Programmation dynamique en deux phases pour résoudre le Hydro Unit Commitment à une usine

Alexandre Heintzmann[1,2], Christian Artigues[2], Pascale Bendotti[1], Sandra Ulrich Ngueveu[2], Cécile Rottner[1]

[1] EDF Lab Paris-Saclay, 7 Bd. Gaspard Monge, 91120 Palaiseau, France
{alexandre.heintzmann,pascale.bendotti,cecile.rottner}@edf.fr

[2] LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
{alexandre.heintzmann,christian.artigues,sandra.ulrich.ngueveu}@laas.fr

Mots-clés : *programmation dynamique, programmation multi-objectif, Hydro Unit Commitment.*

Le **HUC** est un problème de planification journalière de production électrique propre aux usines hydroélectriques. Ce problème s'inscrit dans le problème de planification de production électrique, nommé **Unit Commitment Problem** (**UCP**). L'**UCP** est résolu aujourd'hui par EDF par une décomposition Lagrangienne, produisant un sous-problème par unité de même nature : thermique, hydraulique (**HUC**), solaire... Plus précisément, le cas étudié est celui du **HUC** à une usine, noté **1-HUC**. Ses caractéristiques sont les suivantes. L'usine est située entre deux réservoirs. Le temps est discrétisé en T pas de temps. L'usine possède N points de fonctionnement, qui sont des couples de puissance générée et de débit d'eau utilisé. L'eau débitée par l'usine passe du réservoir amont au réservoir aval au cours du même pas de temps. Plusieurs contraintes sont considérées. Chaque réservoir possède un volume initial, ainsi qu'un volume maximal et minimal par pas de temps. L'usine présente des contraintes de gradient, limitant la variation de débit entre deux pas de temps consécutifs. L'usine comporte aussi des contraintes de palier avant changement de sens : lorsqu'il y a une augmentation de débit, le débit doit rester au moins aussi élevé pendant Θ pas de temps, et inversement lorsqu'il y a une diminution du débit. Additionnellement, à chaque pas de temps, il y a un apport extérieur en eau dans les réservoirs, positif, négatif ou nul. Le cas d'un problème de maximisation de revenu, pour des prix donnés, est considéré. Les prix correspondent à la valeur unitaire de l'énergie produite, variable à chaque pas de temps, et la valeur unitaire de l'eau à la fin de l'horizon de temps. Ces prix proviennent de la décomposition Lagrangienne de l'**UCP**.

Ce problème peut être représenté graphiquement. Soit un graphe G, dans lequel un sommet est représenté par $(t, i, \theta,$haut/bas), avec t le pas de temps, i le point de fonctionnement, θ la durée restante de la contrainte de palier, et "haut" ou "bas" indiquant si on peut accéder aux sommets de point de fonctionnement $i' \geq i$ ou $i' \leq i$ au pas de temps suivant. Un arc est ajouté à G d'un sommet du pas de temps t vers un sommet du pas de temps $t+1$, si les contraintes de gradient et de palier sont satisfaites. Dans un tel graphe, possédant $2TN\Theta$ sommets, toutes les contraintes du problème sont représentées, mis à part les contraintes de volume. En considérant le volume comme une ressource, alors le 1-HUC peut se ramener à trouver un plus long chemin avec ressource dans le graphe G [1]. Etant donné que G est un graphe orienté sans circuit (DAG), le problème du plus long chemin est équivalent à un problème de plus court chemin. Ainsi, le 1-HUC revient à un problème de plus court chemin avec contrainte de ressource (RCSPP), qui peut être résolu par programmation dynamique, avec un algorithme d'extension d'étiquette (labelling). L'algorithme de labelling s'appuie sur des dominances entre solutions partielles qui ne peuvent pas être appliquées tant que la borne inférieure de la ressource n'est pas atteinte par ces solutions. La difficulté dans le cas du 1-HUC vient du fait que la ressource est bornée supérieurement mais aussi inférieurement. Il existe un certain nombre de cas pathologiques du 1-HUC où l'algorithme de labelling du RCSPP ne peut pas appliquer de dominances rapidement, menant à un nombre de solutions partielles à explorer très élevé et donc à des temps de calcul prohibitifs.

