École des Ponts
ParisTech

ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES

Recherche Opérationelle

Projet

Nicolas BESSIN
Erwann ESTEVE
Tidiane POLO

1 Linéarisations

Q1 Les contraintes suivantes sur μ permettent la linéarisation de $\mu = \alpha\beta$ avec $\alpha \in \{0,1\}$ et $\beta \in [0, M]$:

$$\begin{cases} \mu \in [0, M] & \text{(domaine de définition)} \\ \mu \leq \alpha M & \text{(impose } \mu = 0 \text{ si } \alpha = 0) \\ \beta - (1-\alpha)M \leq \mu \leq \beta & \text{(impose } \mu = \beta \text{ si } \alpha = 1) \end{cases}$$

Q2 Les contraintes suivantes sur μ et α permettent la linéarisation de $\mu = [\beta]^+$ avec $\beta \in [-M, M]$:

$$\begin{cases} \mu \in [0, M], \alpha \in \{0,1\} & \text{(domaine de définition)} \\ \beta \leq M\alpha \leq M + \beta & \text{(impose } \alpha = 1 \text{ si } \beta > 0, \alpha = 0 \text{ si } \beta < 0) \\ \beta - (1-\alpha)M \leq \mu \leq \beta + (1-\alpha)M & \text{(impose } \mu = \beta \text{ si } \alpha = 1) \\ \mu \leq \alpha M & \text{(impose } \mu = 0 \text{ si } \alpha = 0) \end{cases}$$

Notons que α est indéterminé pour $\beta = 0$, mais qu'on a tout de même $\mu = 0 = [\beta]^+$.

Q3 On introduit les variables binaires y_α et y_β, et la variable continue y. Le problème s'écrit alors :

$$\begin{aligned} &\min_{\alpha,\beta,y_\alpha,y_\beta,y} y \\ &\text{s.c. } y \geq \alpha - 2My_\alpha, \quad y \geq \beta - 2My_\beta, \quad y_\alpha + y_\beta = 1 \end{aligned} \tag{1}$$

La contrainte $y_\alpha + y_\beta = 1$ assure $y = \alpha$ ou $y = \beta$. De plus, puisque c'est un problème de minimisation, on a bien $y = min(\alpha, \beta)$.
Q4 On introduit la variable continue δ et les contraintes suivantes : $\delta \geq \alpha, \quad \delta \geq \beta$
Le problème initial s'écrit alors :

$$\begin{aligned} &\min_{\alpha,\beta,\gamma,\delta} \delta + \gamma \\ &\text{s.c. } A(\alpha,\beta,\gamma)^T \leq b, \quad \alpha \leq \delta, \quad \beta \leq \delta \end{aligned} \tag{2}$$

Q5 Pour linéariser les termes en -min, on utilise : $-min(a,b) = max(-a,-b)$
Pour linéariser les termes $p_f(v,x,y)c^c(C^f(v,x,y,z,\omega))$, on doit distribuer la multiplication.
N.B. : la justification de cette formulation du PLNE est trouvable étape par étape dans les commentaires du document linearSolverCode.jl

$$\begin{aligned} &\min_{x,y,z} cost^{cons} + cost^{ope} \\ &(x_{v,s}) \in \{0,1\}^{V^s \times S}, (y^{land}_{e,q}) \in \{0,1\}^{E^0 \times Q}, (z_e) \in \{0,1\}^{E^t}, (y^{sub}_{e,q}) \in \{0,1\}^{E^S \times Q} \\ &\text{s.c. } (1), (2), (3), (4) \text{ (conditions du sujet)} \end{aligned}$$

$$\begin{cases} minusCapa_v \geq -\sum_{s \in S} r_s * x_{v,s}, \quad minusCapa_v \geq -\sum_{q \in Q_0} r_q * y^{land}_{e,q} & \text{for } v \in V^s, e = (v_0, v) \\ C^n_{v,\omega} \geq 0, \quad C^n_{v,\omega} \geq \pi_\omega * \sum_{t \in V^t} z_{v,t} + minusCapa_v & \text{for } v \in V^s, \omega \in \Omega \\ C^n_\omega = \sum_{v \in V^s} C^{nf}_{v,\omega} & \text{for } \omega \in \Omega \end{cases} \tag{3}$$

$$\begin{cases} C^{ownFail}_{v,\omega} \in R_+^{V^s \times \Omega} \\ C^{ownFail}_{v,\omega} \geq 0, \quad C^{ownFail}_{v,\omega} \geq \pi_\omega * \sum_{t \in V^t} z_{v,t} - \sum_{e=(v,\tilde{v}) \in E^S, q \in Q_0} r_q * y^{sub}_{e,q} & \text{for } v \in V^s, \omega \in \Omega \end{cases} \tag{4}$$

