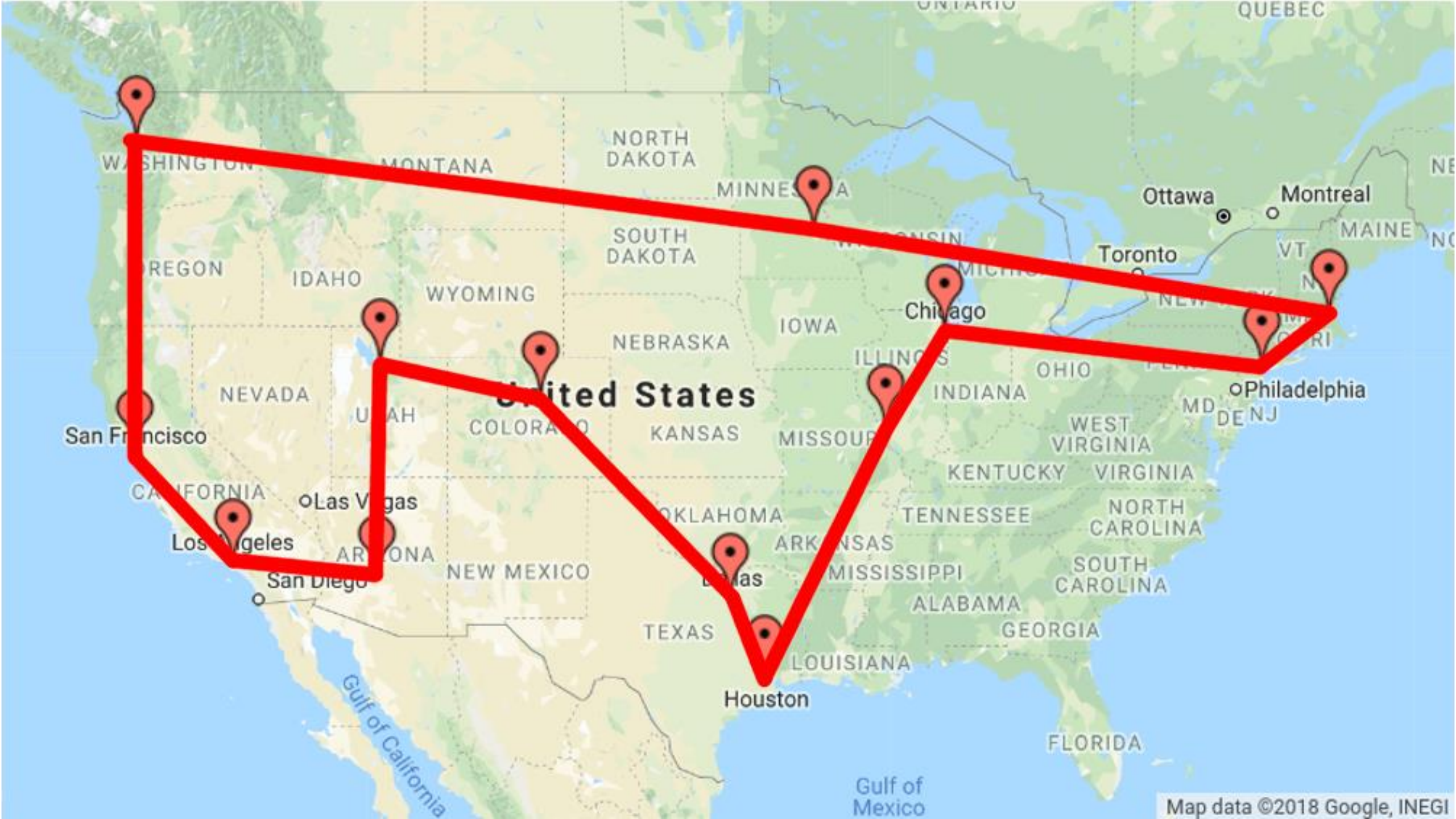
