Projet Optimisation dans les Graphes

Plus Court Chemin Robuste

Réalisé par

Zineddine KERMADJ

Oualid CHABANE

Daryl DOKU

Table des matières

1. Partie 1 : Analyse du problème

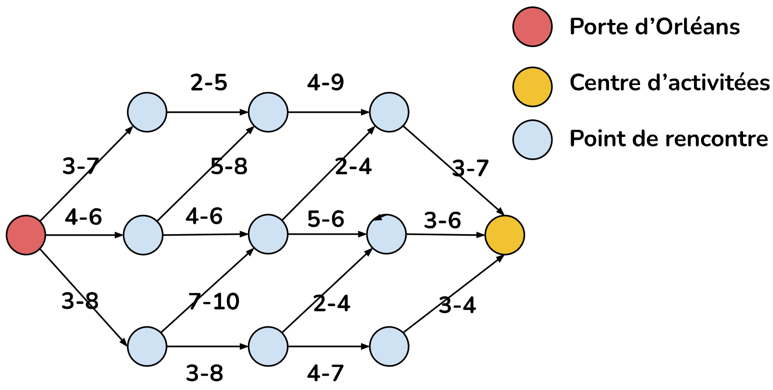

Figure 01: Graphe modélisant les points de rencontres, le point de départ ainsi que le centre d'activité et les durées maximales et minimales de transport entre chaque deux points.

1.1. Spécificités

- Les poids associés aux arêtes du graphe représentent des intervalles de temps exprimés en minutes, notés sous la forme :t_{max} , t_{min} avec $t_{max} > t_{min}$ où :

  t_{min} : correspond au temps de trajet minimal observé sur le tronçon considéré (conditions de circulation favorables),

  t_{max} : correspond au temps de trajet maximal (conditions de circulation défavorables).

- Nous travaillons donc avec des valeurs réelles strictement positives, car les durées de déplacement ne peuvent être ni nulles ni négatives.

Chaque arête est ainsi caractérisée non pas par une seule pondération fixe, mais par un intervalle de valeurs traduisant l'incertitude liée aux conditions de trafic. Cette particularité fait que le graphe n'est plus simplement pondéré, mais pondéré par intervalles, ce qui introduit une dimension de variabilité et de robustesse dans la recherche du chemin optimal.

Ainsi, selon la stratégie adoptée (optimiste, pessimiste, prudente ou stable), l'algorithme prendra en compte respectivement la borne minimale, la borne maximale ou la différence entre ces deux bornes comme coût effectif de chaque arête. Plusieurs notions d'optimalité peuvent alors être envisagées :

- **Itinéraire optimiste** : on suppose que le trafic est fluide sur tous les tronçons et on considère les temps minimaux tmin.
- **Itinéraire pessimiste** : on suppose que les conditions sont défavorables et on considère les temps maximaux t_{max} .
- **Itinéraire prudent** : on cherche un compromis entre les deux extrêmes pour garantir une arrivée sûre.
- **Itinéraire stable** : on cherche à minimiser la **marge de fluctuation** sur le trajet, c'est-à-dire la différence entre les durées maximales et minimales cumulées.

Et bien plus à découvrir !

1.2. Choix Algorithmique

Pour ce projet, on doit trouver le meilleur chemin dans un graphe où chaque trajet a une durée variable (intervalle min-max). Plusieurs approches étaient possibles :

DFS (parcours en profondeur) : Pratique pour lister tous les chemins possibles, mais beaucoup trop lent pour trouver le meilleur itinéraire. Dans le pire des cas, il faudrait tester tous les chemins un par un.