$$\begin{cases} (P^{sent}_{u,v,\omega}) \in R^{Vs^2\times\Omega}, 1^{powerIsMin}_{u,v,\omega} \in \{0,1\}^{Vs^2\times\Omega}, 1^{cableIsMin}_{u,v,\omega} \in \{0,1\}^{Vs^2\times\Omega} \\ 1^{powerIsMin}_{u,v,\omega} + 1^{cableIsMin}_{u,v,\omega} = 1 \qquad \text{for } (u,v) \in V^{s^2}, \omega \in \Omega \\ P^{sent}_{u,v,\omega} \geq \sum_{t\in V^t} z_{u,t} * \pi_\omega - 1^{cableIsMin}_{u,v,\omega} * P^{max} * N^{turbine} \\ P^{sent}_{u,v,\omega} \geq \sum_{q\in Q^S} y^{sub}_{e,q} * r_q - 1^{powerIsMin}_{u,v,\omega} * P^{max} * N^{turbine} \qquad e=(u,v) \end{cases} \tag{5}$$

$$\begin{cases} C^{otherFail}_{u,v,\omega} \in R^{Vs^2\times\Omega}_+ \\ C^{otherFail}_{u,v,\omega} \geq 0, \quad C^{otherFail}_{u,v,\omega} \geq \pi_\omega * \sum_{t\in V^t} z_{v,t} - P^{sent}_{u,v,\omega} + minusCapa_v \end{cases} \tag{6}$$

$$\begin{cases} C^f(v,\omega) \in R^{V^s\times\Omega} \\ C^f(v,\omega) = C^{ownFail}_{v,\omega} + \sum_{e=(v_0,\bar{v}),v\neq\bar{v},\tilde{e}=(v,\bar{v})} C^{otherFail}_{v,\bar{v}\omega} \text{for } v \in V^s, \omega \in \Omega \end{cases} \tag{7}$$

$$\begin{cases} (cost^n_\omega) \in R^{V_s\times\Omega}, (cost^f_{v,\omega}) \in R^{V^s\times\Omega} \\ cost^{cons} = \sum_{v\in V^s}\sum_{s\in S} c_s x_{vs} + \sum_{e\in E^0}\sum_{q\in Q_0} c_{eq} y^{land}_{eq} + \sum_{e\in E^S}\sum_{q\in Q_S} c_{eq} y^{sub}_{eq} + \sum_{e\in E^t} c_e z_e \\ cost^n_\omega \geq c^0 C^n_\omega, cost^n_\omega \geq c^0 C^n_\omega + c^p(C^n_\omega - C^{max}) \qquad \text{for } \omega \in \Omega, v \in V^s \\ cost^f_{v,\omega} \geq c^0 Curt^{fail}_{v,\omega}, \quad cost^f_{v,\omega} \geq c^0 C^f_{v,\omega} + c^p(C^f_{v,\omega} - C^{max}) \qquad \text{for } \omega \in \Omega, v \in V^s \end{cases} \tag{8}$$

$$\begin{cases} (x^{timesCostF}_{v,s,\omega}) \in R^{V^s\times S\times\Omega}, (y^{timesCostF}_{e,q,\omega}) \in R^{V^s\times Q\times\Omega} \\ cost^{max} = c^0 P^{max} N^{turbine} + c^p[P^{max} N^{turbine} - C^{max}]^+ \qquad \text{Pas une variable} \\ x^{timesCostF}_{v,s,\omega} \leq x_{v,s} * cost^{max}, \quad x^{timesCostF}_{v,s,\omega} \leq cost^f_{v,\omega} \qquad \text{for } \omega \in \Omega, v \in V^s, s \in S \\ x^{timesCostF}_{v,s,\omega} \geq cost^f_{v,\omega} - (1-x_{v,s}) * cost^{max}, \quad x^{timesCostF}_{v,s,\omega} \geq 0 \qquad \text{for } \omega \in \Omega, v \in V^s, s \in S \\ y^{timesCostF}_{e,q,\omega} \leq y^{land}_{v,q} * cost^{max}, \quad y^{timesCostF}_{e,q,\omega} \leq cost^f_{v,\omega} \qquad \text{for } \omega \in \Omega, v \in V^s, q \in Q \\ y^{timesCostF}_{e,q,\omega} \geq cost^f_{v,\omega} - (1-x_{v,s}) * cost^{max}, \quad y^{timesCostF}_{e,q,\omega} \geq 0 \qquad \text{for } \omega \in \Omega, v \in V^s, q \in Q \end{cases} \tag{9}$$

