ÉTUDE DE CAS

Énergie, Transport, Environnement

Xavier Olive Olivier Poitou Gabriel Jarry

OBJECTIFS DU MODULE

- Connaître plusieurs cadres mathématiques disponibles pour faire de l'optimisation
- Prendre en main différents outils d'optimisation
- Pratiquer toute la chaîne : compréhension du problème, formalisation des données d'entrée, modélisation, résolution analyse et présentation des résultats.

Acquérir une expérience pratique de l'optimisation

SUJET À TRAITER

The Pollution Routing Problem

Problème d'optimisation d'une tournée de flotte de véhicules.

Données proches d'un cadre réaliste (open data)

Premières contraintes :

- travail en trinôme ;
- au moins un SD par groupe ;
- au plus un étudiant non francophone

DIFFICULTÉS

La fonction à optimiser n'est pas posée clairement.

La modélisation est basée sur **votre** culture générale.

Le choix des méthodes d'optimisation à appliquer est libre.

Les SD vont devoir former les autres.

PRÉREQUIS

Le bagage **de base** en optimisation pour un ingénieur ISAE

Tronc commun :

- Optimisation non linéaire, descentes de gradient, 1A
- Programmation linéaire (Simplexe), 1A
- Programmation linéaire en nombre entiers, 2A

Filière :

- Programmation par contraintes
- Métaheuristiques

et un peu plus de courage que d'habitude

Déroulement des séances

- 11 janvier : kick-off
- 17 janvier : prise en main/support général
- 24 janvier : choix des pistes à explorer
- 8 février (journée) : mise en forme du projet
- 22 février : soutenances

Le saviez-vous ? Le plus gros du travail est à fournir **entre** les séances.

Le choix du langage/format pour la soutenance est libre, **mais**

- Les ressources sont fournies en Python ;
- Le support aussi.

Pour commencer, dans un terminal :

```
git clone --recurse-submodules -j4
  https://github.com/xoolive/edu_pollution
```

(en une seule ligne)

CONTENU DU REPOSITORY

Dans `notebooks` :

- un squelette du projet avec proposition MILP ;
- un squelette pour `ipyleaflet`

Dans `resources`, les supports utilisés dans d'autres modules :

- programmation non linéaire ;
- programmation linéaire en nombres entiers ;
- programmation par contraintes ;
- métaheuristiques.

DERNIERS CONSEILS

Sont valorisés :

- le travail en équipe : utilisez les compétences de chacun, ne travaillez pas à trois derrière un clavier ;
- les problèmes bien posés… et correctement traités. Préférez les problèmes simples aux complexes, et les problèmes complexes aux compliqués ;
- proof of concept : un code lisible, modulaire et qui permet de reproduire les résultats présentés

Commencez par un problème simple, et complexifiez pas à pas.