Bellman-Ford : Algorithme robuste qui accepte même les poids négatifs, mais ici c'est du gaspillage. Plus long que certains algorithmes [C = O((V+E) log V), avec V le nombre de sommets et E le nombre d'arêtes] et on n'a pas besoin de sa capacité à gérer les poids négatifs puisque tous nos temps de trajet sont positifs. En plus, dans notre cas, on peut avoir des graphes non dirigés, ce qui est surtout le cas récurrent, alors que l'algorithme de Bellman-Ford ne fonctionnera pas correctement dans cette situation. Les graphs non dirigés avec des cycles négatifs sont extrêmement difficiles à gérer, selon le livre de JeffEricson Algorithms, on peut calculer ce type de chemins en $O(VE + V^2 \log V)$ en temps par une réduction du problème du couplage dans les graphs.

Dijkstra : L'algorithme idéal pour notre cas. Il explore intelligemment le graphe en privilégiant toujours les chemins les plus prometteurs [C = O(V.E) avec V le nombre de sommets et E le nombre d'arêtes]. Rapide et optimal pour des poids positifs même en présence des cycles qui est le cas fréquent dans notre problème.

1.3. Question 1

1.3.1. Nombre d'itinéraires possibles est: 8

1.3.2. Méthode permettant de l'obtenir:

Le nombre de chemins possibles est représenté par le cardinal de l'ensemble

$$\{P \ -> \ s_1 \ -> \ .. \ -> \ s_n \ -> \ C \mid s_i, s_j \in S \ et \ s_i \neq s_j \ et \ (s_i, s_j) \in E \}$$

On peut le déterminer avec un simple parcours en profondeur comme le suivant:

Fonction count_routes

Entrées

g: graphe
start: sommet de départ P
end: sommet d'arrivé C
visited: sommets déjà vus, initialisé à vide initialement

Sorties

Le cardinal de l'ensemble E.

Algorithme

$SI\ start\ =\ end$

$retourner\ 1\ \ //Le\ chemin\ vide$

$SI\ start\ \in\ visited$

$retourner\ 0$ //L'ensemble E ne contient pas de cycles,
//l'ensemble E induit par les nœuds déjà visités est
//vide d'où son cardinal est nulle.

SI

$r\ <-\ 0$

$$FORALL \quad n \in neighbours(start) \quad DO$$
$$r \ <- \ r + count\_routes(g, n, end, visited \cup \{start\})$$
$$retourner\ r$$

On peut démontrer par récurrence sur la taille du chemin que l'algorithme suivant calcule bien le cardinal de l'ensemble énoncé précédemment. L'algorithme fonctionne bien même en présence des cycles grâce à cette condition $s_i \neq s_j$.

1.4. Question 2 (approche pire cas)

1.4.1. Énoncé du problème

On doit chercher le chemin le plus long en termes de borne supérieure du temps mis sur une arête.

1.4.2. Itinéraire dans le pire cas

le chemin est de taille 30min, et qui est le suivant

$$1 \ -> \ 4 \ -> \ 6 \ -> \ 9 \ -> \ 11$$

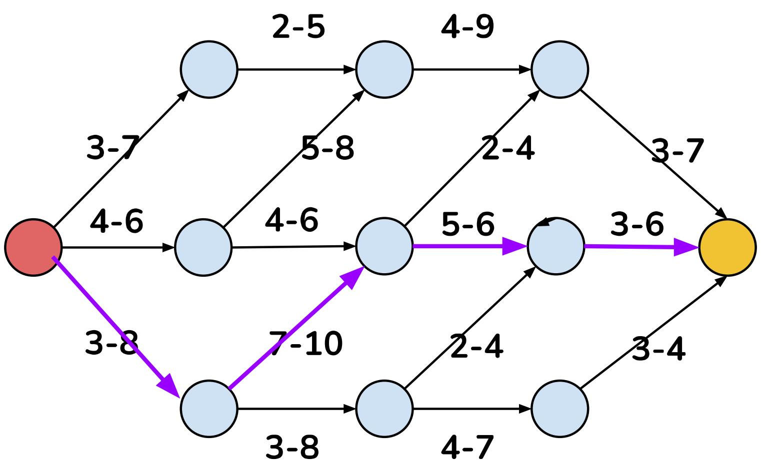

Figure 02: Le chemin le plus adéquat dans le pire cas, on prends le (max, max)

1.5. Question 3 (approche optimiste)

1.5.1. Énoncé du problème

L'approche optimiste consiste à considérer que l'on est dans le meilleur cas possible. Il s'agit donc de trouver le plus court chemin de la Porte d'Orléans (noeud 1) au centre d'activités (noeud 11) en supposant que le parcours de chaque tronçon se fait dans le plus petit temps possible. Autrement dit, en supposant que la valeur de chaque arête est le temps minimal sur cette arête, on recherche le plus court chemin.

