

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, jupyter notebook, github...). Le rendu s'effectuera sur Moodle. La date limite de rendu est le lundi 06/05 à 09h00, aucun rendu ne sera possible après cela.

Epaisseur du trait ?

On s'intéresse dans ce sujet à un bâtiment résidentiel équipé de panneaux photovoltaïques, dont on cherche à **maximiser l'autoconsommation**, cad la part d'utilisation sur place de l'énergie électrique produite pour les panneaux (le surplus étant réinjecté sur le réseau électrique). On suppose pour cela que le bâtiment est équipé d'un chauffe-eau électrique intelligent, dont **on peut contrôler la puissance de chauffe**. On considère par ailleurs un horizon de temps de pilotage $[t_0, t_f]$, que l'on partitionne en N intervalles de temps $[t_i, t_{i+1}]$ ($i = 0, \dots, N-1$) de longueur uniforme Δt , avec $\Delta t = (t_f - t_0)/N$, de sorte que $t_N = t_f$.



La température du ballon évolue entre deux pas de temps t_i et $t_{i+1} = t_i + \Delta t$ selon l'équation

$$T_{i+1} = e^{-k\Delta t}T_i + \frac{1 - e^{-k\Delta t}}{k}C(-Q_i + P_i) \quad i = 0, \dots, N-1 \quad (1)$$

où $k, C > 0$ sont deux constantes et Q_i est la puissance soustraite par un éventuel soutirage d'eau, effectué pour des usages non-pilotables des occupants (vaisselle, douche,...), supposée connue et constante sur l'intervalle $[t_i, t_{i+1}]$. P_i est la puissance de chauffage du ballon au temps t_i , supposée également constante sur chaque intervalle de temps. Les conditions initiale/terminale correspondant à cette dynamique sont $T_0 = T_{in}$ et $P_N = 0$.

Le ballon est par ailleurs équipé d'un thermostat limitant la température atteignable, pour des raisons de sécurité. Ce système de régulation assure ainsi que $0 \leq T_i \leq T_{sat}$ pour $i = 0, \dots, N$. La puissance du ballon est par ailleurs limitée

$$0 \leq P_i \leq P_M \quad i = 0, \dots, N \quad (2)$$

L'énergie photovoltaïque auto-consommée sur l'horizon de temps $[t_0, t_f]$ s'écrit

$$\sum_{i=1}^N \min \{E_i^{PV}, w_i + P_i\Delta t\} \quad (3)$$

où E^{PV} est l'énergie produite par les panneaux photovoltaïques et w_i les consommations non-pilotables de la maison.

1 Etude du problème d'optimisation

1. Interpréter la définition de l'énergie auto-consommée (3). Justifier que l'on peut de façon équivalente considérer la quantité

$$\sum_{i=1}^N \min \{\hat{E}_i, P_i\Delta t\} \quad (4)$$

avec $\hat{E}_i = E_i^{PV} - w_i$. Quelle difficulté comporte ce coût (tout comme le précédent) dans le cadre d'un problème d'optimisation ?

2. En lieu et place de (4), on se propose de maximiser la fonction dite de "soft-min"

$$\sum_{i=1}^N h(E_i^{PV}, P_i \Delta t) \text{ où } h : (x_1, x_2) \in \mathbb{R}^2 \mapsto \frac{x_1 e^{-\alpha x_1} + x_2 e^{-\alpha x_2}}{e^{-\alpha x_1} + e^{-\alpha x_2}} \quad (5)$$

avec $\alpha > 0$ constante. Justifier que, pour α grand, h est une bonne approximation de la fonction \min . Quel intérêt a-t-on alors à considérer ce problème approché plutôt que le problème original ?

3. Interpréter l'équation (1). Cette modélisation vous semble-t-elle raisonnable ?
4. Formuler le problème d'optimisation à résoudre sous la forme

$$\begin{aligned} \min_x \quad & f(x) \\ \text{tel que } & c_{eq}(x) = 0, \\ & c_{in}(x) \leq 0 \end{aligned} \quad (6)$$

On précisera les variables de décision x , leur nombre n , les contraintes c_{eq} et c_{in} ainsi que la fonction objectif f à minimiser.

