Méthode exacte dirigée par une relaxation bi-objectif pour le problème du plus court chemin avec fenêtres de ressource

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

[1] LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
[2] EDF Lab Paris-Saclay, 7 Bd. Gaspard Monge, 91120 Palaiseau, France

Mots-clés : *Algorithme de graphe, méthode deux-phases, application au HUC.*

Dans ce papier, nous considérons le problème du plus court chemin avec fenêtres de ressource (WRSPP). Le WRSPP est une généralisation du problème de plus court chemin avec contrainte de ressource (RCSPP), car la quantité de ressource à utiliser est bornée supérieurement mais aussi inférieurement. La présence de ces fenêtres rend le WRSPP particulièrement difficile à résoudre. En effet, certaines dominances classiques du RCSPP ne s'appliquent pas au WRSPP comme précisé dans [1, 3] et montré numériquement dans [3]. Comme indiqué dans le survey [8], peu d'algorithmes ont été développés pour résoudre le WRSPP. Un algorithme de programmation dynamique dédié au WRSPP a été proposé dans [10], mais nécessite une extension de graphe de taille exponentielle qui le rend difficilement utilisable en pratique.

Nous proposons un nouvel algorithme pour le WRSPP dans le cadre de graphes acycliques. L'idée est de relâcher la ressource pour la considérer comme un second objectif afin de diriger une énumération en s'inspirant du principe de l'algorithme en deux-phases pour les problèmes bi-objectif [9], mais dans une zone fortement réduite grâce aux propriétés énoncées ci-après.

1 WRSPP et sa relaxation bi-objectif

Dans cette section, nous définissons trois problèmes considérés sur des graphes acycliques : le plus court chemin (SPP), le plus court chemin bi-objectif (BOSPP) et le WRSPP. Pour ce type de graphes, les variantes du SPP et du plus long chemin (LPP) sont équivalentes. Les cas d'applications définis dans la **Section 3** étant des problèmes de maximisation, les variantes sont définies en tant que plus long chemin.

Problème 1 (LPP)**.** Soit un graphe $G = (V, A)$ acyclique, avec V et A respectivement l'ensemble des sommets et des arcs de G. On note $s \in V$ et $p \in V$ respectivement les sommets source et puits de G. Chaque arc $a \in A$ a une valeur réelle $v(a)$. Pour un chemin C, on note $v(C) = \sum_{a \in C} v(a)$ la valeur du chemin. Résoudre le LPP consiste à trouver le chemin C de s à p qui maximise $v(C)$.

Problème 2 (BOLPP)**.** Le BOLPP est une généralisation du LPP, où un arc $a \in A$ a deux valeurs $v_1(a)$ et $v_2(a)$ correspondant aux deux objectifs. Pour un chemin C, on note $v_i(C) = \sum_{a \in C} v_i(a)$ avec $i \in \{1, 2\}$ la valeur sur l'objectif i. Résoudre le BOLPP consiste à trouver les chemins maximisant les deux objectifs.

Comme il y a deux objectifs, deux solutions peuvent être incomparables ce qui signifie qu'il peut exister plusieurs chemins dits Pareto-optimaux.

Définition 1 (Chemin Pareto-optimal en bi-objectif)**.** Un chemin C dans G de s à p est Pareto-optimal s'il n'existe aucun chemin C' dans G de s à p tel que $\forall i \in \{1, 2\}$, $v_i(C) \leq v_i(C')$ et $\exists i \in \{1, 2\}$ tel que $v_i(C) < v_i(C')$.

alexandre.heintzmann@{laas,edf}.fr

Problème 3 (WRLPP)**.** Le WRLPP est une extension du LPP où chaque arc a une valeur $v(a) \in \mathbb{R}$ et une consommation de ressource $r(a) \in \mathbb{R}_{\geq 0}$. Chaque sommet $u \in V$ a une fenêtre de ressource $[\beta(u); \alpha(u)]$. Pour un chemin C, on note $v(C) = \sum_{a \in C} v(a)$ la valeur et $r(C) = \sum_{a \in C} r(a)$ la quantité de ressource utilisée. Un chemin C de s à u dans G est localement faisable si $r(C) \in [\beta(u); \alpha(u)]$. Un chemin C de s à u dans G est faisable si tous les sous-chemins C' de C partant de s sont localement faisables. Résoudre le WRLPP consiste à trouver le chemin faisable C de s à p dans G qui maximise $v(C)$.

