

Projet de Fin d'Etudes (PFE) 2020-2021

Étude comparée des modes de calcul des durées de
déplacement entre un ensemble d'origines et un
ensemble de destinations

Directeur de recherche :

Mindjid MAIZIA

Léonie MICHARD

PFE/DAE5

UIT/RESEAU

2020-2021

Étude comparée des modes de calcul des durées de déplacement entre un ensemble d'origines et un ensemble de destinations

Résumé :

Dans le domaine des transports, on détermine généralement la durée des déplacements du domicile vers le lieu de travail grâce à un algorithme de plus court chemin. Or, des applications ne fonctionnant pas totalement sur ce type d'algorithme, à l'image de Google Maps ou de Waze, sont souvent employés par les personnes actives pour rejoindre leur lieu de travail. Afin de comprendre pourquoi ces deux utilisateurs n'utilisent pas les mêmes outils de calcul de la plus courte durée, il est nécessaire de comparer ces différentes méthodes de calcul d'une durée de plus court chemin. L'étude des écarts des durées nous amène à penser que ces différents modèles ne fonctionnent pas selon les mêmes critères. Les durées obtenues avec un algorithme de plus courts chemins sont plus faibles que celles fournis par les applications. Cela s'explique notamment par le fait que les algorithmes de plus court chemin ne prennent pas en compte les éléments relatifs à la congestion ou à la perturbation du trafic. Cependant, ce qui intéresse l'industrie du transport, ce ne sont pas les durées de déplacement mais les flux qu'elles génèrent. Nous allons donc, à partir d'un modèle liant les flux à la durée, déterminer à quel point les différences de durées ont un impact sur lors du calcul des flux.

AVERTISSEMENT

Cette recherche a fait appel à des lectures, enquêtes et interviews. Tout emprunt à des contenus d'interviews, des écrits autres que strictement personnel, toute reproduction et citation, font systématiquement l'objet d'un référencement.

L'auteur de cette recherche a signé une attestation sur l'honneur de non-plagiat.

Formation par la recherche, Projet de Fin d'Etudes en génie de l'aménagement et de l'environnement

La formation au génie de l'aménagement et de l'environnement, assurée par le département aménagement et environnement de l'Ecole Polytechnique de l'Université de Tours, associe dans le champ de l'urbanisme, de l'aménagement des espaces fortement à faiblement anthropisés, l'acquisition de connaissances fondamentales, l'acquisition de techniques et de savoir-faire, la formation à la pratique professionnelle et la formation par la recherche. Cette dernière ne vise pas à former les seuls futurs élèves désireux de prolonger leur formation par les études doctorales, mais tout en ouvrant à cette voie, elle vise tout d'abord à favoriser la capacité des futurs ingénieurs à :

- Accroître leurs compétences en matière de pratique professionnelle par la mobilisation de connaissances et de techniques, dont les fondements et contenus ont été explorés le plus finement possible afin d'en assurer une bonne maîtrise intellectuelle et pratique,
- Accroître la capacité des ingénieurs en génie de l'aménagement et de l'environnement à innover tant en matière de méthodes que d'outils, mobilisables pour affronter et résoudre les problèmes complexes posés par l'organisation et la gestion des espaces.

La formation par la recherche inclut un exercice individuel de recherche, le projet de fin d'études (P.F.E.), situé en dernière année de formation des élèves ingénieurs. Cet exercice correspond à un stage d'une durée minimum de trois mois, en laboratoire de recherche, principalement au sein de l'équipe Dynamiques et Actions Territoriales et Environnementales de l'UMR 7324 CITERES à laquelle appartiennent les enseignants-chercheurs du département aménagement.

Le travail de recherche, dont l'objectif de base est d'acquérir une compétence méthodologique en matière de recherche, doit répondre à l'un des deux grands objectifs :

- Développer toute ou partie d'une méthode ou d'un outil nouveau permettant le traitement innovant d'un problème d'aménagement
- Approfondir les connaissances de base pour mieux affronter une question complexe en matière d'aménagement.

Afin de valoriser ce travail de recherche nous avons décidé de mettre en ligne sur la base du Système Universitaire de Documentation (SUDOC), les mémoires à partir de la mention bien.

REMERCIEMENTS

Je souhaiterais particulièrement remercier mon directeur de recherche Monsieur Mindjid Maizia, professeur de l'option RESEAU, Réseaux Espace et Système pour l'Environnement et l'Aménagement Urbain à l'École Polytechnique de l'Université de Tours. Je le remercie pour le temps qu'il m'a consacré afin de mener à bien cette recherche. J'ai ainsi pu réaliser ce projet en utilisant les savoirs, méthodes et réflexions qu'il m'a transmis. Je le remercie aussi pour sa disponibilité et sa réactivité.

Je tiens également à remercier mes compagnons de classe, Lou Lorient, Sabine Madin et Maxime Ott, pour leur aide et leur écoute qui m'a permis de réaliser de ce projet de recherche.

Enfin je tiens à témoigner toute ma reconnaissance à Mathieu Clavier, pour son aide et son soutien tout au long de mes recherches.

Table des matières

1. Enjeux	8
2. État de l'art	9
2.1. Mode de calcul d'une durée	9
2.1.1. Approche générale.....	9
2.1.2. Algorithmes de plus courts chemins.....	10
2.1.3. Synthèse	12
2.2. Modèles de calcul des flux	12
2.3. Discussion	13
3. Méthode	13
4. Étude.....	13
4.1. Travail sur les durées.....	13
4.1.1. Première approche	14
4.1.2. Comparaison deux à deux.....	15
4.2. Travail sur les flux.....	17
4.2.1. Avant-propos	17
4.2.2. Première approche	18
4.2.3. Etude de la fonction du calcul des flux	18
5. Conclusion	19
6. Bibliographie.....	20
7. Annexe 1 : Code du système logique permettant de comparer les durées obtenues par les algorithmes de Dijkstra et de Bellman-Ford.	21
8. Annexe 2 : Bases de données des durées et flux associés.	24
9. Annexe 3 : Création des plus courts chemins avec l'algorithme de Dijkstra	27
10. Annexe 4 : Étude des variations de β	32

Table des illustrations

Figure 1 : Durée obtenue avec l'application Google Maps.....	8
Figure 2 : Durée obtenue avec l'application Waze	8
Figure 3 : Exemple d'approche en profondeur	9
Figure 4 : Exemple approche en largeur.....	9
Figure 5 : Exemple de Floyd-Warshall	10
Figure 6 : Exemple de numérotation.....	11
Figure 7 : Exemple algorithme de Dijkstra.....	12
Figure 8 : Déplacements obtenus avec l'algorithme de Dijkstra	14
Figure 9 : Différence des durées entre Waze et Google Maps en fonction des durées croissantes sur Google Maps.....	14
Figure 10 : Différence en proportion des durées en valeur absolue en fonction des durées croissantes de Maps.....	14
Figure 11 : Différence en proportion des moyennes glissantes des durées en valeur absolue en fonction des durées croissantes de Maps	14
Figure 12 : Comparaison entre les durées obtenues avec l'algorithme de Dijkstra et l'application Google Maps.....	15
Figure 13 : Comparaison entre les durées obtenues avec l'algorithme de Dijkstra et l'application Waze	16
Figure 14 : Comparaison entre les durées obtenues avec l'application Google Maps et l'application Waze	16
Figure 15 : Étude de sensibilité de la variable β	17
Figure 16 : Flux en fonction de la durée de Google Maps	18
Figure 17 : Dérivé des flux en fonction de la durée	18

1. Enjeux

Ce PFE a pour sujet les modes de calcul de durée de déplacement et l'impact des différentes durées sur les flux. Afin de comprendre les enjeux de ce sujet, nous allons l'illustrer.

Lors de déplacement, notamment nouveau, l'utilisation d'une application à l'image de Google Maps ou de Waze est souvent favorisée. Cependant le choix de ces applications peut entraîner une différence dans la durée de parcours annoncée au départ. En effet, les images suivantes montrent deux trajets pour aller de la même origine à la même destination, l'une avec l'application Google Maps (figure 1) l'autre avec l'application Waze (figure 2).

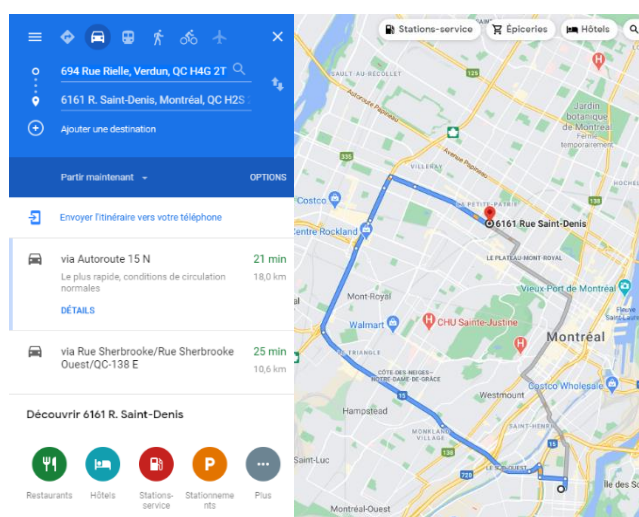


Figure 1 : Durée obtenue avec l'application Google Maps

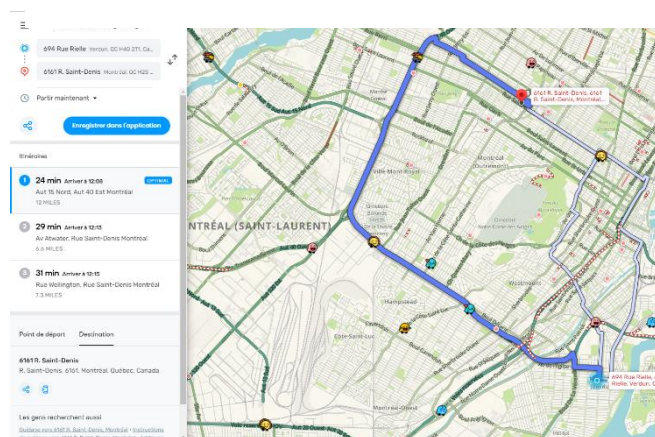


Figure 2 : Durée obtenue avec l'application Waze

Comme on le voit sur la figure 1, il est possible que les deux applications donnent pour un même chemin des durées de parcours différentes. En effet, avec Maps on obtient une durée de trajet de 21 minutes et

pour le même déplacement avec Waze une durée de 24 minutes. L'écart est de 3 minutes et est donc important. Ces deux applications peuvent parfois donner des durées différentes ainsi que des chemins différents. L'études porte sur les durées, il n'est donc pas nécessaire de prendre en compte le chemin que va proposer chaque application.

