



**HAL**  
open science

# Optimisation du parking des avions à Paris Charles de Gaulle

Thibault Falque, Bertrand Mazure, Christophe Lecoutre, Karim Tabia

► **To cite this version:**

Thibault Falque, Bertrand Mazure, Christophe Lecoutre, Karim Tabia. Optimisation du parking des avions à Paris Charles de Gaulle. 23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision, INSA Lyon, Feb 2022, Villeurbanne - Lyon, France. hal-03595354

**HAL Id: hal-03595354**

**<https://hal.science/hal-03595354>**

Submitted on 3 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Optimisation du parking des avions à Paris Charles de Gaulle

Thibault Falque<sup>1,2</sup>, Bertrand Mazure<sup>2</sup>, Christophe Lecoutre<sup>2</sup>, Karim Tabia<sup>2</sup>

<sup>1</sup> Exakis Nelite

<sup>2</sup> CRIL, Artois University & CNRS

{falque,mazure,lecoutre,tabia}@cril.fr

**Mots-clés :** *programmation par contraintes, optimisation, planification, applications, modélisation.*

Avant la crise sanitaire du COVID-19, l'Association internationale du transport aérien (IATA) prévoyait que le nombre de passagers aériens doublerait d'ici 2036, pour atteindre 7,8 milliards de personnes. En France, les infrastructures des aéroports parisiens devaient voir la fréquentation des passagers passer de 35/40 millions à quelques centaines de millions. Même si la crise sanitaire du COVID-19 a considérablement diminué le trafic aérien, la pression concurrentielle reste forte, et les compagnies aériennes et aéroportuaires doivent réduire drastiquement leurs coûts opérationnels. Dans un tel contexte, l'optimisation de la gestion des ressources aéroportuaires est essentielle pour maîtriser les coûts induits tout en conservant une bonne qualité de services.

L'utilisation de techniques d'intelligence artificielle dans les transports et, plus spécifiquement dans le secteur aéroportuaire, n'est pas nouvelle. En effet, il existe de nombreux problèmes de transport aérien où des systèmes d'aide à la décision sont utilisés tout en intégrant des composants d'intelligence artificielle. Les exemples les plus évidents sont l'allocation de ressources telles que la planification des portes d'embarquement pour les avions (en tenant compte de caractéristiques importantes comme, par exemple, les dimensions de l'avion et la satisfaction des compagnies aériennes).

En terme de technologies, notons que la programmation en nombres entiers binaires a été utilisée [12] pour réduire les distances de marche des passagers et que la programmation linéaire en nombres entiers (ILP) pour minimiser la somme des pénalités de retard [10] ou la minimisation de la déviation des heures d'arrivée et de départ à l'aéroport d'Amsterdam Schiphol [5].

Un problème classique du transport aérien est le problème d'affectation des portes d'aéroport (AGAP), qui consiste à affecter chaque vol (avion) à une porte d'embarquement disponible tout en maximisant à la fois le confort des passagers et l'efficacité opérationnelle de l'aéroport. De nombreuses approches ont été proposées pour l'AGAP [1, 4]. On peut citer par exemple, des modèles [8, 9] ont été développés en OPL (*Open programming Language*) [15] et en MIP (*mixed-integer linear programming*) [11].

Dans cette session, nous nous intéressons au problème d'affectation des parkings d'aéroport (APAP) tel que défini à l'aéroport international CDG. Nous proposons une approche de Programmation par Contraintes (PC) tirant avantage (d'un point de vue algorithmique) des progrès réalisés au cours de la dernière décennie sur les contraintes défini en contraintes extensions(ou contraintes tables).

On peut considérer l'affectation d'un vol à un parking pendant une rotation comme une tâche. En fait, selon la durée de la rotation, l'avion peut être déplacé plusieurs fois. Un vol peut alors être placé sur un sur un seul parking pendant toute la durée de sa rotation (1 tâche), sur deux parkings successifs (2 tâches), ou même trois parkings (3 tâches). Les vols secs (composés d'une arrivée ou d'un départ) n'occupent nécessairement que 1 stationnement (tâche). La construction de l'ordonnancement du problème de stationnement consiste à trouver un parking pour chaque tâche de chaque rotation. Notons que la précision de l'emploi du temps est de 5 minutes.