$$\begin{cases} (x^{timesCostNoF}_{v,s,\omega}) \in R^{V^s\times Q_0,\Omega}, (y^{timesCostNoF}_{e,q,\omega}) \in R^{V^s\times Q_0,\Omega} \\ x^{timesCostNoF}_{v,s,\omega} \leq x_{v,s} * cost^{max}, \quad x^{timesCostNoF}_{v,s,\omega} \leq cost^n_\omega \qquad \text{for } \omega \in \Omega, v \in V_s, s \in S \\ x^{timesCostNoF}_{v,s,\omega} \geq cost^n_\omega - (1-x_{v,s}) * cost^{max}, \quad x^{timesCostNoF}_{v,s,\omega} \geq 0 \qquad \text{for } \omega \in \Omega, v \in V_s, s \in S \\ y^{timesCostNoF}_{e,q,\omega} \leq y^{land}_{v,q} * cost^{max}, \quad y^{timesCostNoF}_{e,q,\omega} \leq cost^n_\omega \qquad \text{for } \omega \in \Omega, q \in Q_0, v \in V^s \\ y^{timesCostNoF}_{e,q,\omega} \geq cost^n_\omega - (1-y^{land}_{v,q}) * cost^{max}, \quad y^{timesCostNoF}_{e,q,\omega} \geq 0 \qquad \text{for } \omega \in \Omega, q \in Q_0, v \in V^s \end{cases} \tag{10}$$

$$\begin{cases} cost^{ope} = \sum_{\omega\in\Omega}[\sum_{v\in V^s}(\sum_{s\in S} p_s x^{timesCostF}_{v,s,\omega} + \sum_{q\in Q^0, e=(v_0,v)} p_q y^{timesCostF}_{e,q,\omega}) \\ \qquad + cost^n_\omega - \sum_{v\in V^s}(\sum_{s\in S} p_s x^{timesCostNoF}_{v,s,\omega} + \sum_{q\in Q^0, e=(v_0,v)} p_q y^{timesCostNoF}_{e,q,\omega})] \end{cases} \tag{11}$$

Q6 On obtient pour small la solution optimale suivante, pour une valeur optimale de *3244.15* :

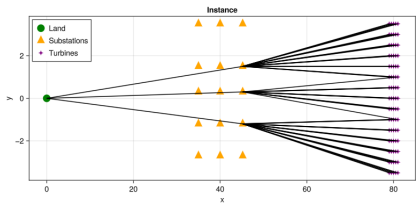

FIGURE 1 – Solution optimale pour l'instance small

Q7 Les *indicator* rallongent le temps de calcul mais la solution optimale est la même. (cf document *solverQuadraticIndicator.jl*, lignes 122 à 145). On peut en déduire que linéariser un problème est plus efficace, même si cela requiert un travail de linéarisation au préalable.

Q8 On constate (cf *Q6*) que les sites de sous-station utilisés sont les plus proches des éoliennes, et que les types de sous-station et de câbles utilisés sont les moins chers (2). De plus, plutôt que de regrouper toutes les éoliennes sur une même sous-station, la solution optimale les répartit sur trois sous-stations différentes, ce qui permet d'améliorer la résilience de la solution.

2 Résolution du problème

Une fois le solveur linéaire mis en place, le problème principal qui empêche de l'utiliser pour les instances medium, large et huge est la taille de celles-ci - en effet, le nombre de variables dans le modèle influe énormément sur le temps de calcul et la mémoire nécessaire (et donc la possibilité de résoudre le problème). Ainsi nous avons implémenté la stratégie suivante :

— Réduire la taille de l'instance avec différentes méthodes (voir ci-dessous).
— Résoudre le problème pour cette sous-instance avec le solveur linéaire
— Transformer la solution obtenue en une solution du problème initial.

Les stratégies de réduction de taille que nous avons implémentées sont motivées par les observations suivantes que nous avons pu faire sur l'instance small : seuls certains sites de sous-station sont utilisés (1), seuls certains types de sous-station et de câbles sont utilisés (2), et le nombre de scénarios est très grand (100 pour small & medium, 1000 pour large & huge).

