesse réglementaire sur ce segment noté V_i^l.

Sur chaque arc i on définit le poids de l'arc p_i en fonction de la distance du segment d_i (dans notre cas est fixe pour tout les segments : $d_i = 100,\ \forall i$) et de la limitation de vitesse V_i^l sur chaque segment i (données par la réglementation).
Le poids sur l'arc i sera alors, le temps de parcours du segment de distance $d_i = d$ à la vitesse constante V_i^l :

$$p_i = t_i = \frac{d_i}{V_i^l}$$

On obtient un graphe non-orienté valué, où tout les poids sont non-négatifs. Le problème posé se résume alors à trouver le plus court chemin (en terme de temps) entre le point de départ $A = (0, 0)$ et l'arrivée $B = (9, 9)$.

2.2.2 Trajets à coût minimal

On considère le même graphe précèdent et on calcule les poids des arc par une fonction coût qui servira de base pour retrouver le chemin le moins coûteux.

Le coût C_i^I sur chaque arc i est donné par deux coûts, l'un lié au temps passé sur le segment, noté C_i^1 et l'autre à la consommation du véhicule C_i^2 .

$C_i^1 = \alpha \ t_i$, où $\alpha = 5$[€/h] et $t_i = \frac{d_i}{v_i}$

$$C_i^1 = \alpha \ \frac{d_i}{v_i} \tag{2.1}$$

C_i^2 est un coût calculé sur chaque arc i en fonction de la consommation de carburant $Conso(v_i)$ par :

$$C_i^2 = \beta \ Conso(v_i).$$

Où α est le prix du carburant au litre, pour ce cas pratique : $\beta = 1.3$ $[€/l]$.

La consommation du véhicule $Conso(v_i)$ est une fonction linéaire sur la vitesse du véhicule sur ce segment v_i avec deux paramètres a et b (dans notre cas $a = 0.0625$ et $b = 1.875$) par :$Conso(v_i) = av_i + b$.

$$C_i^2 = \beta \ (av_i + b). \tag{2.2}$$

Un autre C^p coût vient s'ajouter aux segments dont la limitation de vitesse est de $V_i^l = 130[Km/h]$:

$$C_i^p = \begin{cases} 3[€] & \text{si} V_i^l = 130[Km/h] \\ 0, & \text{sinon.} \end{cases} \quad \text{; pour tout} i.$$

Finalement, le coût C_i^I est donné par la fonction :

$$\begin{aligned} C_i^I(v_i) &= C_i^p + \alpha \ \frac{d_i}{v_i} + \beta \ (a \ v_i + b); & (2.3) \\ &= C_i^p + 5\frac{100}{v_i} + 1.3(0.0625 v_i + 1.875). & (2.4) \end{aligned}$$

Le coût C_i^I sur chaque segment i est alors une fonction de la vitesse du véhicule dont la courbe est de la forme suivante (FIG.2.1) :

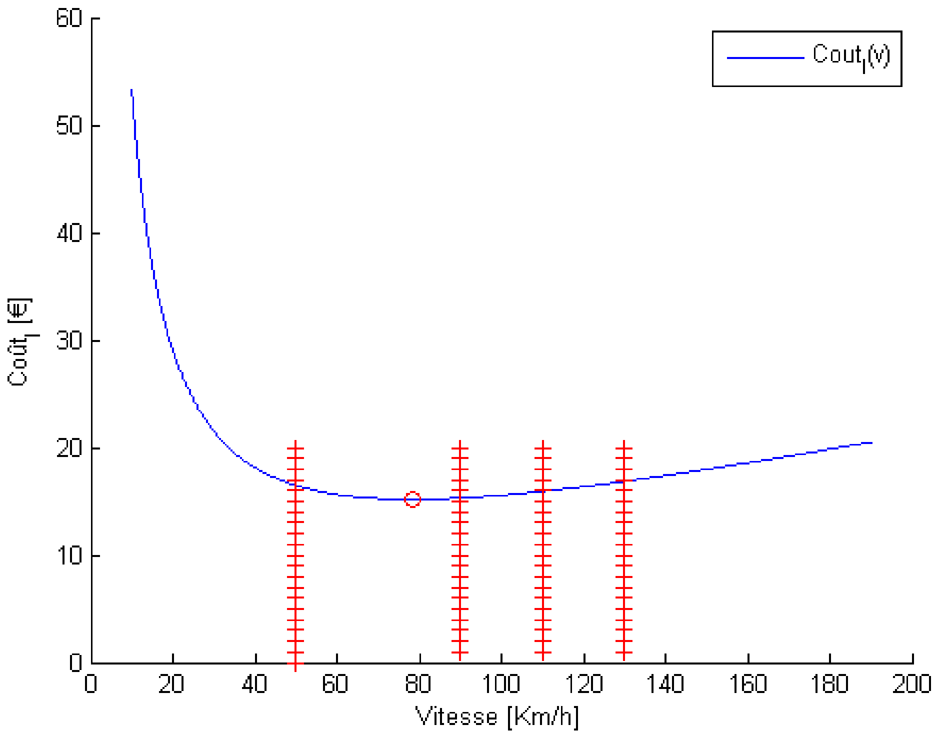


FIG. 2.1 – Le coût sur chaque segment en fonction de la vitesse.

La fonction coût est strictement décroissante sur $]0, \tilde{v}[$ et est strictement croissante sur $]\tilde{v}, +\infty[$,

$$\text{où : } \tilde{v} = \sqrt{\frac{\alpha \ d_i}{\beta \ a}}; \tag{2.5}$$

$$= \sqrt{\frac{500}{0.08125}} \simeq 78.45[Km/h]. \tag{2.6}$$

Le problème traité étant l'identification des chemins de coût minimum, la pondération des arc du graphe sera alors donnée par le coût minimal possible sur ce segment. Il s'agit donc pour l'automobiliste de trouver la vitesse optimale V_i^o de déplacement sur chaque arc i qui minimise le coût sur cet arc.

Nous utiliserons, en suite ces vitesses comme des poids p_i sur les arcs pour l'optimisation par l'algorithme de Dijkstra et trouver ainsi le chemin le moins coûteux de $A = (0,0)$ à $B = (9,9)$, avec $p_i = C_i^I$.

Le calcul des vitesses V_i^o, $i \in \Sigma$ (Σ est l'ensemble de tous les sommets), se fait en fonction de la limitation de vitesse V_i^l sur le segment i, selon l'algorithme suivant :

$$V_i^o = \begin{cases} \tilde{v}, & \text{si : } \tilde{v} \leq V_i^l, \text{ (voir Eq.2.5) ;} \\ V_i^l, & \text{sinon.} \end{cases} \qquad (2.7)$$

Une fois la vitesse optimale V_i^o sur chaque segment est connue, on calcule le coût minimal sur chaque arc i par l'équation (Eq.2.3), avec $p_i = C_i^I(V_i^o)$.

2.3 Méthodes de résolution

Une méthode efficace de calcul des plus courts chemins lorsque tous les poids non-négatifs est l'algorithme de Dijkstra. Il permet de calculer le plus court chemin entre un sommet particulier et tous les autres.
Dans notre cas cet algorithme définira le plus court chemin entre les points de départ et d'arrivée.

2.3.1 L'algorithme de Dijkstra

Le principe de l'algorithme de Dijkstra est de progresser de proche en proche sur la liste des sommets du graphe et construit une arborescence, sous-graphe du graphe initial où tout chemin de l'arborescence entre la racine et tout autre sommet du graphe est le plus court chemin entre ces deux sommets.

Phase 1 : Mise en place :

- Soit I le sommet initial ;
- On désigne par Σ l'ensemble dans lequel on met les sommets au fur et à mesure de leur marquage définitif ;
- A chaque sommet S, on associe le couple $(dist(S), p(S))$ où $dist(S)$ désigne la distance (provisoire ou definitive) de I a S, et $p(S)$ le prédécesseur de S ;

Phase 2 : Initialisation :

- Attribuer au sommet I, le couple $(0, I)$;
- Attribuer à chaque sommet adjacent à I, le couple (poids de l'arc le reliant à I, I) ;
- Attribuer aux autres sommets, le couple $(+\infty, ?)$;

Phase 3 : Fonctionnement :

Tant que tous les sommets ne sont pas dans Σ :

– Choisir parmi les sommets non placés dans Σ, celui dont la distance provisoire est minimale : appelons-le S ;
– Mettre S dans Σ ;
– Pour chacun des sommets Y_i qui lui sont adjacents et qui ne sont pas dans Σ :
  – Calculer $s = dist(S) + p(S, Y_i)$ ($p(S, Y_i)$ poids de l'arc $[S, Y_i]$) ;
  – Si s est inférieur à la distance provisoire de Y_i, attribuer à Y_i le couple (s, S) (ainsi, $dist(Y_i) \hookleftarrow \min\{dist(S) + p(S, Y_i); dist(Y_i)\}$) ;

Phase 4 : Conclusion :
- La longueur du plus court chemin de I à n'importe quel autre sommet S est alors $dist(S)$;
- Et le chemin de poids minimum se lit en partant de S et allant vers les prédécesseurs successifs.

2.3.2 Implémentation et résultats

L'algorithme est implémenté sous le langage de programmation Java dans l'environnement de développement Eclipse.
L'application lit les données à partir du fichier de données, avec une définition de sommet par un couple (x, y) donnant sa position sur la grille. Une fonction de mapping est alors définie pour numéroter les sommets de "0" à "(nombre de sommets -1)" : $I = x + ym$, m étant le nombre de ligne verticales de la grille.
Puis un calcul des poids p_i du graphe est effectué selon les définitions données dans les sections précédentes.

En applicant l'algorithme de Dijkstra (section 2.3.1), on aboutit à l'arborescence des plus court chemin de la racine $A = (0, 0)$ vers tout les sommets du graphe, en particulier vers le sommet $B = (9, 9)$. Il suffit donc de récupérer le coût sur le sommet B et de reconstruire le chemin en commençant par le sommet B et en remontant par la relation de prédécesseur successivement jusqu'à la source A.

