

UnitCommitment Project

Thomas Wawrzyniak, Enzo Sawaya

January 2026

1 Introduction

Dans ce projet, nous aborderons le problème de l'Unit Commitment (UC), qui consiste à déterminer le programme optimal de production d'un système électrique mixte, afin de satisfaire la demande horaire tout en minimisant le coût de production.

Nous considérons deux types d'actifs : des centrales thermiques et des systèmes hydroélectriques en cascade, tous deux associés à différentes contraintes.

Le problème sera formulé comme un problème linéaire mixte entier (MILP), que nous résolverons à l'aide du langage Pyomo et du solveur HiGHS, permettant d'obtenir un programme de production réalisable et économiquement optimal sur l'horizon considéré.

Part I

Formulation du problème

2 Définition des ensembles de contraintes

Chaque actif, thermique comme hydraulique, est associé à des contraintes liées à son fonctionnement réel. Cette partie s'attache à introduire les paramètres et les contraintes permettant de modéliser au mieux ces actifs.

2.1 Centrales thermiques

On considère une centrale thermique $g \in \mathcal{G}$ sur un horizon de temps $t = 1, \dots, T$. Afin de la modéliser au mieux, il est d'abord nécessaire d'introduire des variables que nous présentons ci-dessous.

2.1.1 Variables

Pour chaque instant t , on définit les variables suivantes :

$u_t \in \{0, 1\}$	État de la centrale (ON/OFF)
$p_t \geq 0$	Puissance produite (MW)
$y_t \in \{0, 1\}$	Variable de démarrage (startup)
$z_t \in \{0, 1\}$	Variable d'arrêt (shutdown)

Remarques sur les variables de démarrage et d'arrêt La variable de démarrage y_t indique si une centrale est mise en marche à l'instant t . Elle vaut 1 uniquement lorsque l'unité passe de l'état éteint à l'état allumé entre $t-1$ et t , et elle est donc directement liée aux variables d'engagement u_{t-1} et u_t . Elle est essentielle pour modéliser les coûts de démarrage.

De même, la variable d'arrêt z_t indique si la centrale s'arrête à l'instant t , c'est-à-dire lorsqu'elle passe de l'état allumé à l'état éteint.

Ensemble, les variables y_t et z_t permettent non seulement de suivre précisément les transitions d'état des unités, mais aussi d'exprimer correctement les contraintes de temps minimum de fonctionnement (min ON) et de temps minimum d'arrêt (min OFF), comme détaillé ci-dessous.

2.1.2 Contraintes

Chacune des variables précédentes est soumise à des contraintes afin de simuler au mieux la physique d'une centrale thermique réelle. On recense notamment des contraintes sur les bornes de puissance, le ramping et les temps minimum de fonctionnement/arrêt. Celles-ci sont décrites plus en détail ci-après.

Bornes de puissance

Pour chaque instant t :

$$p_t \leq p_t^{\max} u_t \quad (1)$$

$$p_t \geq p_t^{\min} u_t \quad (2)$$

On remarque l'utilisation de la variable discrète u_t , permettant d'imposer une puissance nulle si la centrale est éteinte.

Transitions ON/OFF

Pour les instants $t = 2, \dots, T$:

$$u_t - u_{t-1} = y_t - z_t \quad (3)$$

Comme $y_t \in \{0, 1\}$ et $z_t \in \{0, 1\}$, la formulation ci-dessus suffit à définir $y_t = 1$ lors d'un démarrage ou $z_t = 1$ lors d'un arrêt au temps t .

Contraintes de ramping

Les contraintes de ramping suivantes modélisent respectivement l'augmentation maximale de puissance par time step et la diminution maximale de puissance par time step.

$$p_t - p_{t-1} \leq RU_t \Delta t u_{t-1} + p_{\min} y_t \quad \forall t = 2, \dots, T \quad (4)$$

$$p_{t-1} - p_t \leq RD_t \Delta t u_t + p_{\max} z_t \quad \forall t = 2, \dots, T \quad (5)$$

$$(6)$$

Le second terme de la première équation permet d'autoriser le passage de la centrale à sa puissance minimale, même si la variation de puissance demandée est supérieure à la limite de ramping-up.