2 Etude et résolution numérique du problème soft-min

5. Quelles méthodes de résolution peuvent être envisagées pour ce problème ?
6. Développer un algorithme de résolution dans le cas de l'horizon de temps $t_0 = 6h$ et $t_f = 19h$ avec $\Delta t = 0.25h$, dans un cas sans usage non-pilotables ($w_i = 0$ pour tout i). On prendra les valeurs numériques suivantes :

$$\alpha = 100, \quad k = 0.2h^{-1}, \quad T_{sat} = T_f = 70^\circ C, \quad T_{in} = 50^\circ \quad (7)$$

$$C = 100C/Wh, \quad P_M = 3kW, \quad E_i^{PV} = 2e^{-(t_i-13)^2/9} \quad \text{et} \quad Q_i = 3W \text{ pour } t_i = 12 \text{ et } 0 \text{ sinon.} \quad (8)$$

On tracera notamment les graphes suivants :

- (a) l'évolution temporelle de la température du ballon, que l'on superposera aux bornes sur la température
- (b) l'évolution temporelle de la puissance fournie au ballon, de la courbe d'énergie photovoltaïque produite et de l'énergie photovoltaïque auto-consommée correspondante au cours du temps

Commenter les résultats obtenus.

3 Etude et résolution numérique du problème avec variable slack

7. On se propose d'introduire $N + 1$ variables additionnelles s_i , telles que $s_i \leq E_i^{PV}$ et $s_i \leq P_i$ pour tout $i = 0, \dots, N$. Reformuler le problème d'optimisation à l'aide de ces variables.
8. Etudier la convexité de ce nouveau problème. Appartient-il à une famille particulière de problème d'optimisation ? Quels avantages/inconvénients présente-t-il par rapport à la formulation précédente ?
9. Résoudre ce nouveau problème avec les mêmes paramètres que précédemment. Comparer les solutions obtenues.

4 Pilotage d'usages blancs et introduction à l'optimisation entière

En plus du chauffe-eau, on suppose que la machine à laver du logement est pilotable et l'on cherche également à optimiser la puissance électrique qui lui est allouée P_i^L ($i = 0, \dots, N$), toujours de sorte à maximiser l'autoconsommation. Pour chaque cycle de lavage, la machine à laver consomme une certaine puissance \bar{P}^L pendant un temps donné fixe et ininterrompible $T = n_L \Delta t$. Pour un cycle lancé au temps t_i , par exemple, on a ainsi

$$P_j^L = \bar{P}^L \mathbb{1}_{\{i, \dots, i+n_L-1\}}(j) = \begin{cases} \bar{P}^L & \text{si } j \in \{i, \dots, i+n_L-1\} \\ 0 & \text{sinon} \end{cases} \quad (9)$$

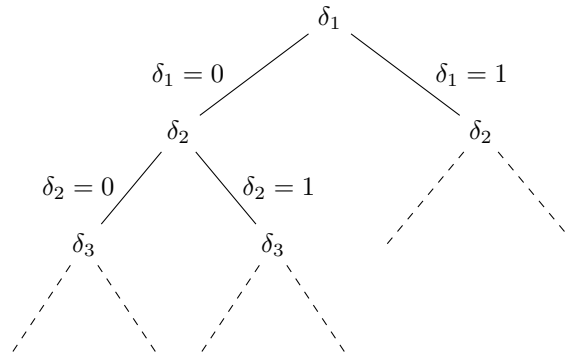
10. On suppose dans un premier temps que le cycle de la machine à laver est lancée à un temps t_{i_0} fixe donné. Ecrire le problème d'optimisation correspondant.
11. A l'aide de ce sous-problème, proposer une approche combinatoire exhaustive permettant de résoudre le problème originel. L'implémenter avec les mêmes paramètres que précédemment ainsi que $n_L = 6$ et $\bar{P}^L = 0.25W$. Commenter le résultat obtenu. Etudier l'extensibilité de cette technique de résolution au cas de plusieurs autres usages pilotables.
12. Au vu de la question précédente, on cherche une autre technique de résolution. Malheureusement, le comportement (9) ne peut être modélisé par des variables continues. On définit donc N variables binaires $\delta_i \in \{0, 1\}$ vérifiant, pour $M \gg 1$ une grande constante,