Le WRSPP diffère du plus court chemin avec fenêtre de temps (TWSPP) [4] pour lequel il est possible d'attendre l'ouverture de la fenêtre à un sommet.

Comme notre algorithme s'inspire de l'algorithme en deux phases dans le cas bi-objectif [9], nous rappelons ce dernier, adapté au cas du BOLPP. Pour celà, nous rappelons des définitions ainsi que le principe des deux phases.

Définition 2 (Chemin Pareto-supporté)**.** Un chemin Pareto-supporté est un chemin Pareto-optimal, qui fait partie de l'enveloppe convexe des solutions dans l'espace des objectifs.

Définition 3 (Valeur de l'objectif agrégé suivant δ)**.** On note $\mu_\delta(C) = \delta_1 \cdot v_1(C) + \delta_2 \cdot v_2(C)$ la valeur des objectifs agrégés suivant $\delta_1 \geq 0$ et $\delta_2 \geq 0$ pour le chemin C.

Première phase La première phase de l'algorithme consiste à trouver tous les chemins Pareto-supportés. Ces chemins peuvent tous être obtenus en résolvant le LPP avec pour objectif la valeur agrégée $\mu_\delta(.)$ pour différents vecteurs δ. On note P l'ensemble des chemins Pareto-supportés, triés par ordre croissant sur le premier objectif.

Deuxième phase Une fois toutes les solutions Pareto-supportées générées, pour chaque paire de chemins successifs de P il est possible de former une région en forme de triangle dans l'espace des objectifs. Toute solution Pareto-optimale se situe nécessairement dans une de ces régions. Ainsi, la deuxième phase de l'algorithme consiste à énumérer toutes les solutions de chacune des régions par un algorithme de Branch & Bound.

Nous présentons l'algorithme BORWin (Bi-Objective Relaxation-based algorithm for the shortest path with Resource Windows), dont l'idée est de considérer la ressource d'un chemin vers p comme un second objectif et de relâcher les fenêtres de ressource. Comme le BOLPP est moins restrictif que le WRLPP, tout chemin du WRLPP est aussi un chemin admissible du BOLPP. Nous pouvons alors obtenir le chemin optimal du WRLPP par énumération des chemins du BOLPP en nous inspirant des principes de l'algorithme en deux phases bi-objectif.

2 Algorithme BORWin

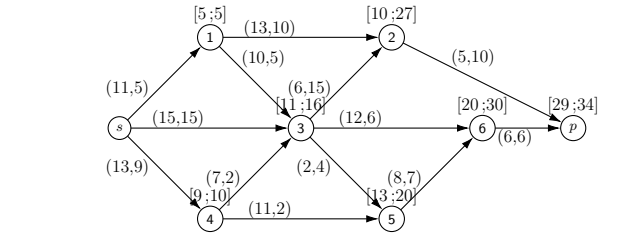
Pour la suite, nous distinguons trois cas du WRLPP : Localement Infaisable par Défaut (LID), Localement Infaisable par Excès (LIE) ; Localement Faisable (LF).

Définition 4 (Cas LID, LIE et LF du WRLPP)**.** Soit C le chemin maximisant $v(C)$ sans prendre en compte les ressources. Le cas LID est celui avec $r(C) < \beta(p)$, le cas LIE celui avec $r(C) > \alpha(p)$ et le cas LF celui avec $r(C) \in [\beta(p); \alpha(p)]$.

Les cas LID et LIE étant symétriques l'un de l'autre, dans la suite nous traiterons uniquement le cas LF et le cas LID.

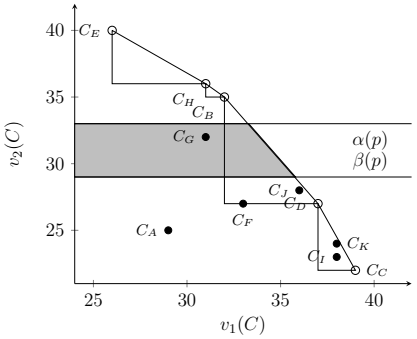
Construction du BOLPP A partir d'un WRLPP, nous définissons un BOLPP en relâchant les fenêtres de ressources, et avec les valeurs $v_1(C) = v(C)$ et $v_2(C) = r(C)$ pour un chemin C.