Pour la deuxième contrainte, cela se discute : si l'on met p_{\max} , la centrale peut s'arrêter brutalement à n'importe quel niveau de puissance. Cela est possible en la déconnectant du réseau, mais engendre probablement une perte économique pour le producteur. Si l'on met p_{\min} , alors la centrale ne peut s'éteindre que lorsque la puissance est redescendue à p_{\min} . Il peut être pertinent d'utiliser $p_{\min} + \varepsilon$ afin d'éviter des problèmes numériques.

Temps minimum de fonctionnement (τ^+)

Une fois la centrale allumée, celle-ci ne peut pas être éteinte immédiatement. Elle doit fonctionner pendant au moins τ^+ pas de temps. Cette contrainte est modélisée par l'équation suivante :

$$\sum_{k=t}^{t+\tau^+-1} u_k \geq \tau^+ y_t \quad \forall t = 1, \dots, T - \tau^+ + 1 \quad (7)$$

Lorsque la centrale démarre, $y_t = 1$. On impose alors que la somme des u_k sur les τ^+ pas de temps suivants soit au moins égale à τ^+ . En effet, la somme des u_k correspond au nombre de pas de temps pendant lesquels la centrale est en fonctionnement.

Remarque importante : Avec cette écriture, la contrainte n'est pas active pour les pas de temps compris entre $T - \tau^+$ et T . Il faudra chercher à l'améliorer par la suite.

Temps minimum d'arrêt (τ^-)

De la même manière que pour le temps minimum de fonctionnement, il existe une contrainte de temps minimum d'arrêt :

$$\sum_{k=t}^{t+\tau^--1} (1 - u_k) \geq \tau^- z_t \quad \forall t = 1, \dots, T - \tau^- + 1 \quad (8)$$

2.2 Système hydraulique en cascade

Le système est représenté par un graphe orienté $G = (V, A)$ où chaque sommet $v \in V$ représente un réservoir et chaque arc $a \in A$ représente une conduite permettant soit une opération de pompage, soit de turbinage entre deux réservoirs. L'arc sortant du dernier réservoir se déverse dans la rivière, dont nous ne contrôlons pas le volume.

2.2.1 Variables

Pour chaque réservoir $v \in V$ et instant t :

$$V_{v,t} \quad \text{volume stocké (hm}^3\text{)}$$

Pour chaque arc $a \in A$ et instant t :

$$\begin{aligned} f_{a,t} & \quad \text{débit (hm}^3\text{/h)} \\ p_{a,t} & \quad \text{puissance (MW)} \end{aligned}$$

2.2.2 Contraintes

Comme pour les centrales thermiques, les variables modélisant le système en cascade sont soumises à plusieurs contraintes. Notamment : des bornes minimales et maximales sur les débits des arcs, des contraintes de rampe sur ces débits, ainsi que des puissances minimales et maximales pour les turbines.

Les volumes des réservoirs sont également contraints, et une équation de bilan de masse décrit l'évolution du volume de chaque réservoir en fonction des apports naturels (*inflows*, par exemple la pluie) ainsi que des débits entrants et sortants.

Chaque turbine est en outre modélisée par une enveloppe (linéarisée par morceaux) permettant de représenter la relation entre le débit qui la traverse et la puissance produite.

Bornes sur les volumes

Pour chaque réservoir $v \in V$ et instant t :

$$V_{v,t}^{\min} \leq V_{v,t} \leq V_{v,t}^{\max} \quad (9)$$

Bilan de masse

Pour chaque réservoir $v \in V$ et instant $t = 1, \dots, T-1$:

$$V_{v,t+1} = V_{v,t} + \Delta t \left(a_{v,t} + \sum_{a \in In(v)} f_{a,t} - \sum_{a \in Out(v)} f_{a,t} \right) \quad (10)$$

Le bilan de masse traduit les variations de volume dans chaque réservoir à chaque instant t . Ces variations sont dues aux inflows, ainsi qu'aux flux entrants et sortants dans chaque réservoir.