Cet exemple nous permet de supposer que les calculs effectués par les applications pour déterminer la durée du plus court chemin ne sont pas les mêmes. En effet, il est facilement possible d'imaginer que différents paramètres sont pris en compte lors des calculs de plus court chemin de ces applications comme la densité du trafic, les potentiels travaux, les routes barrées... Mais aussi que ces paramètres ne sont pas pris en compte de la même façon. Il serait donc intéressant de connaître les modes de calcul utilisés par les différentes applications cela expliquerait les divergences de durées obtenues dans cet exemple.

En effet, ce genre d'application est omniprésente dans notre quotidien, ils sont essentiels puisqu'il permettent de faire gagner un temps précieux, mais aussi d'économiser énormément d'argent pour l'industrie du transport notamment. En effet l'optimisation des durées de trajets est un enjeu stratégique fondamental pour l'industrie du transport. Il est intéressant, notamment pour eux, de pouvoir quantifier les écarts de durées mais aussi de lier ces durées aux flux. En effet, les flux représentant la quantité de véhicules déplacés d'un point à un autre, l'industrie du transport cherche donc à le maximiser. Le choix de l'application pourrait donc avoir un impact sur les flux et donc sur l'économie des industries du transport.

Google Maps et Waze étant des applications commerciales, il nous est impossible de connaître la façon dont elles sont codées. Cependant l'idée de base de ces codes est nécessairement un calcul du plus court chemin, émanant de la modification d'un algorithme de plus court chemin. Afin de bien comprendre ce sujet il est donc nécessaire de savoir comment fonctionne les algorithmes de plus court chemin. De même, il est nécessaire de déterminer la façon dont le calcul des flux peut être lié avec la durée du déplacement.

2. État de l'art

N'ayant pas accès aux codes des applications Google Maps ou Waze citées en exemple, ce PFE se concentre, dans un premier temps, sur le fonctionnement des algorithmes de plus courts chemins. Algorithmes qui, selon l'hypothèse en préambule, compose le code des applications. Cela nous permettra de pouvoir quantifier les écarts de durées.

Le second enjeu concerne le calcul des flux, il faut dans un premier temps déterminer le modèle de calcul des flux lié à la durée avant de pouvoir comprendre l'impact de la durée sur les flux.

La première partie de cet état de l'art sera donc consacré à la démonstration de l'existence plusieurs modes de calcul de durée. La seconde, quant à elle, sera consacrée au modèle de calcul des flux. Tout cela permettra d'aboutir sur la problématique et l'hypothèse de ce PFE.

2.1. Mode de calcul d'une durée

Un problème de plus court chemin est représenté sous forme de graphe. Le graphe est composé de sommets et de branches. Les branches (ou arcs) relient les sommets entre eux et des poids peuvent leur être affectés. Ces poids peuvent être des coûts ou des longueurs mais dans notre cas ce seront des durées en minutes. Trouver le plus court chemin pour aller d'un point à un autre revient à traverser le graphe en passant par les arcs donnant la durée la plus courte pour un trajet donné.

2.1.1. Approche générale

Les différents algorithmes correspondent aux différentes façons de parcourir ce graphe. Il existe 2 approches pour parcourir un graphe que sont l'approche en profondeur et l'approche en largeur.

Pour l'approche en profondeur on considère le graphe de la figure 3. Ce graphe est composé de 6 sommets nommés v_1 jusqu'à v_6 . Le départ est sur le sommet v_1 , il est noté comme découvert (a). Les sommets v_2 et v_3 sont reliés à ce dernier, ils sont donc considérés comme exploré et mis sur liste « d'attente », c'est-à-dire sur la liste des sommets explorés mais non découverts. On considère que le sommet v_2 est premier sur la liste. En tant que premier sur la liste c'est le prochain sommet à être découvert (b), les sommets qui lui sont reliés sont

alors ajoutés avant le sommet v_3 sur la liste dans l'ordre suivant : v_4 puis v_5 .

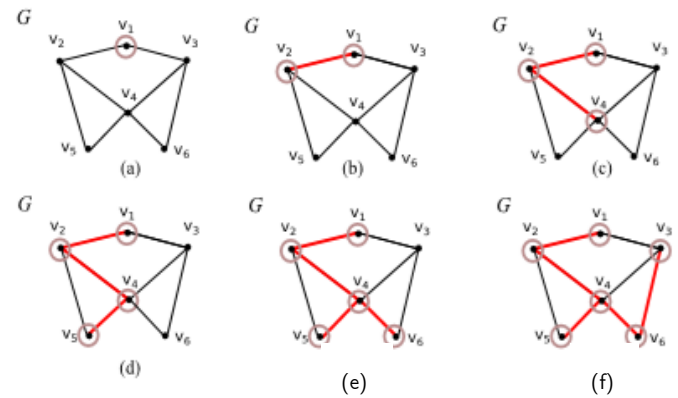


Figure 3 : Exemple d'approche en profondeur

Le sommet v_4 est alors découvert (c) et les sommets qui lui sont liés, v_5 et v_6 , sont placés dans cet ordre en amont de la liste des sommets explorés. Le sommet v_5 est donc le prochain à être découvert (d), sachant que les sommets auxquels il est lié (v_2 et v_4) sont déjà découverts, on ne rajoute rien sur la liste des sommets explorés. Le premier sommet sur la liste est donc le sommet v_6 , il est désormais découvert (e) et permet d'ajouter à la liste le sommet v_3 qui est sur cette liste depuis le début. Enfin, le sommet v_3 est découvert (f) [2].

Une autre méthode de parcours est l'approche en largeur. Cette approche suit le même principe d'exploration/découverte mais n'est pas réalisée de la même façon. On considère le graphe de la figure 4. En effet, les sommets liés sont ajoutés à la liste des sommets explorés après et non plus avant les sommets déjà présents. À l'étape (b) la liste d'attente est la suivante : v_3 , v_5 , v_4 . Cela impose donc que le sommet v_3 est découvert en 3^{ème} (c) après le sommet v_2 mais avant les sommets qui sont liés à v_2 (d), (e).

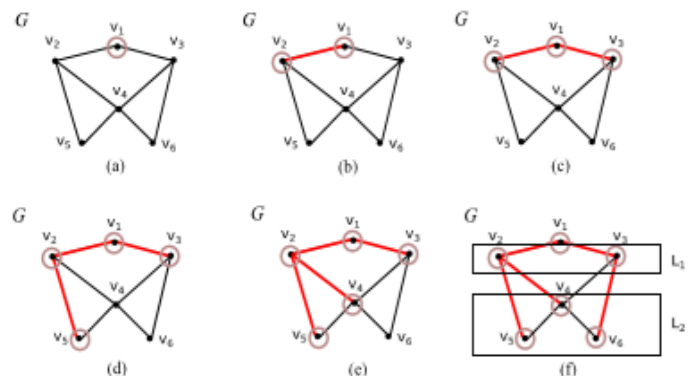


Figure 4 : Exemple approche en largeur

Le sommet v_6 est ajouté à la fin de la liste à l'étape (c), c'est le dernier sommet découvert.

Ces deux approches permettent de déterminer pour des graphes non orientés si un chemin pour aller d'un sommet à un autre existe. Cette première approche permet de se familiariser avec les graphes.

Or le propos est de déterminer la durée du trajet. Pour cela il faut travailler sur des graphes avec des poids sur les arcs. Nous allons nous intéresser aux méthodes pour parcourir ces graphes qui donne la durée de parcours minimal. Nous allons donc travailler sur les algorithmes de plus courts chemins.

2.1.2. Algorithmes de plus courts chemins

Les algorithmes de plus courts chemins sont une solution pour un problème de durée minimale. Ces algorithmes permettent de connaître le plus court chemin pour des graphes orientés et dont les arcs ont un poids. Il existe plusieurs algorithmes, en effet comme on l'a vu dans la partie précédente il y a plusieurs moyens de parcourir un graphe. Nous allons en détailler trois, l'algorithme de Floyd-Warshall, l'algorithme de Bellman-Ford et l'algorithme de Dijkstra. L'algorithme de Floyd-Warshall permet de déterminer le plus court chemin entre chacun des sommets du graphe. Les deux suivants permettent de rejoindre seulement un sommet spécifique à un autre.

Algorithme de Floyd-Warshall :

Cet algorithme permet d'obtenir les plus courts chemins entre chacun des sommets deux à deux. Le résultat présenté sous forme d'une matrice carrée est composé des durées minimales pour aller d'un sommet à un autre [2]. Sachant que les valeurs de la diagonale représentent le trajet d'un sommet jusqu'à lui-même, la matrice est composée d'une diagonale de zéros. Grace au graphique suivant

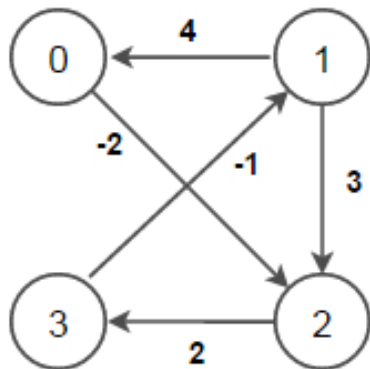


Figure 5 : Exemple de Floyd-Warshall

(Figure 5) nous allons pouvoir expliquer le fonctionnement de l'algorithme concrètement.

Le processus est initialisé grâce aux chemins apparents. Dans cet exemple seul les chemins 0-2 ; 1-0 ; 1-2 ; 2-3 et 3-1 ont pour le moment une durée. Celle-ci est la valeur indiquée sur l'arc. Les autres chemins ont une durée initiale de $+\infty$.

L'idée de l'algorithme est de supposer le passage par chaque sommet k de 0 à 3. Pour chaque k l'algorithme compare la valeur du chemin précédemment enregistrée avec la valeur du chemin incluant le sommet k . Si la durée comprenant le passage par le sommet k est plus faible alors c'est ce dernier qui est retenu. Sinon, rien ne change.