Prenons l'exemple pour le graphe de la **FIG. 1a** où un arc a est annoté par $(v(a), r(a))$ et un sommet u par $[\beta(u); \alpha(u)]$. De plus, les 11 chemins différents de s à p sont énumérés. Ce WRLPP est dans le cas LID, car le chemin maximisant la valeur sans prendre en compte la ressource est C_C avec $r(C_C) < \beta(p)$. La **FIG. 1b** représente chaque chemin du BOLPP dans



C_A=(1, 2), $v(C_A) = 29$, $r(C_A) = 25$ C_B=(1, 3, 2), $v(C_B) = 32$, $r(C_B) = 35$
C_C=(1, 3, 6), $v(C_C) = 39$, $r(C_C) = 22$ C_D=(1, 3, 5, 6), $v(C_D) = 37$, $r(C_D) = 27$
C_E=(3, 2), $v(C_E) = 26$, $r(C_E) = 40$ C_F=(3, 6), $v(C_F) = 33$, $r(C_F) = 27$
C_G=(3, 5, 6), $v(C_G) = 31$, $r(C_G) = 32$ C_H=(4, 3, 2), $v(C_H) = 31$, $r(C_H) = 36$
C_I=(4, 3, 6), $v(C_I) = 38$, $r(C_I) = 23$ C_J=(4, 3, 5, 6), $v(C_J) = 36$, $r(C_J) = 28$
C_K=(4, 5, 6), $v(C_K) = 38$, $r(C_K) = 24$

(a) Graphe du WRLPP



(b) Représentations des chemins du BOLPP dans l'espace des objectifs avec fenêtre $[\beta(p); \alpha(p)]$

FIG. 1 – Graphe et front de Pareto d'un exemple à 8 sommets

l'espace des objectifs, les triangles formés par les chemins Pareto-supportés ainsi que les bornes $\beta(p)$ et $\alpha(p)$. Pour cette instance, seul C_G est faisable et ne fait partie d'aucun triangle.

Comme montré par l'exemple, il n'est pas possible de s'appuyer uniquement sur les triangles pour obtenir le chemin C^* optimal du WRLPP. Cependant, il est possible d'obtenir un trapèze semi-défini (zone grisée de la **FIG. 1b**) en combinant le fait que $r(C^*) \in [\beta(p); \alpha(p)]$ avec un côté des triangles, ce qui permet de définir une zone de recherche réduite, mais aussi une direction d'énumération. Nous introduisons un lemme définissant une borne supérieure sur $v(C^*)$ pour toute paire de chemins successifs ainsi qu'un théorème indiquant quelle paire mène à la borne supérieure la plus basse sur $v(C^*)$, et donc à une zone de recherche restreinte.

Lemme 1. *Soit une paire de chemins successifs (C_1, C_2) de P. Soit δ tel que $\mu_\delta(C_1) = \mu_\delta(C_2)$. Notons $\overline{v}_1(C_1, \delta)$ la valeur v_1 du point à l'intersection de la droite de coefficients δ passant par le point C_1 et la borne inférieure $\beta(p)$. Alors $\overline{v}_1(C_1, \delta)$ est une borne supérieure sur $v(C^*)$ la valeur du chemin optimal du WRLPP.*

Théorème 1. *Dans le cas LF, $\overline{v}_1(C_1, \delta)$ est minimisée pour C_1 le chemin maximisant $v_1(.)$ et $\delta = [1, 0]$. Dans le cas LID, $\overline{v}_1(C_1, \delta)$ est minimisée pour (C_1, C_2) deux chemins successifs de P tels que $\beta(p) \in [v_2(C_2); v_2(C_1)]$ et δ tel que $\mu_\delta(C_1) = \mu_\delta(C_2)$.*

En s'appuyant sur le **Théorème 1**, la première phase consiste alors à trouver la paire $(\widetilde{C}, \widetilde{\delta})$ minimisant $\overline{v}(\widetilde{C}, \widetilde{\delta})$. Pour cela, nous cherchons une seule paire de chemins de P et non toutes les paires comme à la première phase de l'algorithme bi-objectif standard.

