Gabriel Muñoz

Optimisation des opérations au sol lors des escales avion

Modélisation mathématique et formulation du problème

Introduction

Ce document présente la modélisation mathématique d'un problème de planification des opérations d'escale aéroportuaire. Le projet est issu d'un test technique réalisé dans le cadre d'un processus de recrutement, puis retravaillé et intégré à mon portfolio personnel afin d'illustrer une démarche complète d'optimisation, depuis la formalisation mathématique jusqu'à l'implémentation en Python et la visualisation des solutions.

Le cœur du projet repose sur un modèle d'optimisation combinatoire implémenté en Python, accompagné d'une démonstration sur des données d'exemple (notebooks Jupyter), de diagrammes de Gantt représentant les plannings obtenus, et d'outils de visualisation des affectations de véhicules. Le présent document constitue une note de support décrivant les hypothèses, les ensembles, les variables et les contraintes utilisées dans la modélisation.

Contexte et problématique

Contexte opérationnel

Lors d'une escale aérienne, un avion doit subir une séquence structurée de tâches opérationnelles avant son prochain départ. Ces tâches incluent notamment le débarquement et l'embarquement des passagers, la gestion des bagages, le ravitaillement en carburant, le renouvellement en eau potable, le rechargement en restauration, ainsi que les opérations de repoussage (**pushback**).

Une organisation typique de ces tâches est représentée en Figure 1. Certaines tâches peuvent être réalisées en parallèle, tandis que d'autres doivent respecter des précédences strictes (par exemple, l'embarquement ne peut débuter qu'après certaines opérations techniques).

De plus, la nature de certaines tâches et les véhicules requis dépendent du type de stationnement de l'avion :

- sur un poste **on contact**, le débarquement et l'embarquement nécessitent une passerelle (**bridge**),
- sur un poste **off contact**, ces opérations sont réalisées à l'aide d'escaliers mobiles (**mobile stairs**).

Dans ce projet, on suppose que chaque avion doit réaliser l'ensemble des tâches décrites lors de son escale, et que la structure de précédence entre tâches est connue et identique pour toutes les escales.

