[Date]

Van De Velde Luc

[nom de la société]

Projet Big Data

Projet

Table des matières

[Sujet 2](#_Toc12690312)

[CONTEXTE 2](#_Toc12690313)

[SUJET : 2](#_Toc12690314)

[Version de base 3](#_Toc12690315)

[Contraintes supplémentaires 3](#_Toc12690316)

[LIVRABLES : 3](#_Toc12690317)

[Livrable technique 3](#_Toc12690318)

[Notebook 4](#_Toc12690319)

[Soutenance 5](#_Toc12690320)

# Sujet

## CONTEXTE

Depuis les années 90, il y a eu une véritable prise de conscience mondiale de la nécessité de réduire la consommation d’énergie et des émissions de gaz à effet de serre. Les premiers engagements sont apparus lors de la signature du protocole de Kyoto en 1997. Mais son entrée en vigueur n’a finalement eu lieu qu’en 2005 et de nombreux scientifiques ont jugé les efforts insuffisants pour ralentir le réchauffement climatique. Depuis, d’autres engagements plus ambitieux ont vus le jour (division par 4 des émissions d’ici 2050 pour la France par exemple, engagements de certaines grandes villes comme Paris). Mais la tâche est compliquée. Les pouvoirs publics et les collectivités territoriales n’ont pas la possibilité d’obliger les entreprises et les particuliers à changer leurs habitudes pour atteindre ces objectifs. L’action se porte donc avant tout à faire évoluer les comportements. L’économie et le recyclage des matières premières, l’amélioration des modes de transports et des performances énergétiques des bâtiments doivent devenir des priorités.

Dans ce sens, L’ADEME (Agence de l’Environnement et de la Maîtrise de l’Energie) a récemment lancé un appel à manifestation d’intérêt pour promouvoir la réalisation de démonstrateurs et d’expérimentations de nouvelles solutions de mobilité pour les personnes et les marchandises adaptées à différents types de territoires. Votre structure CesiCDP est déjà bien implantée dans le domaine. Aidé de nombreux partenaires, vous avez réalisé plusieurs études sur le thème de la Mobilité Multimodale Intelligente. Les nouvelles technologies de transport, plus économiques et moins polluantes ne sont pas sans poser de nouveaux défis notamment d’un point de vue de l’optimisation de la gestion des ressources. Mais ces problèmes de logistique du transport présentent un enjeu majeur pour l’avenir : ses applications sont nombreuses (distribution du courrier, livraison de produits, traitement du réseau routier, ramassage des ordures) et leur impact sur l’environnement peut être véritablement significatif. Votre étude s’inscrit donc dans le cadre d’une réponse à l’appel de l’ADEME.

## SUJET :

Le but de votre étude est de générer une tournée de livraison (problème du VRP). Le problème algorithmique consiste donc à calculer sur un réseau routier une tournée permettant de relier entre elles un sous-ensemble de villes, puis de revenir à son point de départ, de manière à minimiser la distance totale parcourue.

Vous devrez proposer une méthode issue de la Recherche Opérationnelle pour générer une tournée de livraison correspondant à ce problème. L’implémentation se fera sur une version de base du problème, à laquelle vous pourrez ajouter des contraintes supplémentaires, rendant le problème plus réaliste, mais aussi plus dur à traiter.

Par ailleurs, vous devrez effectuer une étude statistique du comportement de votre méthode de résolution, faisant apparaitre ses performances (qualité de solution, temps de convergence). Idéalement, des statistiques prédictives permettent d’extrapoler ce comportement sur des cas d’usages que vos ordinateurs seuls ne pourraient traiter.

## Version de base

* Le choix du modèle et le code en Python capable de résoudre des instances de taille importante (plusieurs milliers de villes).
* Une étude statistique du comportement expérimental de l'algorithme

## Contraintes supplémentaires

Voici une liste (non exhaustive) de contraintes que vous pouvez considérer. Pour certaines, des versions avancées sont aussi proposées. L’implémentation d’une contrainte et de l’une de ses versions avancées vaut l’implémentation de deux contraintes.

* Fenêtre de temps de livraison pour chaque objet
  + Interdiction de livrer hors de la fenêtre
  + Possibilité d'attendre sur place l'ouverture de la fenêtre temporelle
* k camions disponibles simultanément pour effectuer les livraisons. Le calcul de la tournée devra inclure l’affectation des objets (et donc des points de livraison) aux différents camions disponibles, et minimiser non plus la distance, mais la date de retour des camions à la base.
  + Capacité des camions (2 ou 3 dimensions) et encombrement des objets
  + Certains objets ne peuvent être livrés que par certains camions
* Chaque objet a un point de collecte spécifique

D’autres contraintes peuvent être traitées, si c’est justifié par une application industrielle (pas forcément issue de la tournée d'entretien).