Première phase L'algorithme calcule un chemin C_1 du LPP maximisant le premier objectif. Si nous sommes dans le cas LF, alors la procédure s'arrête avec $\widetilde{C} = C_1$ et $\widetilde{\delta} = [1, 0]$. Si nous sommes dans le cas LID, l'algorithme calcule un second chemin C_2 du BOLPP maximisant le second objectif. Le vecteur δ est calculé de façon à ce que $\mu_\delta(C_1) = \mu_\delta(C_2)$, soit $\delta_1 = 1$ et $\delta_2 = \frac{v_1(C_1) - v_1(C_2)}{v_2(C_2) - v_2(C_1)}$. Soit C_3 le chemin du LPP maximisant $\mu_\delta(.)$. Dans le cas $C_1 = C_3$ ou $C_2 = C_3$ alors la procédure s'arrête. Dans le cas contraire nous distinguons deux cas : $v_2(C_3) \geq \beta(p)$, $v_2(C_3) < \beta(p)$. Suivant le **Théorème 1**, nous gardons C_2 et C_3 dans le premier cas, C_1 et C_3 dans le second. La procédure continue en boucle, en recalculant δ de la même manière avec les deux chemins gardés, puis en obtenant le chemin optimal du LPP avec pour objectif $\mu_\delta(.)$. La procédure s'arrête lorsque le chemin optimal du LPP a déjà été obtenu, avec $\widetilde{C}$ l'un des deux derniers chemins gardés, et $\widetilde{\delta} = \delta$.

A la fin de cette première phase, nous avons obtenu $(\widetilde{C}, \widetilde{\delta})$ minimisant $\overline{v}(\widetilde{C}, \widetilde{\delta})$ ce qui permet d'obtenir un trapèze semi-défini.

Pour la seconde phase, nous lançons une recherche implicite des chemins du BOLPP par une variante de l'algorithme des K-meilleurs chemins [5] pour le LPP avec pour objectif $\mu_{\widetilde{\delta}}(.)$.

Ces chemins sont progressivement changés de manière à construire des chemins faisables pour le WRLPP dans la zone de recherche. Pour celà, nous définissons un chemin hybride puis énonçons deux théorèmes donnant une borne supérieure sur $\mu_{\widetilde{\delta}}(C)$ où C est un chemin faisable du WRLPP, ainsi qu'une borne inférieure sur $\mu_{\delta}(C^*)$ où C^* est le chemin optimal du WRLPP.

Définition 5 (Chemin hybride H)**.** Un chemin hybride H est composé de deux sous-chemins : H_{RW} de s à un sommet v, puis H_{BO} de v à p. La partie H_{RW} vérifie les fenêtres de ressource, alors que la partie H_{BO} les relâche.

Théorème 2. *Soit un chemin hybride $H = (H_{RW}, H_{BO})$, avec H_{BO} maximisant $\mu_{\widetilde{\delta}}(.)$. Pour tout chemin C commençant par H_{RW} faisable pour le WRLPP, $\mu_{\widetilde{\delta}}(H) \geq \mu_{\widetilde{\delta}}(C)$.*

Théorème 3. *Soit un chemin faisable C du WRLPP. Le chemin optimal C^* du WRLPP est tel que $\mu_{\widetilde{\delta}}(C^*) \geq \widetilde{\delta}_1 \cdot v(C) + \widetilde{\delta}_2 \cdot \beta(p)$.*