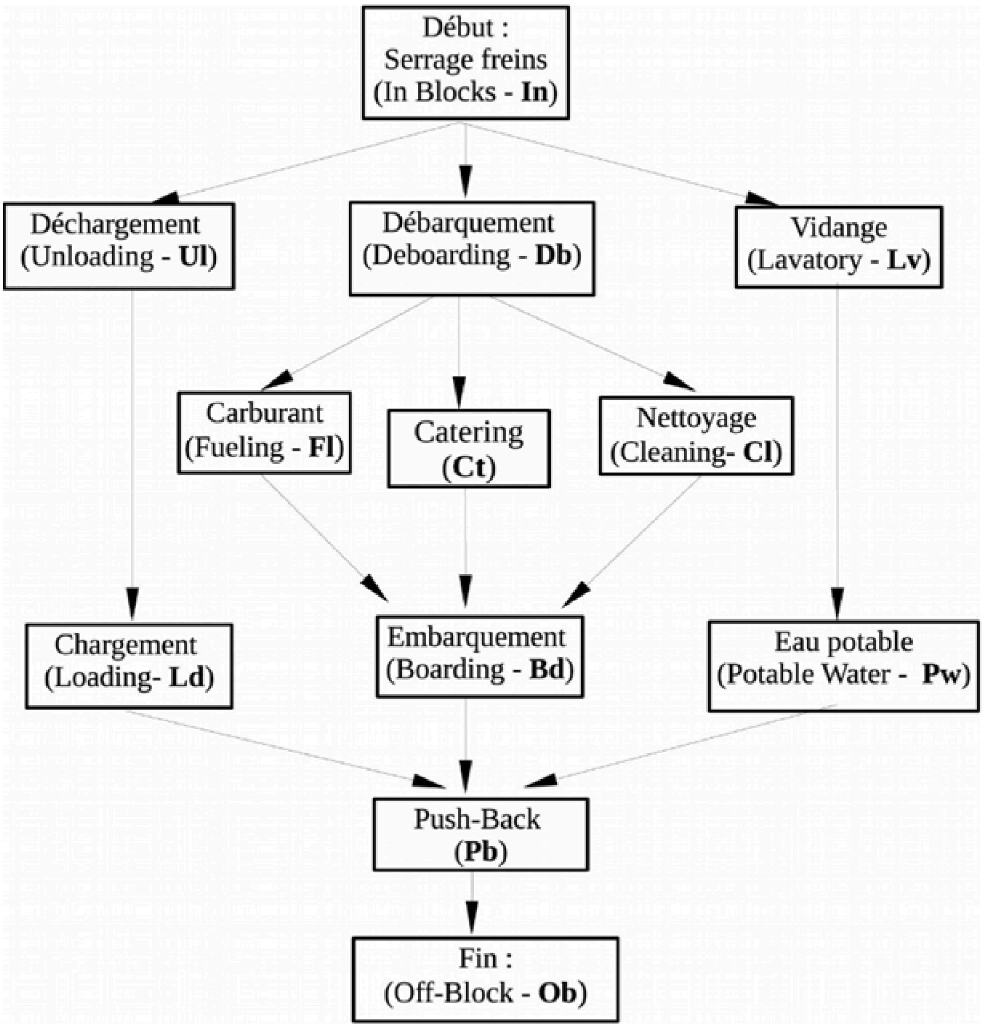

Figure 1: Tâches à effectuer lors d'une escale

Ressources et véhicules

Les tâches sont effectuées par un ensemble de véhicules spécialisés, chacun étant capable de réaliser un sous-ensemble précis des tâches :

- **Bridge** : passerelle pour l'embarquement et le débarquement en poste **on contact**,
- **Mobile stairs** : escaliers mobiles pour l'embarquement et le débarquement en poste **off contact**,
- **Water truck** : vidange et renouvellement de l'eau potable,
- **Fueling truck** : ravitaillement en carburant,
- **Baggage truck** : chargement et déchargement des bagages,
- **Tractor** : opérations de **pushback**,
- **Catering truck** : rechargement en nourriture.

Chaque véhicule démarre depuis une base commune et se déplace entre les différents parkings pour effectuer les tâches qui lui sont assignées.

Objectifs et périmètre du problème

L'objectif de ce travail est de proposer une modélisation mathématique permettant de planifier l'exécution des tâches d'escale et l'allocation des véhicules associés, en minimisant le nombre de véhicules effectivement utilisés tout en respectant les contraintes temporelles et opérationnelles.

Certaines tâches peuvent être priorisées selon des critères opérationnels (par exemple, réaliser le débarquement le plus tôt possible afin de limiter l'attente des passagers).

Les données supposées disponibles incluent :
- les durées d'exécution des tâches (potentiellement différentes selon le type de stationnement),
- les fenêtres temporelles de chaque escale,
- les distances et durées de déplacement entre parkings,

L'affectation des avions aux parkings est considérée comme exogène dans ce travail et n'est pas optimisée ; elle constitue une donnée d'entrée de la modélisation.

Formulation et résolution du problème

Définition des variables

1) Données d'entrée

À partir des données d'entrée, les ensembles suivants sont construits :
- $\mathbf{Avions}$ qui regroupe tous les avions
- $\mathbf{Avions\ On} \subset \mathbf{Avions}$ qui regroupe tous les avions qui sont en on contact
- $\mathbf{Avions\ Off} \subset \mathbf{Avions}$ qui regroupe tous les avions qui sont en off contact
- $\mathbf{Véhicules}$ qui regroupe tous les véhicules
- $\mathbf{Tâches}$ qui regroupe toutes tâches
- $\mathbf{Tâches\ V} \subset \mathbf{Tâches}$ qui regroupe toutes les tâches nécessitant un véhicule
- $\mathbf{Tâches}_{\boldsymbol{v}} \subset \mathbf{Tâches\ V}$ qui regroupe $\forall v \in$ Véhicules toutes les tâches que peut faire ce véhicule (par exemple l'embarquement et le débarquement pour les véhicules de type mobile stairs)
- $\mathbf{Parking}$ qui regroupe tous les parkings et la base
- $\mathbf{Ordre} = [\![0, 4n]\!]$ qui est l'ensemble de l'ordre des endroits ou se situe un véhicule. Un véhicule peut au maximum faire deux tâches par avions, et rentrer à la base entre chacune des ces tâches. Ce qui donne une cardinalité maximum de 4n+1. Cette discrétisation en ordres permet de transformer les déplacements continus des véhicules en une séquence finie de positions ordonnées, rendant possible une formulation linéaire et réifiable des contraintes de déplacement et d'affectation dans un cadre MILP ou CP.