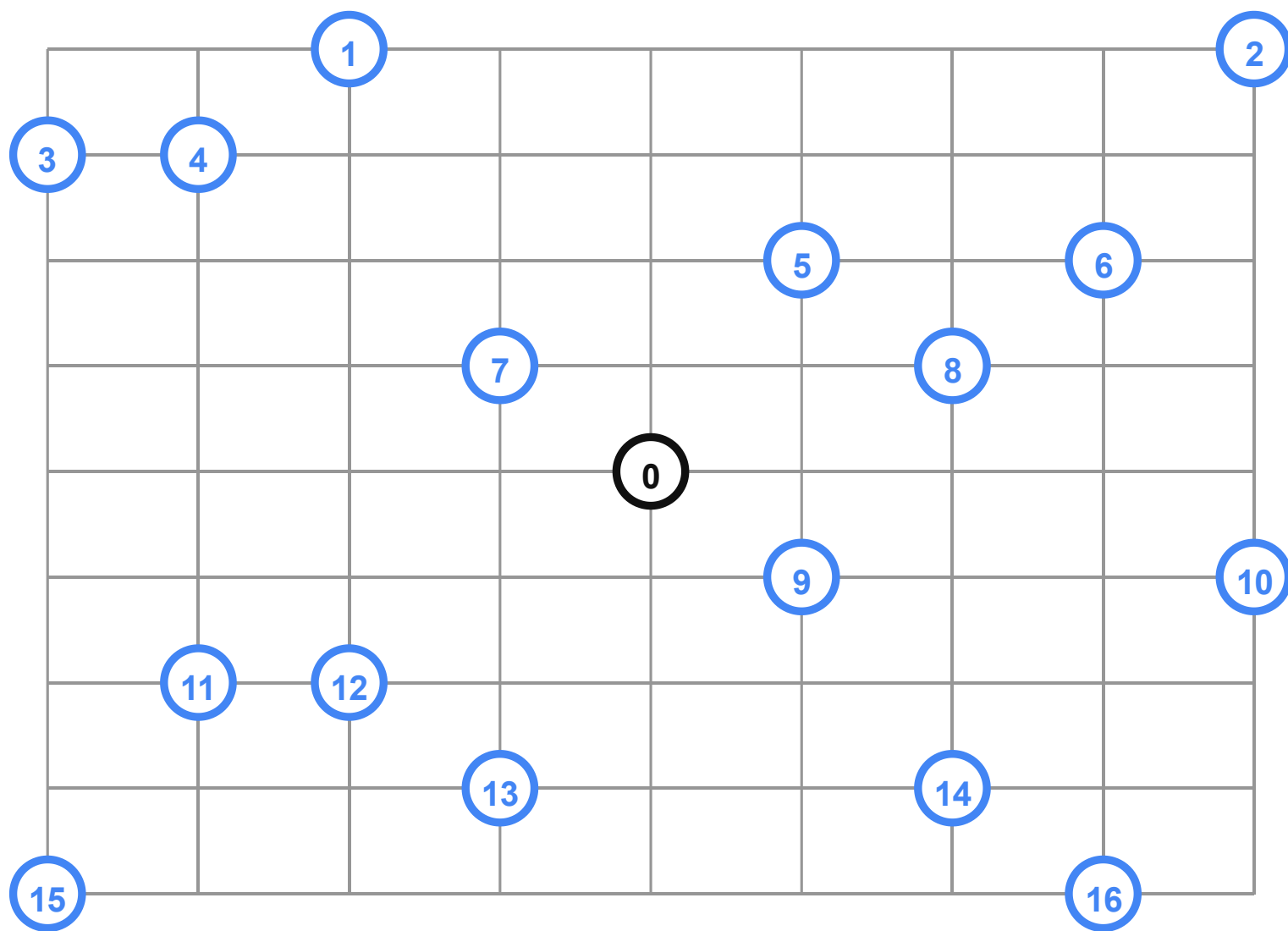