Ainsi, soit $P(i, j, k)$ la fonction qui donne la durée du plus court chemin entre les sommets i et j en passant par les sommets de 1 à k . On obtient :

$$P(i, j, k) = \min(P(i, j, k - 1) ; P(i, k, k - 1) + P(k, j, k - 1))$$

En suivant l'exemple de la figure 5 on obtient les étapes suivantes :

Pour $k = 0$:

Le chemin de 1 vers 2 est plus court en passant par le sommet 0 ($4 - 2 = 2 < 3$), le chemin retenu pour aller de 1 vers 2 est donc 1,0,2 qui a pour valeur 2. Les autres sommets ne sont pas directement reliés à 0.

Pour $k = 1$:

Pour aller de 3 à 0 ou de 3 à 2 les chemins passent par 1. Le sommet 3 est le seul sommet qui admet un arc orienté vers le sommet 1. Sachant que les durées des chemins 3-2 et 3-0 sont initialisées à $+\infty$, les durées retenues sont celles en passant par le sommet 1. En tenant compte de l'étape précédente, le chemin du sommet 3 vers le sommet 2 le plus court est : 3-1-0-2 (durée : 1).

Pour $k = 2$:

Les chemins 0-2-3 (0) et 1-0-2-3 (4) sont retenus puisqu'ils sont les plus courts.

Pour $k = 3$:

On obtient les chemins 2-3-1 (1) ; 2-3-1-0 (5) ; 0-2-3-1 (-1).

Ainsi on a les durées suivantes :

DE/VERS	0	1	2	3
0	0	-1	-2	0
1	4	0	2	4
2	5	1	0	2
3	3	-1	1	0

Cette méthode permet de connaître les plus courts chemins entre chaque sommet donc on peut en déduire le plus court chemin entre deux sommets spécifiques. Or, si l'on ne cherche qu'une valeur entre deux sommets, le fait de calculer tous les plus courts chemins implique des calculs inutiles. En effet le temps de réalisation de cet algorithme est d'autant plus grand qu'il y a de sommet. Le temps de calcul est de S^3 , c'est-à-dire le nombre de sommet au cube [1].

D'autres algorithmes permettent d'obtenir directement le plus court chemin entre deux sommets spécifiés. Nous allons détailler les deux algorithmes fonctionnant de cette façon [2].

Algorithme de Bellman-Ford :

Cet algorithme fonctionne comme tous les autres à partir d'un graphe. Ce graphe ne doit cependant pas comporter de circuit ou « boucle ».

La première étape pour trouver le plus court chemin est de numéroté les sommets. Pour les numéroté, il faut calculer le nombre d'arc incident qu'ils ont. (Figure 6)

Pour deux sommets au même « niveau » dans le graphe, on numérote en premier le sommet ayant le

moins d'antécédents. Le sommet c à 1 antécédent tandis que le sommet a en a 2, le sommet c a donc un numéro antérieur à celui du sommet a . En cas d'égalité des antécédents il faut faire un choix arbitraire entre les sommets concernés afin de les numéroté. Pour chaque sommet, en les étudiant dans l'ordre croissant de la numérotation, la durée minimale de trajet est calculée grâce aux sommets précédents. Sachant que le plus court chemin se calcule grâce aux sommets prédécesseurs, le graphe ne peut pas comporter de circuit. Au sein d'un circuit, les sommets sont prédécesseur les uns des autres, il est donc impossible de déterminer les durées minimales de trajet avec cette méthode. S'il y a plusieurs antécédents, c'est la durée minimale entre les deux qui est retenue. Une fois qu'une valeur est attribuée à un sommet, cette valeur ne change plus [3]. Prenons l'exemple du sommet nommé a , il a 2 prédécesseurs, c et s . Soit $\lambda(x)$ la durée du plus court chemin de s (point de départ du graphe) à x et $\gamma(x,y)$ la durée pour aller du sommet x au sommet y (y et x sont directement lié sur le graphe). Alors pour l'exemple on obtient :

$$\begin{aligned}\lambda(a) &= \min(\lambda(s) + \gamma(s,a) ; \lambda(c) + \gamma(c,a)) \\ &= \min(0 + 5 ; 1 + 2) = 3\end{aligned}$$

En allant de sommet en sommet en suivant la numérotation on obtient donc le plus court chemin pour aller d'un sommet à un autre. L'algorithme s'arrête une fois que le plus court chemin jusqu'au sommet désiré a été trouvé. L'algorithme de Bellman est rapide et son temps de calcul croît comme m , le nombre d'arc.

L'algorithme de Dijkstra :

L'algorithme de Dijkstra permet de connaître le plus court chemin entre un sommet de départ et n'importe quel sommet du graphe [2]. Ce graphe doit être valué et ne doit comporter que des valeurs positives. Le principe de l'algorithme est de partir d'un sommet et d'examiner ses successeurs. La stratégie de l'algorithme de Dijkstra est basée sur les successeurs contrairement à celle de l'algorithme de Bellman qui utilise les prédécesseurs. Pour chaque sommet k on examine ses successeurs. Pour cela on compare la valeur enregistrée de la durée du plus court chemin avec celle du sommet k augmentée de la longueur de l'arc qui relie ce successeur au sommet k . Si cette dernière est inférieure à la valeur enregistrée, alors elle devient la nouvelle valeur enregistrée. Si une nouvelle valeur est enregistrée, alors il faut aussi retenir le nom

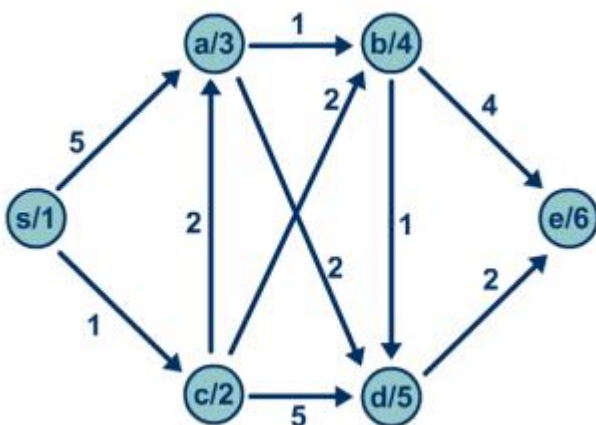


Figure 6 : Exemple de numérotation

du sommet père, afin de pouvoir reconstituer le plus court chemins (Figure 7). Tous les sommets sont initialisés à $+\infty$ sauf le sommet de départ qui lui est initialisé à 0.

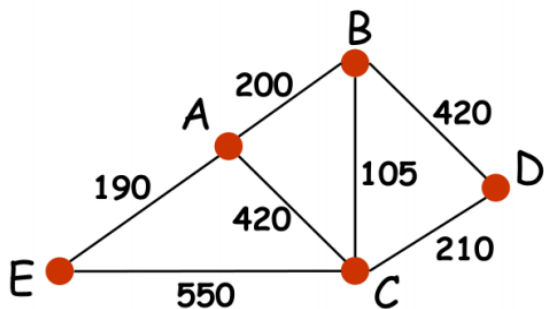


Figure 7 : Exemple algorithme de Dijkstra

En appliquant l'algorithme de Dijkstra pour aller du sommet D au sommet E sur la figure 7 on obtient le tableau suivant :

Itération	Sommet choisi	A	B	C	D	E
Initialisation	D	∞	∞	∞	0	∞
Itération 1	C	∞	420 (D)	210 (D)	-	∞
Itération 2	B	630 (C)	315 (C)	-	-	760 (C)
Itération 3	A	515 (B)	-	-	-	760 (C)
Itération 4	E	-	-	-	-	705 (A)

Le plus court chemin est donc D-C-B-A-E.

Le temps de calcul de cet algorithme dépend du nombre de sommet, n , et à une complexité en $O(n^2)$. Sachant qu'il y a moins d'arc que de sommet au carrée, l'algorithme de Bellman est plus rapide. Cependant il ne peut pas être utilisé si le graphe comporte des cycles, même si les valeurs sont négatives. Les deux ne sont donc pas opposés et peuvent être complémentaires.

2.1.3. Synthèse

Sachant que ces algorithmes ne fonctionnent pas de la même façon, Il est normal de se demander s'il donne des résultats différents. Pour cela, ces deux algorithmes ont été modélisés sur un réseau fictif. Cela permet de savoir s'ils donnent les mêmes résultats en termes de durée ou pas. De façon intuitive, on suppose que les résultats seront les mêmes. En effet même s'il fonctionne de de manière différente leur objectif est de calculer le plus court chemin. Cette intuition est confirmée grâce au système logique de [l'annexe 1](#). En effet, les durées de déplacement calculées sont les mêmes avec les deux algorithmes. Une comparaison des durées de déplacement pourra donc être effectuée entre Waze, Google Maps et un des algorithmes, l'algorithme de Dijkstra par exemple.

2.2. Modèles de calcul des flux

Cette deuxième partie traite le calcul des flux. Mais tout d'abord qu'est-ce qu'un flux ? Le flux peut être compris comme l'expression du déplacement d'une quantité de personne d'un point à un autre. Dans ce projet, il est question de déplacement routier, les quantités déplacées sont donc des véhicules. Afin de pouvoir déterminer l'impact de la durée sur les flux, enjeux énoncés en préambule, il est nécessaire de connaitre une relation liant la durée d'un déplacement aux flux.

Le modèle de calcul des flux le plus répandu, et celui utilisé dans l'industrie du transport, est le modèle gravitaire ou modèle de Reilly. Ce modèle, inspiré de la loi de gravitation universelle Newtonienne, permet de calculer les flux entre un ensemble d'origines et de destinations [\[4\]](#). Ce modèle lie les flux entre une origine et une destination à la durée séparant ces dernières. Ce modèle est applicable et est exprimé grâce à une équation : $t_{ij} = D_{ij}^{-\beta}$. Avec t_{ij} le flux entre les points i et j , D_{ij} la durée entre les points i et j , β est une des variables. La variable β représentera le frein au déplacement aussi appelé degrés de motivation ou résistance au déplacement. Cette équation comprend initialement d'autres paramètre. Cependant l'objectif de cette première partie est de déterminer une relation entre la durée d'un déplacement et les flux pour pouvoir connaître l'impact du premier sur le second. Les autres paramètres n'ont donc pas d'importance pour ce projet, ils sont donc assimilés à 1.