Nous pouvons ainsi construire plusieurs applications
- *parking*: **Avions** $\rightarrow$ **Parking**, a $\rightarrow$ p avec p le parking associé à l'avion a
- *duree*: **Parking²** $\rightarrow \mathbb{R}^+$, (i,j) $\rightarrow$ d(i,j) avec (i,j) deux parking et d(i,j) la durée de déplacement entre le parking i et le parking j. Si i=j alors duree = 0
- *duree_tâche*: **Avions** $*$ **Tâche** $\rightarrow \mathbb{R}^+$ (a,t) $\rightarrow$ τ(a,t) avec τ(a,t) la durée nécessaire à l'avion a pour réaliser la tâche t. En effet selon si l'avion est on ou off, la durée peut varier.

Il est également possible d'obtenir pour chaque escale les dates au plus tôt et au plus tard pour chacune des tâches. Nous notons respectivement $\text{task}^{\min}$ et $\text{task}^{\max}$ les deux matrices de taille $(n, 12)$ indiquant ces heures. Ainsi $\forall t \in$ Tâches, $\forall a \in$ Avions nous pouvons noter respectivement $\text{task}_{a,t}^{\min}$ et $\text{task}_{a,t}^{\max}$.
Enfin nous noterons :
$H_{\max} = \max_{a,t}\left(\text{task}_{a,t}^{\max}\right)$

$T_{\max} = \max_{a,t}(\text{duree\_tâche}(a, t))$

$D_{\max} = \max_{i,j}(\text{duree}(i, j))$.

2) Variables de décision

$\forall v \in$ Véhicules, $\forall a \in$ Avions, $\forall t \in$ Tâches, $\forall (i, j) \in$ Parking², $\forall o \in$ Ordre

- $\text{task}_{a,t} \in \left[\text{task}_{a,t}^{\min}, \text{task}_{a,t}^{\max}\right]$, heure de début effective de la tâche t sur l'avion a
- $h_{v,o} \in [0, H_{\max} + D_{\max} + T_{\max}]$, heure d'arrivée du véhicule v à l'ordre o
- $\text{duree\_deplacement}_{v,o,o+1} \in [0, D_{\max}]$, durée du déplacement du véhicule v entre l'ordre o et l'ordre o+1
- $\text{loc}_{v,o,i} = \begin{cases} 1 \text{ si v se situe en i} \\ 0 \text{ sinon} \end{cases}$
- $u_v = \begin{cases} 0 \text{ si véhicule jamais utilisé} \\ 1 \text{ sinon} \end{cases}$
- $\text{traj}_{v,o,i,j} = \begin{cases} 1 \text{ si v fait trajet i à j entre l'instant o et o+1} \\ 0 \text{ sinon} \end{cases}$
- $\text{b}_{v,a,o} = \begin{cases} 1 \text{ si le véhicule v s'occupe de l'avion a à l'instant o} \\ 0 \text{ sinon} \end{cases}$
- $\text{disp}_{v,a,o,t} = \begin{cases} 1 \text{ si le véhicule v s'occupe de l'avion a à l'instant o pour la tâche t} \\ 0 \text{ sinon} \end{cases}$

Contraintes

Dans les contraintes suivantes un terme M apparait plusieurs fois. Ce dernier est un entier à choisir en fonction des données d'entrée. Chaque M de chaque contrainte est différent

I) Liens entre les variables

1) Lien entre loc et b

Cette contrainte force la position à l'ordre o du véhicule v à être égale à celle du parking(a), s'il s'occupe de l'avion a à l'ordre o