Volumes initiaux

Pour chaque réservoir $v \in V$:

$$V_{v,1} = V_{v,0} \quad (11)$$

Bornes sur les débits

Pour chaque arc $a \in A$ et instant t :

$$\underline{f}_{a,t} \leq f_{a,t} \leq \bar{f}_{a,t} \quad (12)$$

On notera que le signe des bornes définit le type d'arc (turbine ou pompe). Un flux négatif signifie que l'arc est une pompe tandis qu'un flux positif signifie que c'est une turbine.

Ramping des débits

Pour chaque arc $a \in A$ et instant $t = 2, \dots, T$:

$$f_{a,t} - f_{a,t-1} \leq RU_a \Delta t \quad (13)$$

$$f_{a,t-1} - f_{a,t} \leq RD_a \Delta t \quad (14)$$

Avec cette formulation, le débit au premier pas de temps n'est pas limité par une contrainte de rampe. Il est donc choisi librement par le solveur entre sa borne inférieure et sa borne supérieure.

Fonction de production des turbines

Pour chaque arc a modélisant une turbine et instant $t \in T$:

$$p_{a,t} \leq p_j + \rho_j f_{a,t} \quad \forall j \in J \quad (15)$$

Cette contrainte impose uniquement une limite physique supérieure à la puissance produite par la turbine. Imposer directement $p(f) = \min(\cdot)$ n'est pas possible dans un MILP, car l'opérateur minimum introduirait une formulation non linéaire.

À l'optimum, la contrainte d'équilibre offre-demande, combinée à la minimisation des coûts, pousse naturellement la production hydraulique à atteindre cette borne maximale autorisée. En effet, l'hydroélectricité ayant dans ce cadre un coût nulle, le programme cherchera à utiliser au maximum la puissance hydraulique disponible lorsque cela est possible.

Bornes de puissance

Pour chaque arc $a \in A$ et instant $t \in T$:

$$\underline{p}_{a,t} \leq p_{a,t} \leq \bar{p}_{a,t} \quad (16)$$

Chaque turbine et chaque pompe sont également soumises à des contraintes de puissance. Pour une pompe, la puissance maximale est nulle, tandis que la puissance minimale est négative.

Pompes

Pour chaque arc a modélisant une pompe et instant $t \in T$:

$$p_{a,t} = \rho_a f_{a,t} \quad (17)$$

La relation entre le débit et la puissance pour les pompes est linéaire, ce qui autorise une contrainte d'égalité.

2.3 Contrainte d'équilibre offre-demande

$$\sum_i P_{i,t} \geq d_t \quad \forall t \in T, \forall \text{ actif } i \quad (18)$$

La contrainte d'équilibre offre-demande est indispensable pour assurer que la production et la demande coïncide en tout instant t . On vient ici imposer une inégalité pour s'assurer que la production atteigne au minimum la demande. En réalité, le modèle ne va jamais produire plus car cela reviendrait à augmenter la fonction objectif.

3 Formulation MILP du problème UC

L'ensemble des variables thermiques et hydrauliques est regroupé dans un vecteur x , comprenant notamment $p_{g,t}, u_{g,t}, y_{g,t}, z_{g,t}$ pour $g \in \mathcal{G}$ ainsi que $V_{v,t}, f_{a,t}, p_{a,t}$ pour $v \in V, a \in A$. Le problème complet d'Unit Commitment peut alors s'écrire sous la forme suivante.

Problème UC (MILP).

$$\min_x \quad \sum_{t=1}^T \sum_{g \in \mathcal{G}} (c_{g,t} \Delta t p_{g,t} + s_{g,t} y_{g,t}) \quad (19)$$

$$\text{s.c.} \quad x \in X^{\text{th}} \times X^{\text{hyd}} \quad (20)$$

$$\sum_{g \in \mathcal{G}} p_{g,t} + \sum_{a \in A} p_{a,t} \geq d_t \quad \forall t = 1, \dots, T \quad (21)$$

où X^{th} désigne l'ensemble des contraintes associées aux centrales thermiques (bornes de puissance, transitions ON/OFF, ramping, minimum up/down times, etc.) et X^{hyd} l'ensemble des contraintes du système hydraulique en cascade (bilans de masse, bornes de volume, bornes et ramping des débits, fonctions de production, etc.).