Les résultats obtenus pour les deux problèmes sont :

Chemin plus rapide :

Segment	Vitesse	Temps	Coût	Radars	Péage
(0, 0, H)	110 km/h	0,91 h	15,92 €	0	Non
(1, 0, H)	130 km/h	0,77 h	19,85 €	0	Oui
(2, 0, H)	130 km/h	0,77 h	19,85 €	0	Oui
(3, 0, V)	90 km/h	1,11 h	15,31 €	0	Non
(3, 1, H)	110 km/h	0,91 h	15,92 €	0	Non
(4, 1, H)	110 km/h	0,91 h	15,92 €	0	Non
(5, 1, V)	110 km/h	0,91 h	15,92 €	0	Non
(5, 2, V)	110 km/h	0,91 h	15,92 €	0	Non
(5, 3, V)	90 km/h	1,11 h	15,31 €	1	Non
(5, 4, V)	90 km/h	1,11 h	15,31 €	0	Non
(5, 5, H)	110 km/h	0,91 h	15,92 €	0	Non
(6, 5, H)	110 km/h	0,91 h	15,92 €	0	Non
(7, 5, H)	110 km/h	0,91 h	15,92 €	0	Non
(8, 5, V)	50 km/h	2,00 h	16,50 €	0	Non
(8, 6, V)	110 km/h	0,91 h	15,92 €	0	Non
(8, 7, V)	110 km/h	0,91 h	15,92 €	0	Non
(8, 8, H)	110 km/h	0,91 h	15,92 €	0	Non
(9, 8, V)	110 km/h	0,91 h	15,92 €	0	Non

TAB. 2.1 – Solution du plus court chemin en temps.

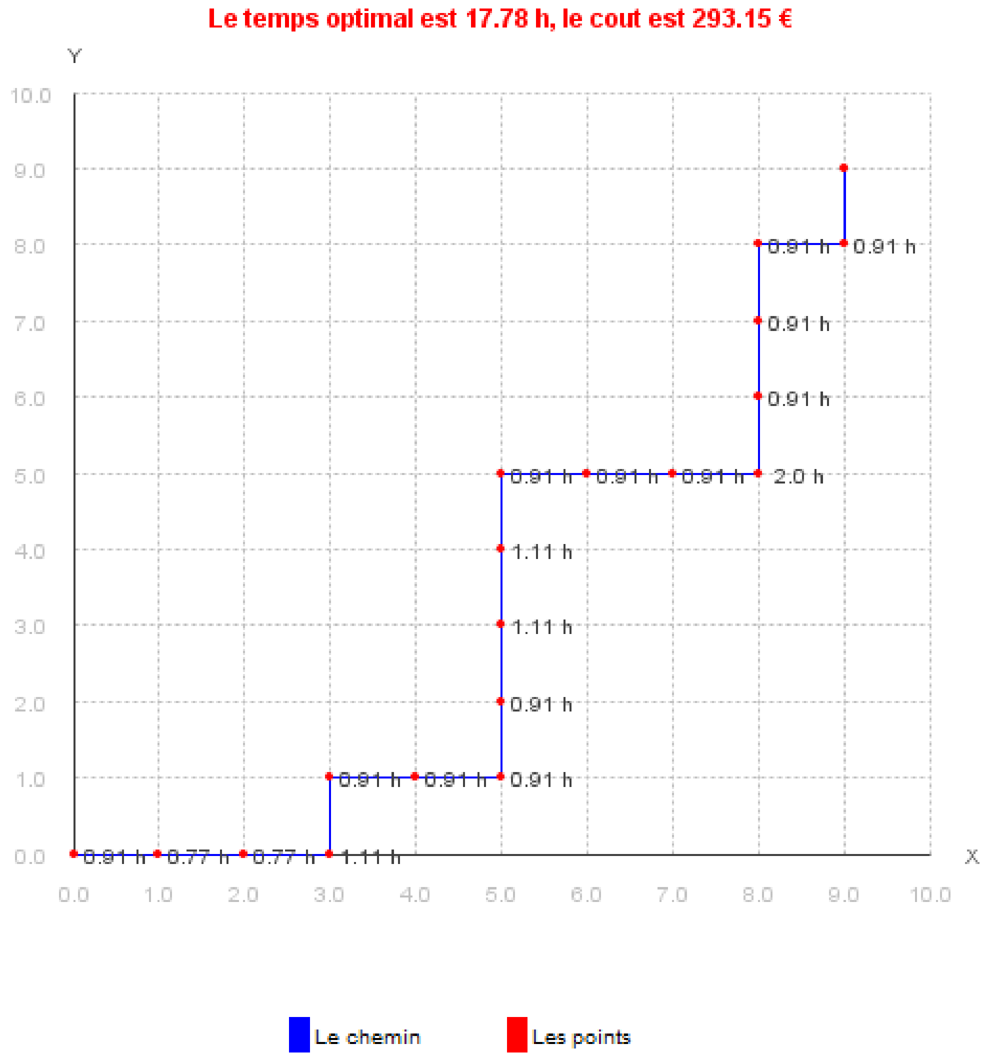


FIG. 2.2 – Trajet le plus rapide

$$T^o = 17.78[\mathrm{h}] = 17h\ 46mn\ 48s. \quad \text{Le coût sur ce trajet est de } 293.15[\text{ €}] \tag{2.8}$$

Chemin moins coûteux :

Segment	V_{Lim}	Vitesse	Temps	Cout	Radars	Péage
(0, 0, H)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(1, 0, V)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(1, 1, H)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(2, 1, H)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(3, 1, H)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(4, 1, H)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(5, 1, H)	90 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(6, 1, V)	90 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(6, 2, V)	90 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(6, 3, H)	90 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(7, 3, V)	90 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(7, 4, V)	90 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(7, 5, H)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(8, 5, V)	50 km/h	50,00 km/h	2,00 h	16,50 €	0	Non
(8, 6, V)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(8, 7, H)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(9, 7, V)	90 km/h	78,45 km/h	1,27 h	15,19 €	0	Non
(9, 8, V)	110 km/h	78,45 km/h	1,27 h	15,19 €	0	Non

TAB. 2.2 – Solution du plus court chemin en coûts.

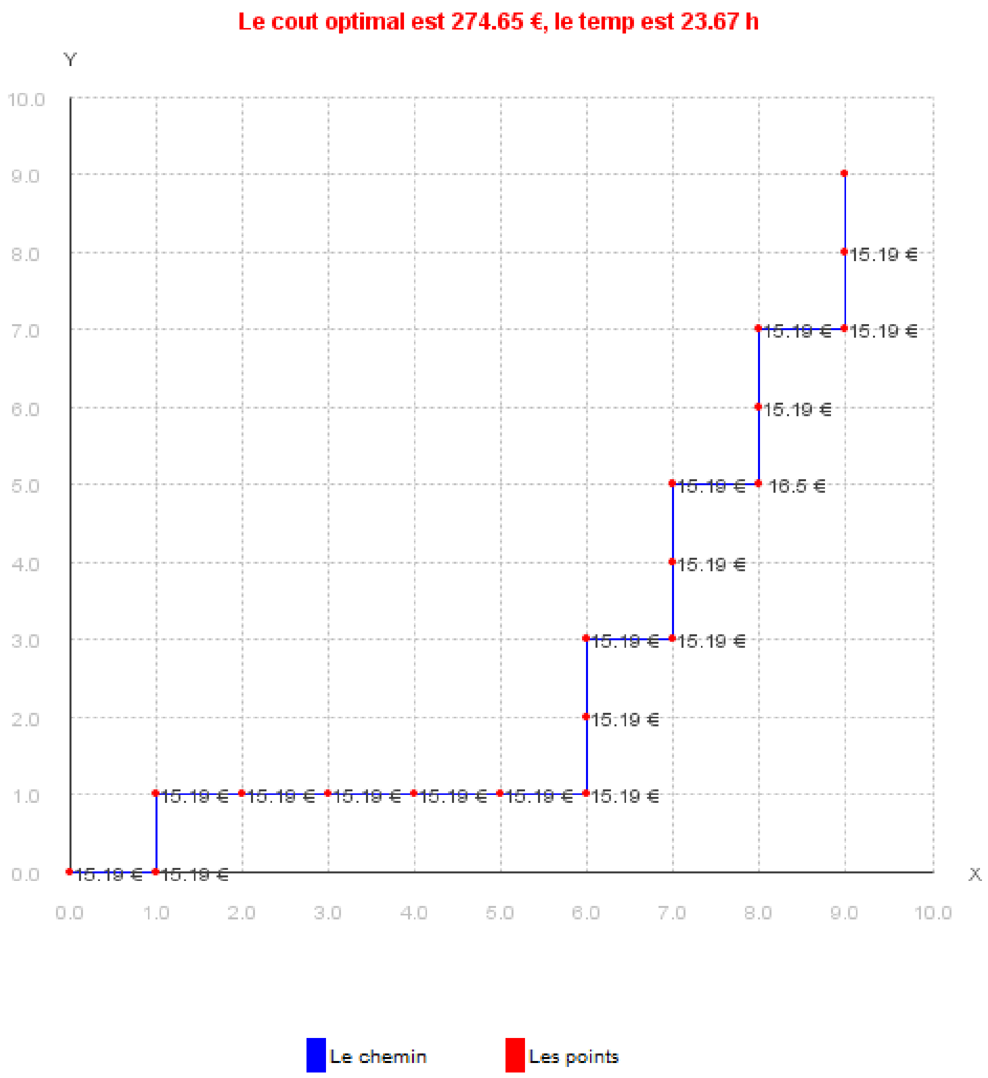


FIG. 2.3 – Trajet le moins coûteux

$$C^o = 274.64[€]. \quad \text{Le temps sur ce trajet est de } 23.67[h] = 23h\ 40mn\ 12s. \tag{2.9}$$

2.3.3 Commentaires sur les résultats

Les chemins obtenus sont optimaux car l'algorithme de Dijkstra est un algorithme exacte [1].

3
Le problème probabiliste

3.1 Introduction

Le respect des limitations de vitesse n'est pas toujours de rigueur chez certains automobilistes.
Un conducteur pressé veut faire le trajet de $A = (0, 0)$ à $B = (9, 9)$ en un temps $t \leq R\ T^o$, où $R = 80\%$ est un ratio représentant la proportion de temps que le conducteur souhaite passer en route et T^o est la durée minimale calculée à l'étape précédente (voir section 2.3.2).

En introduisant les coût liés aux pénalités éventuelles de dépassement de vitesses susceptibles d'être détectés par les différents types de radars, le problème est de trouver un chemin et de définir la vitesse sur chaque segment de manière à minimiser l'espérance du coût total du trajet sans dépasser le temps fixé par le conducteur.

3.2 L'espérance du nombre de radars sur un trajet

Les radars mobiles sont distribués sur tous les segment avec équiprobabilité et indépendamment les uns des autres. Sur un même segment, on peut trouver plusieurs radars mobiles y compris sur les segments munis d'un radar fixe.
On veut quantifier l'espérance du nombre de radars mobiles sur chaque segment.

