SOD321 : Optimisation discrète
Projet - Course d'avions

Bechir Trabelsi
Mohamed Mkaouar

Octobre 2020

1 Introduction

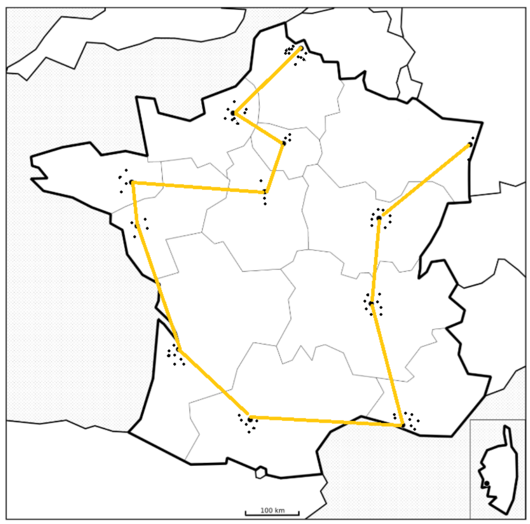
Ce projet s'inspire de la compétition Breitling 100/24. On considère n aérodromes numérotés de $1..n$ et de m régions. Pour chaque aérodrome, on connait ses coordonnées (x, y) dans le plan et le numéro de région à laquelle il appartient. On suppose connus, les aérodromes de départ et d'arrivée, le nombre minimal d'aérodromes à visiter $A_{\min}$ ainsi que la distance R que peut parcourir un avion sans se poser. Nous voulons donc minimiser la distance parcourue pendant les 24 heures .

Notations

Pour modéliser ce problème, on considère les variables suivantes :

- $x_{ij}, i \in \{1..n\}$ et $j \in \{1..n\}$ est une variable binaire égale à 1 si l'avion passe de l'aérodrome i à l'aérodrome j, elle est égale à 0 sinon.
- $d_{ij}, i \in \{1..n\}$ et $j \in \{1..n\}$ correspond à la distance entre deux aérodromes i et j. Cette distance est calculée comme la distance euclidienue arrondie, c'est à dire en notant (x_i, y_i) et (x_j, y_j) les coordonnées des aérodromes i et j respectivement: $d = \left[\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right]$.

- m est le nombre de régions.
- $Reg \in \mathbf{N}^n$ désigne à quelle région appartient chaque aérodrome, i.e $Reg[i] = k \iff$ le i^e aérodrome est dans la k^e région.
- dep désigne l'aérodrome de départ.
- arr désigne l'aérodrome d'arrivée.
- $T = \{1..n\}$
- $M = \{1..m\}$



2 Modèle avec un nombre polynomial de contraintes

$$
\begin{cases}
\min & \sum_{i,j\in T^2} d_{ij}x_{ij} \\
\text{s.t.} & \\
& d_{ij}x_{ij} \leq R \qquad \forall i,j \in T^2, \\
& \sum_{i,j} x_{ij} \geq A_{min} - 1 \qquad \forall i,j \in T^2, \\
& \sum_{i} (x_{ij} - x_{j,i}) = 0 \qquad \forall j \in T \setminus \{dep, ar\}, \\
& \sum_{\substack{i,j \\ Reg_i = k}} x_{ij} \geq 1 \qquad \forall k \in M, \\
& \sum_{i} x_{dep,i} = 1, \\
& \sum_{i} x_{i,dep} = 0, \\
& \sum_{i} x_{i,ar} = 1, \\
& \sum_{i} x_{ar,i} = 0, \\
& x_{i,i} = 0 \qquad \forall i \in T, \\
& u_j \geq u_i + 1 - n\left(1 - x_{ij}\right) \qquad \forall i \in T
\end{cases}
$$