## LIVRABLES :

Vous devez d'une part rendre 24h avant la soutenance deux livrables. L'un est technique, l'autre est un livrable rédigé :

* 1 livrable technique : l'ensemble des codes et des donnes générées (solutions et statistiques)
* 1 livrable rédigé : Notebook présentant l'ensemble du travail et des résultats obtenus

### Livrable technique

Le livrable technique est une archive contenante :

* Les Dataset utilisés
  + Fichiers d'instance de problème
  + Valeur de l'optimal ou borne inférieure
  + Code de génération aléatoire
* Code Python de l'algorithme
* Codes pour générer l'étude expérimentale (langage au choix entre Python et R)
  + Code de calcul des paramètres des graphes et des bornes inférieures
  + Code de calcul des statistiques descriptives, voire prédictives
  + Script de tests de montée en charge

Le code n'a pas à être au standard d'un code à industrialiser (pas besoin d'une grande modularité). Il doit en revanche rester lisible, commenté, et privilégier la performance. Il est fortement recommandé de suivre les recommandations PEP :

<https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/235263-de-bonnes-pratiques>

### Notebook

Le notebook au format Jupyter doit présenter vos travaux et les résultats que vous avez obtenus. L'aspect story-telling doit être privilégié dans la rédaction. Des références d'articles scientifiques ou d'ouvrages spécialisés peuvent être incluses, et les résultats comparés avec ceux obtenus par votre implémentation. Les points abordés incluent (pas nécessairement dans cet ordre) :

1. Description de l'algorithme

* Modélisation du problème algorithmique
  + Définition du problème formel
  + Étude de complexité
* Présentation du choix et description de l'algorithme
  + Fonctionnement, paramètres
  + Spécificités algorithmiques éventuelles ajoutées à la méthode
* Modélisation du problème selon le formalisme de l'algorithme
* Illustration avec différents cas de tests

1. Étude statistique du comportement expérimental, à l'aide de dataset générés aléatoirement, mais aussi éventuellement de dataset issus de la recherche scientifique

* Statistiques descriptives, voire prédictives, du comportement de l'algorithme, mises en regard avec l’industrie
* Exploitation des paramètres de l'instance du problème (taille et largeur du graphe, degré, nombre d'objets, taille et valeur des objets, nombre de véhicules, nombre de points de livraison...), et des paramètres de l'algorithme (température, taille de liste tabou, nombre de mutations etc.)
* Analyse et commentaire des résultats d'analyse (qualité des solutions, temps de convergence, nombre d'itérations, espace mémoire...)

Soutenance :

La soutenance doit être orientée résultats, avec démonstration de l’exécution du code (sur des cas suffisamment petits pour garder la présentation fluide) et présentation des résultats. La présentation se faisant en anglais, il est probable qu'au moins un des membres du jury soit non informaticien. Il est donc recommandé de prêter une attention particulière sur l'aspect vulgarisation, tout en maintenant autant que possible l'exactitude et la rigueur scientifique.

Par ailleurs, l’usage d’outils de visualisation pourrait rendre votre présentation plus dynamique. Des outils comme NetworkX (https://networkx.github.io/documentation/stable/index.html), et matplotlib vous seront surement très utiles. Vous pouvez aussi ajouter un à votre Notebook, si vous préférez. Veillez toutefois à ne pas perdre trop de temps avec ces outils.

Les points suivants doivent être abordés dans la présentation :

* Présentation du problème concret avec les contraintes considérées, et des enjeux industriels
* Présentation concise et vulgarisée de l'approche de résolution
* Démonstration technique sur des instances suffisamment petites pour garder la présentation fluide
* Présentation vulgarisée de l'étude statistique, focus sur les résultats les plus significatifs

La soutenance aura lieu en anglais.

A faire :

**Livrable technique**

Le livrable technique est une archive contenante :

• Les Dataset utilisés

o Fichiers d'instance de problème

o Valeur de l'optimal ou borne inférieure

o Code de génération aléatoire

• Code Python de l'algorithme

• Codes pour générer l'étude expérimentale (langage au choix entre Python et R)

o Code de calcul des paramètres des graphes et des bornes inférieures

o Code de calcul des statistiques descriptives, voire prédictives

o Script de tests de montée en charge

**Notebook**

Étude de complexité(Ludovic)

Présentation du choix et description de l'algorithme

* Fonctionnement, paramètres
* Spécificités algorithmiques éventuelles ajoutées à la méthode
* Modélisation du problème selon le formalisme de l'algorithme
* Illustration avec différents cas de tests

Étude statistique du comportement expérimental, à l'aide de dataset générés aléatoirement, mais aussi éventuellement de dataset issus de la recherche scientifique

* Statistiques descriptives, voire prédictives, du comportement de l'algorithme, mises en regard avec l’industrie
* Exploitation des paramètres de l'instance du problème (taille et largeur du graphe, degré, nombre d'objets, taille et valeur des objets, nombre de véhicules, nombre de points de livraison...), et des paramètres de l'algorithme (température, taille de liste tabou, nombre de mutations etc.)
* Analyse et commentaire des résultats d'analyse (qualité des solutions, temps de convergence, nombre d'itérations, espace mémoire...)

|  |  |  |  |
| --- | --- | --- | --- |
| Ludovic | Denis | Quentin | Luc |
| Etude de complexité | Dataset | L’étude expérimentale | Code Python de l'algorithme |