Loi de probabilité de la distribution des radars

Pour un segment donné i soit l'expérience aléatoire suivante :
$X_i^j =$ " Apparition du j^e radar mobile " ;
La réponse est $"oui"$ ou $"non"$ $(0,1)$ avec une probabilité $\wp = \frac{1}{180}$.

$$X_i^j \rightsquigarrow Bernoulli(\wp).$$

On répète cette expérience pour chacun des 20 radars indépendamment les uns des autres. On aura sur ce segment, $\mathscr{X}_i$ le nombre de radars définit par :

$$\mathscr{X}_i = \sum_{j=1}^{20} X_i^j.$$

$$\mathscr{X}_i \rightsquigarrow \mathscr{B}(20, \wp).$$

Espérance du nombre de radars

$$\begin{aligned} E(\mathscr{X}_i) &= E(\sum_{j=1}^{20} X_i^j); \\ &= \sum_{j=1}^{20} E(X_i^j); \\ &= 20\ \wp; \\ &= \frac{20}{180} = \frac{1}{9}. \end{aligned}$$

3.3 Coûts incluant les radars

Le coût C_i^{II} sur chaque segment i est vu comme la somme d'un coût $C_i^I(v_i)$ (ref. Eq.2.3, section 2.2.2), lié à la consommation du véhicule, le temps passé sur le segment, un coût éventuel de péage et un coût $C_i^R(v_i)$ qui est le coût lié aux pénalités d'excès de vitesse détectés par les radars.

Le coût $C_i^R(v_i)$ s'écrit :

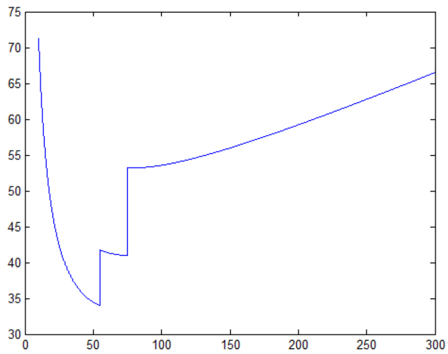
$$C_i^R(v_i) = \left[\mathbb{1}_{(R_{fix})} + E(\mathcal{X}_i)\right] penalite(v_i)$$

Où :

$$\mathbb{1}_{(R_{fix})} = \begin{cases} 1, & \text{si il y a un radar fixe sur le segment ;} \\ 0, & \text{sinon.} \end{cases} \tag{3.1}$$

$$penalite(v_i) = \begin{cases} 0, & \text{si} \quad v_i \leq 1.1V_i^l; \\ 100 * 0.7, & \text{si} \quad 1.1V_i^l < v_i \leq 1.5V_i^l; \\ 200 * 0.9, & \text{si} \quad 1.5V_i^l < v_i. \end{cases} \tag{3.2}$$

La fonction du coût C_i^{II} est discontinue car $penalite(v_i)$ est une fonction en escalier de v_i. L'allure des courbes de coût tracées en utilisant MATLAB se présentent comme suit :



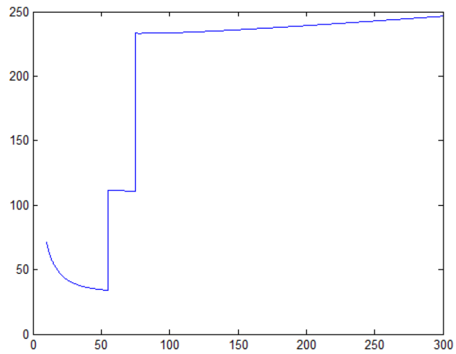


FIG. 3.1 – Le coût C_i^{II} en fonction de la vitesse sur les segments limités à 50Km/h avec et sans radar fixe.

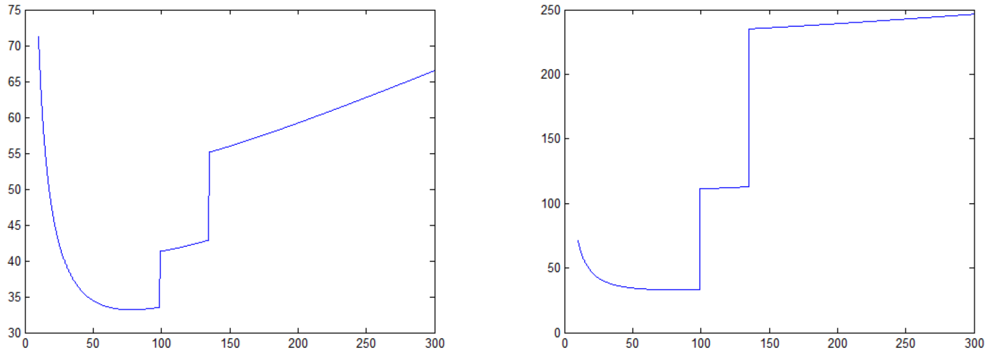


FIG. 3.2 – Le coût C_i^{II} en fonction de la vitesse sur les segments limités à 90Km/h avec et sans radar fixe.

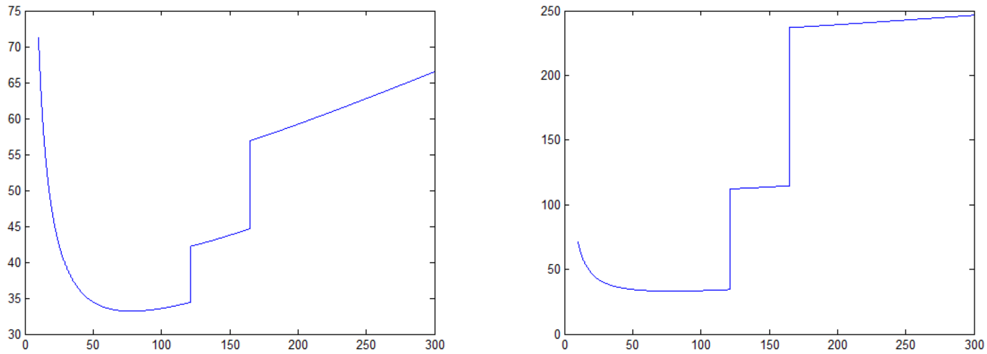


FIG. 3.3 – Le coût C_i^{II} en fonction de la vitesse sur les segments limités à 110Km/h avec et sans radar fixe.

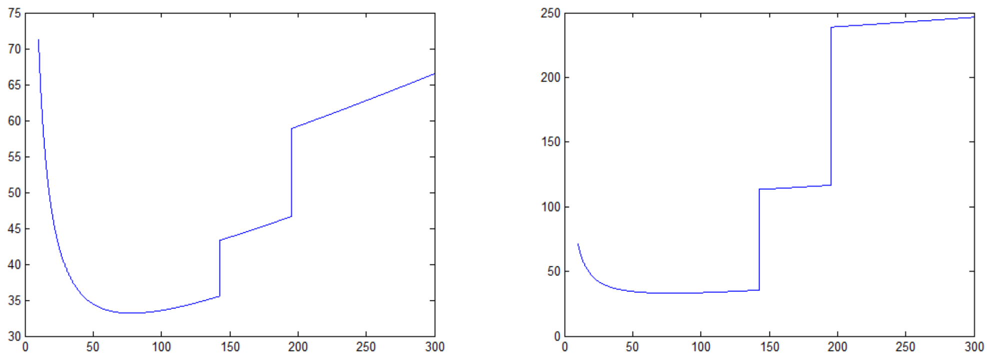


FIG. 3.4 – Le coût C_i^{II} en fonction de la vitesse sur les segments limités à 130Km/h avec et sans radar fixe.

3.4 Méthode de résolution

3.4.1 Discrétisation de la vitesse

Le passage d'un sommet à un sommet adjacent se fait à travers un segment sur lequel l'automobiliste choisit une vitesse de déplacement propre à ce même segment sur un ensemble continue ($]0, V_{\max}]$, où $V_{\max}$ est la vitesse maximale que peut atteindre le véhicule).

La forme particulière de la courbe du coût total en fonction de la vitesse suggère que certains choix de vitesses sont à privilégier. Dans le cas générale, la courbe du coût est en trois paliers :

Palier I : Le premier palier ($]0, 1.1V_i^l]$), représente une très forte pente décroissante au début, pour des vitesses vitesses. Ceci est inintéressant pour les deux critères : coût et temps. Un point intéressant à considéré est l'autre extrémité de ce palier qui correspond à la vitesse $\bar{v}_1 = 1.1V_i^l$.

Palier II : L'extrémité initiale du palier n'est pa fermé, donc, on ne considère que le point $\bar{v}_2 = 1.5V_i^l$.

Palier III : Le troisième palier et un ensemble ouvert. La première extrémité est traitée dans le palier précédent. La limite physique du véhicule sur chaque segment qui va représenter l'extrémité finale du palier est prise par hypothèse à deux fois la valeur de la limitation réglementaire de la vitesse, $\bar{v}_3 = 2V_i^l$.

Point particulier : Le point $\tilde{v}$, définit dans la relation 2.5 (section 2.2.2) représente un minimum local de la fonction coût, il se retrouver sur l'un des trois palier en fonction de la valeur de la vitesse maximale autorisée V_i^l. On le traite donc comme une valeur pour la discrétisation de la vitesse : $\bar{v}_4 = \tilde{v}$.

Résumé : Dans la version discrétiseé du problème, les valeurs possibles pour la vitesse sont :

$$\bar{v} = \begin{cases} \bar{v}_1 & = & 1.1V_i^l \\ \bar{v}_2 & = & 1.5V_i^l \\ \bar{v}_3 & = & 2V_i^l \\ \bar{v}_4 & = & \tilde{v} \end{cases} \tag{3.3}$$

3.4.2 Construction du graphe

On considère, pour la résolution de ce problème, un graphe constitué de l'ensemble des sommets du quadrillage représentant la carte des déplacements du véhicule.
Chaque segment de route sera représenté par quatre arcs différents, pondérés par les six composantes du vecteur $\bar{v}$. Ceci signifie que l'automobiliste peut se déplacer d'un sommet à un autre sommet adjacent en choisissant l'un des six arcs, qui est concrètement une des six vitesses de déplacement *possibles*.

Remarquons qu'un choix de la vitesse sur un segment induit un choix du temps passé et du coût total sur ce segment.

3.4.3 Algorithme de résolution

À ce stade, nous disposons d'un graphe valué, chaque arc i_j (i pour le segment et j pour le j^e arc parmi les six) est pondéré par un couple (temps, coût), $\bar{p}_i{}^j = (t_i, c_i)^j$.