Dans cette formulation, la variable u_i indique le nombre de sommets visités depuis le sommet de départ s.
La contrainte

$$u_j \geqslant u_i + 1 - n\left(1 - x_{ij}\right) \quad \forall i, j; i \neq j$$

assure la connexité de la solution.
Par ailleurs, dans ce modèle ainsi que dans le 2ème modèle (modèle exponentiel énoncée dans la section suivante) , nous avons modélisé le problème comme suit:

- $d_{ij}x_{ij} \leqslant R$ montre que l'avion ne peut parcourir la distance entre les aérodromes i et j à moins que cette distance est inférieure que la distance qu'un avion peut parcourir sans se poser.
- $\sum_{i,j} x_{ij} \geqslant A_{\min} - 1$ indique que l'avion doit visiter au moins $A_{\min}$ aérodrome.
- $\sum_i \left(x_{ij} - x_{ji}\right) = 0$ assure la continuité du chemin.
- $\sum_{\substack{i,j \\ Reg_i = k}} x_{ij} \geq 1$ assure le passage de l'avion de toutes les régions au moins une fois.
- $\sum_i x_{is} = 0$, $\sum_i x_{ti} = 0$, $\sum_i x_{it} = 1$ et $\sum_i x_{si} = 1$ assurent que le départ soit en s et que l'arrivée soit en t.

3 Modèle avec un nombre exponentiel de contraintes

Dans cette partie on applique une autre approche pour la formulation du problème où on remplace la contrainte de connexité vérifiée avec les variables u_i par une contrainte de sous-tours:

$$\sum_{i,j} x_{ij} \leq |S| - 1 \, \forall i, j \in S^2, \tag{1}$$

où $|S|$ est le cardinale de l'ensemble S.

$$(\mathcal{P}_{matre}) \begin{cases} \min & \displaystyle\sum_{i,j \in T^2} d_{ij} x_{ij} \\ \text{s.t.} & \\ & d_{ij} x_{ij} \leq R \qquad \forall i, j \in T^2, \\ & \displaystyle\sum_{i,j} x_{ij} \geq A_{min} - 1 \quad \forall i, j \in T^2, \\ & \displaystyle\sum_{i} (x_{ij} - x_{j,i}) = 0 \qquad \forall j \in T \setminus \{dep, ar\}, \\ & \displaystyle\sum_{\substack{i,j \\ Reg_i = k}} x_{ij} \geq 1 \qquad \forall k \in M, \\ & \displaystyle\sum_{i} x_{dep,i} = 1, \\ & \displaystyle\sum_{i} x_{i,dep} = 0, \\ & \displaystyle\sum_{i} x_{i,ar} = 1, \\ & \displaystyle\sum_{i} x_{ar,i} = 0, \\ & x_{i,i} = 0 \qquad \forall i \in T, \\ & \displaystyle\sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad \forall S \subset T, \ \ S \neq \emptyset, T \end{cases}$$

Ces contraintes empêchent les solutions d'avoir des sous-tours. Néanmoins, ces contraintes sont en nombres exponentielles. Pour pouvoir résoudre une telle formulation, on utilise un algorithme de génération des contraintes inspiré par l'algorithme de génération des colonnes, l'idée est d'ajouter des contraintes au fur et à mesure en résolvant un sous-problème à chaque itération, pour cela on introduit($\mathcal{P}_{secondaire}$).
On note $z_i = \mathbf{I}_{(i \in S)}$

$$(\mathcal{P}_{secondaire}) \begin{cases} \max & \displaystyle\sum_{(i,j) \in T^2} w_{ij} z_i z_j - \sum_{i \in T} z_i + 1 \\ \text{s.t.} & \\ & z_i \in \{0, 1\} \quad \forall i \in T \end{cases}$$