$\forall v \in \text{Véhicules}, \forall a \in \text{Avions}, \forall o \in \text{Ordre}$

$\text{loc}_{v,o,\text{parking(a)}} \geq b_{v,a,o}$

2) Lien entre u et b

Si un véhicule n'est jamais utilisé ($u_v = 0$) alors il n'est jamais utilisé pour aucun a, o $\forall v \in \text{Véhicules}$

$M.u_v \geq \sum_{(a,o) \in \text{Avions} * \text{Ordre}} b_{v,a,o}$

Ici on peut poser $M = 2n$ car un véhicule pourra au maximum s'occuper des toutes les tâches de tous les avions. Or un type de véhicule peut soit faire 1 ou 2 tâche. (Les contraintes II.5 et II.6 s'assurent de cela)

3) Lien entre b et disp

Un véhicule v s'occupe de l'avion a à l'ordre o, s'il effectue une tâche t sur l'avion a à l'ordre o

$\forall v \in \text{Véhicules}, \forall a \in \text{Avions}, \forall o \in \text{Ordre}$

$b_{v,a,o} = \sum_{t \in \text{Tâches}_v} \text{disp}(v, a, o, t)$

4) Lien entre duree et traj

La durée de déplacement d'un véhicule entre l'ordre o et l'ordre $o + 1$ est égale à la durée du trajet entre i et j avec i et j le déplacement effectué par le véhicule v à l'ordre o

$\forall v \in \text{Véhicules}, \forall o \in \text{Ordre} \setminus \{4n\}$

$\text{duree\_deplacement}_{v,o,o+1} = \sum_{(i,j) \in \text{ Parking}^2} \text{duree}(i, j).\text{traj}_{v,o,i,j}$

5) Lien entre loc et traj

$\forall v \in \text{Véhicules}, \forall o \in \text{Ordre} \setminus \{4n\}$

$\sum_{(i,j) \in \text{ Parking}^2} \text{traj}_{v,o,i,j} = 1$; un véhicule fait exactement 1 trajet à chaque ordre (peut être d'un lieu vers le même)

$\sum_{i \in \text{ Parking}} \text{loc}_{v,o,i} = 1$; un véhicule se situe exactement à un seul endroit à la fois

$\forall i \in \text{Parking}, \text{loc}_{v,o,i} = \sum_{j\in \text{ Parking}} \text{traj}_{v,o,i,j}$; la position de v à o est égale au point de départ de son trajet
$\forall j \in \text{Parking}, \text{loc}_{v,o+1,j} = \sum_{i\in \text{ Parking}} \text{traj}_{v,o,i,j}$; la position de v à $o+1$ est égale au point de d'arrivée de son trajet

II) Les contraintes métiers

1) Ordre des tâches
Si l'on numérote les tâches de cette manière :