Seconde phase Nous calculons un premier chemin hybride $H = (H_{RW}, H_{BO})$, avec $H_{RW} = \emptyset$ et H_{BO} le chemin optimal de s à p du LPP avec pour objectif $\mu_{\widetilde{\delta}}(.)$. Nous initialisons une liste $L = [H]$ de chemins hybrides triés par ordre décroissant de valeur $\mu_{\widetilde{\delta}}(.)$. Tant que L n'est pas vide, alors nous procédons aux étapes suivantes. Nous sélectionnons H le premier chemin hybride de L. Si H est faisable, alors c'est un chemin faisable du WRLPP. Sinon, H n'est pas faisable sur sa partie H_{BO}, nous cherchons alors à étendre la partie H_{RW}. Soit $a = (u, v)$ le premier arc de H_{BO}. Une extension d'étiquette classique consisterait à construire H' avec $H'_{RW} = H_{RW} \cup \{(u, v')\}$, $v' \neq v$ et H'_{LW} le chemin optimal du LPP de v' à p avec pour objectif $\mu_{\widetilde{\delta}}(.)$. Ici nous généralisons cette extension pour pouvoir étendre plusieurs arcs à la fois. Soit un entier $k \in \{0, |H_{BO}| - 1\}$. Soit $a = (u, v)$ le $k + 1^{\text{ème}}$ arc de H_{BO} et C les k premiers arcs de H_{BO}. L'extension consiste à construire H' avec $H'_{RW} = H_{RW} \cup C \cup (u, v')$, $v' \neq v$ et H'_{LW} le chemin optimal du LPP de v' à p avec pour objectif $\mu_{\widetilde{\delta}}(.)$. Pour tout chemin H' obtenu par l'extension, si H'_{RW} vérifie les fenêtres de ressource, alors H' est ajouté à L. Dans le cas contraire, H' est écarté. Dès lors qu'un chemin C faisable pour le WRLPP est trouvé, nous pouvons supprimer des chemins de L en nous appuyant sur les **Théorèmes 2** et **3** : nous supprimons de L tout chemin hybride H tel que $\mu_{\widetilde{\delta}}(H) < \widetilde{\delta}_1 \cdot v(C) + \widetilde{\delta}_2 \cdot \beta(p)$.

Cette seconde phase est améliorable dans le cas où il existe une borne supérieure $h_1(C)$ pour la valeur $v_1(C)$ d'un chemin faisable. Dès qu'un chemin C faisable pour le WRLPP est trouvé, nous supprimons de L tout chemin hybride H pour lesquels $h_1(H) < v_1(C)$.

3 Cas d'application

Il existe plusieurs cas d'applications du WRLPP. L'un d'eux entre dans le cadre de la réponse à un service de flexibilité de la demande d'électricité. Cela consiste à contrôler la charge thermostatique (TCL) de réfrigérateurs ou de chauffe-eau, où la ressource est la température [7]. Le cas d'application considéré ici est le problème du Hydro Unit Commitment à une usine (1-HUC) comme décrit dans [3] auquel on ajoute des contraintes de gradient et de palier. L'usine est située entre deux réservoirs. Le temps est discrétisé en T pas de temps. L'usine possède $N + 1$ points de fonctionnement, qui sont des couples entre un débit d'eau, et une puissance associée. Chaque réservoir possède un volume initial, avec pour chaque pas de temps un volume maximal et minimal. 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 sur L pas de temps avant changement de sens (min-up/down). Le cas d'un problème de maximisation de revenu pour des prix donnés est considéré.

Le 1-HUC peut être représenté par un WRLPP. Soit le graphe $G_{\text{1-}HUC} = (V_{\text{1-}HUC}, A_{\text{1-}HUC})$ défini comme suit. Chaque sommet $u \in V_{\text{1-}HUC}$ est défini par un triplet $(t,\ i,\ l)$, où t est le pas de temps, i le point de fonctionnement et l le temps restant pour vérifier les contraintes de min-up (si $l > 0$) ou de min-down (si $l < 0$). Nous ajoutons aussi un sommet source $s = (0, 0, 0)$ et un sommet puits $p = (T + 1, 0, 0)$.

Pour tout sommet $u = (t, i, l)$, les arcs sont les suivants. **Min-up/down :** Si $l \geq 1$ il y a un arc vers $(t+1,\ i,\ l-1)$, si $l \leq -1$ vers $(t+1,\ i,\ l+1)$ et si $l = 0$ vers $(t+1,\ i,\ 0)$. **Gradient :** Si $l \geq 0$, pour tout $i' > i$ tel que la différence de débit entre les points de fonctionnement i et i' vérifie la contrainte de gradient, il y a un arc vers $(t+1,\ i',\ L-1)$ et inversement si $l \leq 0$ La valeur d'un arc vers $(t,\ i,\ l)$ prend en compte la valeur de l'énergie produite sur le point de fonctionnement i au pas de temps t et la valorisation de l'eau débitée dans le réservoir à la fin de l'horizon. La **FIG. 2** donne un exemple de graphe $G_{1\text{-}HUC}$ pour $T = 5$, $N = 2$, $L = 3$ et des contraintes de gradient ne permettant pas de passer du point de fonctionnement 0 à 2. Dans ce graphe, les sommets sont en noir s'ils peuvent être atteints depuis s, et en gris sinon.

