Rapport du projet de SOD321

David DAHAN MONSONEGO

October 24, 2022

1 Présentation du problème

1.1 Contexte

Nous allons nous inspirer d'un problème que doivent résoudre les compétiteurs de la Coupe Breitling 87/24 . Ces compétiteurs sont des équipes de pilotes qui doivent, en 24 heures, sillonner la France à bord de petits avions et effectuer un maximum de posé-décollé (si possible 87) sur des aérodromes différents. Ils partent tous d'un aérodrome donné et doivent arriver à un autre aérodrome. S'ils n'arrivent pas à visiter 87 aérodromes, ils doivent minimiser la distance parcourue pendant les 24 heures. Afin d'assurer une certaine disparité géographique, les organisateurs définissent des régions qui doivent toutes être visitées au moins une fois.

1.2 Problème à résoudre

Nous allons résoudre le problème suivant : On connaît le nombre **n** d'aérodromes numérotés $1, , n$ et le nombre **m** de régions. Pour chaque aérodrome, on a ses coordonnées (x, y) dans le plan et le numéro de région auquel il appartient (0 veut dire qu'il n'est affecté à aucune région). On connaît aussi les numéros des aérodromes de départ et d'arrivée, le nombre minimal d'aérodromes à visiter **Amin** ainsi que la distance **R** que peut parcourir un avion sans se poser. L'objectif est que la distance minimale parcourue soit minimale.

2 Modélisation du problème sans prise en compte des sous-tours

On considère un graphe $G = (V, A)$ où V représente l'ensemble des sommets et A celui des arrêtes et le coût d_{ij} de l'arrête $(i, j) \in A$. On s'intéresse au problème du plus court chemin entre le noeud départ **d** et celui d'arrivé **f**. On a donc que V a pour cardinal n.

On note $\delta^+(i)$ et $\delta^-(i)$ l'ensemble des arcs entrants et sortant du noeud i, par $\delta^+(S)$ et $\delta^-(S)$ les arcs sortants et entrants de l'ensemble $S \subseteq V$ et par $A(S)$ l'ensemble des arcs dont les deux extrémités sont dans $S \subseteq V$. On définit aussi $V_i := V \setminus \{i\}$, $V_{ij} := V \setminus \{i, j\}$.

Une formulation standard de programmation en nombres entiers pour déterminer un chemin le plus court du noeud **d** au noeud **f** est $\min \sum_{(i,j)\in A} d_{ij}x_{ij}$ où $d_{ij} \in \mathbb{R}$ et $x_{ij} \in \{0, 1\}$ représente respectivement le coût de l'arc (i, j) et la variable qui prend la valeur 1 si l'arc (i, j) appartient au chemin. Les contraintes sont les suivantes :

- Conservation du flow :

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} - \sum_{(j,i)\in\delta^-(i)} x_{ji} = \mathbb{1}_{\{i=d\}} - \mathbb{1}_{\{i=f\}} \quad \forall i \in V$$

- Le degré sortant de chaque nœud est au plus un :

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} \leq 1 \quad \text{et} \quad \sum_{(j,i)\in\delta^-(i)} x_{ji} \leq 1 \quad \forall i \in V$$

- Les variables sont binaires : $x_{ij} \in \{0,1\} \quad \forall (i,j) \in A$

A ces contraintes, il faut ajouter les conditions qui sont propres à notre problème de course d'avions :

- Le sommet **d** est le départ :

$$\sum_{(d,j)\in\delta^+(d)} x_{dj} = 1 \quad \text{et} \quad \sum_{(i,d)\in\delta^-(d)} x_{id} = 0$$

- Le sommet **f** est l'arrivée :

$$\sum_{(i,f)\in\delta^-(f)} x_{if} = 1 \quad \text{et} \quad \sum_{(f,j)\in\delta^+(f)} x_{fj} = 0$$

- Il faut visiter au minimum **Amin** aérodromes :

$$\sum_{(i,j)\in A} x_{ij} \geq Amin - 1$$

- L'avion peut parcourir au maximum une distance **R** sans se poser :

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} * d(i,j) \leq R \quad \forall i \in V$$

- Toutes les régions doivent être visitées au moins une fois :
  On note $\mathbf{N_r}$ le nombre de régions à visiter et W_k les indices des sommet du graphe qui correspondent aux aérodromes appartenant à la région k.

$$\sum_{i\in W_k} \Big(\sum_{(i,j)\in\delta^+(i)} x_{ij} + \sum_{(j,i)\in\delta^-(i)} x_{ji} \Big) \geq 1 \quad \forall k \in \{1,\ldots,N_r\}$$

3 Modélisation des sous-tours

Dans ce projet, pour pallier à l'apparition de sous-tours, nous avons utlisé des contraintes, définies dans l'article de recherche, de deux types :

1. Polynomiales : Sequential formulation (MTZ), Single-flow formulation (SF), Multicommodity-flow formulation (MCF).
2. Exponentielles : Dantzig–Fulkerson–Johnson (DFJ), Generalized cutset inequalities (GCS).

