TIPE Sur l'optimisation du traffic routier.

MOULAT Mathéo
NODET Gaëtan
DURAND Ulysse
MARGUET Emeric

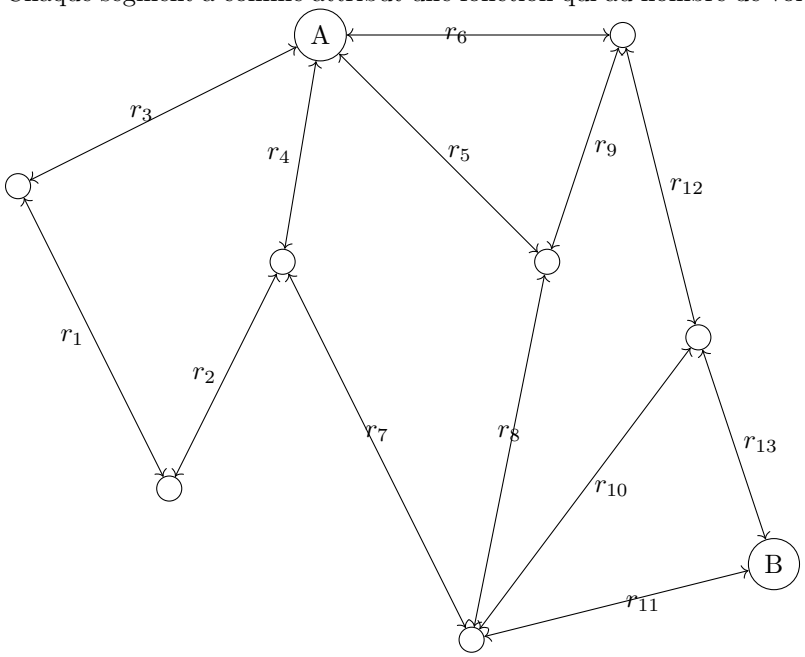
December 17, 2020

1 Abstract

Notre but est de proposer un outil de fluidification du traffic routier en supposant que l'on peut imposer un chemin à suivre à chaque véhicule.

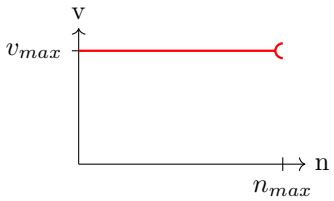
2 Modèle 1

Le réseau routier sera un graphe où les sommets sont des croisements et les arrêtes des portions de route.
Le graphe est orienté et une route à double sens est représentée par deux arrêtes distinctes.
Les voitures vont toutes d'un sommet A du graphe à un autre sommet B.
Pour une voiture, le temps de parcours d'un croisement est nul.
Le régime est permanent, c'est à dire qu'on considère un nombre fixe de voitures présentes a chaque instant sur le graphe.
Un ensemble des voitures sera modélisé par un flux (on pourra donc parler de *4,3* voitures sur la portion de route entre les sommets X et Y), ce nombre sera alors constant au cours du temps.
L'objectif est de minimiser le temps de parcours maximal.
Chaque segment a comme attribut une fonction qui au nombre de voiture sur le segment associe leur vitesse.



2.1 Modeles

2.1.1



Ce modèle très simpliste considère que les voitures roulent a vitesse constante (v_{max}) jusqu'à un nombre maximum de voitures pouvant utiliser la route en même temps (n_{max}). Peu réaliste, il prend tout de même en compte qu'une route ne peut pas accueillir un nombre illimité de voitures. De plus, il permet une implémentation relativement simple de l'algorithme de résolution.

Algorithme de résolution : On utilisera l'algorithme de Dijkstra légerement modifié.
But de l'algorithme de Dijkstra : résoudre le problème de plus court chemin. Ce problème est de trouver le plus court chemin dans un graphe pondéré orienté.