Ceci a conduit à la proposition d'un algorithme de marquage basé sur une liste de labels sur chaque sommet :

1. <u>Définition d'un label :</u>
   Soit $\mathfrak{L}_u$ une liste de labels pour le sommet u. $l = (t_i, c_i, u_i) \in \mathfrak{L}_u$ signifie qu'il existe un chemin de durée t_i, de coût c_i arrivant de u_i entre la source (S) et le sommet u.
2. <u>Définition de la dominance entre labels :</u>
   Soient $\dot{l} = (\dot{t}, \dot{c}, \dot{u}), \ddot{l} = (\ddot{t}, \ddot{c}, \ddot{u}) \in \mathfrak{L}_u$ deux labels sur le sommet u.
   – On dit que le label $\dot{l}$ est dominé par le label $\ddot{l}$ <u>si et seulement si</u> $\ddot{t} \geq \dot{t}$ <u>et</u> $\ddot{c} \geq \dot{c}$.
3. <u>Étendre un label :</u>

   – Parcourir la liste des voisins du sommet u ;
   – Soit v un voisin de u, $\bar{l} = (t_i + t_{\overrightarrow{uv}}, c_i + c_{\overrightarrow{uv}}, u)$ est un label possible pour le sommet v ;
   – Si l' n'est dominé par aucun label de $\mathfrak{L}_v$, alors : ajouter l' à $\mathfrak{L}_v$ et continuer avec le voisin suivant.
4. <u>Algorithme :</u>

Initialisation : $\mathfrak{L}_u = \emptyset, \ \forall u \in V - \{S\}$; $\mathfrak{L}_S = \{(0,0,S)\}$;

Boucle : Tant qu'il existe un label $l \in \mathfrak{L}_u$ non examiné, *étendre* l, ie :

Pour chaque voisin v de u faire : Si $\bar{l} = (t + t_{\overrightarrow{uv}}, c_i + c_{\overrightarrow{uv}}, u)$ n'est pas dominé par un label $\tilde{l} \in \mathfrak{L}_v$, alors ajouter $\bar{l}$ à la liste des labels de $\mathfrak{L}_v$.

Fin Renvoyer le label de la destination T qui a le coût le plus petit parmi tous ceux pour lesquels $t \leq 0.8T^o$.

3.5 Résultats de l'implémentation

L'implémentation de cet algorithme aboutit à des labels discrets représentants des points de l'espace Pareto-optimal qui minimisent les deux critères coût et temps.
Comme le conducteur a la volonté d'aller à un temps $t \leq 0.8T^o$ il suffit de prendre le point dont le temps est le plus proche de t ainsi donné.

La chemin obtenu est :

Segment	V^{lim}	Vitesse	Temps	Coût	Radars	Péage
(0, 0, H)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(1, 0, V)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(1, 1, H)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(2, 1, V)	90 km/h	135,00 km/h	0,74 h	24,89 €	0	Non
(2, 2, V)	90 km/h	135,00 km/h	0,74 h	24,89 €	0	Non
(2, 3, V)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(2, 4, V)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(2, 5, H)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(3, 5, H)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(4, 5, H)	110 km/h	165,00 km/h	0,61 h	26,65 €	0	Non
(5, 5, H)	110 km/h	165,00 km/h	0,61 h	26,65 €	0	Non
(6, 5, H)	110 km/h	165,00 km/h	0,61 h	26,65 €	0	Non
(7, 5, H)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(8, 5, V)	50 km/h	100,00 km/h	1,00 h	35,56 €	0	Non
(8, 6, V)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(8, 7, V)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(8, 8, H)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non
(9, 8, V)	110 km/h	121,00 km/h	0,83 h	16,40 €	0	Non

TAB. 3.1 – Chemin le moins coûteux avec un temps inférieur à $0.8T^o$.

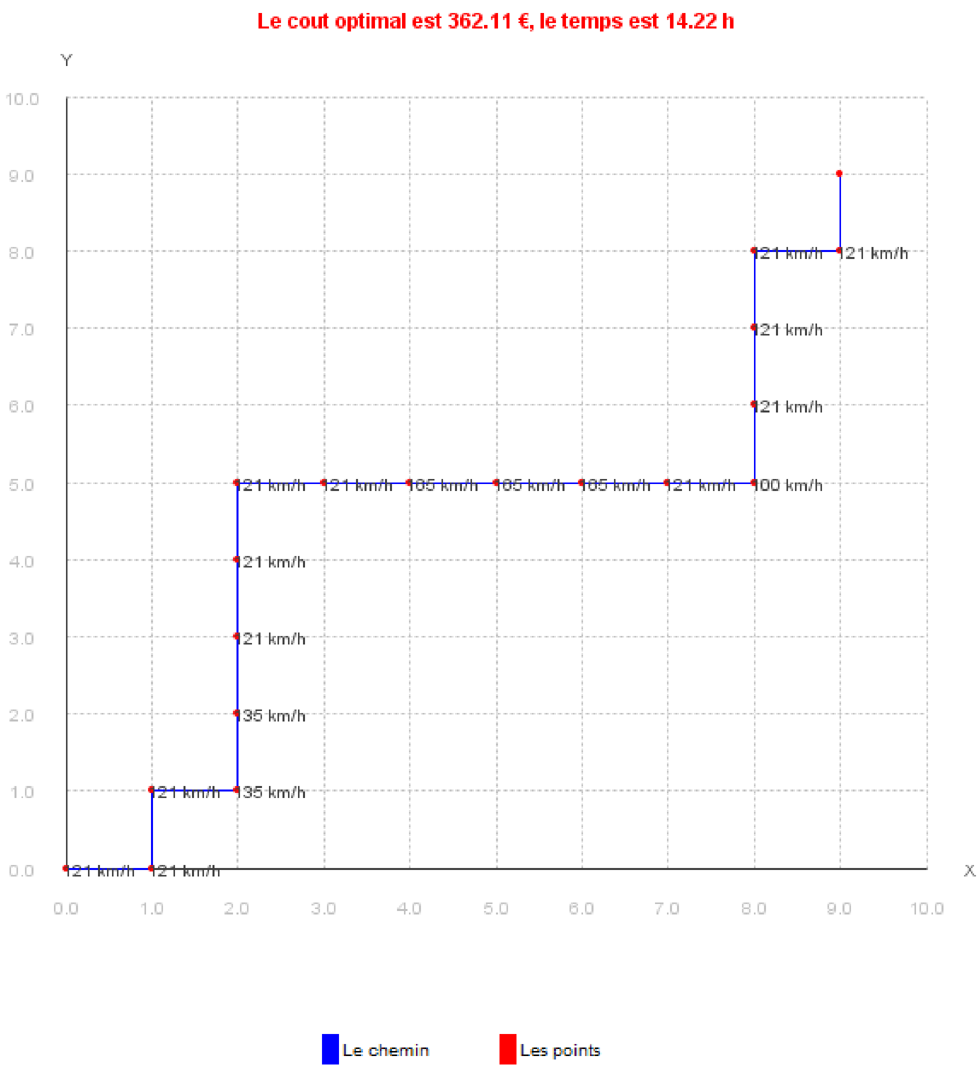


FIG. 3.5 – Trajet le moins coûteux en moins de $0.8T^o$

$$(T^{opt}, C^{opt}) = (14.21, 362.10)[h, €] \tag{3.4}$$

Nous remarquons que cet algorithme peut être amélioré en essayant de se rapprocher de la valeur $0.8T^o = 14.22$ $[h]$.

4
Présentation du logiciel

4.1 Lecture et visualisation des données

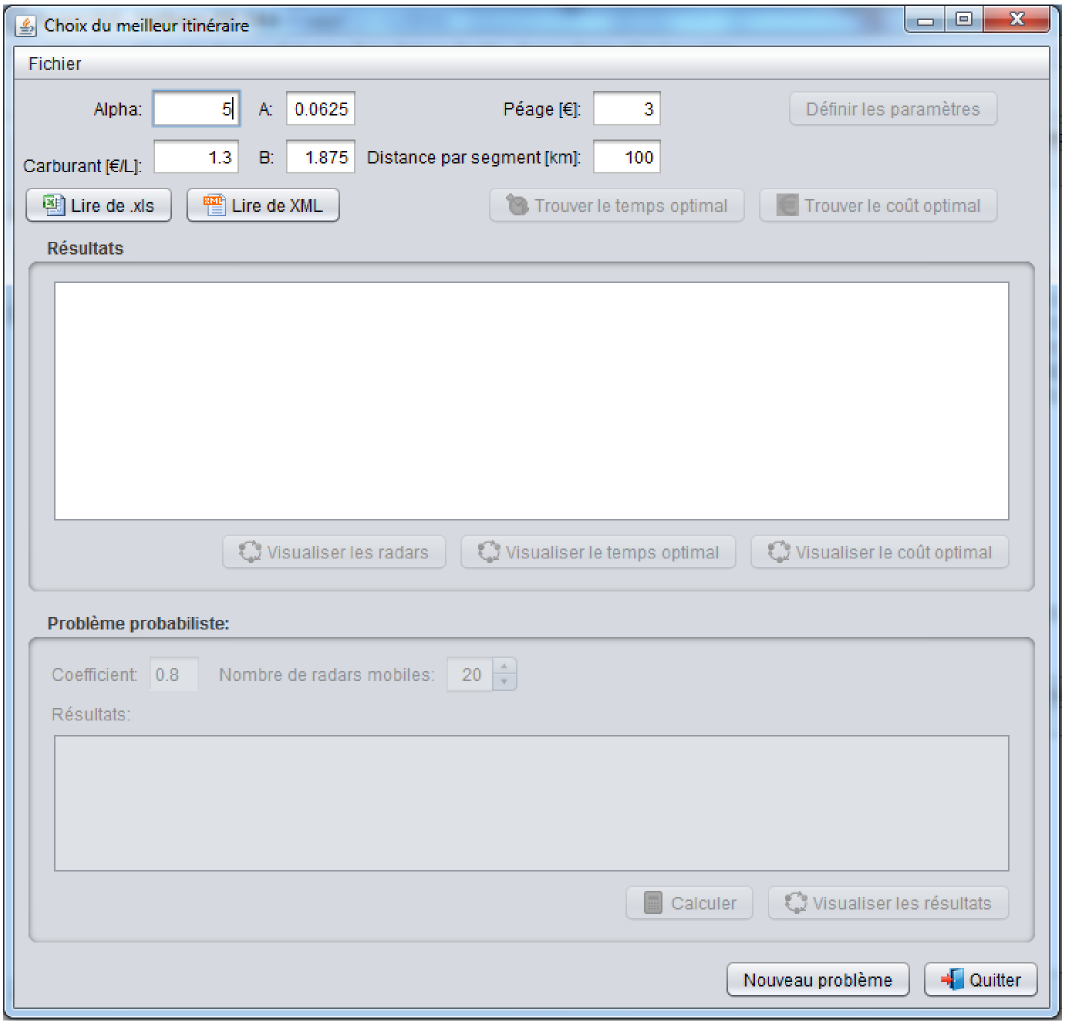


FIG. 4.1 – Interface de l'application.