3.1 Contraintes Polynomiales

3.1.1 MTZ

On introduit, pour chaque nœud, une variable auxiliaire qui peut être vue comme la position du nœud le long du chemin et une contrainte pour chaque arc. Elle ne nécessite que $|A|$ contraintes supplémentaires.

$$u_i + 1 - n * (1 - x_{i,j}) \leq u_j \quad \forall (i,j) \in A,\ i \neq d, j \neq f$$
$$1 \leq u_i \leq n-1 \quad \forall i \in V$$
$$u_d = 1, u_f = n-1$$

3.1.2 SF

On introduit un flux auxiliaire q à délivrer aux nœuds appartenant au chemin de d à f. Les contraintes assurent que le flux auxiliaire partant du noeud d a une valeur égale au nombre de noeuds atteints par le chemin de d à f et que l'équilibre du flux auxiliaire sur chaque noeud vaut 1 si le noeud se trouve dans le chemin de d à f et 0 sinon. Cette formulation étendue nécessite $|A|$ contraintes.

$$
\begin{aligned}
& q_{ij} \leq (n-1)x_{ij} \quad \forall (i,j) \in A \\
& \sum_{(s,j)\in\delta^+(s)} q_{sj} = \sum_{k\in V_s} z_k \\
& \sum_{(i,k)\in\delta^-(k)} q_{ik} - \sum_{(k,j)\in\delta^+(k)} q_{kj} = z_k \quad \forall k \in V_s \\
& \sum_{(i,k)\in\delta^-(k)} x_{ik} = z_k \quad \forall k \in V_s \\
& q_{ij} \geq 0 \quad \forall (i,j) \in A \\
& z_k \in \{0,1\} \quad \forall k \in V_s
\end{aligned}
$$

3.1.3 MCF

Une extension de la formulation à flux unique est obtenue en désagrégeant le flux auxiliaire en $n-1$ flux unitaires. Les sous-tours sont empêchés en appliquant une unité d'un flux auxiliaire distinct de d à chaque noeud appartenant au chemin de d à f. La formulation inclut $O(n|A|)$ contraintes supplémentaires.

$$
\begin{aligned}
& q^k_{ij} \leq x_{ij} \quad \forall k \in V_s, \quad (i,j) \in A \\
& \sum_{(i,j)\in\delta^+(i)} q^k_{ij} - \sum_{(j,i)\in\delta^{-(i)}} q^k_{ji} = \begin{cases} z_k & \text{if } i = s \\ -z_k & \text{if } i = k \\ 0 & \text{else} \end{cases} \qquad \forall i \in V, \\
& \sum_{(i,k)\in\delta-(k)} x_{ik} = z_k \quad \forall k \in V_s \\
& \sum_{(s,j)\in\delta^+(s)} x_{sj} = 1 \\
& \sum_{(i,t)\in\delta^-(t)} x_{it} = 1 \\
& q^k_{ij} \geq 0 \quad \forall k \in V_s, \quad (i,j) \in A \\
& z_k \in \{0,1\} \quad \forall k \in V_s.
\end{aligned}
$$

3.2 Contraintes Exponentielles

3.2.1 DFJ

Dans chaque sous-ensemble S, les sous-tours sont empêchés en s'assurant que le nombre d'arcs dans S qui sont sélectionnés est inférieur au nombre de noeuds dans S. Cette formulation inclut $O(2^n)$ contraintes.

$$\sum_{(i,j)\in A(S)} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V_{st}, |S| \geq 2$$

Le nombre de contraintes élevé rend difficile l'implémentation de ces dernières. Pour cela nous avons considéré le problème tiers suivant :

$$
(ST) \begin{cases} \max \sum_{(i,j)\in A} x_{ij} * a_i * a_j - (\sum_{i\in V} a_i - 1) \\ a_i \in \{0,1\} \quad \forall i \in V \end{cases}
$$

La résolution de ce problème permet de trouver les contraintes qui sont violées par la solution obtenue

x. Les $(a_i)_{i \in V}$ représente les aérodromes qui ont violé la contrainte. On ajoute donc seulement celles qui sont violées.

Ainsi pour la résolution de notre problème maître nous procédons de la manière suivante :

1. **On résolve le problème maître une première fois sans contraintes de sous-tours.**
2. **Tant que l'objectif du problème tiers (ST) est strictement positif (il existe des aérodromes qui ont violé la contrainte) :**
   2.1. **On ajoute les contraintes violées.**
   2.2. **On résolve le problème maître avec l'ajout des contraintes violées.**
3. **On obtient une approximation de notre solution.**
4. **On obtient une solution finale en ajoutant l'un des types de contraintes polynomiales.**

3.2.2 GCS

Pour chaque sous-ensemble S, le nombre d'arcs sélectionnés quittant S n'est pas inférieur au nombre d'arcs sélectionnés sortant de n'importe quel noeud de S. Le nombre de contraintes est $O(n2^n)$.

