TIPE Sur l'optimisation du trafic routier.

DURAND Ulysse

MPX-Lycée Blaise Pascal

2020-2021

Sommaire

1. Enoncé du problème
2. Modélisation - structure de donnée
3. Résolution approchée : se rammener à minimiser une fonction
4. Résolution dans un cas particulier : les graphes série parallèle

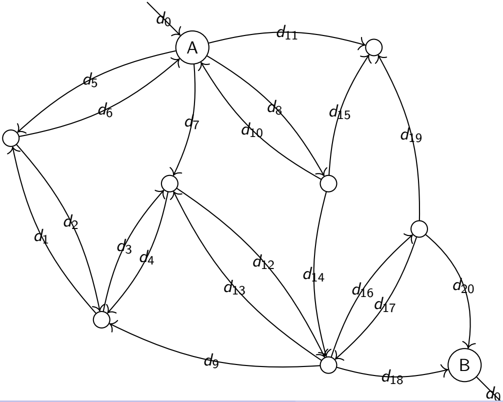
Le problème

On souhaite fluidifier un trafic routier

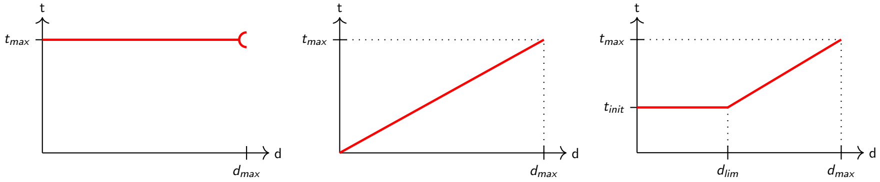
- On a un certain débit de voitures à acheminer d'un point A à un point B
- On peut agir sur le chemin pris par les voitures
- On souhaite réduire le temps de trajet moyen des voitures
- Plus le déit de voitures sur une route est important plus le temps nécessaire pour parcourir la route est élevé
-

Modèle - Structure de donnée

- Le réseau routier : un graphe orienté pondéré.
- Une route : une arète
- Une intersection : un noeud
- Modèle de ralentissement : une fonction du temps de parcours de la route en fonction du débit sur la route. (fonction de ralentissement)



Quelques fonctions de ralentissement



Expression du temps de parcours moyen total

Du noeud A au noeud B il y a n chemins, $c_1, ..., c_n$

Le flux total, de 1, est répartit en flux sur chaque chemin ($d_1, d_2, ..., d_n$ avec $\sum_{i=1}^{n} d_i = 1$)

f_i la fonction de ralentissement de la route r_i Les arètes disposent aussi d'un débit passant ($D_1, ..., D_m$ tels que $D_k = \sum_{i / r_k \in c_i} d_i$)

t_i : temps de parcours du chemin i
Le temps de parcours moyen du graphe est la moyenne pondérée du temps de parcours de chaque chemin par le débit du chemin.

$$T_{tot} = \sum_{i=0}^{n} d_i t_i \text{ et } t_i = \sum_{k / r_k \in c_i} f_k(D_k)$$

Intoduisons $\xi_{i,k} = 1$ si l'arrête k appartient au chemin i et 0 sinon.

$$D_k = \sum_{i=0}^{n} \xi_{i,k} d_i \text{ et } t_i = \sum_{k=0}^{m} \xi_{i,k} f_k(D_k)$$

On a finalement l'expression de la fonction T_{tot} qui a n-1 variables $d_1, ..., d_{n-1}$ ($d_n = 1 - \sum_{i=0}^{n} d_i$)

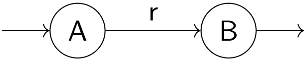
dont on tentera d'approcher le minimum.

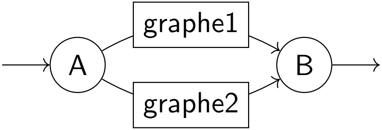
$$T_{tot} = \sum_{i=0}^{n} d_i \sum_{k=0}^{m} \xi_{i,k} f_k\left(\sum_{i=0}^{n} \xi_{i,k} d_i\right)$$

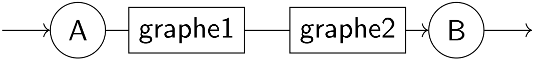
Graphes serie parallele

Un graphe série-parallèle est construit récursivement, de la manière suivante :

C'est soit

-une simple route 

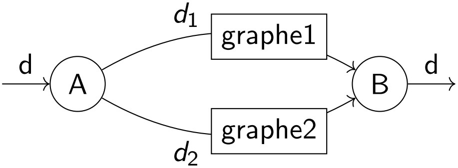
-deux sous graphes en parallèle 

-deux sous graphes en série 

En ocaml :

```
type graph_ser_paral =
     Route of arrete
    |Parallel of (graph_ser_paral*graph_ser_paral)
    |Serie of (graph_ser_paral*graph_ser_paral);;
```

Resolution sur graphe serie parallele



f_1, f_2 : fonctions de ralentissement des graphes 1, 2.

$\widetilde{F}(d, d_1) = \dfrac{d_1}{d} f_1(d_1) + \dfrac{d - d_1}{d} f_2(d - d_1)$: expression du ralentissement du graphe total.

$\dfrac{\partial \widetilde{F}}{\partial d_1}(d, d_1) = \dfrac{1}{d}(f_1(d_1) + d_1 f_1'(d_1) - f_2(d - d_1) + (d - d_1) f_2'(d - d_1)$: utile pour trouver le minimum de $\widetilde{F}$

En appliquant un algorithme de minimisation de fonction, on trouve d_s qui minimise $\widetilde{F}$

On obtient $F(d) = f_1(d_s(d)) + f_2(d - d_s(d))$ la fonction du ralentissement du graphe total et $d_s(d)$ le dACCebit a diriger sur le graphe1