Remarque : Dans (21), la somme $\sum_{a \in A} p_{a,t}$ inclut les turbines (contribution positive) et les pompes (contribution négative), ce qui permet de représenter naturellement la consommation électrique des pompes dans l'équilibre offre-demande.

Part II

Implémentation numérique du problème

4 Données d'entrées

Afin de tester notre modèle, nous avons utilisé les données de centrales hydro-électriques et thermiques disponibles sur Internet. Chaque centrale est modélisée comme une UnitBlock pour laquelle on fournit différentes données :

4.1 Centrales thermiques

Chaque centrale thermique est décrite par les paramètres suivants :

- double MinPower : p_t^{\min}
- double MaxPower : p_t^{\max}
- double DeltaRampUp ; RU_t
- double DeltaRampDown ; RD_t
- double QuadTerm : terme quadratique de la fonction coût
- double LinearTerm : terme linéaire de la fonction coût
- double ConstTerm : terme constant de la fonction coût
- double StartUpCost : c_{startup}
- double InitialPower : p_1
- int64 InitUpDownTime : temps depuis lequel la centrale est OFF/ON
- uint64 MinUpTime : τ^+
- uint64 MinDownTime : τ^-

4.2 Centrales hydro-électriques

Quant aux centrales hydro-électriques, on retrouve les paramètres suivants :

- int StartArc(NumberArcs) : réservoir de départ de chaque arc
- int EndArc(NumberArcs) : réservoir d'arrivée de chaque arc
- double Inflows(NumberReservoirs, One) : $a_{v,t}$
- double InitialVolumetric(NumberReservoirs) : $V_{v,0}$
- double MaxVolumetric(NumberReservoirs, One) : $V_{v,t}^{\max}$
- int InitialFlowRate(NumberArcs) : $f_{a,0}$
- double DeltaRampUp(One, NumberArcs) : RU_a
- double DeltaRampDown(One, NumberArcs) : RD_a
- double MinFlow(One, NumberArcs) : $\underline{f}_{a,t}$

- `double MaxFlow(One, NumberArcs) : $\bar{f}_{a,t}$`
- `double MinPower(One, NumberArcs) : $\underline{p}_{a,t}$`
- `double MaxPower(One, NumberArcs) : $\bar{p}_{a,t}$`
- `int NumberPieces(NumberArcs) : nombre de segments de chaque arc`
- `double LinearTerm(TotalNumberPieces) : ρ_j`
- `double ConstantTerm(TotalNumberPieces) : p_j`

5 Structure des données pour la création du modèle

Afin de pouvoir construire le modèle, nous avons créé une fonction permettant de récupérer des données à partir de fichiers externes (notamment au format NetCDF4) et de les traiter afin de les structurer sous la forme d'un dictionnaire Python facile à utiliser. La structure de ce dictionnaire est détaillée ci-dessous.

5.1 Temps et ensembles

- **Temps :**
 - `data["time"]["T"]` : horizon de temps T (nombre de pas de temps)
 - `data["time"]["delta_t"]` : durée de chaque pas de temps (heures)
- **Ensembles :**
 - `data["SETS"]["T"]` : ensemble des pas de temps $t = 1, \dots, T$
 - `data["SETS"]["G"]` : ensemble des unités thermiques
 - `data["SETS"]["V"]` : ensemble des réservoirs
 - `data["SETS"]["A"]` : ensemble des arcs (turbines et pompes)
 - `data["SETS"]["A_turb"]` : sous-ensemble des arcs turbines
 - `data["SETS"]["A_pump"]` : sous-ensemble des arcs pompes
 - `data["SETS"]["J"]` : segments des turbines (pour HPF piecewise)

5.2 Demande électrique

Pour chaque time step $t \in T$:

- `data["demand"][t]` : demande totale (MW) au pas de temps t

5.3 Unités thermiques

Pour chaque centrale $g \in G$ et time step $t \in T$

- `data["thermal"]["p_min][(g,t)]` : puissance minimale (MW)
- `data["thermal"]["p_max][(g,t)]` : puissance maximale (MW)
- `data["thermal"]["const_cost][(g,t)]` : coût constant de production (€/MW)
- `data["thermal"]["lin_cost][(g,t)]` : coût linéaire de production (€/MW)
- `data["thermal"]["startup_cost"][g]` : coût de démarrage (€)

- `data["thermal"]["RU"][g]` : ramp-up maximal (MW/h)
- `data["thermal"]["RD"][g]` : ramp-down maximal (MW/h)
- `data["thermal"]["min_up"][g]` : temps minimum ON (pas de temps)
- `data["thermal"]["min_down"][g]` : temps minimum OFF (pas de temps)

5.4 Système hydroélectrique

5.4.1 Réservoirs

Pour chaque réservoir $v \in V$ et time step $t \in T$:

- `data["reservoirs"]["V0"][v]` : volume initial (hm^3)
- `data["reservoirs"]["Vmin"][v]` : volume minimal (hm^3)
- `data["reservoirs"]["Vmax"][v]` : volume maximal (hm^3)
- `data["reservoirs"]["inflow"][(v,t)]` : débit entrant naturel (hm^3/h)

5.4.2 Arcs (turbines et pompes)

Pour chaque arc $a \in A$ et time step $t \in T$:

- `data["arcs"]["from"][a]` : réservoir source
- `data["arcs"]["to"][a]` : réservoir destination
- `data["arcs"]["f_min"][a]` : débit minimal (hm^3/h)
- `data["arcs"]["f_max"][a]` : débit maximal (hm^3/h)
- `data["arcs"]["p_min"][a]` : puissance minimale (MW)
- `data["arcs"]["p_max"][a]` : puissance maximale (MW)
- `data["arcs"]["RU"][a]` : ramp-up débit ($(\text{hm}^3/\text{h})/\text{h}$)
- `data["arcs"]["RD"][a]` : ramp-down débit ($(\text{hm}^3/\text{h})/\text{h}$)
- `data["arcs"]["p"][(a,j)]` : constante du segment j de la turbine/pompe (MW)
- `data["arcs"]["r"][(a,j)]` : pente du segment j de la turbine/pompe ($\text{MW}/(\text{hm}^3/\text{h})$)

5.4.3 Graphe du réseau hydraulique

Pour chaque réservoir $v \in V$:

- `data["graph"]["In"][v]` : arcs entrants vers le réservoir v
- `data["graph"]["Out"][v]` : arcs sortants depuis le réservoir v

6 Création du modèle et résolution du problème

Une fois le dictionnaire créé, nous avons créé le modèle grâce à la fonction *build_model.py*. Nous avons décidé d'utiliser le module Pyomo afin de construire le modèle en Python. On définit l'ensemble des paramètres du modèle, la fonction objectif, les variables de décision, ainsi que toutes les contraintes définies ci-dessus. Le modèle est ensuite résolu par le solveur HiGHS.

Part III

Résultats

Nous présentons ici une partie de nos résultats. D'autres résultats peuvent être visualisés dans le dossier `outputs` de notre projet. Les dossiers `plots.T` contiennent des visualisations correspondant à différentes simulations réalisées sur un horizon de temps T .

Il est également possible de visualiser un GIF représentant le graphe des réservoirs, où l'on peut observer ces derniers se remplir et se vider au travers des arcs au cours du temps.

Par ailleurs, les données utilisées ont été obtenues par agrégation de différentes sources citées en référence. Les données relatives aux centrales thermiques proviennent du dépôt GitLab [2], tandis que celles concernant les centrales hydrauliques sont issues des travaux de Matej Lalik [1].

7 Test 1 : configuration simple

Nous avons testé notre modèle sur la configuration suivante :

- 10 centrales thermiques
- 2 centrales hydro-électriques (6 réservoirs / 12 arcs & 2 réservoirs / 4 arcs)

Le premier graphique obtenu représente la production thermique et hydraulique, ainsi que l'équilibre offre-demande.