$$\sum_{(i,j) \in \delta^+(S)} x_{ij} \geq \sum_{(k,j) \in \delta^+(k)} x_{kj} \quad \forall k \in S, \forall S \subseteq V_{st}, \quad |S| \geq 2$$

On procède de même que précédemmen en considérant le problème tiers :

$$(ST_k) \begin{cases} \max \sum_{(i,j) \in A} x_{ij} * a_i * a_j - \sum_{j \in V} x_{kj} * a_j \\ a_i \in \{0,1\} \quad \forall i \in V \end{cases}$$

On résolve le problème maître de la manière suivante :

1. **On résolve le problème maître une première fois sans contraintes de sous-tours.**
2. **Tant que le minimum des objectifs des problèmes tiers $(\mathbf{ST_k})_{\mathbf{k \in V}}$ est strictement négatif (il existe des aérodromes qui ont violé la contrainte) :**
   2.1. **On ajoute les contraintes violées.**
   2.2. **On résolve le problème maître avec l'ajout des contraintes violées.**
3. **On obtient une approximation de notre solution.**
4. **On obtient une solution finale en ajoutant l'un des types de contraintes polynomiales.**

4 Résultats et conclusion

Pour avoir un aperçu des performances des différents types de contraintes polynomiales, nous avons utilisé le fichier test **"aerodrome_20_1.txt"** avec 20 villes. Les résultats concernant la résolution vers la solution finale sont les suivants : 1 secondes pour SF, 4 secondes pour MCF et 8 secondes pour MTZ. De plus, la solution obtenue par les trois formulations est identique, la fonction objectif vaut 89.

Figure 1: Solution obtenue pour **"aerodrome_20_1.txt"**

Après constatation, les contraintes GCS sont plus robustes que les DFJ, ce qui est en accord avec l'article de recherche. De plus, comme il est possible de le constater précédemment, la formulation SF est la plus rapide en terme de complexité de calcul. Donc pour les contraintes exponentielles, nous ne présenterons que la formulation GCS accompagné de la formulation SF.

On a dressé le tableau suivant qui résume les principales caractéristiques des cinq formulations pour arriver à la solution "obtenue" en fonction des fichiers tests : le temps de calcul **T** en secondes, la solution optimale **Opt** et l'écart entre la valeur du noeud courant et la meilleur borne obtenue **gap**.

	MTZ	SF	MCF	GCS + SF
20_1	T = 8, Opt= 89, gap= 0	T = 1, Opt= 89, gap= 0	T = 4, Opt= 89, gap= 0	T = 0, Opt= 89, gap= 0
20_2	T = 0, Opt= 77, gap= 0	T = 1, Opt= 77, gap= 0	T = 3, Opt= 77, gap= 0	T = 1, Opt= 77, gap= 0
20_3	T = 0, Opt= 87, gap= 0	T = 0, Opt= 87, gap= 0	T = 1, Opt= 87, gap= 0	T = 0, Opt= 87, gap= 0
30_1	T = 10, Opt= 124, gap= 0	T = 0, Opt= 124, gap= 0	T = 1, Opt= 124, gap= 0	T = 0, Opt= 124, gap= 0
40_1	T = 5, Opt= 112, gap= 0	T = 1, Opt= 112, gap= 0	T = 11, Opt= 112, gap= 0	T = 4, Opt= 112, gap= 0
50_1	T = 20, Opt= 163, gap= 0	T = 1, Opt= 163, gap= 0	T = 20, Opt= 163, gap= 0	T = 1, Opt= 163, gap= 0
70_1	T = 225 , Opt= 220.44, gap= 4.98%	T = 4, Opt= 224, gap= 0	T = 11, Opt= 224, gap= 0	T = 2, Opt= 224, gap= 0
80_1	T = 225 , Opt= 456, gap= 20%	T = 10, Opt= 399, gap= 0	T = 22, Opt= 399, gap= 0	T = 14, Opt= 399, gap= 0

Table 1: Tableau des principales caractéristiques des cinq formulations en fonction des fichiers tests

De plus on dresse le tableau avec GCS accompagné des trois formulations polynomiales en indiquant pour chacune l'optimum à la racine.

	Optimum	MTZ	SF	MCF
70_1	224	101.2	116.4	216
80_1	399	154	198.8	395

Table 2: Tableau de l'optimum à la racine de GCS accompagné des trois formulations polynomiales en fonction des fichiers tests

On constate donc que la formulation GCS+SF se démarque largement par rapport aux autres formulations sur le point de vu complexité. De plus on constate que la formulation SF est plus efficace en termes de complexité que la formulation GCS+SF pour des dimensions assez grandes ($n = 80$).

Ainsi il est préferrable d'utiliser la formulation GCS+SF pour des dimensions faibles ($n \leq 70$) car elle permet d'obtenir une solution optimale en un temps qui est le plus faible et surtout avec une grande rapidité. Par contre pour des dimensions élevées ($n = 80$), on préférera la formulation SF.