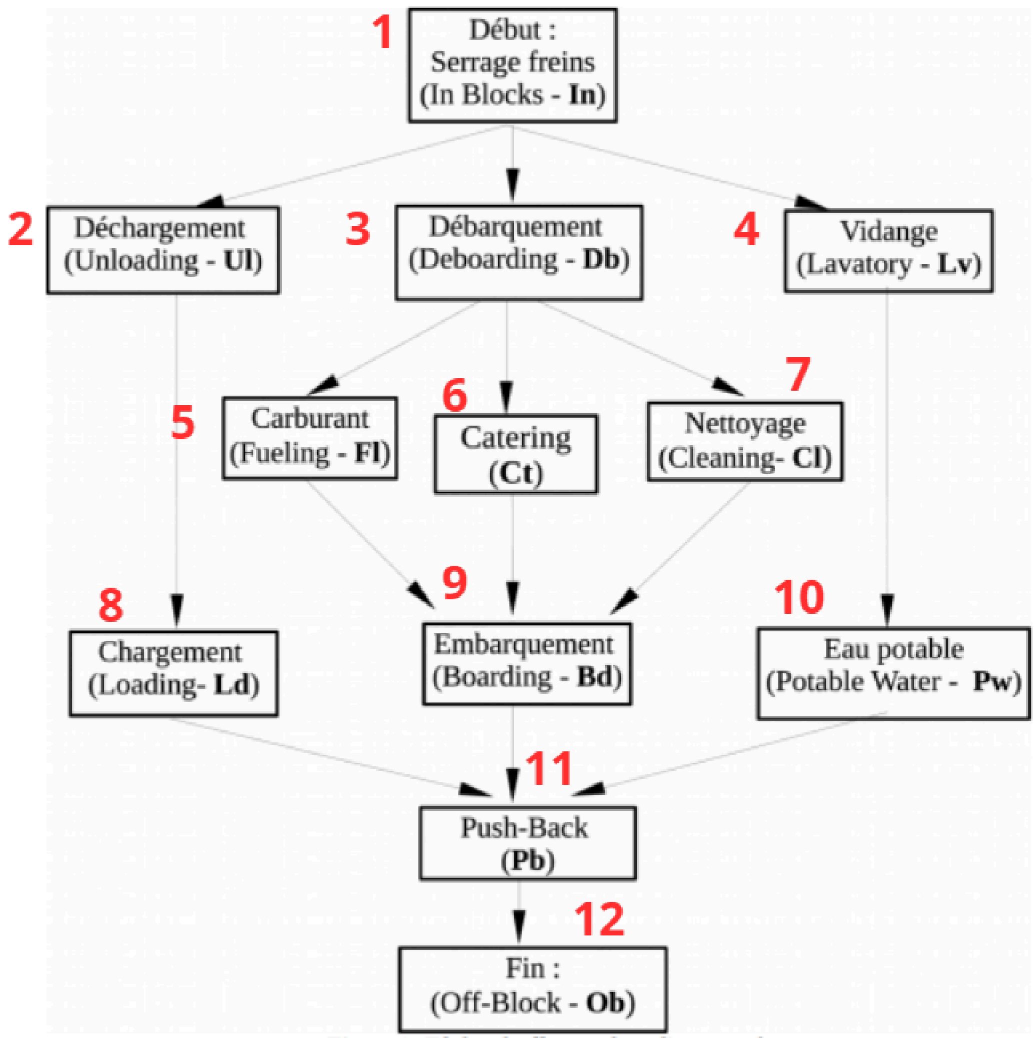

Figure 1: Tâches à effectuer lors d'une escale

$\forall a \in \text{Avions}$
$\text{task}_{a,[2,3,4]} \geq \text{task}_{a,1}$ + duree_tâche$(a, 1)$
$\text{task}_{a,[5,6,7]} \geq \text{task}_{a,3}$ + duree_tâche$(a, 3)$
$\text{task}_{a,8} \geq \text{task}_{a,2}$ + duree_tâche$(a, 2)$
$\text{task}_{a,10} \geq \text{task}_{a,4}$ + duree_tâche$(a, 4)$
$\text{task}_{a,9} \geq \text{task}_{a,5}$ + duree_tâche$(a, 5)$
$\text{task}_{a,9} \geq \text{task}_{a,6}$ + duree_tâche$(a, 6)$
$\text{task}_{a,9} \geq \text{task}_{a,7}$ + duree_tâche$(a, 7)$
$\text{task}_{a,11} \geq \text{task}_{a,8}$ + duree_tâche$(a, 8)$

$\text{task}_{a,11} \geq \text{task}_{a,9}$ + duree_tâche$(a, 9)$
$\text{task}_{a,11} \geq \text{task}_{a,10}$ + duree_tâche$(a, 10)$
$\text{task}_{a,12} \geq \text{task}_{a,11}$ + duree_tâche$(a, 11)$

2) Un véhicule ne peut s'occuper que d'un seul avion à la fois
$\forall v \in \text{Véhicules}, \forall o \in \text{Ordre}$
$\sum_{a \in \text{ Avions}} b_{v,a,o} \leq 1$
$\sum_{(a,t) \in \text{ Avions} * \text{Taches}_v} \text{disp}_{v,a,o,t} \leq 1$
La seconde est impliquée par la première donc non obligatoire, cependant cela peut aider à la relaxation LP