2.3. Discussion

Ainsi il existe plusieurs méthodes de calcul de durée de plus court chemin, avec un algorithme de plus court chemin ou avec des applications comme Google Maps ou Waze. Ces différentes méthodes peuvent entraîner des résultats différents. On peut tout d'abord noter les différences de durée entre les applications Google Maps et Waze exposées en introduction. Il semble donc qu'il y est des différences de durées entre les méthodes de calcul d'un plus court chemin. Il serait donc intéressant de comparer ces différentes méthodes et de mettre en lumière les potentielles divergences. Ces divergences pourront ainsi se retrouver après le calcul des flux. Cette étude nous permet donc d'établir une problématique sur le sujet du PFE et de donner une hypothèse pour y répondre.

Problématique : À quel point le choix d'une application par rapport à une autre a-t-il un impact lors du calcul des flux ?

Hypothèse : Il y a plusieurs méthodes pour calculer une durée qui donnent des résultats différents, il y aura donc des impacts différents selon la durée choisie.

3. Méthode

D'une part, il existe plusieurs façons de déterminer une durée. Selon le mode de calcul employé, on peut supposer que les durées diffèrent, comme le montre les figures 1 et 2. Avec un échantillon de durées pour les mêmes déplacements calculé avec les différents modèle et applications, une potentiel tendance des écarts de durées pourra être mise en lumière. De plus cela va permettre de connaître l'importance des écarts. Pour cela, il faut dans un premier temps créer une base de données puis l'exploiter. Cette base de données va être réaliser à partir du réseau routier de la ville de Montréal pour des véhicules de types voiture ([Annexe 2](#)). Cette méthode de résolution de problème par l'exemple permettra de conclure quant à un modèle mathématique liant les méthodes de calcul d'une durée.

D'autre part, pour déterminer à quel point la durée impact les flux il est nécessaire d'étudier la fonction décrivant les flux. Pour cela il faut déterminer l'inconnu β mais aussi l'impact de la durée sur le calcul. Cela permettra donc de déterminer à quel point la durée à un impact sur les flux. La

méthode utilisée dans cette seconde partie est une méthode logique reposant sur des connaissances mathématiques.

Formalisons mathématiquement l'objectif :

Soit x un trajet

Soit f_i l'ensemble des fonctions de calcul de la durée d'un trajet avec $i \in \mathbb{N}$

Soit g , la fonction de calcul des flux en fonction de la durée

- ✓ Dans un premier temps, il s'agit comparer $\forall i \in \mathbb{N}$ les f_i
- ✓ Dans un second temps, il s'agit de déterminer :

$$\forall i \in \mathbb{N} \text{ et les flux } K_i \text{ tel que } K_i = g(f_i(x)) \\ = (f_i \circ g)(x)$$

Nous allons donc nous intéresser à la fonction composée : $(f_i \circ g)(x)$

4. Étude

4.1. Travail sur les durées

En amont de cette étude nous avons donc réaliser une base de données. Waze et Google Maps étant des applications, il est très facile d'obtenir des durées de déplacement, il faut cependant réaliser un code pour pouvoir obtenir des durées avec l'algorithme de Dijkstra. En effet, le système logique de [l'annexe 3](#) nous permet d'obtenir les durées de déplacement depuis deux origines jusqu'à six destinations. Ce système logique permet aussi de tracer les chemins empruntés par l'algorithme de Dijkstra (figure 8), ce qui nous permet de reporter les origines et les destinations dans les applications Waze et Google Maps. Le fond de carte utilisé est celui fourni par la ville de Montréal.

On obtient donc 12 durées de déplacement pour chaque modèle de calcul. De plus, afin d'avoir un échantillon plus conséquent pour lesquelles le temps de réalisation est faible, nous avons réalisé une base de données de 64 durées du déplacement avec les applications Waze et Google Maps.



Figure 8 : Déplacements obtenus avec l'algorithme de Dijkstra

4.1.1. Première approche

Lorsque l'on observe, notamment sur le plus gros échantillon, les différences entre les durées observées avec Waze et celle avec Google Maps semblent aléatoire. En effet comme le montre la figure 9 aucune tendance ne se dégage.

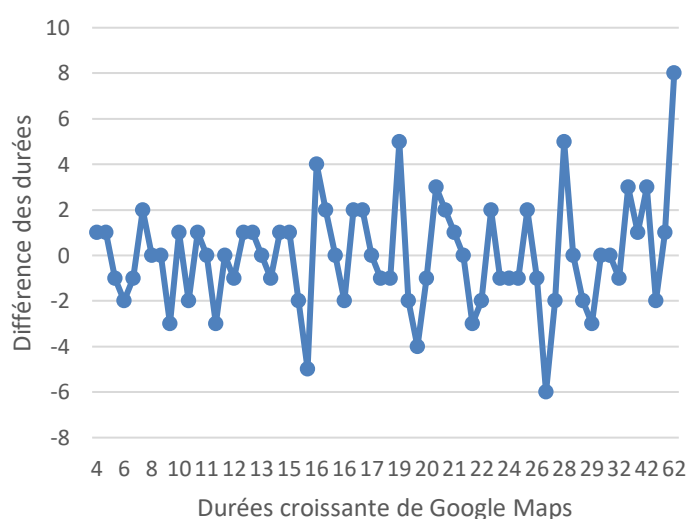


Figure 9 : Différence des durées entre Waze et Google Maps en fonction des durées croissantes sur Google Maps

Cependant lorsque l'on regarde plus précisément ce graphique on remarque que les écarts ne dépassent pas 10 minutes tandis que les durées de déplacement vont de 4 à 45 minutes. En effet, avant toute étude on peut supposer que les écarts pour de grandes valeurs semblent moins important que pour les plus faibles valeurs. Dans le but de vérifier cette hypothèse, il est possible de tracer la proportion de la différence par rapport à la durée initiale. Dans ce cas nous

prendrons comme durée initiale celle de Google Maps, ce choix est totalement arbitraire mais il n'a pas d'impact sur le résultat puisque les différences n'excèdent pas moitié de la durée initiale. Ce calcul est réalisé en valeur absolue afin de travailler seulement avec des valeurs positives. On obtient le graphique suivant :

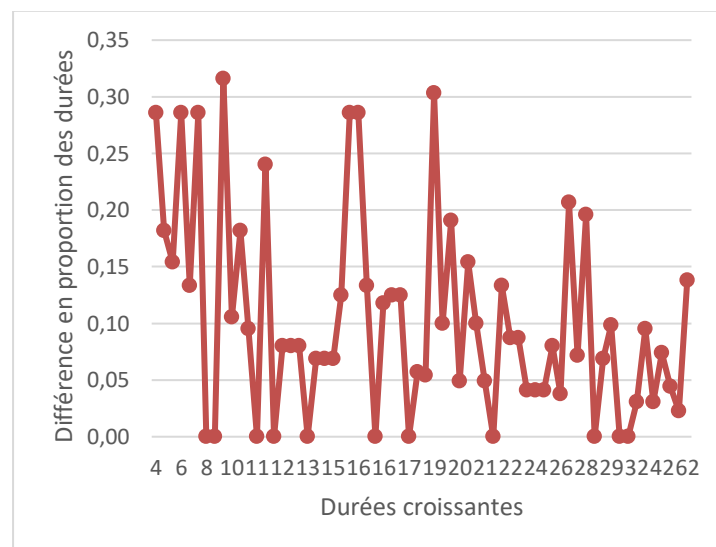


Figure 10 : Différence en proportion des durées en valeur absolue en fonction des durées croissantes de Maps

On remarque, qu'en proportion, plus la durée est élevée moins la différence est grande. C'est-à-dire que plus le temps de trajet est long plus, l'écart entre les différentes applications est faible. Afin de confirmer cette hypothèse, la moyenne mobile de ces proportions a été tracée. C'est une moyenne mobile de dix valeurs.

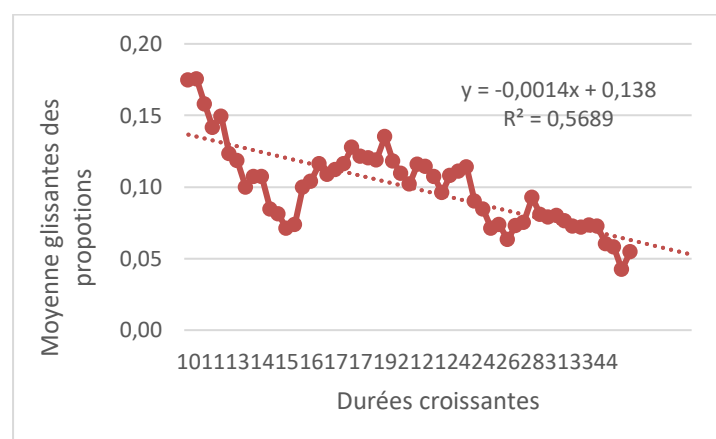


Figure 11 : Différence en proportion des moyennes glissantes des durées en valeur absolue en fonction des durées croissantes de Maps

Avec la figure ci-dessus, l'hypothèse se confirme. Une tendance négative peut être observée avec un R^2 de 0, 57. Ce résultat reste tout de même volatile, un échantillon plus important aurait sans doute donné de

meilleurs résultats. Cette figure confirme tout de même que les différences de durée sont en proportion plus faible pour des grandes durées. Cela peut s'expliquer par le fait que pour une grande durée, même si les algorithmes ne prennent pas en compte les mêmes éléments les divergences ont le temps de « corriger », ce qui n'est pas le cas pour les faibles durées.

4.1.2. Comparaison deux à deux

Comme évoqué en préambule de cette partie, les comparaisons avec les durées obtenues avec l'algorithme de Dijkstra seront faites sur la base de 12 durées répertoriées dans le tableau suivant selon les couleurs de la figure 8. Certaines couleurs n'apparaissent pas, ou très peu puisque qu'ils utilisent des arcs empruntés par d'autres chemins, dans ces cas, l'épaisseur de l'arc est plus grande.