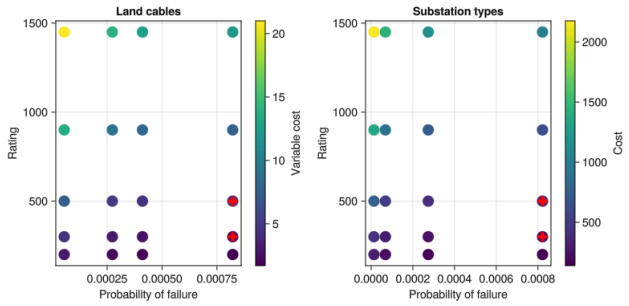

FIGURE 2 – Types des sous-stations et des câbles dans l'instance small - en rouge, les types effectivement utilisés dans la solution optimale

— Ne garder que les sites de sous-station situés sur une même "colonne" - comme pour l'instance small, où les sous-stations sont toutes situées sur la même colonne, la plus proche des éoliennes.
— Remplacer les multiples scénarios par un unique scénario (e.g. un scénario dont la puissance est $p = p_{min} + x(p_{max} - p_{min})$ avec x un coefficient choisi. Avec $x = 1$, la solution est sur-dimensionée : les coûts de contruction sont trop élevés, et avec $x = 0$, la solution est sous-dimensionnée : les coûts d'opération sont trop élevés. On trouve une valeur "optimale" de x à environ $x = 0.99$.)
— Éliminer certains types possibles de sous-stations et / ou de câbles, en fonction de leurs probabilités de panne et de leurs coûts.

Algorithm 1 Réduction de taille d'instance par élimination de scénarios

Require: instance $I = (V^s, V^t, E^0, E^S, E^t, S, Q^0, Q^S, \Omega, ...)$ ▷ ... : autres paramètres, restent inchangés
1: $\pi_{\min} \leftarrow \min_{\omega \in \Omega} \pi_\omega$ ▷ Puissance minimale
2: $\pi_{\max} \leftarrow \max_{\omega \in \Omega} \pi_\omega$ ▷ Puissance maximale
3: $\omega_0 \leftarrow \omega(\pi = \pi_{\min} + x(\pi_{\max} - \pi_{\min}), \text{prob} = 1.0)$ ▷ Déclaration du scénario unique, avec x un coefficient choisi
4: **return** $I' \leftarrow (V^s, V^t, E^0, E^S, E^t, S, Q^0, Q^S, \{\omega_0\}, ...)$ ▷ Création de la nouvelle instance

Nom	x_{vs}	y_{eq}	z_e	Curtailing of v under ω
Quantité	$\|V^s\| \times \|S\|$	$\|Q^0\| \times \|V^s\|$ $+\|V^s\|^2 \times \|Q^s\|$	$\|V^s\| \times \|V^t\|$	$\|V^s\|\|\Omega\|$
Nom	$C^n(x, y, z, \omega)$	Curtailing of v under ω and failure of v	Power sent from v to $\tilde{v}$	$C^f(v, x, y, z, \omega)$
Quantité	$\|\Omega\|$	$\|V^s\|\|\Omega\|$	$\|V^s\|^2\|\Omega\|$	$\|V^s\|\|\Omega\|$

TABLE 1 – Nombre de variables du problème / contraintes en fonction des tailles caractéristiques de l'instance

Le nombre de variables / contraintes - et donc le temps de résolution - du problème est donc en $O(|V^s||S| + |V^s|^2|Q^s| + |V^s||Q^0| + |V^s||V^t| + |V^s|^2|\Omega|)$.
L'intérêt de la stratégie de réduction de taille d'instance est donc d'obtenir des instances dont on peut trouver la solution optimale en un temps raisonnable, à l'aide du solveur linéaire. Ces sous-instances étant extraites des instances initiales, la solution optimale de la sous-instance est une solution réalisable pour l'instance initiale, de coût supérieur à l'optimal. (Mais relativement proches si la réduction de taille est bien faite.)

Instance	small $\|V^s\| = 15, \|\Omega\| = 100$	small reduced $\|V^s\| = 5, \|\Omega\| = 1$
Time (s)	2237	1.4
Cost	3244	3244

TABLE 2 – Temps de calcul & résultat selon la réduction appliquée

En utilisant les différentes stratégies d'agrégation décrites ci-dessus (voir *agregator.jl*), on obtient les résultats suivants :

size	small (optimal)	medium	large	huge
cost	3244	6472	8566	5583
gap	$\leq 3.8\%$	$\leq 7.5\%$	Unknown	Unknown

TABLE 3 – Meilleurs résultats obtenus pour les différentes instances.

Le gap est calculé par rapport à la solution du relaché linéaire, et est donc une sur-estimation du gap par rapport à la solution optimale entière. (Pour large & huge, nous n'avons pas pu calculer le gap, 16Go de RAM n'étant pas suffisant pour résoudre ces instances.)

Le code source pour le projet est disponible sur Github : github.com/Nicolas-Bessin/ProjetREOP.