

Ce mini-projet, à effectuer en binôme au sein du même groupe de PC, fera l'objet d'un rapport incluant notamment équations et graphiques obtenus par des simulations sous Python. La forme de ce rapport est laissée libre (pdf, notebook, version papier...). Plusieurs rendus intermédiaires sont attendus auprès de votre chargé de PC.

Charge d'une flotte de véhicules électriques

On s'intéresse dans ce sujet à la charge de plusieurs véhicules électriques connectés à une station de recharge. Chaque véhicule est décrit par une heure et un état de charge d'arrivée et l'utilisateur indique au système d'ordonnancement une heure et un état de charge de récupération. On cherche dans ce contexte à assurer ces demandes, tout en minimisant le coût électrique de recharge et en assurant que la puissance électrique fournie aux différents véhicules ne dépasse pas la puissance maximale fournie par le réseau. On s'appuiera sur des données disponibles sur Oasis.

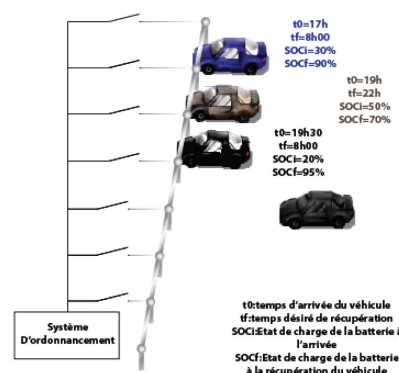


Fig.1 : Exemple d'états de flotte à une station de recharge.

1 Modélisation

1. Formuler la fonction objectif du problème. On pourra supposer l'existence de deux tarifs électriques (heures pleines et heures creuses).
2. Formuler un modèle dynamique reliant l'état de charge d'un véhicule et la puissance qui lui est fournie.
3. Proposer une méthodologie d'identification de ce modèle basée sur les données fournies sur Oasis et la mettre en oeuvre.
4. Formaliser les contraintes restantes du problème et formuler le problème de minimisation correspondant. On précisera en particulier les variables de décision, les contraintes (au nombre de six) et la fonction objectif du problème.

2 Etude et résolution numérique

On s'intéresse dans un premier temps au cas d'un seul véhicule.

1. Etudier le problème (convexité, conditionnement, existence et unicité d'une solution) par la méthode, analytique ou numérique, de votre choix. Proposer une méthode de résolution adaptée aux caractéristiques du problème.
2. Proposer des ordres de grandeurs pour les variables du problème et développer un algorithme de résolution. Commenter les résultats obtenus.

3 Etude avancée

On se propose maintenant d'étudier le problème de plusieurs véhicules.

3. Etudier la faisabilité d'une méthode de décomposition-coordination (présentée en annexe).
4. Appliquer cette méthodologie dans le cas de deux véhicules. L'implémenter. Commenter les résultats obtenus.

Annexe : Algorithme de décomposition/coordination

L'algorithme de décomposition/coordination par méthode d'Uzawa permet de résoudre un problème d'optimisation dans le cas où :

- le coût f se décompose comme une somme de différents termes ne dépendant que d'une variable de décision

$$f(x) = f_1(x_1) + \dots + f_n(x_n)$$

- les contraintes inégalités (et égalités) c se décomposent comme une somme de différents termes ne dépendant que d'une variable de décision

$$c(x) = c_1(x_1) + \dots + c_n(x_n)$$

Le principe de l'algorithme est alors de "décomposer" le problème à l'aide du Lagrangien, de résoudre n problèmes d'optimisation sous contraintes à l'aide des multiplicateurs de Lagrange et de "coordonner" les solutions pour mettre à jour les variables couplées (les multiplicateurs).

Il prend la forme suivante :

Algorithme 1 (Décomposition/coordination)

1. Initialisation ($k = 0$) : on choisit le multiplicateur λ_0 et le pas ρ .
2. Décomposition : on résout les n problèmes :

$$\min_{x_i} [f_i(x_i) + \lambda^k c_i(x_i)] \tag{1}$$

dont on note x_i^k les solutions.

3. Coordination : effectuer les mises à jour des multiplicateurs

$$\lambda^{k+1} = P(\lambda^k + \rho [c_1(x_1^k) + \dots + c_n(x_n^k)]) \tag{2}$$

où P est la projection sur $(\mathbb{R}^+)^m$

4. Test d'arrêt : si $|\lambda^{k+1} - \lambda^k|$ est suffisamment petit, on arrête l'algorithme. Sinon, $k = k + 1$ et on revient à l'étape 2.