1.5.2. Itinéraire optimiste

le chemin est de taille 11min, et qui est le suivant

$$1 \ -> \ 4 \ -> \ 7 \ -> \ 9 \ -> \ 11$$

Figure 03: Le chemin le plus adéquat dans le cas optimiste, on prends le (max, max)

1.5.3. Marge de fluctuation

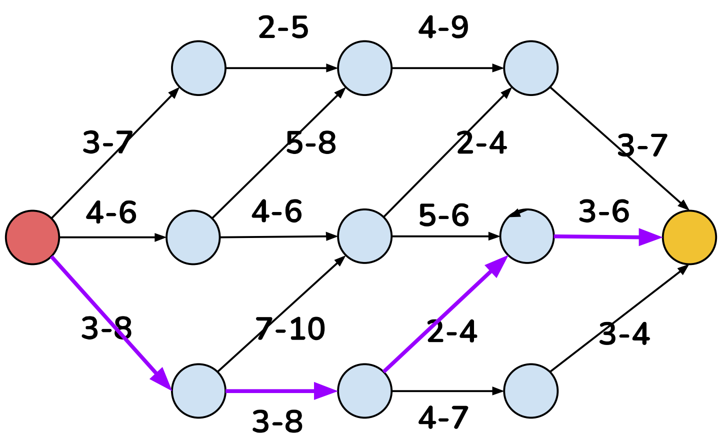

Figure 03: Le chemin le plus adéquat dans le cas optimiste.

$marge \ = \sum d_{max} - \ \sum d_{min} \ = \ 26 \ - \ 11$ donc

$marge \ = \ 15.$

1.6. Question 4 (approche prudente)

1.6.1. Énoncé du problème

L'approche stable consiste à choisir l'itinéraire qui donne le plus court chemin en terms de borne supérieure

1.6.2. Itinéraire prudent

le chemin est de taille **23 min**, et qui est le suivant

$$1 \ -> \ 3 \ -> \ 6 \ -> \ 8 \ -> \ 11$$

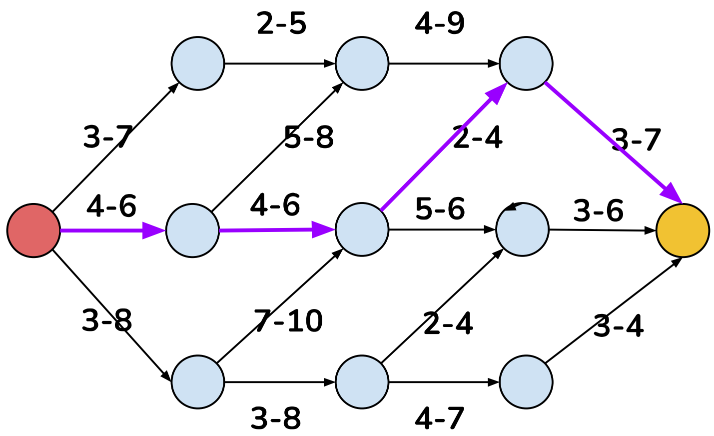

Figure 04: Le chemin le plus adéquat dans le cas prudent

1.7. Question 5 (approche stable)

1.7.1. Énoncé du problème

L'approche stable consiste à choisir l'itinéraire qui donne la plus petite marge de fluctuation en temps de parcours, c'est strictement équivalent à calculer le chemin qui minimise l'amplitude des intervalles de temps sur chaque arête.