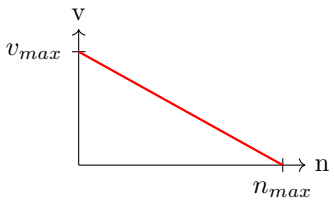
Algorithme 7: Algorithme de Dijkstra

Données : Un graphe orienté pondéré $G = (X, A, W)$ et un sommet $s \in X$
Résultat : Le plus court chemin de s vers tous les autres sommets de G

```
// V : Tableau stockant les étiquettes des sommets de G
Initialiser V à +∞
V[s] = 0
// P : Tableau permettant de retrouver la composition des chemins
Initialiser P à 0
P[s] = s
répéter
    // Recherche du sommet x non fixé de plus petite étiquette
    V_min = +∞
    pour y allant de 1 à N faire
        si y non marqué et V[y] < V_min alors
            x ← y
            V_min ← V[y]
    // Mise à jour des successeurs non fixés de x
    si V_min < +∞ alors
        Marquer x
        pour tout successeur y de x faire
            si y non marqué et V[x] + W[x, y] < V[y] alors
                V[y] = V[x] + W[x, y]
                P[y] = x
jusqu'à V_min = +∞
```

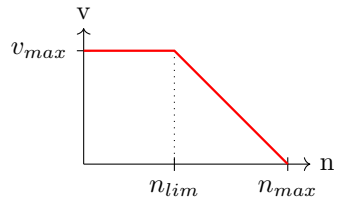
On sort de Dijkstra le plus court chemin de A á B puis si celui ci est plein on peut réappliquer ce même programme pour trouver le suivant. C'est donc fold fulkerson.

2.1.2



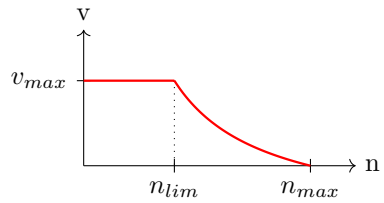
Ce modèle permet de représenter la décroissance de vitesse, avec l'augmentation du nombre de voitures. La vitesse initiale est v_{max}, et elle atteint 0 en n_{max}. Cependant, la vitesse décroît dès la première voiture, ce qui n'est pas très réaliste. Ce modèle permet aussi une implémentation assez simple.

2.1.3



Combinaison des deux précédents, ce modèle considère une vitesse constante (v_{max}) jusqu'à un nombre de voitures limite (n_{lim}) qui déclenche une diminution de la vitesse. Celle ci décroît jusqu'à atteindre 0 pour le nombre maximum de voitures que peut atteindre la route (n_{max}). Ce modèle est bien plus réaliste, même si la décroissance linéaire reste très approximative.

2.1.4



2.2 Expression de la fonction d'évaluation

voitures indicées $1, ..., i, ...N$
routes empruntées indicées $1, ..., l, ..., R$
chemins indicés $1, ..., k, ..., C$
temps de parcours $T_1, ..., T_k, ..., T_R$
nombre de voitures sur le chemin k $\lambda_k = \sum_{i=1}^{N} \delta_{c(i),k}$
avec c(i) l'indice du chemin emprunté par la voiture i
$\chi_{l,k} = 1$ si la route l est dans le chemin k et 0 sinon
T(k) : temps de parcours du chemin k $= \sum_{l=1}^{R} T_l \chi_l(k)$

2.3 resolution par descente de gradiant

pour ce qui est de la résolution pour des fonctions de vitesse compliquées, on se rammène à minimiser la fonction qui a une repartition sur les différents chemins associe le temps de parcours de la plus lente des voitures.
Considérons les chemins $c_1, \cdots, c_n$ dans un certain ordre et considérons en entrée de notre fonction d'évaluation la répartition des voitures $\lambda_1, \lambda_2, \cdots, \lambda_n \in [0, 1]$, avec $\sum_{i=1}^{n} \lambda_i = 1$. Si $\lambda_i = 0.2$ on dirige 20% des voitures sur le chemin i.