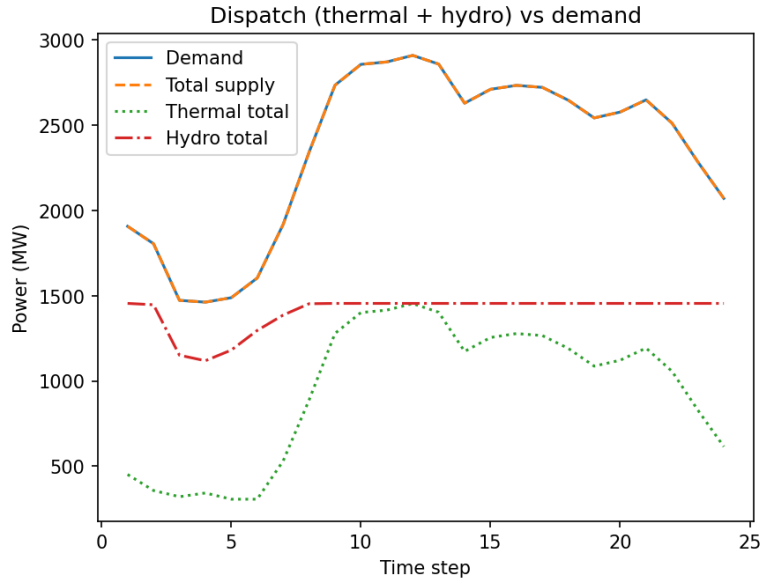


Figure 1: Dispatch thermique / hydro et équilibre offre-demande

On observe que l'équilibre offre-demande est respecté, et assuré par la production thermique et hydro. La production hydro est privilégiée car elle n'a pas de coût associé. Lorsqu'on atteint la production maximale, elle est complétée par le thermique afin d'assurer l'équilibre.

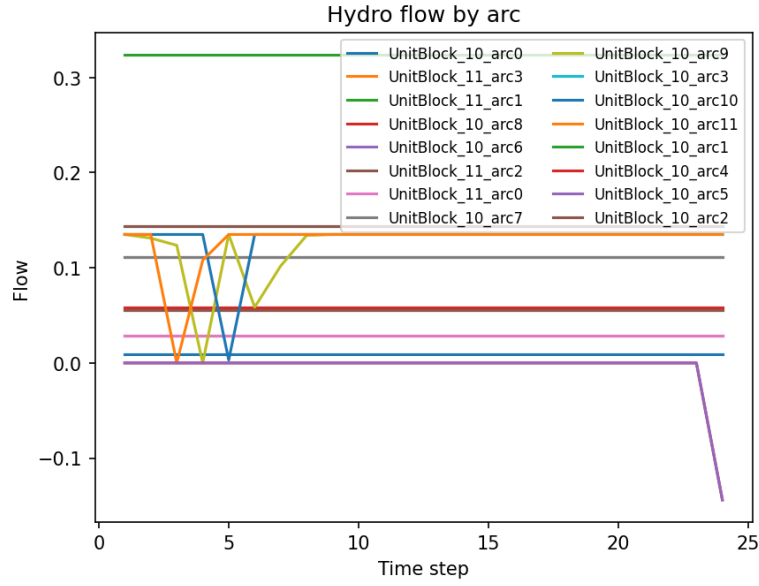


Figure 2: Flux de chaque arc (en hm^3/h)

Le deuxième graphique représente le flux hydro sur chaque arc des deux centrales. On observe que le flux est constant et atteint sa valeur maximale. Cela est cohérent car l'hydro n'a pas de coût donc on vient produire au maximum avec chaque unité. Lorsque la demande baisse, les flux de quelques arcs deviennent nuls pour ajuster l'équilibre.