Couleur	Durées avec Dijkstra	Durées avec Google Maps	Durées avec Waze
Vert	17	24	24
Bleu clair	6	12	13
Saumon	8	17	19
Fuchsia	9	20	21
Orange	16	26	28
Jaune	18	25	25
Violet	13	19	20
Cyan	7	16	17
Bleu	11	17	17
Mauve	5	15	18
Vert foncé	10	20	23
Rouge	14	19	21

On remarque tout d'abord sur ce tableau que les durées obtenues avec l'algorithme de Dijkstra sont moins importantes que celles obtenues avec les deux applications. Afin de quantifier ces écarts, ces trois méthodes de calcul d'une durée seront comparées deux à deux. Ne connaissant pas de fonction les liant les uns aux autres, nos déductions seront donc appuyées par l'expérimentation.

Ainsi il est possible de réaliser une comparaison entre la durée fournie par l'algorithme de Dijkstra et celle fournie par l'application Google Maps sur la figure suivante. En abscisse, les durées selon l'algorithme de Dijkstra et en ordonné celles obtenue avec l'application Google Maps. En jaune, la régression linéaire tracée à un coefficient de détermination $R^2 = 0,79$. Cette régression permet

donc d'expliquer 79 % des variations de la distribution. Ce coefficient est satisfaisant, l'interprétation de l'équation de la droite de régression permet de déterminer plus de 50% de la distribution des points. Cette équation, $y = 0,85x + 9,7$ permet d'obtenir l'écart théorique de durée, l'écart est ici de 10 minutes. Afin de donner du sens à cet écart, il faut qu'il soit relatif à une durée initiale. Les durées maximales mesurées pour cette application sont de 26 minutes pour Google Maps et de 18 minutes pour l'algorithme de Dijkstra, cet écart est donc important puisqu'il représente 55% de la durée maximum obtenue avec l'algorithme de Dijkstra. Le fait d'avoir seulement 12 valeurs de durée peut expliquer l'obtention d'un écart aussi important à l'échelle des durées manipulées. Afin d'obtenir un modèle plus fin il faudrait donc développer la base de données des durées.

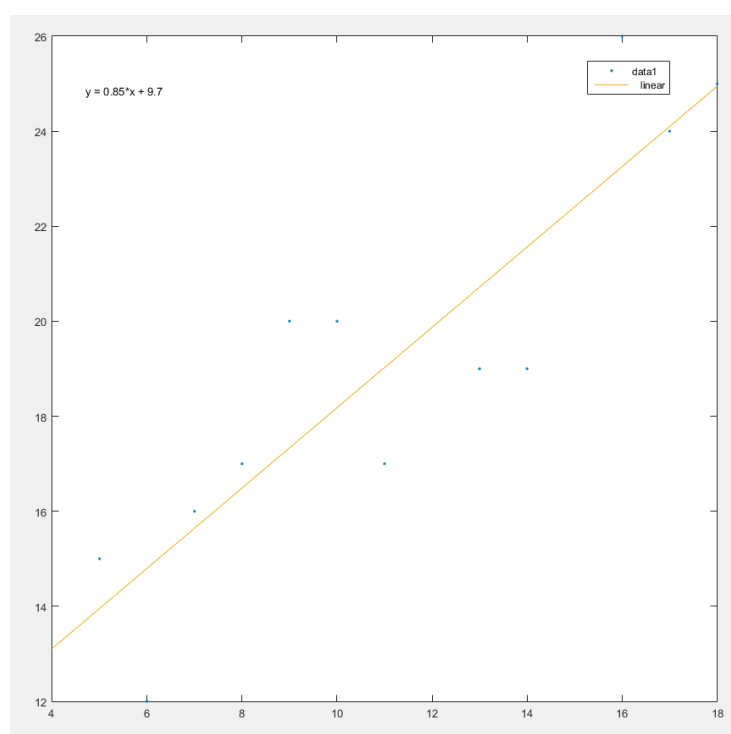


Figure 12 : Comparaison entre les durées obtenues avec l'algorithme de Dijkstra et l'application Google Maps

De même, toujours à partir du tableau précédent, la comparaison entre les durées obtenues avec l'algorithme de Dijkstra et celles obtenues avec l'application Waze est présentée avec la figure ci-dessous. Cette régression linéaire présente un coefficient de détermination $R^2 = 0,63$, ce coefficient est lui aussi satisfaisant. De même, on obtient l'écart théorique de 12 minutes. Cet écart est encore plus important et représente 66 % de la durée maximal obtenue avec l'algorithme de Dijkstra. Là encore, le

faible nombre de données de comparaison met en question la validité de l'équation de la droite obtenue, cependant le fort coefficient de détermination permet de supposer qu'avec plus de valeur la tendance restera la même.

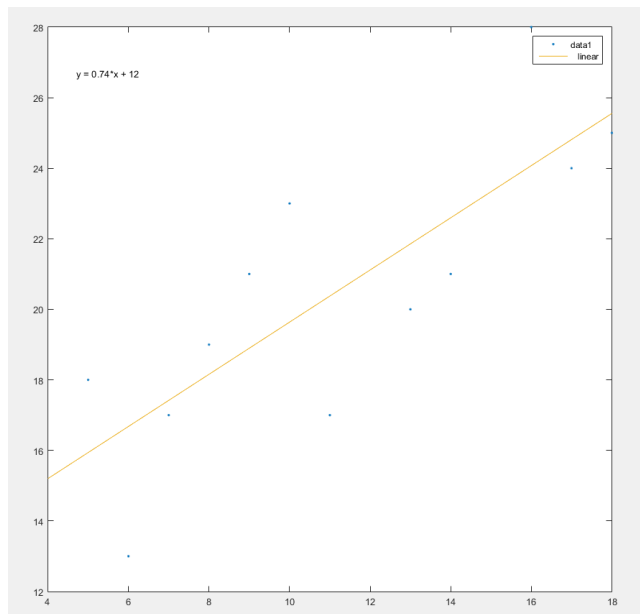


Figure 13 : Comparaison entre les durées obtenues avec l'algorithme de Dijkstra et l'application Waze

La comparaison en durée a donc été réalisée deux à deux avec l'algorithme de Dijkstra, il faut désormais à comparer les durées obtenues avec les deux applications. Pour cela, une autre régression linéaire a été réalisée. Dans ce cas, afin de corriger la limite évoquée précédemment, ce sont les 65 valeurs de la base de données qui ont été utilisées. La distribution semble encore une fois suivre une régression linéaire. Ici, Le coefficient de détermination $R^2 = 0,95$, qui est tout à fait satisfaisant. Cela s'explique notamment par le fait que la distribution est plus importante. De plus l'écart théorique est seulement d'une minute. La distribution comprenant des durées sur une plage plus élevée, l'écart théorique est donc plus affiné. Cela nous permet de conclure que les applications Waze et Google Maps donnent des durées proches. De plus le coefficient direct de cette droite est proche de 1, donc pour chaque augmentation d'une minute sur l'application Google Maps une augmentation de quasiment d'une minute est observable sur l'application Waze. Cela montre que même si les algorithmes de ces applications ne fonctionnent pas tout à fait de la même manière, les résultats obtenus sont très similaires.

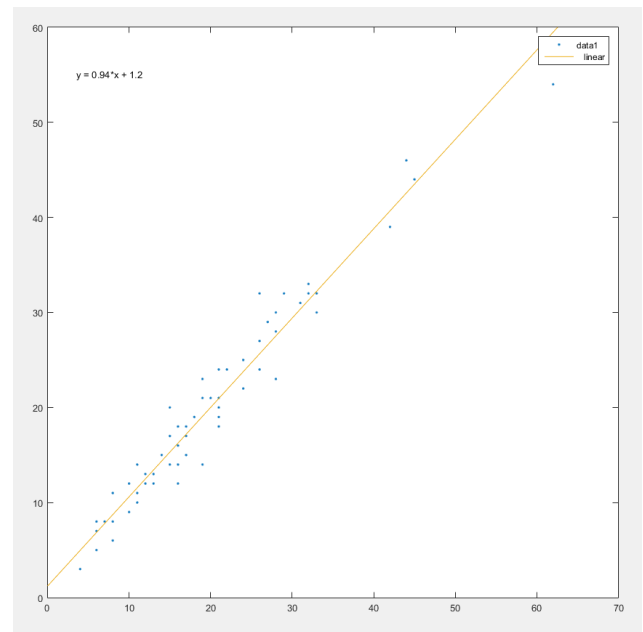


Figure 14 : Comparaison entre les durées obtenues avec l'application Google Maps et l'application Waze

Cette étude montre qu'il existe des écarts importants lorsque l'on compare la durée obtenue avec l'algorithme de Dijkstra par rapport à celle des applications. Cela permet, dans un premier temps, de valider l'hypothèse émise en préambule qui était le fait que les applications n'utilisent pas uniquement un algorithme de plus court chemin. Les durées de déplacement étant plus longue avec les applications on peut en déduire que ces dernières prennent d'autres éléments en compte par rapport à l'algorithme de Dijkstra. Cette étude montre aussi que les écarts, bien que différents entre chaque modèle, sont tout de même très importants lors des comparaisons avec l'algorithme de Dijkstra. L'écart de durée entre les méthodes est 10 fois moins important lorsque l'on compare l'application Waze à l'algorithme de Dijkstra par rapport à la comparaison entre l'application Waze et l'application Google Maps. La différence entre les écarts théoriques peut être expliquée par la faible taille de l'échantillon de comparaison avec l'algorithme de Dijkstra. Cela n'explique cependant pas l'importance de cet écart qui est donc dû à la prise en compte d'éléments extérieurs pour les applications Google Maps et Waze.

Cette première partie permet donc de conclure qu'il existe des écarts de durée entre les différents modèles de calcul d'une durée. Tout d'abord, la différence entre les durées est plus forte en proportion lorsque les durées sont faibles. Cela s'explique « simplement » par le fait que pour une longue durée un choix différents entre les applications

va être « lissé ». Cette étude basée sur des exemples permet de déterminer une relation entre les différentes méthodes de calcul d'une durée. Ces relations issues du modèle empirique sont généralisables pour tous cas d'étude. Nous pouvons aussi affirmer que les applications Google web Maps et Waze fonctionnent de façon assez similaire puisque les écarts entre les durées sont faibles. L'algorithme de Dijkstra quant à lui donnent des durées plus faibles et un écart important 55 % à 66 % de sa valeur maximale par rapport aux applications Google Maps et Waze. La question qui se pose désormais est de savoir si les divergences observées dans cette partie sont observables à l'échelle des flux. Il est désormais question d'étudier les écarts entre les flux selon la durée initiale utilisées.