En lançant notre application, la fenêtre (FIG.4.1) s'ouvre. Par l'intermédiaire de cette l'interface graphique on peut lire un fichier contenant les données concernant la limitation de vitesse et le nombre de radar sur chaque segment. Ce fichier peut soit être de format excel, soit de format XML.

Si le fichier est au format XML, alors il doit respecter la *dtd* suivante (TAB.4.1) :

```
<!ELEMENT   grille (taille,horizontals,verticals)>
<!ELEMENT taille EMPTY>
<!ATTLIST taille abscisse CDATA #REQUIRED>
<!ATTLIST taille ordonnee CDATA #REQUIRED>
<!ELEMENT horizontals     (horizontal*)>
<!ELEMENT horizontal       (segment*)>
<!ATTLIST horizontal num CDATA #REQUIRED>
<!ELEMENT verticals       (vertical*)>
<!ELEMENT vertical       (segment*)>
<!ATTLIST vertical num CDATA #REQUIRED>
<!ELEMENT segment   EMPTY>
<!ATTLIST segment num CDATA #REQUIRED>
<!ATTLIST segment limVitesse CDATA #REQUIRED>
<!ATTLIST segment nbRadar CDATA #REQUIRED>
```

TAB. 4.1 – DTD du fichier XML

Le fichier de données XML doit avoir l'élément *grille* comme racine. Cette élément à comme fils :

- L'élément *taille* qui a la taille en abscisse et en ordonnée de la grille;
- L'élément *horizontals* qui a comme fils un ensemble d'élément *horizontal* représentant chacun une ligne horizontal de la grille. Chaque élément *horizontal* a un ensemble d'élément *segment* comme fils, ayant chacun la limitation de vitesse et le nombre de radar sur ce segment comme attribut. Si un élément *horizontal* a comme attribut, $num = i$, et comme fils un élément *segment* ayant comme attribut $num = j$, alors ce segment est de coordonnées (j, i, H) ;
- L'élément *verticals* qui a comme fils un ensemble d'élément *vertical* représentant chacun une ligne vertical de la grille. Chaque élément *vertical* a un ensemble d'élément *segment* comme fils, ayant chacun la limitation de vitesse et le nombre de radar sur ce segment comme attribut. Si un élément vertical a comme attribut $num = i$ et comme fils un élément *segment* ayant comme attribut $num = j$, alors ce segment est de coordonnées (i, j, V).

Une fois les données du fichier lues, on peut visualiser dans le champs Résultats les matrices des limitations de vitesse et du nombre de radar fixes sur chaque segment :

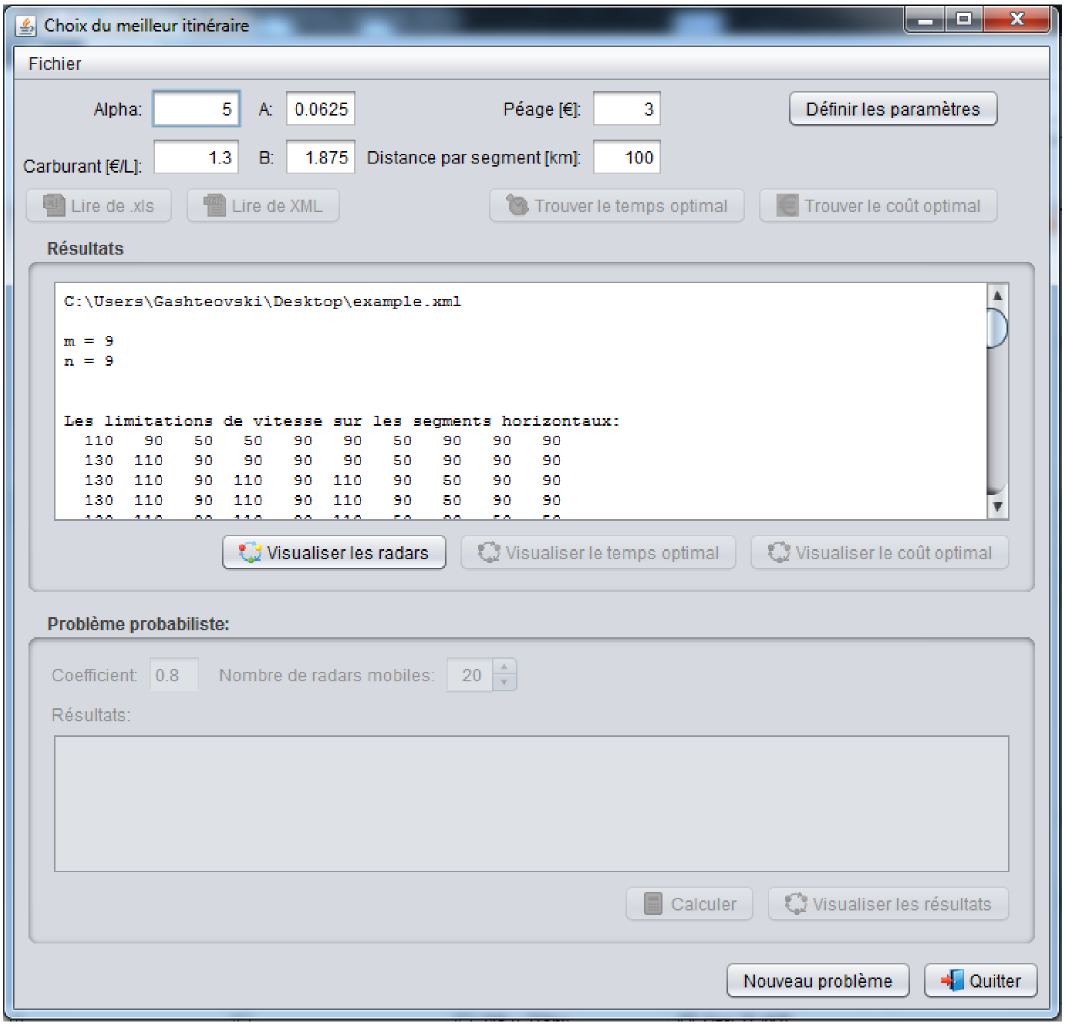


FIG. 4.2 – Visualisation des données.

On peut également visualiser les radars fixes en cliquant sur Visualiser les radars. Un segment est surligné en rouge s'il possède un radar :

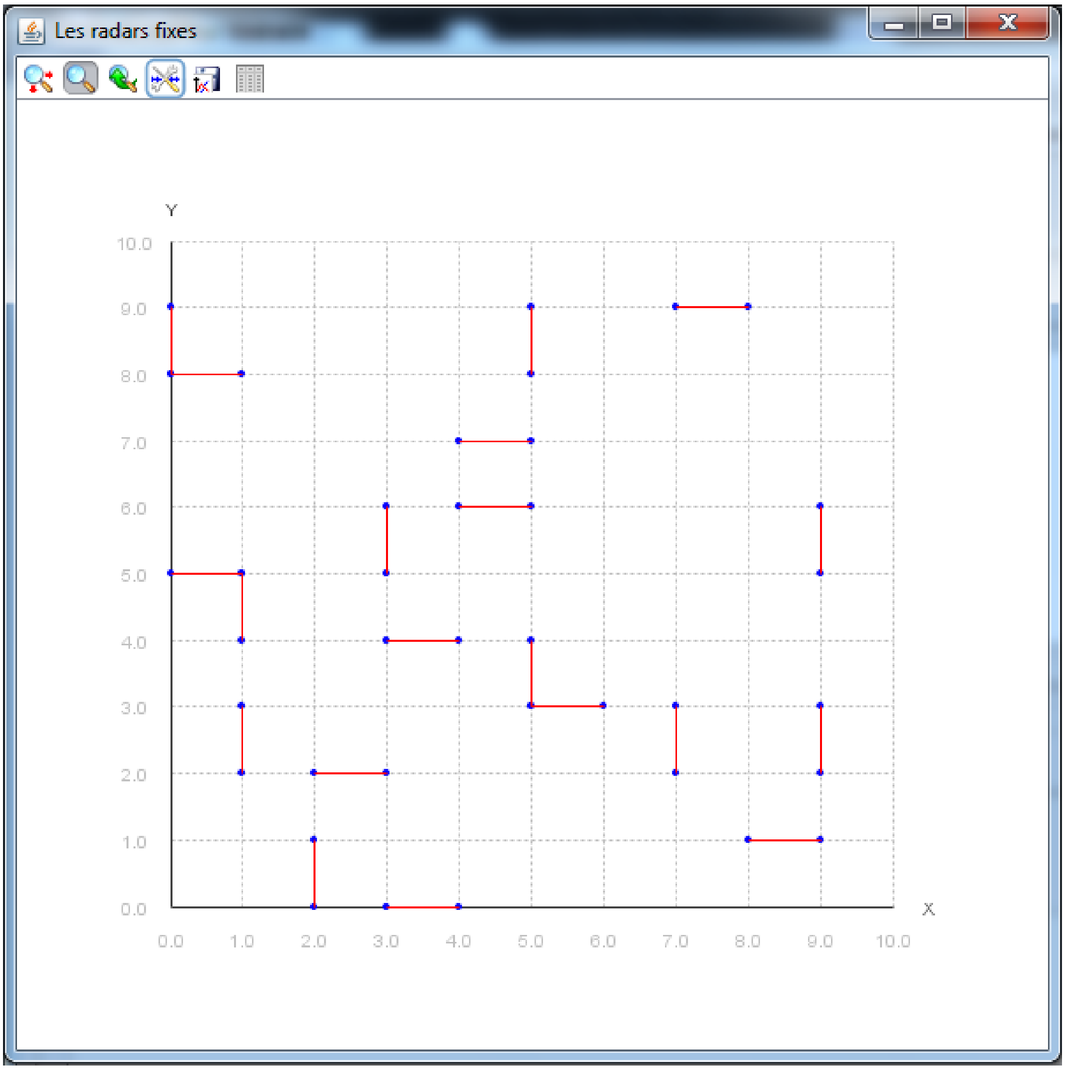


FIG. 4.3 – Visualisation des radars fixes.