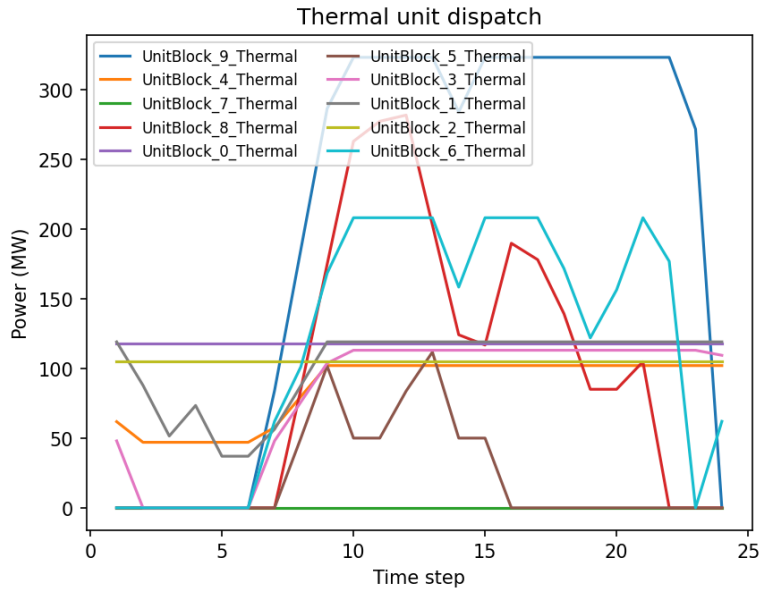


Figure 3: Production des différentes centrales thermiques

Enfin nous avons représenté la production de chacune des centrales thermiques. Lorsque la demande est faible, toutes les unités ne fonctionnent pas. Puis lorsque la demande augmente et l'hydro ne peut pas produire davantage, les autres centrales thermiques s'allument et produisent (pour certaines à leur production maximale).

8 Test 2 : Augmentation de la temporalité et contraintes

Après analyse des résultats de la configuration 1, nous avons décidé de modifier les contraintes de volume minimal pour les réservoirs ainsi que d'étudier une plus longue période. En effet, les réservoirs ne se vidaient que très peu. De plus nous avons tenté de modéliser une vague de froid durant les premiers jours, se traduisant par une variation de la demande.

On obtient les résultats suivants :

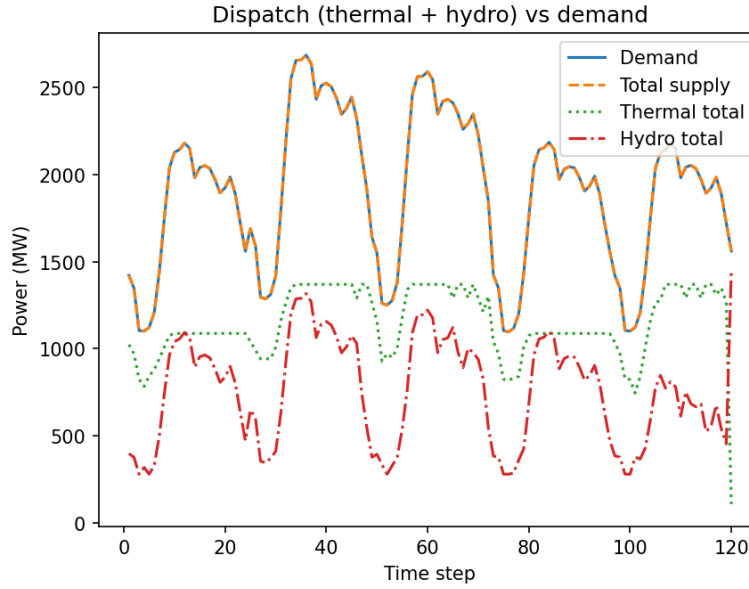


Figure 4: Dispatch thermique / hydro et équilibre offre-demande

On observe encore une fois que l'équilibre est respecté. L'hydro et le thermique produisent tous les deux, mais il semblerait que la production hydro diminue vers la fin, traduisant l'épuisement progressif des réservoirs. Nous pouvons ainsi représenter le volume des réservoirs au cours du temps :