4.2. Travail sur les flux

Pour la suite de ce projet nous utiliserons les mêmes bases de données. Il est désormais question des flux. Dans l'état de l'art, il a été montré qu'il existe un modèle permettant de relier la durée au flux. Ce modèle a été simplifié puisqu'il est seulement question ici de comprendre l'impact que peut avoir la durée sur les flux. Avant toute chose, il est tout de même question de revenir sur le modèle de calcul des flux.

4.2.1. Avant-propos

La formule utilisée dans cette partie est la suivante : $t_{ij} = D_{ij}^{-\beta}$. Les valeurs de D_{ij} sont explicitées dans les bases de données et sont des durées comprises entre 1 minutes et 70 minutes, cependant la variable β reste une inconnue. Cette variable représente le frein au déplacement ou degrés de motivation. Nous allons donc observer l'impact de cette variable sur les flux. Pour cela, il faut fixer une durée et observer l'effet des variations du flux vis-à-vis de l'évolution du frein au déplacement. Les durées obtenues avec les différentes applications étant comprises entre 1 minutes et 70 minutes, la durée fixée pour déterminer l'impact de la variable β sera comprise entre ces deux valeurs. La réalisation d'une étude de sensibilité de la variable β procure le résultat suivant.

$$\frac{\delta F_r}{\delta} = -1, \text{ Card}(S) = 1$$

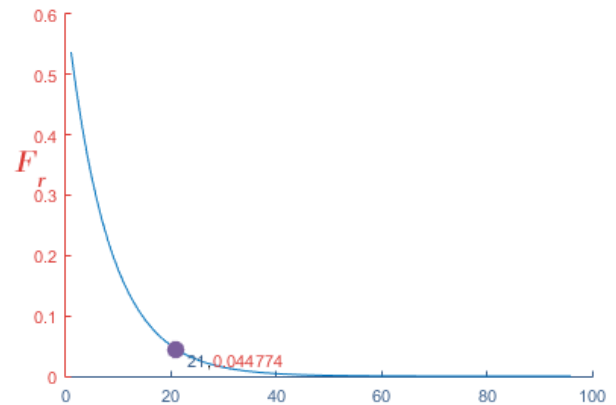


Figure 15 : Étude de sensibilité de la variable β

L'axe des abscisses représente le nombre de valeurs que peut prendre β . L'axe des ordonnées indique les flux obtenus après le calcul avec une durée fixée. Pour cette étude de sensibilité les valeurs que peut prendre β sont comprises entre 0,2 et 5. Ce graphique montre que plus la valeur de β plus les flux obtenus sont proches de 0. Cela s'explique au travers de la formule. En effet la fonction qui décrit le flux à partir d'une durée est une fonction inverse. De plus le β est en exposant pour le dénominateur, donc plus β est grand, plus le numérateur est grand. Ainsi, plus β est grand, plus les flux obtenus seront proches de 0. C'est ici l'allure générale que l'on observe et non les valeurs des flux. La formule ayant été modifiée afin de seulement répondre à la problématique, les valeurs sont donc toujours proches de 0.

Afin de déterminer la valeur optimale de β on cherche à déterminer l'optimum de cette courbe, ici à 1,11. Cependant, selon la durée utilisée la forme de la courbe ne change pas mais la valeur optimale du β change. Dans le but de déterminer une valeur de β significative pour toutes les durées, cette étude de sensibilité doit être approfondie. Après avoir calculé une valeur de β pour plusieurs durées, on obtient un coefficient de variation de 10 % ([Annexe 4](#)). La valeur moyenne peut donc être retenue pour la suite de l'étude. La suite de cette étude sera donc réalisée pour un $\beta = 1,1$.

4.2.2. Première approche

Lorsque l'on poursuit l'étude de la fonction $t_{ij} = D_{ij}^{-\beta}$ pour un β fixé, la fonction inverse impose que plus la durée est grande plus les flux seront proches de 0. Ce sont donc les durées les plus faibles qui vont engendrer les plus grands flux. Lorsque l'on trace les flux obtenus avec les durées de Google Maps en fonction des durées de Google Maps, on remarque que l'impact de la distance est exponentiel. Les faibles durées entraîneront des flux plus importants.

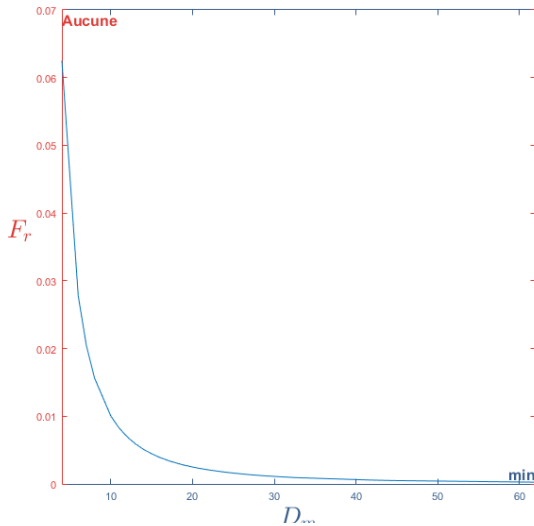


Figure 16 : Flux en fonction de la durée de Google Maps

L'objectif de cette partie est d'observer si les écarts de durées entraînent des écarts dans les flux. La suite de cette étude sera donc consacrée à l'étude de l'impact de la durée sur les flux.

4.2.3. Etude de la fonction du calcul des flux

L'objectif de cette dernière partie est de montrer à quel point la durée a un impact sur les flux. L'état de l'art nous permet de connaître la fonction mathématique qui lie la durée au flux. Nous n'aurons donc pas contrairement à la partie précédente, une approche empirique.

La durée d'un déplacement et les flux sont liés par la formule suivante :

$$f : \text{duree} \rightarrow \text{flux}$$

$$x \mapsto \frac{1}{x^\beta}$$

avec $x > 0$

Connaître l'impact de la durée sur les flux revient à observer pour un petit changement de durée la manière dont le flux change. Cette reformulation de la problématique nous conduit donc à l'étude de la dérivée de f . Cette dernière étant définie de la façon suivante :

$$f' : \text{duree} \rightarrow \text{flux}$$

$$x \mapsto \frac{-\beta}{x^{\beta+1}}$$

Après calcul de la dérivée on obtient la courbe suivante :

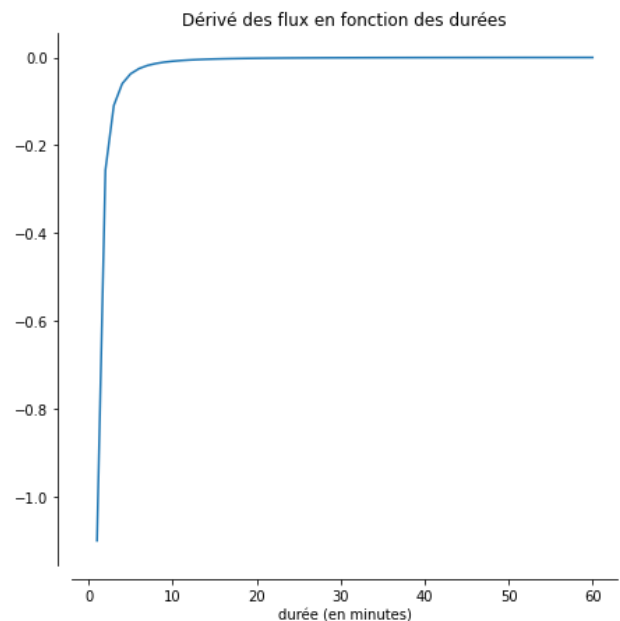


Figure 17 : Dérivé des flux en fonction de la durée

En complément du phénomène d'écrit plus tôt, plus la durée est grande plus le flux est faible, cette dérivée permet de dire que plus la durée est grande plus l'impact qu'il a sur les flux est minime. Cette fonction croît rapidement avant de se stabiliser aux alentours de 0. On en déduit donc que l'impact est extrêmement important jusqu'à une certaine valeur à partir de laquelle l'impact de l'augmentation de la durée sur les flux est atténué. Cette valeur est de 6 minutes. Ce sont donc les durées en dessous de 6 minutes qui ont le plus d'impact sur les flux. Pour en revenir à la problématique, la question on se posait sur l'impact des différences entre les méthodes par

rapport au calcul des flux. Ayant déterminé dans la première partie les écarts théoriques entre les différents modèles de calcul de durée, il est possible d'accéder à plusieurs conclusions. Tout d'abord, une des conclusions de la première partie est fait que l'algorithme de Dijkstra fournis des durées moins importantes que les deux applications. Ces écarts étaient à la hauteur d'une dizaine de minutes. Compte tenu du fait qu'au-delà de 6 minutes les durées ont un impact faible sur les flux, la taille des écarts permet de dire que lorsque les durées obtenues avec l'algorithme de Dijkstra ont un impact fort sur les flux ce n'est pas le cas pour les durées obtenus avec les deux applications. Même si la durée obtenue avec Dijkstra est proche de 1 (impact très élevé sur les flux), la durée sera donc aux alentours de 10 et n'aura qu'un faible impact. Les deux applications, quant à elle, étaient très proches en termes de durée avec un écart théorique de l'ordre de la minute. Ces deux applications ont donc toujours le même effet. Lorsque les durées seront grandes l'impact sur les flux sera faible et lorsque les durées seront inférieures à 6 minutes les effets seront c'est important.

5. Conclusion

Cette étude a permis de mettre en place des fonctions permettant de relier les durées obtenues avec l'algorithme de Dijkstra, celle obtenue avec l'application Google Maps et celle obtenue avec l'application Waze. Ses fonctions ont été déterminées grâce à un modèle empirique, c'est-à-dire à partir d'exemples, elles sont donc généralisables. Cependant la quantité de valeur présente pour les comparaisons avec l'algorithme de Dijkstra reste faible. Le modèle supposé linéaire reste valable, les coefficients de régression étant satisfaisant, mais pourrait être améliorée avec un nombre de valeurs plus conséquent.