$$P_j^L = \delta_i \bar{P}^L, \quad \sum_{i=0}^N \delta_i = n_L \quad \text{et} \quad n_L - \sum_{j=0}^i \delta_j + M\delta_i \leq M(\delta_{i+1} + 1) \quad \text{pour } j = 0, \dots, N-1 \quad (10)$$

- (a) Justifier que (10) est bien équivalent à (9). Ce type d'écriture de contraintes porte le nom de "technique du grand M ". Justifier ce nom.
- (b) Ecrire le problème d'optimisation correspondant, avec des variables de décision continues et binaires. Ce type de problème est nommé "Mixed-Integer" et peut être résolu par l'algorithme de Branch and Bound proposé ci-dessous. Détailler comment l'appliquer au cas présent.
13. *Bonus* : Résoudre ce problème avec un solveur pré-existant, implémentant l'algorithme de Branch-and-Bound. On pourra se référer à Moodle pour des instructions complémentaires sur le choix d'un solveur approprié.

Annexe : Algorithme de Branch and Bound

L'algorithme de Branch and Bound cherche à résoudre un problème d'optimisation en séparant l'ensemble de recherche (ensemble des contraintes) et en bornant les sous-problème d'optimisation correspondants. Il est particulièrement central en optimisation combinatoire, où il permet d'éviter le parcours exhaustif de l'arbre de décision correspondant aux variables binaires.



Si l'on note f la fonction à minimiser (sous certaines contraintes) et $x \in \mathbb{R}^n$ et $\delta \in \{0, 1\}^m$ les variables de décision, il prend la forme suivante :

Algorithme 1 (Branch and Bound)

1. *Initialisation : on considère une solution (x^0, δ^0) admissible, de coût f_0 et on prend $f^b = f_0$ et $\delta^b = \delta_0$. On initialise $k = 1$ et $\delta_f = 0$.*
2. *Déterminer si le problème avec $\delta_i = \delta_i^f$ ($i = 1, \dots, k$) fixé est faisable.*
 - (a) *S'il l'est,*
 - i. *si $k < n$ (on est à un noeud intermédiaire de l'arbre), calculer une borne inférieure du coût optimal correspondant, noté f_{inf} .*
 - A. *Si $f_{inf} < f^b$, $\delta^f \rightarrow (\delta^f, 0)$, $k \rightarrow k + 1$ (on descend la branche) et retourner à l'étape 2. Sinon :*
 - B. *Si $\delta_k^f = 0$, prendre $\delta_k^f = 1$ (on coupe la branche et change de branche) et retourner à l'étape 2; sinon, $\delta_{k-1}^f = 1$, $k \rightarrow k - 1$ (on remonte la branche, coupe et change de branche) et retourner à l'étape 2.*
 - ii. *si $k = n$, (on est arrivé au bout d'une branche) résoudre le problème d'optimisation avec toutes les variables binaires fixées et égales à δ_f . Si le cout f^* obtenu est inférieur à f^b , $f^b \rightarrow f^*$ et $\delta^b \rightarrow \delta_f$. Si $\delta^f = 1$ (on a parcouru la dernière branche), l'algorithme s'arrête. Sinon, aller en étape 2.(a)i.B.*
 - (b) *S'il ne l'est pas, et que $\delta_i^f = 1$, l'algorithme s'arrête. Sinon, aller en étape 2.(a)i.B.*