En considérant le volume comme un second objectif, le problème considéré peut être vu comme un problème de plus court chemin bi-objectif. L'idée de la méthode proposée est d'adapter l'algorithme en deux phases, proposé initialement pour le problème du sac-à-dos bi-objectif [4]. Initialement, cet algorithme énumère les solutions non-dominées, c'est-à-dire des solutions meilleures que n'importe quelle autre solution sur au moins un objectif. La première phase de cet algorithme permet d'obtenir les solutions supportées en résolvant le problème mono-objectif agrégeant les objectifs par combinaison convexe. Connaissant les solutions supportées, des sous-espaces de solutions sont définis, garantissant que toutes les solutions non-dominées se situent dans un de ces espaces. La seconde phase consiste à énumérer des solutions non-dominées de ces sous-espaces. L'algorithme est davantage décrit dans [2].

Plusieurs raisons nous ont menés à adapter cet algorithme. Premièrement, dans le cas du 1-HUC, on souhaite en réalité uniquement obtenir une seule solution : la solution maximisant le profit, en respectant les contraintes sur le volume. Ainsi, on peut orienter la première phase pour générer un seul sous-espace, contenant la solution voulue, puis limiter l'énumération de la seconde phase à ce seul sous-espace. Deuxièmement, l'algorithme en deux phases de la littérature est très efficace sur les problèmes multi-objectifs lorsque le problème mono-objectif associé est facile à résoudre, ce qui est le cas d'un problème de plus court chemin. En effet, la première phase consiste à résoudre plusieurs instances du problème mono-objectif associé. De plus, l'énumération où on cherche une unique solution peut être effectuée par un algorithme des K-meilleurs [3] pour le problème mono-objectif associé. Lors de l'énumération, dès qu'une solution réalisable est trouvée, on peut réduire l'espace de recherche, ce qui accélère l'obtention de la solution optimale du **1-HUC**.

Des expérimentations numériques préliminaires ont été effectuées, comparant le temps de résolution de la méthode en deux phases avec un modèle MILP résolu par CPLEX. Nous avons mis en avant des instances avec une faible variation des prix, difficilement résolues par CPLEX. Pour ces instances, l'algorithme en deux phases est en général plus efficace que CPLEX. Dans le cas de variabilité plus grande, CPLEX résout quasi-instantanément les instances. L'algorithme en deux phases est alors plus lent, mais ne nécessite généralement que quelques secondes. Ces résultats montrent une robustesse de l'algorithme en deux phases vis-à-vis du temps de calcul supérieur à CPLEX. Sur un jeu d'une centaine d'instances, CPLEX nécessite un temps total deux à trois fois supérieur à celui de l'algorithme en deux phases.

Références

[1] Wim van Ackooij, Claudia d'Ambrosio, Dimitri Thomopulos et Renan Spencer Trindade. “Decomposition and shortest path problem formulation for solving the hydro unit commitment and scheduling in a hydro valley”. In : *European Journal of Operational Research* 291.3 (2021), p. 935-943.

[2] Alexandre Heintzmann, Christian Artigues, Pascale Bendotti, Sandra Ulrich Ngueveu et Cécile Rottner. “Dynamic programming for the single unit hydro unit commitment problem”. working paper or preprint. Déc. 2022. url : https://hal.laas.fr/hal-03916388.

[3] Eugene L Lawler. “A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem”. In : *Management science* 18.7 (1972), p. 401-405.

[4] Marc Visée, Jacques Teghem, Marc Pirlot et Ekunda L Ulungu. “Two-phases method and branch and bound procedures to solve the bi–objective knapsack problem”. In : *Journal of Global Optimization* 12.2 (1998), p. 139-155.