3) Une tâche ne peut commencer que lorsque le véhicule nécessaire est présent
$\forall a \in \text{Avions}, \forall v \in \text{Véhicules}, \forall o \in \text{Ordre}, \forall t \in \text{TâchesV}$
$h_{v,o} \leq \text{task}_{a,t} + \left(1 - \text{disp}_{v,a,o,t}\right).M$
Si le véhicule v s'occupe de l'avion a pour la tâche t à l'ordre o, alors son heure d'arrivée doit être inférieure à l'heure de début de la tâche. Sinon $h_{v,o}$ n'est pas contraint
Ici on peut choisir $M = H_{\text{max}} + D_{\text{max}} + T_{\text{max}}$
Cette contrainte peut se reformuler sous forme indicatrice comme ceci :

$\text{disp}_{v,a,o,t} = 1 \Rightarrow h_{v,o} \leq \text{task}_{a,t}$

4) Prise en compte du temps de déplacement
$\forall v \in \text{Véhicules}, \forall o \in \text{Ordre} \setminus \{4n\}$
$h_{v,o+1} \geq h_{v,o} + \text{duree\_deplacement}_{v,o,o+1}$

4 bis) Prise en compte du temps pour effectuer la tâche + déplacement pour le calcul du temps d'arrivée à l'ordre o+1
$\forall v \in \text{Véhicules}, \forall a \in \text{Avions}, \forall t \in \text{Tâches V}, \forall o \in \text{Ordre} \setminus \{4n\}$
$h_{v,o+1} \geq \text{task}_{a,t} + \text{duree\_tâche}(a,t) + \text{duree\_deplacement}_{v,o,o+1} - \left(1 - \text{disp}_{v,a,o,t}\right).M$
Ici on peut choisir $M = H_{\text{max}} + T_{\text{max}}$
Cette contrainte peut se reformuler sous forme indicatrice comme ceci :
$\text{disp}_{v,a,o,t} = 1 \Rightarrow h_{v,o+1} \geq \text{task}_{a,t} + \text{duree\_tâche}(a,t) + \text{duree\_deplacement}_{v,o,o+1}$

5) Disponibilité des véhicules pour chaque avion pour chaque tâche
$\sum_{(v,o) \in \text{ Véhicules} * \text{Ordre}} \text{disp}_{v,a,o,t} = 1 \forall a \in \text{Avions}, \forall t \in \text{Tâches V}$

6) Un véhicule peut accomplir que ses tâches associées
$\text{disp}_{v,a,o,t} = 0 \quad \forall a \in \text{Avions}, \forall t \in \overline{\text{Tâches}_v}$
Ici $\overline{\text{Tâches}_v}$ correspond bien à l'ensemble complémentaire de Tâches_v par rapport à Tâches V

Objectif

L'objectif est formulé à l'aide des ensembles suivants :
- $\textbf{Tâches Prio} \subset \textbf{Tâches}$ qui regroupe toutes les tâches qui doivent être effectuées le plus tôt (exemple : le débarquement doit être fait le plus tôt possible pour empêcher les passagers d'attendre dans l'avion)
- $\textbf{Tâches Anti Prio} \subset \textbf{Tâches}$ qui regroupe toutes les tâches qui doivent être effectuées le plus tard (exemple : l'embarquement doit être fait le plus tard afin de ne pas faire attendre les passagers dans l'avion). Attention $\text{Tâches Anti Prio} \neq \overline{\text{Tâches Prio}}$

$\lambda_1 * \sum_{v \in \text{ Véhicules}} u_v + \lambda_2 * \sum_{(a,i) \in \text{ Avions } * \text{ Tâches Prio}} \text{task}_{a,i} - \lambda_3 * \sum_{(a,i) \in \text{ Avions } * \text{ Tâches Anti Prio}} \text{task}_{a,i})$
Ici λ_1, λ_2 et λ_3 sont des constantes > 0 à calibrer selon les priorités métier