Il existe certaines contraintes physiques, imposées par l'infrastructure. Les contraintes de *capacité* empêchent certains types de avions d'être placés sur certains parkings. Les contraintes *shadow* bloquent le positionnement d'un avion sur certains parkings proches (quel que soit le type d'avion). Les contraintes *réductions* sont assez similaires aux contraintes *shading* sauf qu'elles prennent en compte les types avions. Finalement, contraintes *NoOverlap* reflètent l'impossibilité physique d'affecter deux tâches (deux vols) au même parking.

Ce travail présentera ce problème plus formellement au moyen d'un réseau de contraintes en utilisant un exemple complet de planification et présentera le modèle réalisé à l'aide de la librairie Python PyCSP<sup>3</sup> [7] qui permet de générer le problème au format XCSP3 [2, 3] qui est reconnu par des solveurs standards (AbsCon Essence) [6], Choco [14], OsaR [13], et PicatSAT [16]. Enfin nous présenterons le gain de cette approche par rapport à la solution utilisée actuellement par Aéroport de Paris.

## Références

- [1] A Bouras, Mageed A. Ghaleb, Umar S. Suryahatmaja, and Ahmed M. Salem. The Airport Gate Assignment Problem : A Survey. *The Scientific World Journal*, 2014 :e923859, November 2014.
- [2] F. Boussemart, C. Lecoutre, G. Audemard, and C. Piette. XCSP<sup>3</sup> : An Integrated Format for Benchmarking Combinatorial Constrained Problems. Technical Report arXiv :1611.03398, Specifications, CoRR, 2016.
- [3] F. Boussemart, C. Lecoutre, G. Audemard, and C. Piette. XCSP<sup>3</sup>-core : A Format for Representing Constraint Satisfaction/Optimization Problems. Technical Report arXiv :2009.00514, Specifications 3.0.6, CoRR, 2020.
- [4] Gülesin Sena Daş, Fatma Gzara, and Thomas Stützle. A review on airport gate assignment problems : Single versus multi objective approaches. *Omega*, 92 :102146, April 2020.
- [5] Guido Diepen, J.M. Akker, J.A. Hoogeveen, and J.W. Smeltink. Using column generation for gate planning at Amsterdam Airport Schiphol. January 2007.
- [6] C. Lecoutre. *ACE : A Generic Constraint Solver*. Technical Report. v1 on CoRR, to appear, October 2021.
- [7] C. Lecoutre and N. Szczepanski. *PYCSP<sup>3</sup> : Modeling Combinatorial Constrained Problems in Python*. Technical Report. v1 on CoRR, arXiv :2009.00326, September 2020. 100 pages.
- [8] C Li. Airport Gate Assignment : New Model and Implementation. *arXiv :0811.1618 [cs]*, November 2008.
- [9] C Li. Airport Gate Assignment A Hybrid Model and Implementation. *arXiv :0903.2528 [cs]*, March 2009.
- [10] A. Lim, B. Rodrigues, and Y. Zhu. Airport Gate Scheduling with Time Windows. *Artif Intell Rev*, 24(1) :5–31, September 2005.
- [11] J. L'Ortye, M. Mitici, and H.G. Visser. Robust flight-to-gate assignment with landside capacity constraints. *Transportation Planning and Technology*, 44(4) :356–377, May 2021.
- [12] R. S. Mangoubi and Dennis F. X. Mathaisel. Optimizing Gate Assignments at Airport Terminals. *Transportation Science*, 19(2) :173–188, 1985.
- [13] OsaR Team. OsaR : Scala in OR, 2012.
- [14] C. Prud'homme, J.-G. Fages, and X. Lorca. Choco-solver, TASC, INRIA Rennes, LINA, Cosling S.A. 2016.
- [15] P. van Hentenryck. *The OPL Optimization Programming Language*. The MIT Press, 1999.
- [16] N. F. Zhou, H. Kjellerstrand, and J. Fruhman. *Constraint Solving and Planning with Picat*. Springer, 2017.