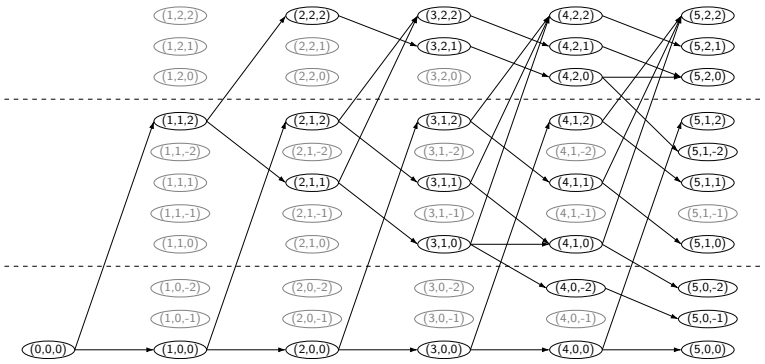


FIG. 2 – Exemple de graphe $G_{1\text{-}HUC}$

Par construction, tout chemin dans $G_{1\text{-}HUC}$ correspond à une solution du 1-HUC de même valeur, vérifiant les contraintes de gradient et de min-up/down. Seules les bornes sur les volumes des réservoirs ne sont pas représentées par le graphe $G_{1\text{-}HUC}$. Pour ce faire, nous introduisons une ressource, qui est la quantité d'eau débitée. Pour un arc $a \in A_{1\text{-}HUC}$ vers un sommet $(t,\ i,\ l)$, la ressource $r(a)$ est le volume d'eau débité au point de fonctionnement i. De plus, pour tout sommet $(t,\ i,\ l)$ nous introduisons la fenêtre de ressource correspondant à la borne minimale et maximale pour le volume cumulé du pas de temps 1 à t, comme indiqué dans [3].

4 Résultats numériques

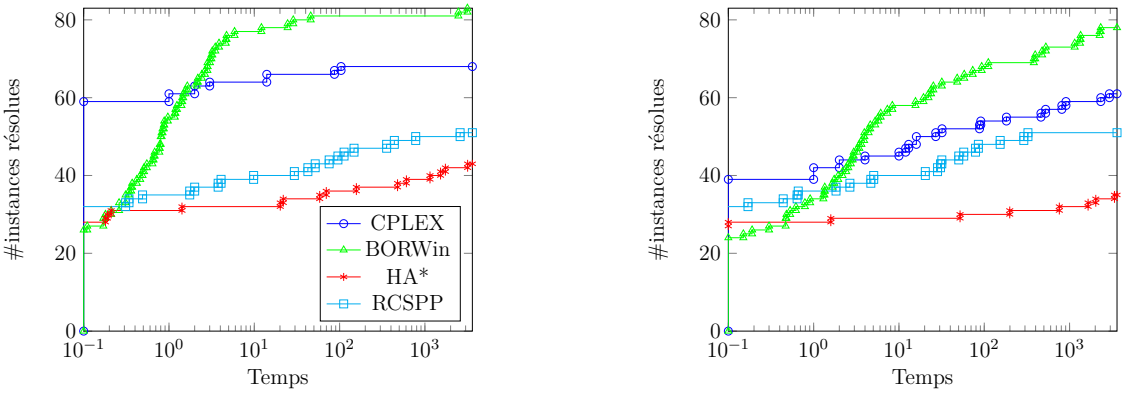
Dans cette section, nous comparons quatre approches : l'approche actuelle d'EDF qui consiste à résoudre un modèle MILP du 1-HUC par CPLEX ; un algorithme classique du RCSPP adapté au 1-HUC [1], la variante exacte de A* (HA*) dédié au 1-HUC [3] et notre algorithme BORWin, amélioré par l'heuristique optimiste de HA*.

Nous comparons ces approches sur deux jeux d'instances. Le premier est un jeu de 83 instances EDF, pour lesquelles le volume initial ou les contraintes sur le volume au dernier pas de temps ont été modifiées de manière à ce que toutes les instances sont restrictives sur la ressource. Le second est identique au premier jeu d'instance, mais où la valorisation de l'énergie a été modifiée de manière à ce que la différence de la valorisation de l'énergie pour toute paire de pas de temps est d'au plus 10%.