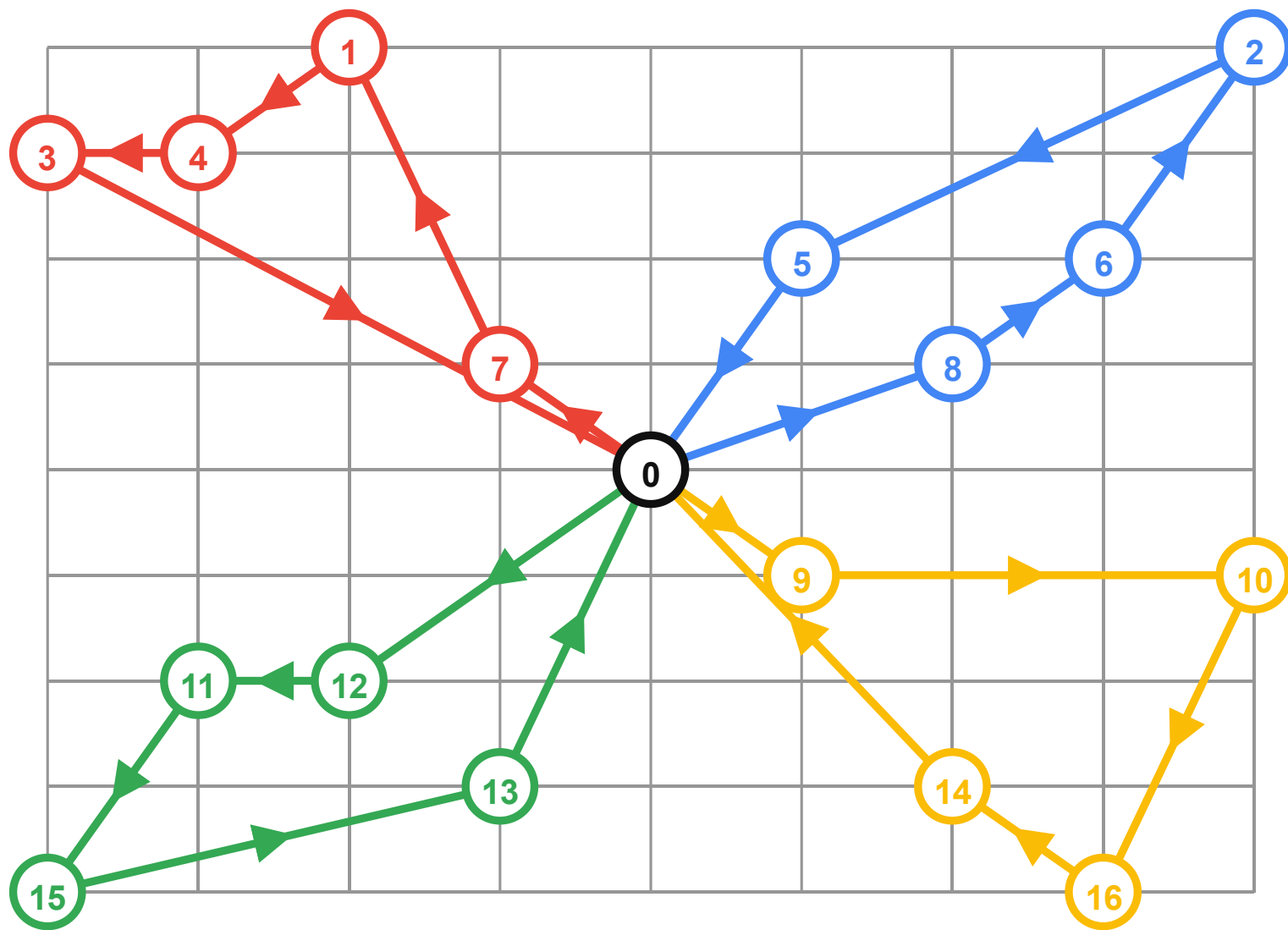
Optimisation de livraison pour une boutique en ligne avec Google OR-Tools