En cliquant sur l'icône set scales de la barre d'outils on obtient cette fenêtre :

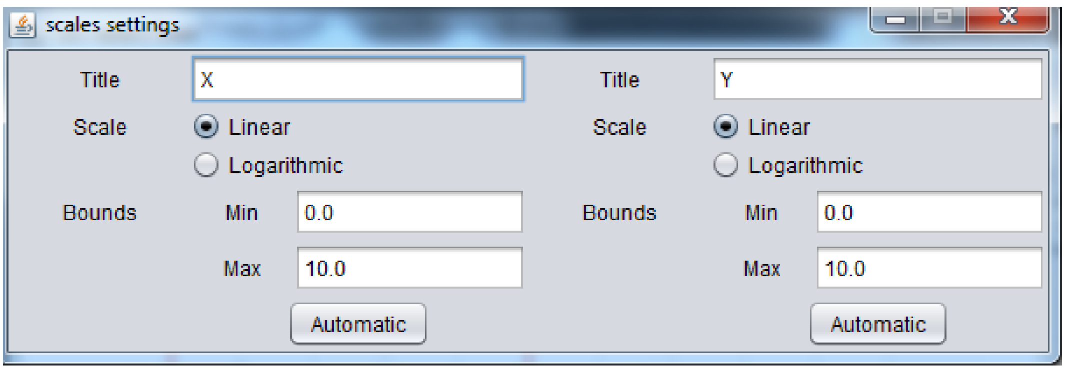


FIG. 4.4 – Paramétrage de l'affichage.

On a deux champs Title qui permettent de nommer l'abscisse et l'ordonnée sur la fenêtre de visualisation des radars fixes, ainsi que deux champs Min et deux champs Max qui définissent les coordonnées minimum et maximum de la grille.

En cliquant sur l'icône get Datas de la barre d'outils on a la fenêtre suivante :

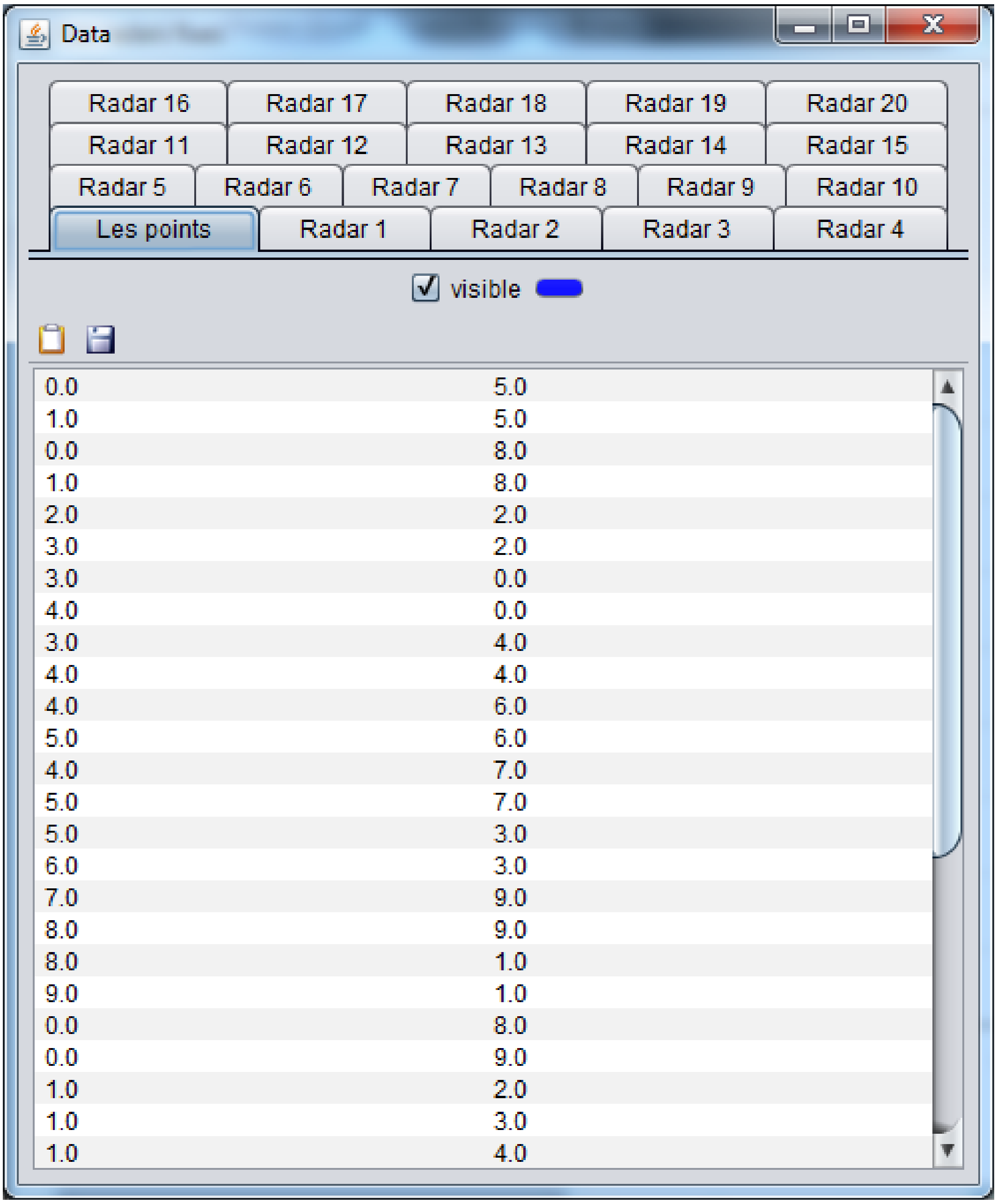


FIG. 4.5 – Paramètres de l'affichage des radars.

En allant dans l'onglet Les Points, si on décoche la case "visible", alors, dans la fenêtre de visualisation des radars fixes, les points adjacents aux segment possédant un radar fixe n'apparaissent plus. Si on va dans un onglet "radar" et que l'on décoche la case visible, alors les segments possédant un radar fixe n'apparaissent plus en rouge.

4.2 Paramètres du problème

On peut maintenant, via l'interface graphique, rentrer les différents paramètres du problème, à savoir :

- Les coefficients a et b de la fonction de coût liée à la consommation du véhicule.
- Le coefficient c de la fonction de coût C1 = ct liée au temps passé sur un segment.
- Le prix d'un litre d'essence.
- Le prix du péage pour les segments dont la limitation de vitesse est de 130Km/h.
- La distance en Km sur chaque segment.
- Le nombre de radar mobile.

On valide ces paramètres en cliquant sur Définissez les paramètres.

Ces différentes données sont lues et utilisée par le programme pour déterminer aux choix :

- Le trajet de temps minimal si l'automobiliste respecte les limitations de vitesse.
- Le trajet de coût minimal si l'automobiliste respecte les limitations de vitesse.
- Le chemin minimisant l'espérance du coût du trajet et dont le temps de trajet est inférieur ou égal à $0{,}8 * T^0$, ou T^0 est le temps minimal obtenu en respectant les limitations de vitesse.

4.3 Temps optimal

Maintenant que les données du fichier (XML ou XLS) et les paramètres rentré via l'interface graphique sont lues par le programme, on peut en cliquant sur trouver le temps optimale déterminer et afficher le trajet de temps minimal si l'automobiliste respecte les limitations de vitesse :

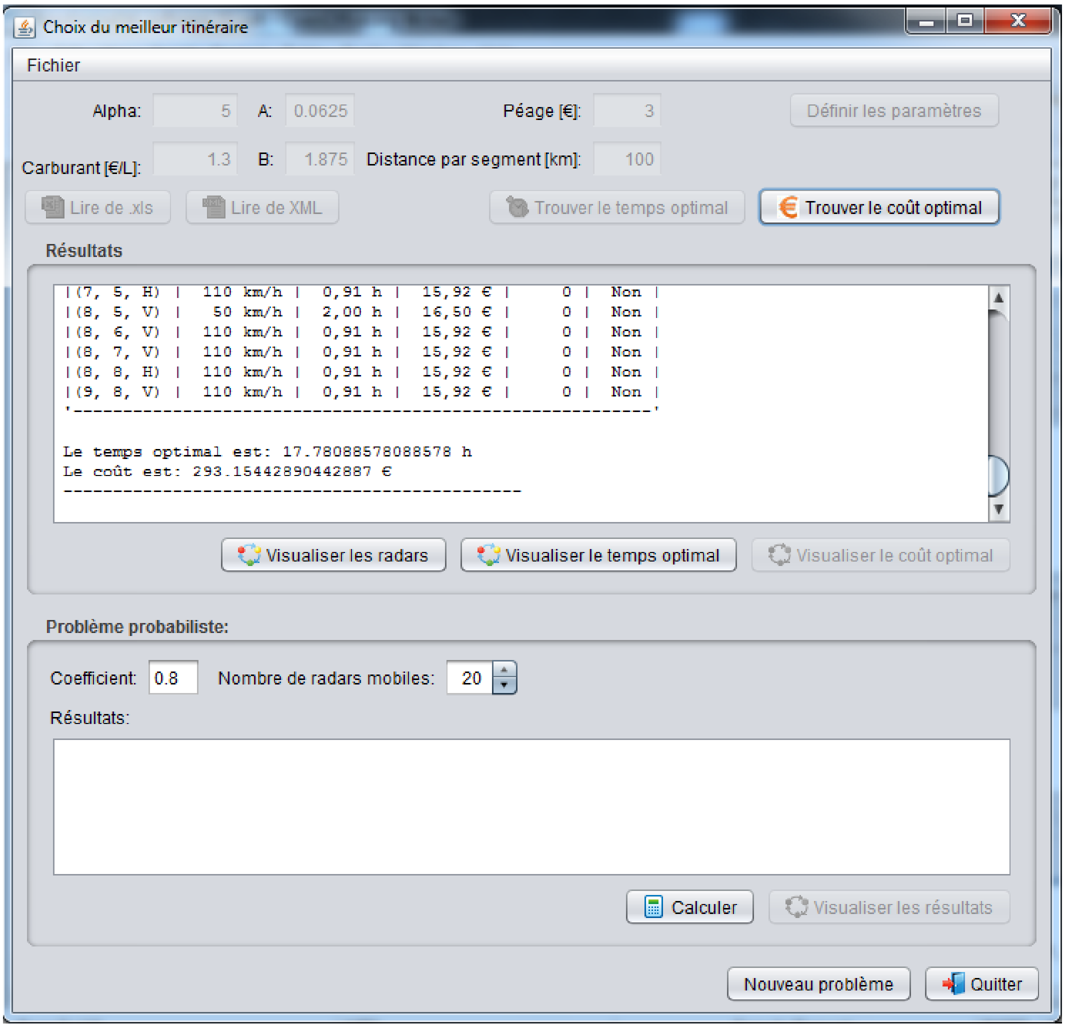


FIG. 4.6 – Calcul du chemin le plus rapide.

