Laplace
LABORATOIRE PLASMA ET CONVERSION D'ÉNERGIE

Formation toolbox pour le design et controle des micro-reseaux

Séance Bonus

Corentin Bœnnec
Groupe Genesys, Laplace, Université de Toulouse

15 mai 2024

cnrs
TOULOUSE INP N7
UNIVERSITÉ TOULOUSE III PAUL SABATIER

Sommaire

Rappel sur l'optimisation bi-objectif

Cadre général

Un problème d'optimisation c'est :

- Des variables de décisions.
- Un domaine de définition pour ces variables ($\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{N}$, 0,1...).
- Des contraintes fonction de ces variables.
- Un ou **plusieurs objectifs** fonctions de ces variables à minimiser ou maximiser.

Comparaison de solution (introduction)

Dans le cadre d'un problème de minimisation. Si une solution $S1$ dont la valeur des objectifs est (x_1, y_1) est comparée à une solution $S2$ dont la valeur des objectifs est (x_2, y_2). Plusieurs situations sont possibles :

- $S1$ « est meilleure que » $S2$ si $x_1 < x_2$ et $y_1 < y_2$
- $S2$ « est meilleure que » $S1$ si $x_2 < x_1$ et $y_2 < y_1$
- $x_2 > x_1$ et $y_2 < y_1$ ou $x_2 < x_1$ et $y_2 > y_1$ aucune n'est meilleure.

Dominance au sens de Pareto

Dominance au sens de Pareto

De façon informelle, lorsque l'on compare 2 solutions d'un problème multi-objectif. On dira que $S1$ domine $S2$ noté ($S1 \succ S2$) si tous les objectifs de $S1$ sont au moins aussi bons que ceux de $S2$ et au moins un est strictement meilleur.
De façon plus formelle et dans le cadre d'un problème de minimisation à n variables et m objectifs avec f la fonction objectif.

$$S1 = \{x_1^1, ..., x_n^1\}, S2 = \{x_1^2, ..., x_n^2\} \quad (1)$$

$$f(x) : \mathcal{R}^n \rightarrow \mathcal{R}^m \quad (2)$$

$$f(S1) = \{y_1^1, ..., y_m^1\}, f(S2) = \{y_1^2, ..., y_m^2\} \quad (3)$$

$$S1 \succ S2 \implies \forall i \quad y_i^1 \leq y_i^2 \quad (4)$$

$$\exists j \quad y_j^1 < y_j^2 \quad (5)$$

Représentation des solutions sous forme de Front

Définition du front

- La comparaison de 2 solutions peut conduire à 2 scénarios, soit l'une domine l'autre soit aucune ne domine l'autre.
- En répétant l'opération et en comparant chaque solution à toutes les autres, on obtient la liste des solutions non dominées.
- Les solutions dominées sont par nature moins bonnes que d'autres solutions qui les ont dominées (du moins dans le cadre du problème défini).
- Les solutions non dominées forment, elles, ce que l'on appel un front de Pareto (dont une représentation en 2D est fourni ci contre).
- Il n'est alors mathématiquement pas possible de classer les solutions restantes. Il revient alors au décideur de choisir, parmi les solutions non dominées, celle qui lui parait être le meilleur compromis.

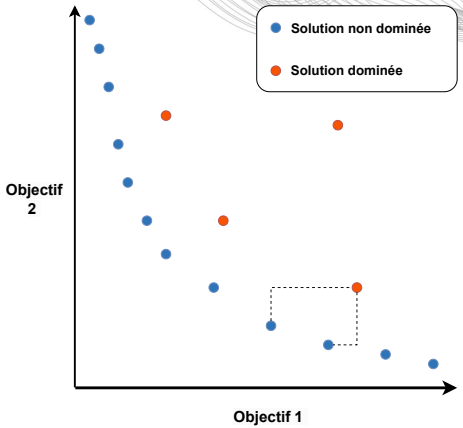

Figure – Exemple d'illustration d'un front dans le cadre d'un problème de minimisation bi-objectif. Le front se compose des solutions non dominées, ici en bleu.

Principe général du NSGAII

Quelques élément et charactéristiques

Le Non Sorting Genetique Agorithm II est un algorithme métaheuristique évolutionniste qui vise à l'obtention du front de Pareto pour un problème donné.
C'est un algorithme à base de population avec du nichage sur les objectifs. Càd qu'un ensemble de solutions courantes est constamment conservé afin d'évoluer (on conserve également les meilleures solutions rencontrées). Le nichage dans le NSGA-II vise à maintenir la diversité des solutions via une *crowding distance* sur les objectif pour encourager une exploration étendue du front.

Le déroulement de l'algorithme

1. Initialisation de la population
2. Pour chaque génération
   - 2.1 Calcul de la dominance dans la population
   - 2.2 Affectation des solutions à des niches
   - 2.3 tri des solutions en fonction de la dominance et des niches
   - 2.4 Sélection des individus pour reproduction
   - 2.5 Croisement des individus sélectionnés
   - 2.6 Mutation
   - 2.7 Calcul des critères pour les nouvelles solutions

Utilisation du NSGAII pour le design de microgrid

Les points pratiques à retenir :

- La taille recommandée de la population est de $n_{ind} = 100$ individus (d'autres valeurs peuvent bien sûr être explorée)
- Le nombre de génération doit être suffisant pour converger ($n_{gen} \geq n_{ind}$)
- Notre implémentation de l'algorithme va chercher à minimiser les différents objectifs. Si certains objectifs doivent être maximisés, cela peut être accompli par un changement de signe dans la fonction objectif.
- La liste des éléments à fournir est : Un microgrid, les scénarios associés, les bornes des variables, le dictionnaire de correspondance des variables, une fonction objectif, la taille de la population, le nombre de génération et enfin un controller.*
- Une fois les solutions obtenus il est possible (comme dans le script d'exemple) de sélectionner une solution et de la mettre en oeuvre via le designer manuel.