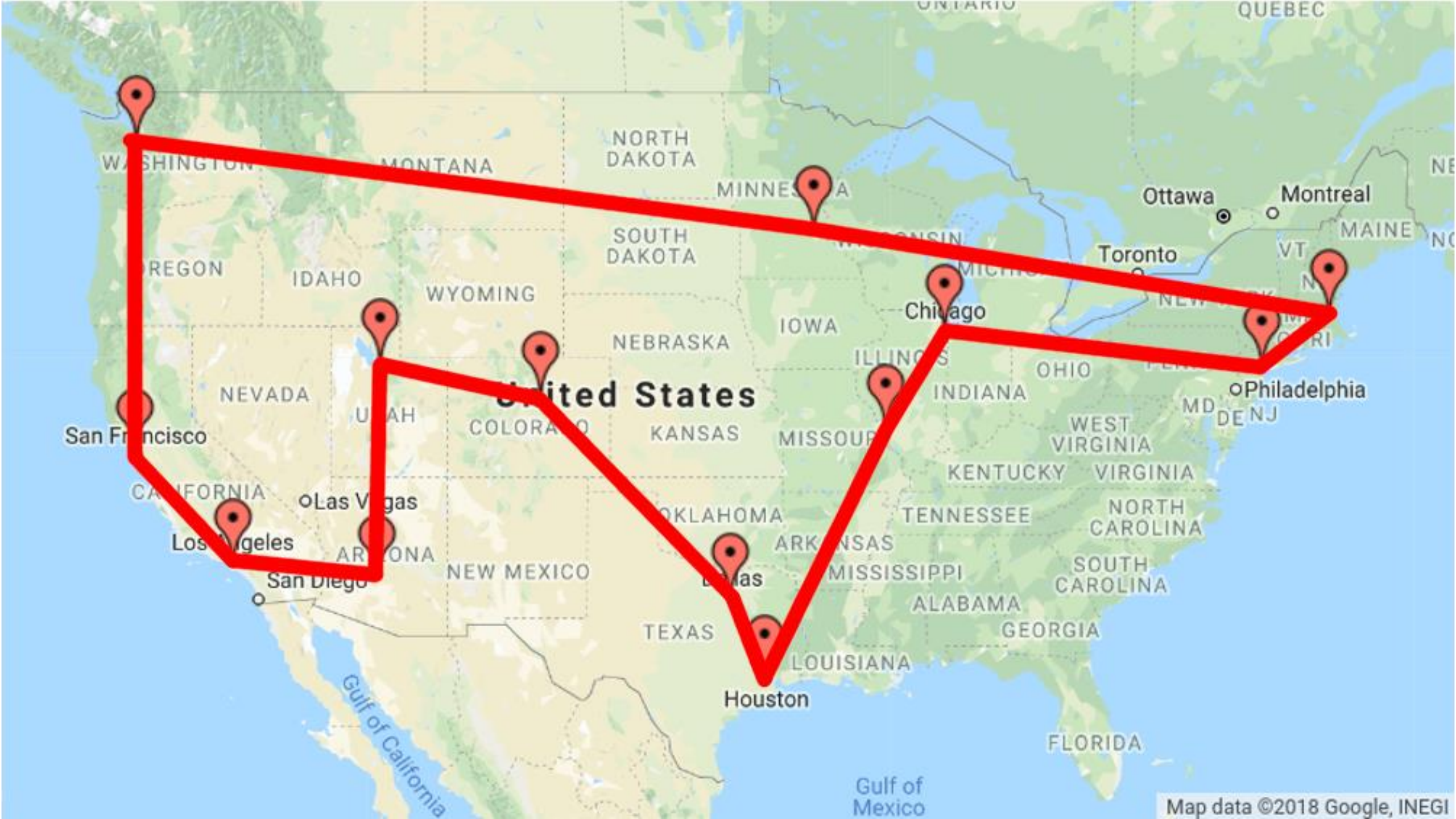
Projet de semestre 5 – Noah Godel













Contexte

Besoins

- Enterprise-XY -> super marché en ligne
- Effectuent les livraisons par des tiers (DHL, Post, ...)
- Veulent faire une partie des livraisons eux même
 - Livraison proche du dépôt -> Livraison directe
 - Livraison loin du dépôt -> Livraison par tiers
- Projet précédant -> focus sur matrice de distance