Conditions Initiales

$\forall v \in \text{Véhicules } \text{loc}_{v,0,\text{base}} = 1$

$\text{loc}_{v,4n,\text{base}} = 1$

$\forall v \in \text{Véhicules } h_{v,0} = 0$

Limites de la modélisation

Par souci de lisibilité, j'ai volontairement posé quelques hypothèses simplificatrices. D'abord, tous les véhicules sont supposés aller à la même vitesse et les temps de déplacement $\text{duree}(i, j)$ sont exogènes et indépendants de l'encombrement : on ne modélise ni trafic ni congestion en circulation. Les fenêtres temporelles sont « dures » (une tâche ne peut pas démarrer avant task^{min} ni après task^{max}), ce qui peut rendre le problème infaisable si les données sont tendues ; on pourrait introduire des marges d'avance/retard pénalisées. Côté durées d'exécution, j'ai exprimé $\text{duree\_tâche}(a, t)$ au niveau de l'avion a (et pas seulement selon on/off) pour rester plus général ; on ignore toutefois l'incertitude (aléas d'arrivée, météo, variabilité opératoire) ainsi que durées de tâches dépendantes de la séquence qui pourraient s'ajouter selon la tâche précédente/suivante.

Sur le catalogue des tâches et les ressources, certaines familles d'avions peuvent nécessiter des tâches supplémentaires, et, pour une même tâche, le type d'avion peut imposer un type de véhicule spécifique : aujourd'hui ces compatibilités fines « véhicule ↔ avion ↔ tâche » ne sont gérées que via l'éligibilité Tâches_v, pas par des ensembles dédiés $V_{\{a,t\}}$. De même, le modèle ne traite pas les tâches multi-ressources (exiger simultanément plusieurs véhicules) ni les contraintes de sécurité inter-tâches (co-activité interdite à proximité d'un ravitaillement), ni les ressources humaines (équipages, pauses, multi-compétences). Ces dimensions pourraient être intégrées par des contraintes de simultanéité / non-chevauchement et des bilans de capacité.

Du point de vue spatial, j'autorise par défaut tous les arcs (i, j) entre parkings (graphe complet). En réalité, certains trajets directs $i \rightarrow j$ n'existent pas ou ne sont pas autorisés et doivent passer par des nœuds intermédiaires (sens uniques, barrières physiques, points de passage réglementés, fermetures temporaires…), d'où l'intérêt de restreindre à un sous-graphe $E \subset \text{Parking}^2$ (orienté si besoin). Par ailleurs, il n'y a pas de capacité de parking ni de capacité de circulation : plusieurs véhicules peuvent occuper le même parking et emprunter le même arc au même instant, ce qui peut « attirer » artificiellement tout le monde vers un parking central. L'objectif ne pénalise pas explicitement la distance/attente (ni un coût fixe d'occupation) : en l'état, il privilégie le respect des fenêtres/priorités et l'ouverture minimale de véhicules, sans internaliser les zigzags ou les stationnements prolongés. Enfin, comme précisé, les placements des avions sont pris comme données d'entrée ; on pourrait aller plus loin et co-optimiser l'affectation aux parkings/gates avec l'ordonnancement, au prix d'un problème couplé plus lourd.

Sur la structure temporelle et l'échelle, j'utilise une discrétisation en $4n + 1$ « pas » avec exactement un arc par pas : c'est simple mais un peu rigide ; une formulation **event-based** ou l'introduction d'un indicateur d'activité permettrait de réduire le nombre de pas réellement exploités. La taille du modèle croît en $O(|\text{Véhicules}| \cdot |\text{Ordre}| \cdot |\text{Parking}|^2)$ via les variables traj, ce qui peut devenir coûteux quand le nombre de parkings ou d'ordres augmente. On n'impose pas non plus une heure de fin maximale par véhicule (contrainte de fin de service), qui peut être pertinente en exploitation. Enfin concernant les contraintes activables, j'ai conservé les deux écritures (avec Big-M et en version indicatrice) dans le document ; en pratique, la forme **indicatrice** est préférable (plus propre et plus stable numériquement), la version Big-M restant une alternative documentée.

Mise en oeuvre

Dans l'implémentation Python, cette formulation est traduite en programmation par contraintes via OR-Tools CP-SAT, en remplaçant les variables de trajectoire explicites par des contraintes d'ordre et des temps de transition.