De manière générique, en proportion, plus la durée de trajet est élevée, plus la différence entre les méthodes s'estompe. En effet, cela s'explique par le lissage des différents choix des algorithmes pour des trajets long. Pour une courte durée, un paramètre pris en compte différemment est difficilement rattrapable puisque peut de choix sont faisable lorsque l'on s'engage sur un court chemin.

L'étude des applications montre tout de même que Google Maps et Waze fonction de façon très similaire. Même s'il existe de nombreux exemples, comme ceux présentés en introduction pour lesquelles les résultats qu'ils fournissent sont différents, le modèle obtenu montre que leur écart est en fait de seulement 1 minutes. Ces algorithmes fonctionnent de façons semblables et prennent en compte des paramètres en plus par rapport à l'algorithme de Dijkstra. Ainsi les écarts de durées sont de l'ordre de 10 minutes entre ces deux méthodes. Cet écart impose parfois que les durées des applications soient deux fois plus importantes que celles de l'algorithme de Dijkstra.

Concernant les flux, il faut tout d'abord noter que la durée à un effet inverse sur eux, c'est-à-dire que plus la durée est élevée, plus le flux est faible. De plus ces durées n'ont un impact significatif seulement pour les durées faibles. Ainsi au-dessus de 6 minutes de durées de déplacement les durées ont un impact sur les flux peu significatif. Les deux applications fonctionnant de façon similaire, leurs impacts seront toujours les mêmes. Cependant, l'algorithme de Dijkstra fourni des durées pouvant avoir un fort impact sur les flux tandis que l'écart de durée avec les applications pour le même déplacement imposent que ces dernières aient peu d'impact.

Enfin pour conclure vis-à-vis de l'ingénierie du transport, les durées critiques énoncés sont trop faible pour avoir un enjeu dans ce domaine. Ainsi, à l'échelle de grand trajet pour lesquelles les durées se comptent en heure, la différence entre les modèles de calcul de durées n'aura pas d'impact sur les flux. Il est donc cohérent que l'industrie du transport utilise des algorithmes de plus court chemin, à l'image de l'algorithme de Dijkstra, plutôt que des applications. Ces applications qui prennent en compte des éléments de densité de circulation et de perturbation du trafic sont plus adapté aux faibles durées comme pour les trajets domicile-travail puisque l'impact sur les flux y est plus important.

6. Bibliographie

- [1] Backhouse R.C, Van den Eijnde J.P.H.W, Van Gasteren A.J.M, 1994, Calculating path algorithms, www.sciencedirect.com/science/article/pii/0167642394900051, consulté le 04/03/2020
- [2] Dondi R, Mauri G, Zoppis I, 2019, Graph Algorithms, www.sciencedirect.com/science/article/pii/B978012809633820424X, consulté le 04/03/2020
- [3] Goldberga A.V, Radzik T, 1993, A heuristic improvement of the Bellman-Ford algorithm, www.sciencedirect.com/science/article/pii/089396599390022F, consulté le 21/02/2020
- [4] Maizia M, 2014, Réseau, transport et flux, https://b0ad29e6-fb46-404d-8ea1-52cfaa356ac6.filesusr.com/ugd/3f240d_065e8e6d1a844a7ea105b664c10f11ed.pdf, consulté le 13/01/2020

7. Annexe 1 : Code du système logique permettant de comparer les durées obtenues par les algorithmes de Dijkstra et de Bellman-Ford.

```
function [ D_b_f , D_d ] = pfe( R ) ;
% [ D_b_f , D_d ] = pfe( R ) ;
% Calcule les différences entre les modèles de calcul de durée
%
% input (variables exogènes) :
% R : Réseau en Aucune
%
% output (finalités)
% D_b_f : Durée du trajet avec l'algorithme de Bellman-Ford en h
% D_d : Durée du trajet avec l'algorithme de Dijkstra en h
% G : Graphe du réseau en Aucune
% I2J : Liens en Aucune
% L : Longueur des arcs en km
% v : Vitesse sur les arcs en km/h
% ij : Ensemble des origines et des destinations en Aucune
% E : Matrice d'adjacence en Aucune
% C : Matrice des coûts en h
% I : Ensemble ordonné des origines en Aucune
% J : Ensemble ordonné des destinations en Aucune
%
% Exemple :
% % R est une variable à importer d'un fichier spécifique ;
% [ D_b_f , D_d ] = pfe( R ) ;
%
%
% Généré automatiquement par toaster systems - Copyright : Mindjid Maïzia - PolyTech
Tours 2018
% L'auteur n'est nullement responsable des éventuelles erreurs de fonctionnement de
% l'application
% Tout usage des produits toaster doit être signalé par la référence :
% Mindjid Maïzia, toaster integral, tools for model computing, PolyTech Tours, 2016-2022

R = input.R ;
G = F384582199 ( R ) ;
I2J = F229022946 ( R ) ;
L = F549879145 ( G ) ;
v = F450517462 ( G ) ;
ij = F507266724 ( I2J ) ;
E = F638545736 ( G ) ;
C = F492511776 ( L , v ) ;
I = F285175798 ( ij ) ;
J = F659450255 ( ij ) ;
```

```

D_d = F274817223 ( E , C , I , J ) ;
D_b_f = F306412010 ( E , C , I , J ) ;
output.G = G;
output.I2J = I2J;
output.L = L;
output.v = v;
output.ij = ij;
output.E = E;
output.C = C;
output.I = I;
output.J = J;
output.D_b_f = D_b_f;
output.D_d = D_d;

function G = F384582199 ( R )
% Graphe du réseau
    G = R.G ;
end

function I2J = F229022946 ( R )
% Liens
    I2J = R.I2J ;
end

function L = F549879145 ( G )
% Longueur des arcs
    L = G.L ;
end

function v = F450517462 ( G )
% Vitesse sur les arcs
    v = G.K ;
end

function ij = F507266724 ( I2J )
% Ensemble des origines et des destinations
    ij = I2J.ij ;
end

function E = F638545736 ( G )
% Matrice d'adjacence
    E = G.E ;
end

function C = F492511776 ( L , v )
% Matrice des coûts
    C=L./v

```

```

end

function I = F285175798 ( ij )
% Ensemble ordonné des origines
    I=ij(:,1);
end

function J = F659450255 ( ij )
% Ensemble ordonné des destinations
    J=ij(:,2);
end

function D_d = F274817223 ( E , C , I , J )
% Durée du trajet avec l'algorithme de Dijkstra
    E=[E;E(:,[2 1])];
    C=[E,[C;C]];
    a=[];
    for i=1:numel(I)
        [c,p]=dijkstra(E,C,I(i),J(i));
        a =[a;c];
    end

    D_d=[I J a];

end

function D_b_f = F306412010 ( E , C , I , J )
% Durée du trajet avec l'algorithme de Bellman-Ford
    E=[E;E(:,[2 1])];
    C=[E,[C;C]];
    H=sparse(C(:,1)',C(:,2)',C(:,3)');
    a=[];
    for i=1:numel(I)
        [dist, path, pred] = graphshortestpath(H, I(i), J(i), 'Method', 'Bellman-Ford');
        a=[a;dist];
    end

    D_b_f = [I J a];
end

end

% Copyright : Mindjid MAIZIA 2016-2022 , PolyTech Tour

```

8. Annexe 2 : Bases de données des durées et flux associés.

a. Avec Dijkstra

Déplacement	Couleur	Durées avec Dijkstra	Durées avec Google Maps	Durées avec Waze	Flux induit par les durées de Dijkstra	Flux induit par les durées de Google Maps	Flux induit par les durées de Waze
1	Vert	17	24	24	0,044	0,030	0,030
2	Bleu clair	6	12	13	0,139	0,065	0,060
3	Saumon	8	17	19	0,102	0,044	0,039
4	Fuchsia	9	20	21	0,089	0,037	0,035
5	Orange	16	26	28	0,047	0,028	0,026
6	Jaune	18	25	25	0,042	0,029	0,029
7	Violet	13	19	20	0,060	0,039	0,037
8	Cyan	7	16	17	0,118	0,047	0,044
9	Bleu	11	17	17	0,072	0,044	0,044
10	Mauve	5	15	18	0,170	0,051	0,042
11	Vert foncé	10	20	23	0,079	0,037	0,032
12	Rouge	14	19	21	0,055	0,039	0,035

b. Entre Google Maps et Waze

Déplacement	Durées avec Google Maps	Durées avec Waze	Différence entre les durées	Proportion de la différence	Flux avec les durées de Google Maps	Flux avec les durées de Waze	Différence de Flux
1	16	12	4	0,29	0,0474	0,0650	-0,0176
2	32	32	0	0,00	0,0221	0,0221	0,0000
3	28	28	0	0,00	0,0256	0,0256	0,0000
4	12	12	0	0,00	0,0650	0,0650	0,0000
5	24	22	2	0,09	0,0303	0,0334	-0,0030
6	21	20	1	0,05	0,0351	0,0371	-0,0019
7	24	25	-1	0,04	0,0303	0,0290	0,0013

8	21	18	3	0,15	0,0351	0,0416	-0,0065
9	28	23	5	0,20	0,0256	0,0318	-0,0062
10	32	33	-1	0,03	0,0221	0,0214	0,0007
11	45	44	1	0,02	0,0152	0,0156	-0,0004
12	8	11	-3	0,32	0,1015	0,0715	0,0300
13	12	13	-1	0,08	0,0650	0,0595	0,0055
14	15	20	-5	0,29	0,0509	0,0371	0,0138
15	19	23	-4	0,19	0,0392	0,0318	0,0074
16	14	15	-1	0,07	0,0549	0,0509	0,0040
17	21	19	2	0,10	0,0351	0,0392	-0,0041
18	31	31	0	0,00	0,0229	0,0229	0,0000
19	19	14	5	0,30	0,0392	0,0549	-0,0157
20	26	27	-1	0,04	0,0278	0,0266	0,0011
21	7	8	-1	0,13	0,1176	0,1015	0,0161
22	18	19	-1	0,05	0,0416	0,0392	0,0024
23	28	30	-2	0,07	0,0256	0,0237	0,0019
24	16	14	2	0,13	0,0474	0,0549	-0,0075
25	4	3	1	0,29	0,2176	0,2987	-0,0810
26	13	12	1	0,08	0,0595	0,0650	-0,0055
27	13	12	1	0,08	0,0595	0,0650	-0,0055
28	15	14	1	0,07	0,0509	0,0549	-0,0040
29	27	29	-2	0,07	0,0266	0,0246	0,0020
30	10	12	-2	0,18	0,0794	0,0650	0,0144
31	42	39	3	0,07	0,0164	0,0178	-0,0014
32	26	32	-6	0,21	0,0278	0,0221	0,0057
33	11	11	0	0,00	0,0715	0,0715	0,0000
34	17	18	-1	0,06	0,0443	0,0416	0,0027
35	17	17	0	0,00	0,0443	0,0443	0,0000
36	21	21	0	0,00	0,0351	0,0351	0,0000
37	16	16	0	0,00	0,0474	0,0474	0,0000
38	17	15	2	0,13	0,0443	0,0509	-0,0065