Contexte

Google OR-Tools

- Librairie Open-Source
- Recherche opérationnelle
- Outils pour VRP



Google OR-Tools

Contexte

Contraintes

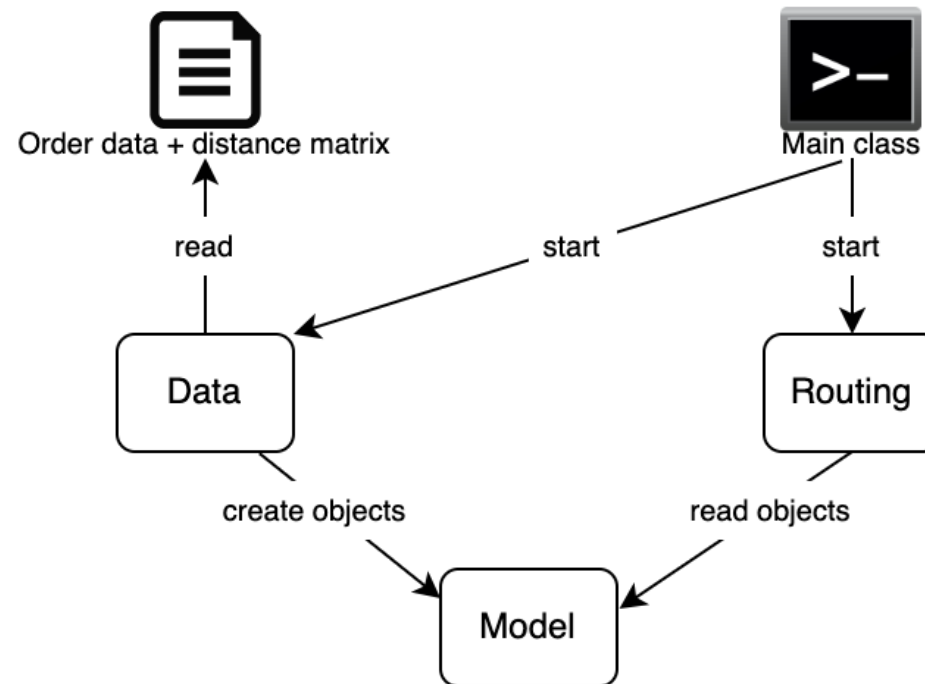
- Nombre de véhicules à disposition
- Fenêtres de livraison sur les commandes
- Capacité (en volume et poids) des véhicules
- Pris total d'une solution ne doit pas déplacer une livraison par tiers totale

Objectives

- Création d'une application fonctionnelle qui :
 - Lis des données de commandes (seulement NPA)
 - Calcule un plan de livraison qui respecte les contraintes
- Test de l'application et comparaison l'ancienne implémentation
- Conception d'une approche pour utiliser des adresses complètes

Travail effectué

- Application Java avec Maven pour importer Google OR-Tools
- 3 Parties :
 - Parsing des données géographiques et des commandes
 - Définition des contraintes
 - Tests automatisés



Travail effectué

Parsing

- Commandes sous format CSV
- Données géographiques -> matrice de distance en JSON
 - Pré généré dans le projet précédant

	Chicago	Detroit	Indianapolis	Milwaukee	St. Louis
Chicago		281	184	92	297
Detroit	283		287	382	530
Indianapolis	183	287		280	242
Milwaukee	92	382	279		373
St. Louis	298	531	244	375	