En cliquant sur l'icône Visualisez le temps optimale, On obtient une visualisation du chemin obtenu avec le temps optimale :

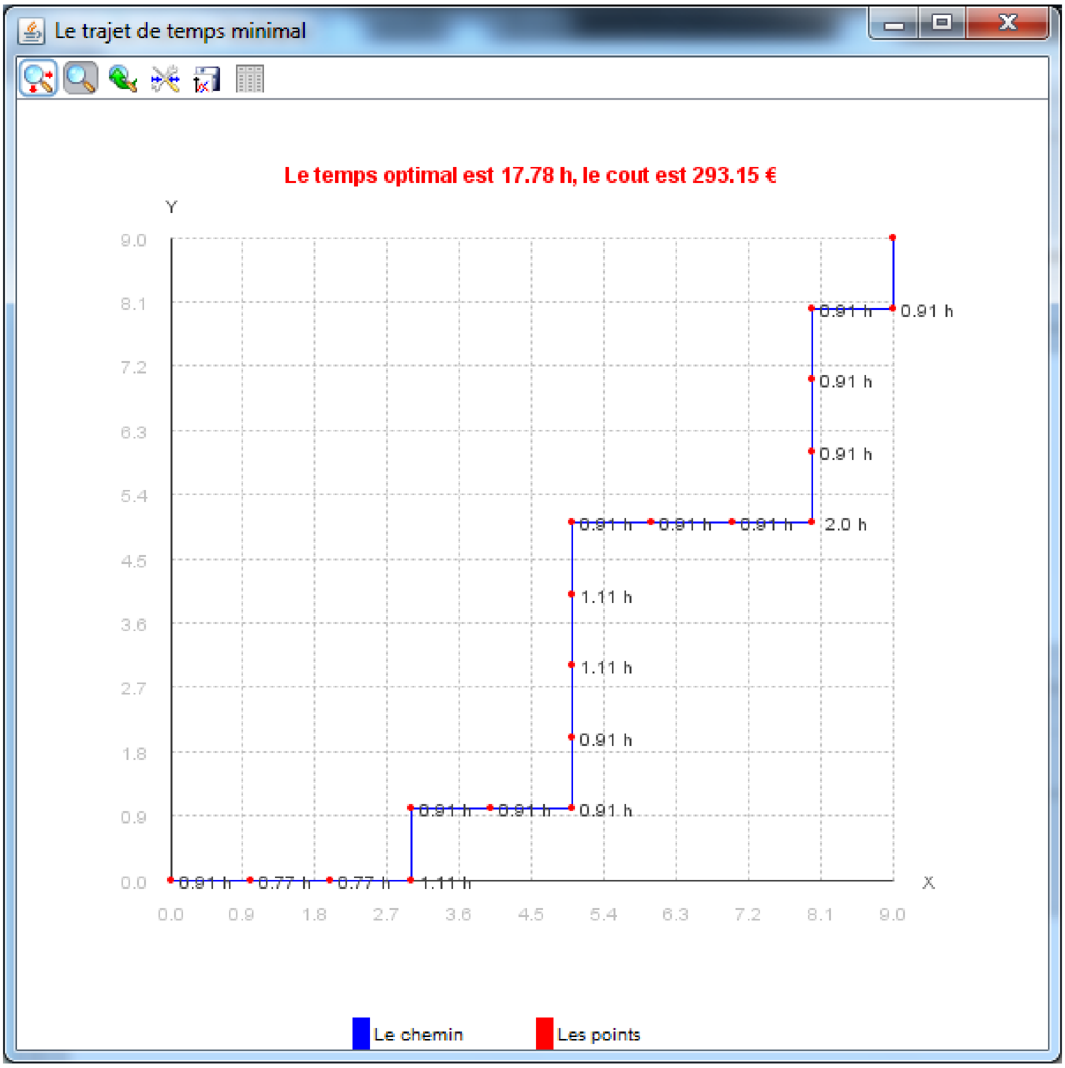


FIG. 4.7 – Visualisation du chemin le plus rapide.

4.4 Coût optimal

On peut aussi déterminer le trajet de coût minimal si l'automobiliste respecte les limitations de vitesse en cliquant sur l'icône Trouver le coût optimal :

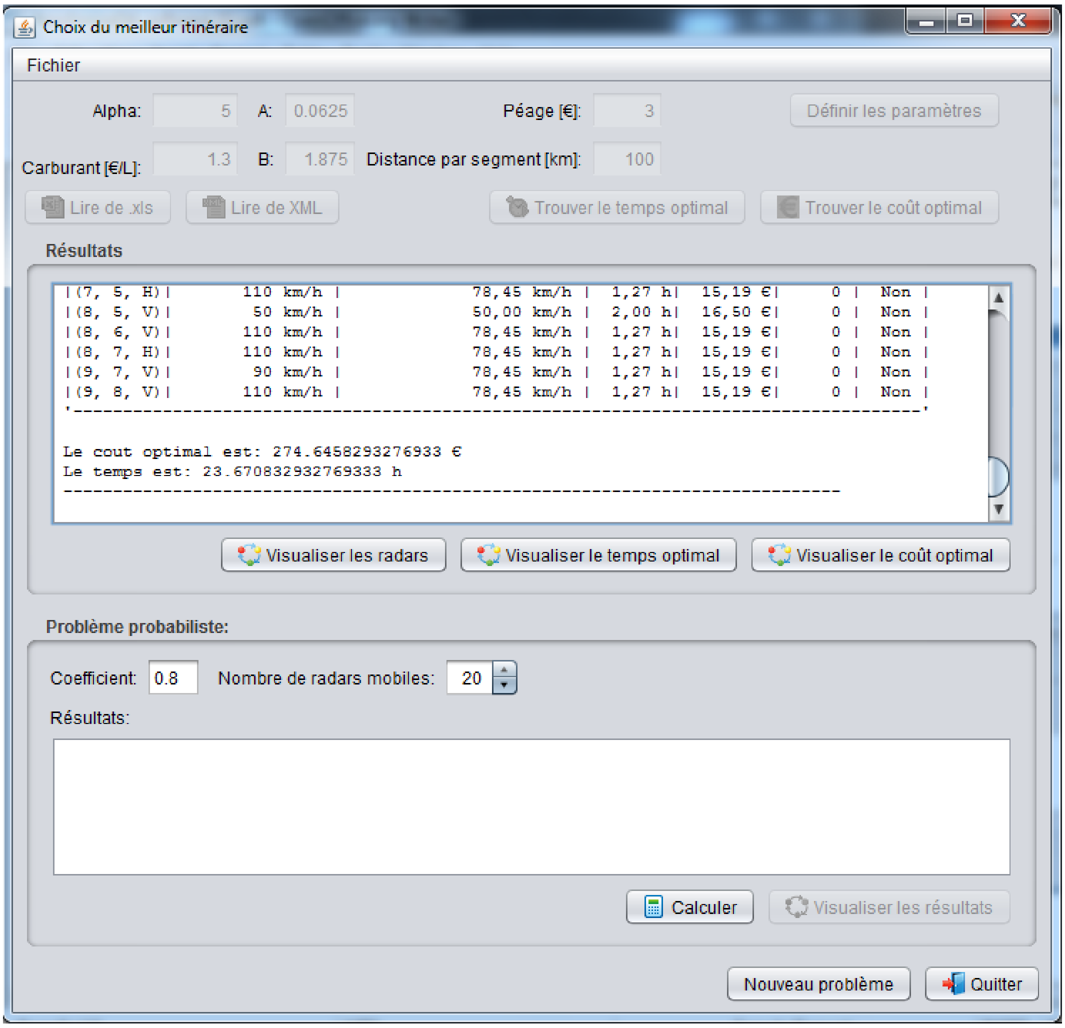


FIG. 4.8 – Calcul du chemin le moins coûteux.

En cliquant sur Visualisez le coût optimal, on obtient une visualisation du chemin obtenu avec le coût optimale :

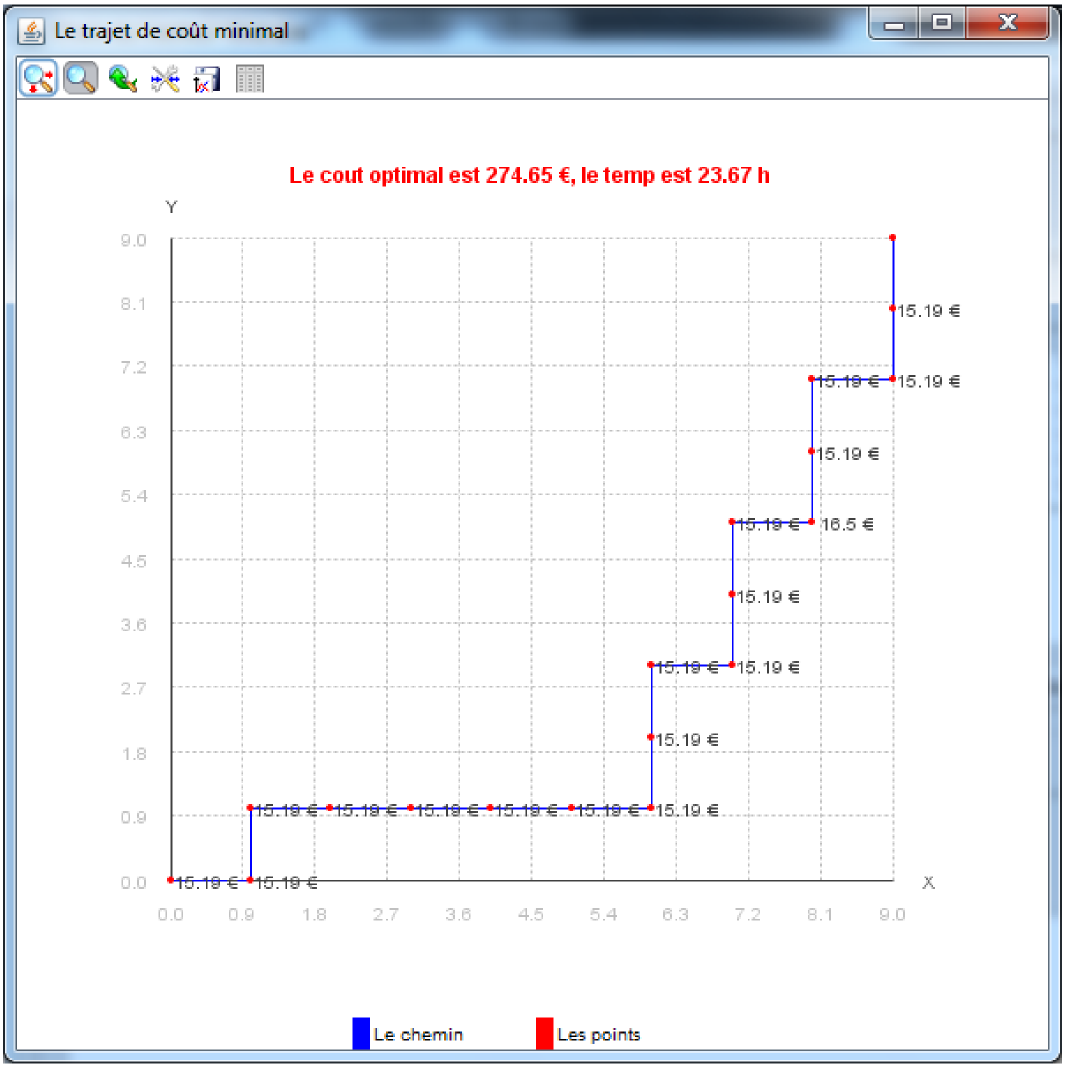


FIG. 4.9 – Visualisation du chemin le moins coûteux.