1.7.2. Itinéraire stable (et 3- Marge de fluctuation)

la marge de fluctuation est de **8 min,** un temps max de **24 min** et un temps minimale de **16 min**, représenté par le chemin suivant

$$1 \;->\; 3 \;->\; 6 \;->\; 9 \;->\; 11$$

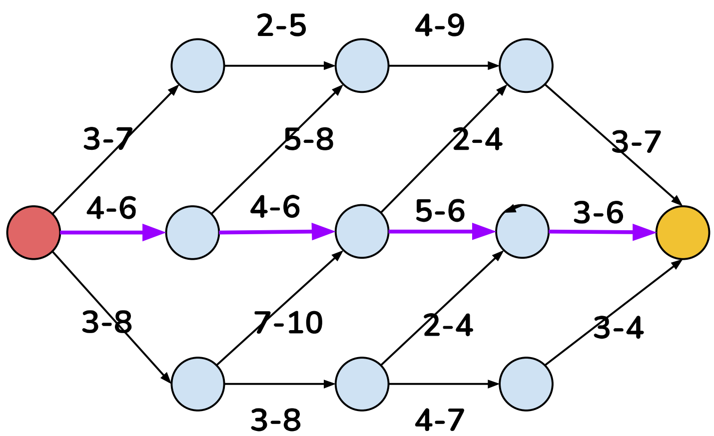

Figure 05: Le chemin le plus adéquat dans le cas de marge de fluctuation

2. Partie 2: Code

Le code est accessible sur ce github.

3. Simulations stochastiques : modélisation probabiliste du trafic

1. Motivation

Les heuristiques déterministes (optimiste, prudente, stable, pire cas) reposent exclusivement sur les bornes de l'intervalle t_{min}, t_{max} . Cette approche présente une limitation majeure : elle ne capture pas la distribution réelle des temps de trajet à l'intérieur de cet intervalle.

En pratique, les temps de parcours ne suivent pas une distribution uniforme entre les bornes, mais présentent des patterns liés aux conditions de circulation. Nous proposons donc deux approches stochastiques permettant d'estimer le comportement moyen des arêtes du graphe par simulation Monte Carlo.

Approche 1 : Distribution gaussienne avec positionnement Beta

- **Principe**

Cette méthode utilise une distribution Beta(α, β) pour positionner le centre μ d'une gaussienne entre tmin et tmax. Le choix de la Beta est motivé par sa flexibilité pour modéliser différents régimes de trafic sur un intervalle borné.

- **Méthodologie**

Pour chaque arête (u, v ,t_{min}, t_{max}), l'algorithme procède comme suit :

1. Sélection aléatoire d'un profil de trafic p ∈ {fluide, normal, dense}
2. Attribution des paramètres Beta correspondants :
   - Fluide : (α, β) = (2, 5) → distribution biaisée vers tmin
   - Normal : (α, β) = (2, 2) → distribution symétrique
   - Dense : (α, β) = (5, 2) → distribution biaisée vers tmax

3. Génération de la position du centre : x ~ Beta(α, β), puis $\mu = t_{min} + x \cdot (t_{max} - t_{min})$
4. Échantillonnage de n = 1000 valeurs selon N(μ, σ²) avec $\sigma = (t_{max} - t_{min}) / (5 + \eta)$, η étant à par défaut mais choisit "arbitrairement"
5. Filtrage des valeurs hors intervalle et calcul de l'estimateur empirique E[T]

- **Limites**

Cette approche suppose un régime de trafic stationnaire sur la période d'observation, ce qui constitue une hypothèse forte dans un contexte urbain réel.