Travail effectué





























Contraintes

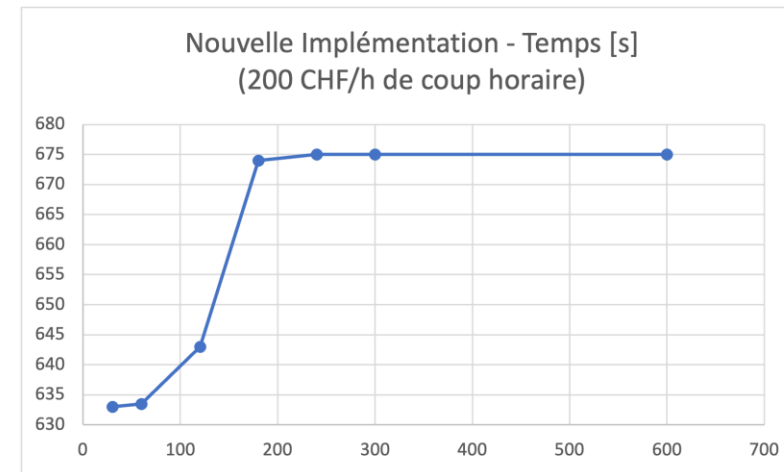
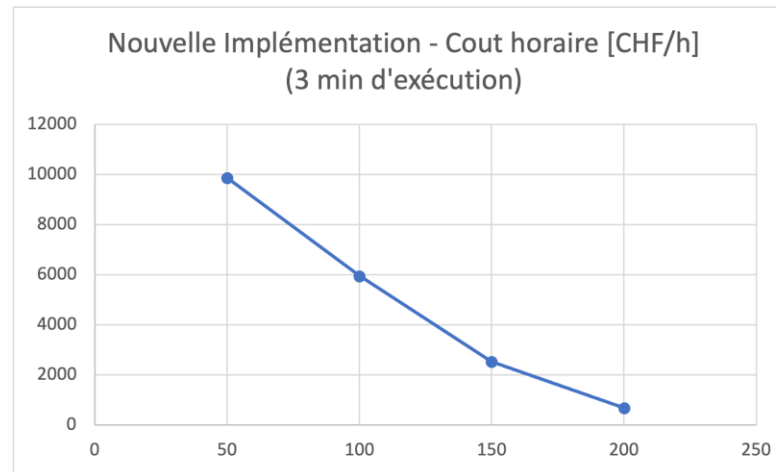
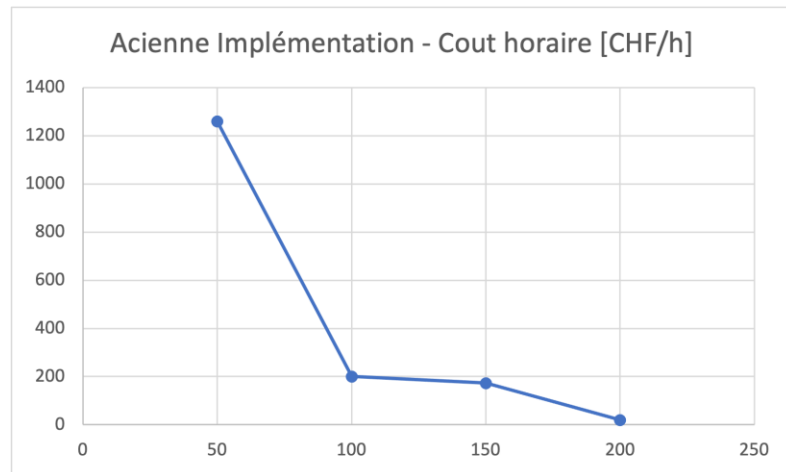
- Fenêtres de livraison
 - Temps d'arrivée + durée de livraison -> dans la fenêtre
- Respect des capacités des véhicules
 - Total de poids de toutes les commandes < capacité
 - Total du volume de toutes les commandes < capacité
- Pas de livraison directe pour des commandes trop éloignées
 - Coût d'une solution -> coût des livraisons directes + coûts livraison tiers
 - Configurer le solveur à minimiser le coût

Travail effectué

Test automatisé

- Pipeline Gitlab-CI pour automatiser les tests
- Test sur des données synthétique
 - Résultat prédictible donc test complet
- Test sur un jeu de données réduit
 - Résultat pas prédictible donc en surface

 Passed ⌚ 00:01:47 📅 1 month ago	refactor: split the config class #116041  main  0621c582 		 
 Passed ⌚ 00:01:49 📅 1 month ago	test: implement unit tests for the tim... #114511  main  c895c7d9 		 
 Passed ⌚ 00:01:41 📅 1 month ago	feat: implement the time window co... #114376  main  e85bf158 		 
 Passed ⌚ 00:01:49 📅 1 month ago	refactor: restructure the code #114374  main  aef2fa16 		 