La **FIG. 3** affiche pour chaque jeu d'instances, le nombre d'instances résolues à l'optimalité pour un temps donné par approche. Ces résultats montrent que BORWin est la seule alternative permettant de résoudre toutes les instances du jeu A et presque toutes les instances du jeu B en moins d'une heure. Plus précisément, BORWin résout au moins 20 instances de plus que les autres approches en une heure et devient l'alternative la plus efficace à partir de 2 secondes.

L'algorithme HA* n'est pas adapté car son heuristique optimiste ne prend pas en compte les contraintes de gradient et min-up/down. L'algorithme du RCSPP est inefficace car lorsqu'il y a une borne inférieure sur la ressource, les règles de dominance ne s'appliquent pas. Pour la résolution du MILP, il y a 12 instances du jeu A, et 5 instances du jeu B où la résolution du MILP retourne une solution infaisable, qui ne vérifie pas les bornes sur les ressources. Cette violation est dans la majorité des cas de moins de 0.1%, mais peut atteindre 10% : un volume

du réservoir de 29175.04 pour une borne supérieure de 26380. Ces erreurs numériques ont été observées en exploitation et une analyse numérique a également révélé que certaines instances atteignaient la limite des calculs en flottant [2, 6]. De plus, il y a 2 instances du jeu A et 6 instances du jeu B où la résolution du MILP retourne une solution sous-optimale avant la fin du temps alloué. Ces erreurs sont d'au plus 0.01%, par exemple avec solution de valeur 3838650 obtenue par résolution du MILP et de valeur 3838990 obtenue par BORWin. Ces erreurs correspondent au gap d'optimalité de CPLEX qui est de 10^{-4} par défaut.



(a) Jeu d'instances A

(b) Jeu d'instances B

FIG. 3 – Nombre d'instances résolues par rapport au temps

5 Conclusion et perspectives

Ces résultats montrent l'efficacité de BORWin pour résoudre le 1-HUC représenté en WRLPP. Il serait envisageable de remplacer la résolution d'un MILP par l'utilisation de BORWin pour les vallées à 1 usine. Une perspective serait d'utiliser BORWin au sein d'un schéma de décomposition d'une vallée à plusieurs usines en sous-problèmes de 1-HUC et de l'éprouver sur d'autres applications du WRLPP.

Références

[1] W. van Ackooij, C. d'Ambrosio, L. Liberti, R. Taktak, D. Thomopulos et S. Toubaline. "Shortest path problem variants for the hydro unit commitment problem". In : *Electronic Notes in Discrete Mathematics* 69 (2018), p. 309-316.

[2] P. Bendotti et F. Févotte. "Impact de l'arithmétique flottante sur la résolution de programmes linéaires". In : *10èmes Journées Polyèdres et Optimisation Combinatoire*. 2016.

[3] A. Heintzmann, C. Artigues, P. Bendotti, S. U. Ngueveu et C. Rottner. "Efficient exact A* algorithm for the single plant Hydro Unit Commitment problem". In : *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*. T. 35. Annals of Computer Science and Information Systems. IEEE, 2023, p. 533-543. doi : 10.15439/2023F5158.

[4] S. Irnich et G. Desaulniers. "Shortest path problems with resource constraints". In : *Column generation*. Springer, 2005, p. 33-65.

[5] E. 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.

[6] Y. Sahraoui, P. Bendotti et C. d'Ambrosio. "Real-world hydro-power unit-commitment : Dealing with numerical errors and feasibility issues". In : *Energy* 184 (2019), p. 91-104.

[7] S. H. Tindemans, V. Trovato et G. Strbac. "Decentralized control of thermostatic loads for flexible demand response". In : *IEEE Transactions on Control Systems Technology* 23.5 (2015), p. 1685-1700.

[8] L. Turner. "Variants of the shortest path problem". In : *Algorithmic Operations Research* 6.2 (2011), p. 91-104.

[9] E. L. Ulungu et J. Teghem. "The two phases method : An efficient procedure to solve bi-objective combinatorial optimization problems". In : *Foundations of computing and decision sciences* 20.2 (1995), p. 149-165.

[10] X. Zhu et W. E. Wilhelm. "A three-stage approach for the resource-constrained shortest path as a sub-problem in column generation". In : *Computers & Operations Research* 39.2 (2012), p. 164-178.