Approche 2 : Processus Beta dynamique

- **Principe**

Cette méthode modélise l'évolution temporelle du trafic comme un processus stochastique paramétré par (αt, βt) variant dans le temps. Elle intègre trois phénomènes observés empiriquement :

- Drift progressif : tendance naturelle à la congestion
- Relâchements périodiques : dissipation cyclique du trafic
- Bruit continu : micro-variations locales

- **Modèle dynamique**

Soit (αt, βt) l'état du système à l'instant t. L'évolution est régie par :

1. Drift (à chaque pas) :

$$\alpha_{t+1} = \alpha_t + \delta (\alpha_{target} - \alpha_t)$$
$$\beta_{t+1} = \beta_t + \delta (\beta_{target} - \beta_t)$$

où $\delta = 0.01 \cdot \lambda_{congestion}$ contrôle la vitesse de convergence

2. Relâchements (tous les τ pas) :

Si t mod τ = 0 :

$$\alpha_t \leftarrow \alpha_t + \gamma \cdot (\alpha_0 - \alpha_t)$$

$\beta_t \leftarrow \beta_t + \gamma \cdot (\beta_0 - \beta_t)$
où $\gamma \in [0, 1]$ contrôle l'intensité du relâchement

3. Bruit gaussien (à chaque pas) :
$\alpha_t \leftarrow \alpha_t + \varepsilon_1, \beta_t \leftarrow \beta_t + \varepsilon_2$ avec $\varepsilon_1, \varepsilon_2 \sim N(0, \eta^2)$

- **Génération des échantillons**

À chaque instant t, on tire $X_t \sim$ Beta(αt, βt) et on calcule :
$T_t = t_{min} + X_t \cdot (t_{max} - t_{min})$

L'estimateur final est la moyenne empirique sur T échantillons.

- **Application au graphe**

Une fois les temps moyens $E[T_{uv}]$ estimés pour chaque arête (u, v), on construit un nouveau graphe G' où chaque arête possède un poids unique $\hat{E}[T_{uv}] \approx E[T_{uv}]$. L'algorithme de Dijkstra est alors appliqué sur G' pour obtenir le chemin optimal selon le critère du temps moyen.

- **Analyse comparative**

Méthode	Réalisme	Hypothèse principale
Bornes (min/max)	Faible	Cas extrêmes déterministes
Gaussienne Beta	Moyen	Régime stationnaire
Processus Beta	Élevé	Dynamique temporelle du trafic

- **Interprétation**

Pour un trajet avec intervalle [10, 30] minutes :

- Heuristique optimiste : 10 min (borne inférieure)

- Heuristique prudente : 30 min (borne supérieure)
- Simulation gaussienne : ≈ 18 min (moyenne statique)
- Simulation Beta dynamique : ≈ 22 min (intègre congestion progressive)

Le processus Beta produit systématiquement des estimations plus conservatives que l'approche gaussienne, car il capture l'accumulation naturelle du trafic et ses variations cycliques. Cette propriété en fait un outil pertinent pour des applications nécessitant une modélisation réaliste des conditions de circulation en environnement urbain.

- **Complexité**

Chaque simulation nécessite $O(n \cdot |E|)$ opérations, où n est le nombre d'échantillons par arête (typiquement $n = 1000$). Cette étape de prétraitement reste raisonnable car elle est effectuée une seule fois avant l'application de Dijkstra.

4. Conclusion

Ce projet a exploré le problème du plus court chemin robuste dans un graphe pondéré par intervalles, modélisant l'incertitude des temps de trajet urbains.

Nous avons implémenté quatre heuristiques déterministes basées sur l'algorithme de Dijkstra (complexité $O((V+E) \log V)$), offrant des garanties selon différents critères : optimiste (11 min), prudent (23 min), stable (marge de 8 min) et pire cas (30 min).

Au-delà de ces bornes, nous avons développé deux approches stochastiques : une distribution gaussienne statique et un processus Beta dynamique intégrant drift, relâchements périodiques et bruit. Le processus Beta (≈22 min) produit des estimations plus réalistes que l'approche gaussienne (≈18 min) en capturant l'accumulation naturelle du trafic.

Ce travail illustre la complémentarité entre garanties déterministes et modélisation probabiliste pour la prise de décision en contexte incertain, une problématique centrale en optimisation de graphes et systèmes de navigation intelligents.