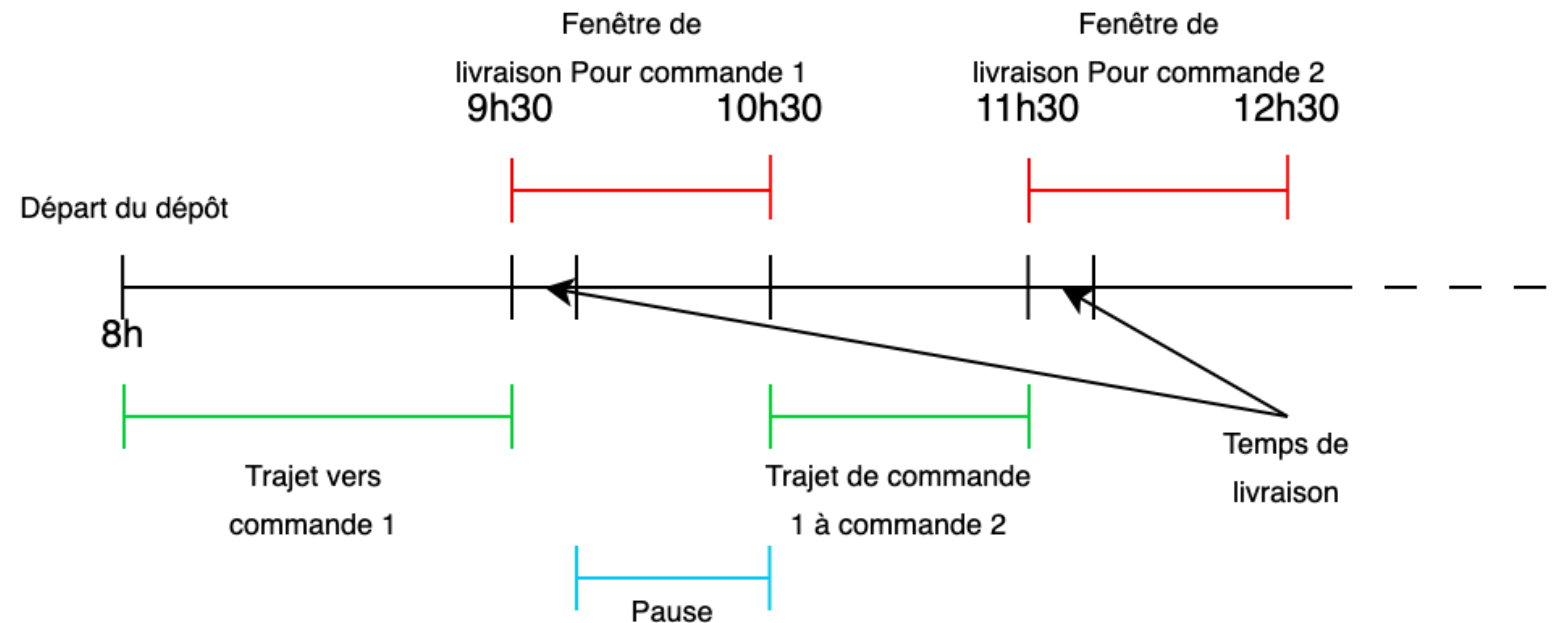


Resultats

Conclusion

Difficultés

- Implémentation des commandes optionnels
 - Par défaut le solveur essaye de trouver une solution valide
 - Mais il faut qu'il optimise pour trouver une bonne solution
- Manque de pauses dynamiques pour les fenêtres de livraison
 - Pas de pause -> contrainte additionnelle
 - Avec pause -> meilleurs résultats



Conclusion

Idées d'extension

- Visualisation des données sur Google Maps
- Modification du calcul de coups
 - Facteur le plus contraignant
 - Très simple pour l'instant
- Utilisation de vraies adresses pour les commandes
 - Approche 1 : Créer une matrice de distance avec tous les adresses commandes
 - Approche 2 : Utiliser directement l'API Google Maps
 - Approche 3 : Faire la planification pour les NPA et ajouter les adresses après-coup