

# ÉTUDE DE CAS

Énergie, Transport, Environnement

---

Xavier Olive   Olivier Poitou

- ▶ 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

## Energy Production Planning

Problème d'optimisation d'un système de fourniture d'électricité.

Données proches d'un cadre réaliste (open data)

Premières contraintes :

- ▶ travail en binôme ;
- ▶ les SDD évitent de se regrouper ;
- ▶ au plus un étudiant non francophone

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.

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

- ▶ 6 novembre : prise en main/support général
- ▶ 20 novembre : fin exercice PPE
- ▶ 21 janvier : présentation/discussion de l'extension du projet par groupe
- ▶ 12 février : présentation du travail effectué, points bloquants, préparation soutenance
- ▶ 5 mars : 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
```

<https://github.com/poitou/ETE>

(en une seule ligne)

Dans **project** :

- ▶ un squelette du projet avec proposition MILP;

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

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



Sont valorisés :

- ▶ le travail en équipe : utilisez les compétences de chacun ;
- ▶ les problèmes bien posés... et correctement traités et défendus ;
- ▶ 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.