Sommaire

L'exemple du compresseur

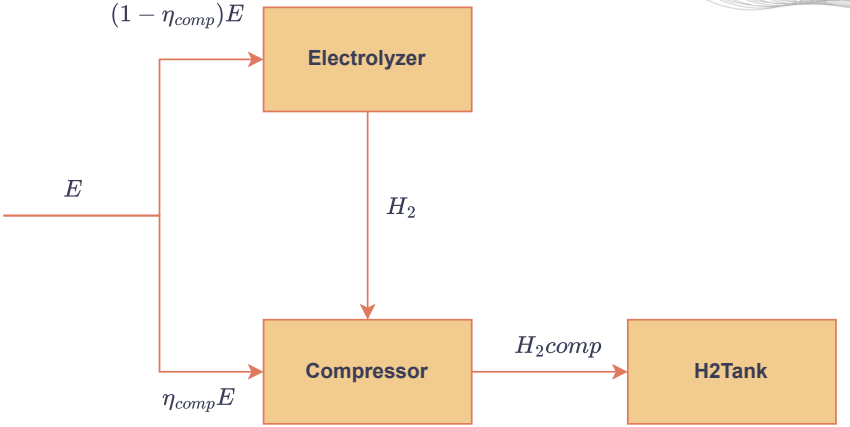

Figure – Intégration énergétique du compresseur

Problématique associée à la modélisation

Équation

L'équation de la puissance nécessaire au compresseur est de la forme suivante :

$$P_{H2}^{cmp} = \frac{Cp_{H2} \times T_{amb}}{\eta_{H2}^{cmp}} \times [(\frac{P_{stockage}}{P_{elyz}})^{\frac{\gamma-1}{\gamma}} - 1] \times m_{H2}^{dot} \quad (6)$$

Avec P_{H2}^{cmp} en [W] la puissance requise par le compresseur, $P_{stockage}$ la pression dans le H2Tank et m_{H2}^{dot} le débit de dihydrogène en [kg/s].

Problématique

Les variables en bleu (le reste de l'équation est ici considéré constant) sont nécessaires au calcul de répartition de la puissance entre électrolyseur et compresseur. Il faut donc réaliser une interaction logicielle qui permet de déterminer la répartition des puissanceS lors de l'utilisation de l'électrolyseur pour cela compresseur, H2tank et électrolyzeur doivent échanger différentes informations

Intégration logicielle

Les étapes :

1. Création d'une structure `Compresseur` (ou plusieurs type de compresseur) et d'un super-type `AbstractAuxiliary`.
2. Redéfinition des structures de modèle d'efficacité pour l'électrolyseur avec l'ajout d'un compresseur.
3. Initialisation du tableau qui recueille la valeur de pression actuelle du tank (1 valeur par scénario).
4. Ajout de la communication entre électrolyseur et H2Tank dans la boucle interne de simulation. On ajoute ce partage d'information entre `update_operation_informations` et `compute_operation_decisions`.
5. Définition de la fonction de partage d'informations `share_info_bw_assets!(h, y, s, mg)`.
6. Définition de la fonction de pression du tank `get_pression`.
7. Définition de la fonction de calcul de la part d'énergie allouée au compresseur pour chaque type de compresseur `get_compressor_share(power_E::Float64, compresseur::Compresseur_Fix, s::Int64)`.
8. Définition de la fonction de calcul du rendement de l'électrolyseur en prenant en compte la part du compresseur `compute_operation_efficiency` .

Intégration logicielle

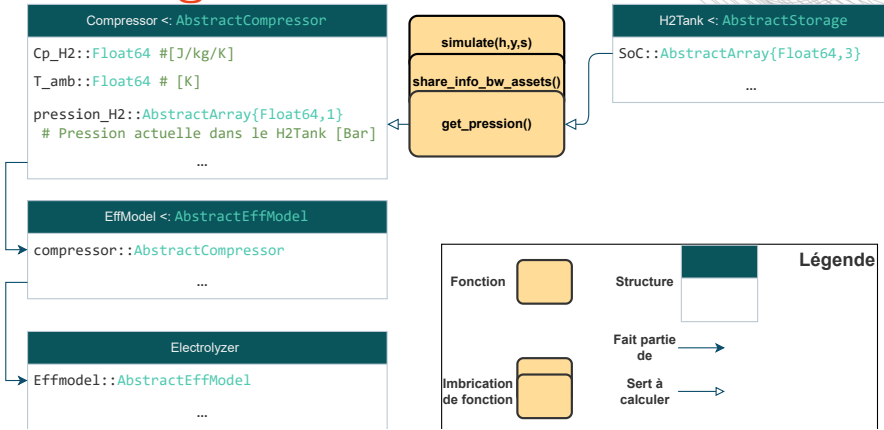

Figure – Intégration du compresseur et de ses interactions. Les champs des structures sont non exhaustifs.