On linéarise le problème en posant $y_{ij} = z_i z_j$ et $y_{jj} = z_j z_j$

$$(\mathcal{P}_{secondaire}) \begin{cases} \max & \sum_{(i,j)\in T^2} w_{ij} y_{ij} - \sum_{i\in T} y_{i,i} + 1 \\ \text{s.t.} & \\ d_{ij}x_{ij} \leq R & \forall i,j \in T^2, \\ y_{ij} \leq y_{i,i} & \forall i,j \in T^2, \\ y_{ij} \leq y_{j,j} & \forall i,j \in T^2, \\ y_{ij} \geq y_{i,i} + y_{j,j} - 1 & \forall i,j \in T^2, \\ z_i \in \{0,1\} & \forall i \in T \end{cases}$$

l'algorithme de génération des contraintes se déroule comme suit:

Algorithm 1 Algorithme de génération des contraintes

-initialisation: On initialise un ensembles des contraintes.
while ***TRUE*** **do**
    -Résoudre $(\mathcal{P}_{matre})$
    -Résoudre $(\mathcal{P}_{secondaire})$ avec la valeur optimale de $(\mathcal{P}_{matre})$: $\bar{x}$
    if $Val(\mathcal{P}_{secondaire}) \geq 0$ **then**
        -La solution obtenue caractérise un ensemble S' où (1) est violé par $\bar{x}$ donc on ajoute la contrainte de sous-tour dans $(\mathcal{P}_{matre})$.
    else
        toutes les inégalités de sous-tour sont vérifiés.
        BREAK
    end
end

4 Résultat

Pour les deux modèle on trouve les chemins identiques suivants :

instance	chemin
instance-0	1 – 3 – 2 – 5
instance-20-1	19 – 8 – 11 – 15 – 1 – 5 – 14 – 10
instance-20-2	1 – 15 – 5 – 20 – 14 – 8 – 19 – 10
instance-20-3	1 – 15 – 12 – 5 – 11 – 8 – 19 – 10
instance-20-4	1 – 5 – 20 – 14 – 10 – 8 – 11 – 15
instance-30-1	19 – 8 – 11 – 25 – 5 – 21 – 29 – 12 – 17 – 18 – 20 – 14 – 22 – 10
instance-40-1	27 – 7 – 10 – 23 – 11 – 32 – 3 – 4 – 9 – 28 – 22 – 24 – 35 – 8
instance-50-1	49 – 17 – 48 – 3 – 27 – 9 – 13 – 1 – 29 – 31 – 40 – 11 – 32 – 50 – 2 – 34

4.1 Modèle 1: Polynomiale

instance	Temps d'exécution(sec)	valeur opt	borne inf	B&B	nb d'itérations
instance-0	0.149	12	8	3	52
instance-20-1	1.438	91	34.4	10973	57025
instance-20-2	0.202	79	58	393	2372
instance-20-4	0.528	96	38.1	5363	34916
instance-30-1	24.147	130	39.933	99496	780808
instance-40-1	3.819	114	77.25	10844	84169
instance-50-1	23.68	168	91.32	73341	713537

4.2 Modèle 2: Exponentiel

instance	Temps d'exécution(sec)	valeur opt	borne inf	B&B	nb d'itérations
instance-0	0.052	12	8	0	12
instance-20-1	67.61	91	56.45	2947	18808
instance-20-2	2.308	79	67	208	1046
instance-20-4	962.772	96	74.39	3248	28919
instance-40-1	190.031	114	97.33	2875	19594
instance-50-1	1096.994	168	126.68	31932	339032

5 Conclusion

Les deux modèles fournissent la même valeur optimale. Ceci est bien évident vu qu'ils sont issus du même problème. Cependant, la taille de la deuxième formulation est très importante (un nombre exponentiel de contraintes) par rapport au premier qui en a un nombre polynomial. Ceci explique la différence du temps d'exécution entre les deux algorithmes. En fait, plus l'instance est grande plus la différence augmente. Ceci dit, le nombre d'itération, quant à lui, joue le rôle inverse. L'exécution du modèle exponentiel se fait en un nombre d'itération considérablement inférieure à celle du modèle polynomial. De surcroît, la borne inf est aussi meilleure pour le modèle exponentiel.
C'est alors à l'utilisateur de choisir le modèle à utiliser en fonction de ses préférences.