4.5 Problème probabiliste

Pour déterminer un chemin dont l'espérance du coût est minimum parmi les chemins dont le temps de parcours est inférieur ou égal à T0 (T0 est le chemin de temps minimal en respectant les limitations de vitesse), On rentre le nombre de radars mobiles (20 par défaut), et (0,8 par défaut) puis on clique sur calculez :

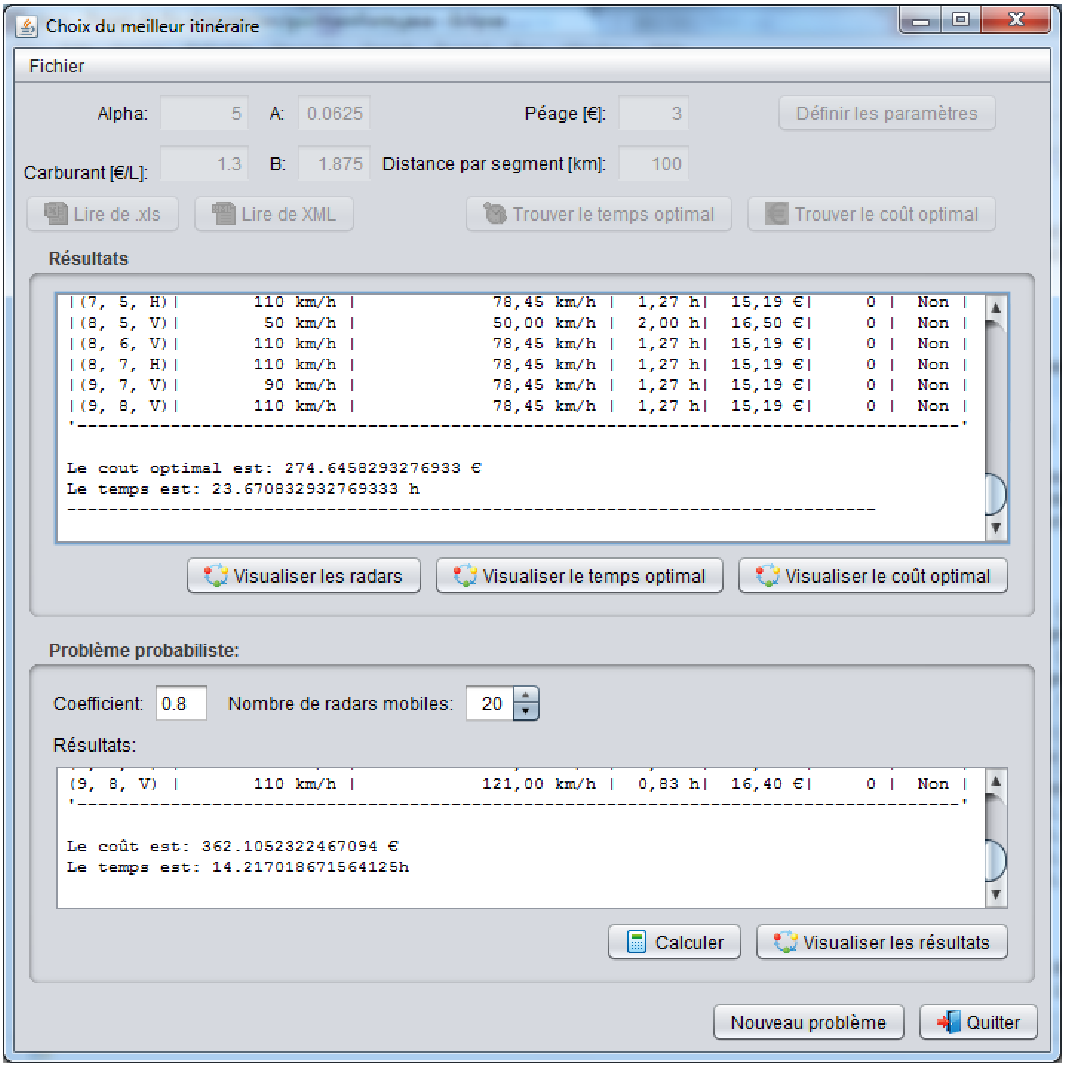


FIG. 4.10 – Paramétrage de la partie probabiliste.

On peut enfin visualiser le chemin obtenu en cliquant sur visualiser les résultats :

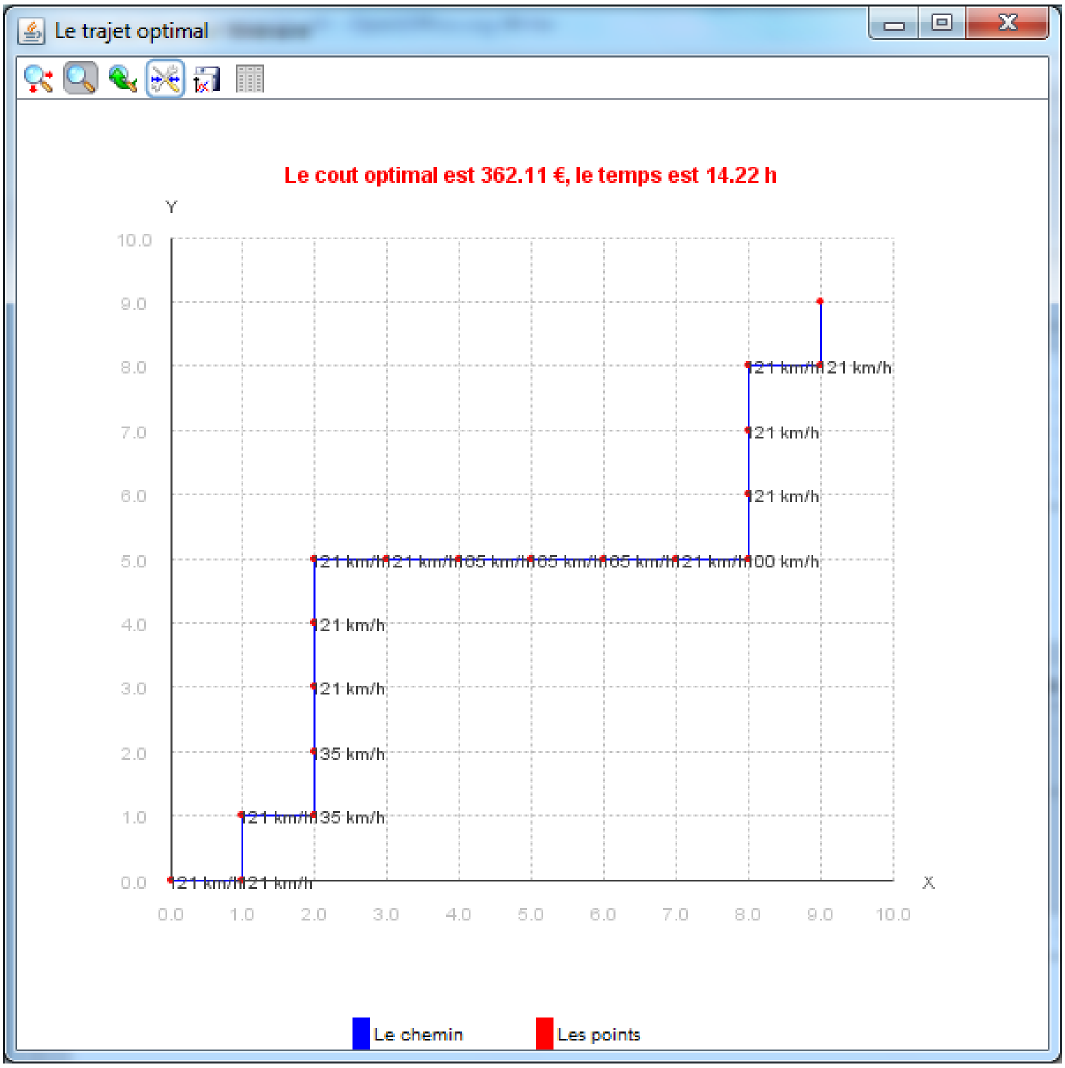


FIG. 4.11 – Visualisation du meilleur chemin obtenu.

4.6 Sauvegarde des résultats

Les résultats sous forme graphique peuvent être sauvegardés à partir de leur visualisation soit sous forme graphique ou texte.

Les sorties texte ne peuvent être sauvegardés qu'en applicant un copier/coller. Étant pris par le temps, nous nous somme penché sur le développement des méthodes qui représentent le cœur de notre travail.

Les fonctions sauvegarde et lecture de solutions pour les comparer et les amélioré par d'éventuels hybridation avec des heuristiques telle la recherche local, seraient une prochaine étape qui vient juste après le travail ici présenté.

Conclusion

Dans ce travail nous avons étudié une problématique concrète et nous avons mis en pratique l'application de l'algorithme de Dijkstra pour retrouver le plus court chemin en termes de temps puis en termes de coût.

Nous avons aussi, mis en œuvre un algorithme basé sur des labels pour un problème à deux critères. Cette algorithme se base sur une discrétisation particulière des vitesse afin d'obtenir des solutions les plus avantageuses.
Une solution est ensuite choisit le plus proche des attentes du conducteur : de temps proche de sa volonté d'aller à 80% plus vite que si il respectait les limitations de vitesse et qui offre le coût minimal.

Notre solution offre un temps inférieur au temps espéré du conducteur, ce qui suggère qu'on peut l'améliorer (diminuer le coût en s'approchant de la valeur du temps de référence).

Bibliographie

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 2nd ed., 2001.

[2] Roseaux, *préface de Robert Faure, Exercices et problèmes résolus de recherche opérationnelle*, Paris, 1986.

[3] G. SAPORTA, *Probabiltés, analyse de données et statistique*, Technip, Paris, 2006.