39	10	9	1	0,11	0,0794	0,0892	-0,0098
40	29	32	-3	0,10	0,0246	0,0221	0,0025
41	8	8	0	0,00	0,1015	0,1015	0,0000
42	8	8	0	0,00	0,1015	0,1015	0,0000
43	21	24	-3	0,13	0,0351	0,0303	0,0048
44	24	25	-1	0,04	0,0303	0,0290	0,0013
45	20	21	-1	0,05	0,0371	0,0351	0,0019
46	16	18	-2	0,12	0,0474	0,0416	0,0058
47	22	24	-2	0,09	0,0334	0,0303	0,0030
48	8	6	2	0,29	0,1015	0,1393	-0,0378
49	15	14	1	0,07	0,0509	0,0549	-0,0040
50	6	8	-2	0,29	0,1393	0,1015	0,0378
51	13	13	0	0,00	0,0595	0,0595	0,0000
52	6	5	1	0,18	0,1393	0,1703	-0,0309
53	15	17	-2	0,13	0,0509	0,0443	0,0065
54	26	24	2	0,08	0,0278	0,0303	-0,0026
55	33	32	1	0,03	0,0214	0,0221	-0,0007
56	11	14	-3	0,24	0,0715	0,0549	0,0167
57	17	15	2	0,13	0,0443	0,0509	-0,0065
58	6	7	-1	0,15	0,1393	0,1176	0,0217
59	19	21	-2	0,10	0,0392	0,0351	0,0041
60	11	10	1	0,10	0,0715	0,0794	-0,0079
61	24	25	-1	0,04	0,0303	0,0290	0,0013
62	33	30	3	0,10	0,0214	0,0237	-0,0024
63	62	54	8	0,14	0,0107	0,0124	-0,0018
64	44	46	-2	0,04	0,0156	0,0148	0,0007

9. Annexe 3 : Création des plus courts chemins avec l'algorithme de Dijkstra

```
function [ fg ] = drawcapacity( R ) ;
% [ fg ] = drawcapacity( R ) ;
% Calcul et dessine la capacité des arcs d'un réseau
%
% input (variables exogènes) :
% R : Définition géométrique du réseau en Aucune
%
% output (finalités)
% fg : Figure représentant la capacité des arcs en Aucune
% G : Graphe représentant le réseau R en Aucune
% I2J : Propriétés des liaisons relatives aux flux en Aucune
% l : Longueur des arcs de G en km
% v : Vitesse pratiquée sur chaque arc de G en km/h
% ij : Ensemble des couples "Origine-Destination" en Aucune
% E : Arcs de G en Aucune
% I_f : Ensemble ordonné des origines des flux en Aucune
% J_f : Ensemble ordonné des destinations des flux en Aucune
% C : Valuation des arcs de G en h
% C_e_p : Charge instantanée sur les arcs de G en q/h
% K_e : Capacité des arcs de G en q/h
% V : Sommet ou nœud de G en Aucune
%
% Exemple :
% % R est une variable à importer d'un fichier spécifique;
% [ fg ] = drawcapacity( R ) ;
%
%
% Généré automatiquement par toaster systems - Copyright : Mindjid Maïzia - PolyTech
Tours 2018
% L'auteur n'est nullement responsable des éventuelles erreurs de fonctionnement de
% l'application
% Tout usage des produits toaster doit être signalé par la référence :
% Mindjid Maïzia, toaster integral, tools for model computing, PolyTech Tours, 2016-2022

R = input.R ;
G = F468206154 ( R ) ;
I2J = F514534421 ( R ) ;
l = F874424628 ( G ) ;
v = F290516522 ( G ) ;
ij = F881683175 ( I2J ) ;
E = F6376410 ( G ) ;
I_f = F446945900 ( ij ) ;
J_f = F819058429 ( ij ) ;
```

```

C = F893455595 ( l , v ) ;
C_e_p = F409787087 ( E , I_f , J_f , C ) ;
K_e = F751684027 ( E , C_e_p ) ;
V = F142211972 ( G ) ;
fg = F672650514 ( E , I_f , J_f , K_e , V ) ;
output.G = G;
output.I2J = I2J;
output.l = l;
output.v = v;
output.ij = ij;
output.E = E;
output.I_f = I_f;
output.J_f = J_f;
output.C = C;
output.C_e_p = C_e_p;
output.K_e = K_e;
output.V = V;
output.fg = fg;

```

```

function G = F468206154 ( R )
% Graphe représentant le réseau R
    G = R.G ;
end

```

```

function I2J = F514534421 ( R )
% Propriétés des liaisons relatives aux flux
    I2J = R.I2J ;
end

```

```

function l = F874424628 ( G )
% Longueur des arcs de G
    l = G.L ;
end

```

```

function v = F290516522 ( G )
% Vitesse pratiquée sur chaque arc de G
    v = G.K ;
end

```

```

function ij = F881683175 ( I2J )
% Ensemble des couples "Origine-Destination"
    ij = I2J.ij ;
end

```

```

function E = F6376410 ( G )

```

```

% Arcs de G
    E = G.E ;
end

function I_f = F446945900 ( ij )
% Ensemble ordonné des origines des flux
    I_f=ij(:,1);
end

function J_f = F819058429 ( ij )
% Ensemble ordonné des destinations des flux
    J_f=ij(:,2);
end

function C = F893455595 ( l , v )
% Valuation des arcs de G
    C=1./v;
end

function C_e_p = F409787087 ( E , I_f , J_f , C )
% Appartenance des arcs à un déplacement
    fig=figure('WindowStyle','modal');
    axes;
    txt=text(0,0,'Patienter','HorizontalAlignment','center','FontSize',14);
    axis off, xlim([-1 1]), ylim([-1 1]), drawnow

    E=[E;E(:,[2 1])]; % On rend symétrique le graphe si nécessaire
    C=[E,[C;C]]; % On réalise la concaténation correspondante
    D=[];
    C_e_p=zeros(size(E,1)+1,size(I_f,1));
    for f=1:size(I_f,1)
        [c,p]=dijkstra(E,C,I_f(f),J_f(f));
        set(txt,'string',['Patienter dijkstra du flux ' num2str(f) ' sur '
num2str(size(I_f,1))])
        drawnow
        D=[D c];
        if ~isempty(p)
            e=[p(1:end-1);p(2:end)]';
            k=ismember(E,e,'rows');
            C_e_p(k,f)=f;
        end
    end
    delete(fig)
    C_e_p(end,:)= D;
end

```

```

function K_e = F751684027 ( E , C_e_p )
    % Appartenance des arcs à un déplacement et nombre de déplacements auxquels ils
    appartiennent

    C_e_p(end,:)=[]
    K_e = max(C_e_p,[],2);
    l=C_e_p==0;
    lw=sum(l-1,2);
    id=K_e==0;
    K_e=[E;E(:,[2 1])] K_e lw];
    K_e(id,:)=[];
end

function V = F142211972 ( G )
    % Sommet ou nœud de G
    V = G.V ;
end

function fg = F672650514 ( E , I_f , J_f , K_e , V )
    % Figure représentant les déplacements sur les arcs
    e=K_e(:,[1 2]);
    k=K_e(:,3);
    l=K_e(:,end);
    lw=5*l/max(l)

    fg = figure('WindowStyle','modal','color','w');

    patch('vertices',V,'faces',E,'EdgeColor',[0.6 0.6 0.6])
    for p=1:size(e,1)
        if k(p)==1
            patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor','g');
        elseif k(p)==2
            patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[0
191/255 1]);
        elseif k(p)==3
            patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[240/255
128/255 128/255]);
        elseif k(p)==4
            patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[199/255
21/255 133/255]);
        elseif k(p)==5
            patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[1
165/255 0]);
        elseif k(p)==6
            patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[1
215/255 0]);
        elseif k(p)==7
            patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[147/255
112/255 219/255]);
        elseif k(p)==8

```

```

        patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[0 1
1]);
    elseif k(p)==9
        patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor','b');
    elseif k(p)==10
        patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[221/255
160/255 221/255]);
    elseif k(p)==11
        patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor',[0
128/255 0]);
    else
        patch('vertices',V,'faces',e(p,:), 'LineWidth',lw(p), 'EdgeColor','r');
    end;
end;
patch('vertices',V,'faces',I_f, 'EdgeColor','none', 'Marker','o',...
'FaceColor','none', 'MarkerFaceColor','b', 'MarkerSize',12)
patch('vertices',V,'faces',J_f, 'EdgeColor','none', 'Marker','o',...
'FaceColor','none', 'MarkerFaceColor','r', 'MarkerSize',12)
axis image off
zoom on
fg=inf;
end

end

% Copyright : Mindjid MAIZIA 2016-2022 , PolyTech Tour

```

Published with MATLAB® R2015a

10. Annexe 4 : Étude des variations de β

Durées en entrée	Valeurs optimales de β				
27	1,11				
12	1,256				
25	1,112				
14	1,208				
68	0,988				
23	1,06	Ecart type	0,1128		
32	1,06	Moyenne	1,1023		
57	1,06				
37	1,06	Coefficient de variation	0,1023	Variation en %	10,23
55	1,016				
52	1,016				
17	1,16				
43	1,016				
60	1,016				
8	1,3				
61	1,016				
47	1,016				
20	1,16				
7	1,4				
48	1,016				