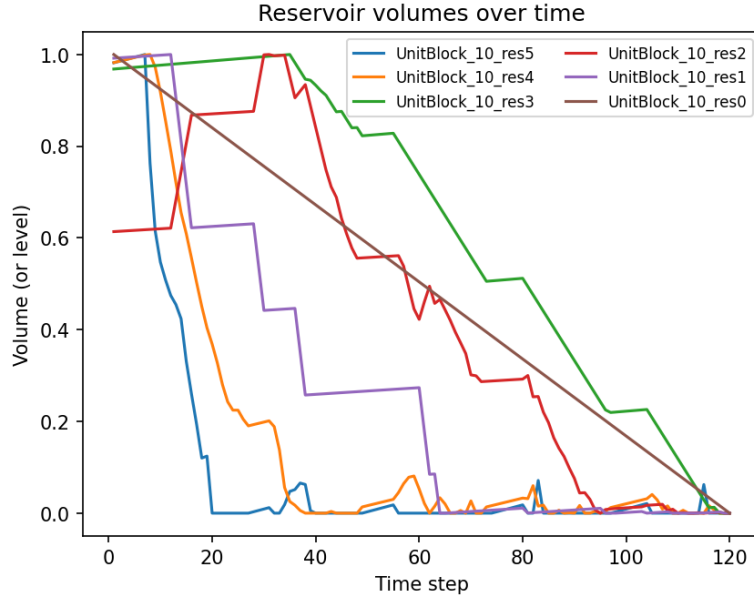


Figure 5: Volume des réservoirs au cours du temps

Afin de faciliter la lecture, nous avons mis à l'échelle le graphe en représentant en ordonnée :

$$\frac{V - V_{\min}}{V_{\max} - V_{\min}}.$$

Il semblerait que les volumes des réservoirs se vident au cours du temps pour produire de l'électricité, jusqu'à atteindre leur valeur minimale à la fin de la simulation. On peut d'ailleurs observer les relations entre les différents réservoirs. En effet, le réservoir 2 est directement lié au réservoir 4. Ainsi jusqu'à ce que le réservoir 4 se vide complètement, le réservoir 2 se remplit.

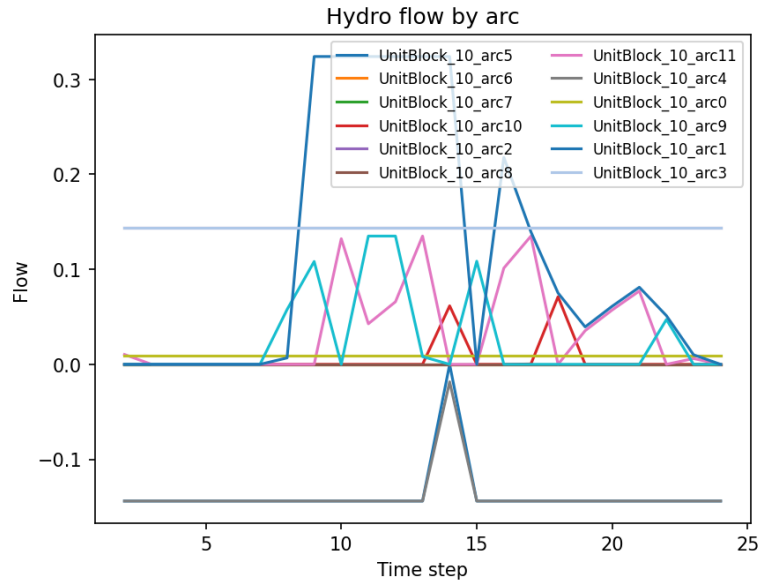


Figure 6: Volume des réservoirs au cours du temps

Enfin, nous avons représenté les flux de chaque arc sur une période de 24 heures. Lorsque la demande est importante, les flux semblent atteindre leur valeur maximale. Il est aussi intéressant de remarquer que certains arcs sont en mode pompage (flux négatif)

References

- [1] NVE Database Spreadsheet (Hydro data), <https://gitlab.com/lalikm1999/smspp-hydro-units/-/blob/main/NVE>
- [2] SMS++ Project – UCBlock data (thermal), https://gitlab.com/smspp/ucblock/-/tree/develop/netCDF_files/UCData/T-Ramp,
- [3] W. van Ackooij, I. Danti Lopez, A. Frangioni, F. Lacalandra, and M. Tahanan, *Large-scale unit commitment under uncertainty: an updated literature survey*, Annals of Operations Research, vol. 271, no. 1, pp. 11–85, 2018.