



HAL
open science

Calcul des dates d'atterrissage d'une séquence d'avions pour des fonctions de coût convexes et affines par morceaux

Maurice Diamantini, Alain Faye, Julien Khamphousone

► **To cite this version:**

Maurice Diamantini, Alain Faye, Julien Khamphousone. Calcul des dates d'atterrissage d'une séquence d'avions pour des fonctions de coût convexes et affines par morceaux. ROADEF 2019, Feb 2019, Le Havre, France. hal-02354405

HAL Id: hal-02354405

<https://hal.science/hal-02354405>

Submitted on 7 Nov 2019

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.

Calcul des dates d'atterrissage d'une séquence d'avions pour des fonctions de coût convexes et affines par morceaux

Maurice Diamantini¹, Alain Faye², Julien Khamphousone²

¹ UMA-ENSTA 91762 Palaiseau, France

`maurice.diamantini@ensta-paristech.fr`

² CEDRIC-ENSIIE F-91025 Evry, France

`{alain.faye,julien.khamphousone}@ensiie.fr`

Mots-clés : *OR in Airlines, Aircraft Landing Problem, dates d'atterrissage, programmation dynamique, algorithme polynomial.*

1 Présentation du problème

Ce papier étudie le problème du séquençage des avions lors de leur arrivée à l'aéroport, problème connu dans la littérature sous le nom de Aircraft Landing Problem [1]. Il s'agit de séquencer les avions arrivant sur la piste d'atterrissage tout en respectant des conditions de sécurité entre les avions. Les avions créent des turbulences et une durée minimum entre deux atterrissages successifs doit être respectée. La durée de séparation dépend du type des avions qui se suivent. Un petit avion qui atterrit derrière un gros avion doit attendre plus longtemps qu'un gros avion qui atterrit à la suite d'un petit. Chaque avion i peut atterrir dans une certaine fenêtre de temps $[E_i, L_i]$. E_i est la date au plus tôt à laquelle l'avion peut atterrir, L_i est la date au plus tard. Dans cette fenêtre, T_i est la date préférée d'atterrissage qui correspond à la date à laquelle l'avion arriverait sur la piste s'il allait à sa vitesse de croisière. Si l'avion i était seul il atterrirait à la date T_i mais en présence d'autres avions un arbitrage est nécessaire. Les avions doivent soit accélérer pour atterrir plus tôt ou au contraire ralentir voire faire des boucles pour arriver plus tard afin de respecter les contraintes de sécurité. Une déviation par rapport à la date préférée d'atterrissage engendre un coût. Dans la littérature on considère généralement qu'une avance ou un retard engendre un coût linéaire en fonction de l'écartement à la date préférée d'atterrissage. L'objectif est de minimiser le coût total de déviation. Sur chaque piste, le problème se décompose en deux phases : d'abord choisir l'ordre des avions et ensuite calculer les dates d'atterrissage. Ce dernier problème peut se résoudre par un PL (Programme Linéaire) [3, 5, 6, 7]. A. Faye [4] propose un algorithme de complexité quadratique en fonction du nombre d'avions. Cependant, un coût linéaire peut s'avérer assez éloigné des coûts réels encourus. Par exemple, un retard peut avoir des conséquences sur les passagers en correspondance et sur les vols ultérieurs sur lesquels le personnel de bord devra prendre place. On conçoit facilement que plus le retard est grand, plus les complications sont nombreuses et que plus l'impact sur le coût s'accroît. Il est donc légitime de modéliser la fonction coût par une fonction convexe et affine par morceaux centrée en la date préférée d'atterrissage et avec des pentes croissantes de part et d'autre de cette date. Pour des raisons d'équité entre compagnies aériennes, une fonction de ce type a été introduite par Soomer et Franx [6]. Pour un ordre fixé des avions, le coût total était calculé par un PL. Ici, nous proposons un algorithme polynomial dont la complexité dépend à la fois du nombre d'avions et du nombre de pentes de la fonction de coût d'un avion. Ainsi, si n est le nombre d'avions et si b est le nombre maximum de pentes que peut comporter le coût d'un avion, la complexité de l'algorithme est $O(n^2b^2)$. Cet algorithme est basé sur la programmation dynamique et est une généralisation de l'algorithme proposé dans [4].

2 Résultats numériques

Les tests numériques sont effectués sur la base des instances de la littérature [2] à partir desquelles des fonctions de coût ont été générées pour différents nombres de pentes. L’algorithme polynomial est codé en Julia et CPLEX est utilisé pour résoudre le PL. Une étude préliminaire montre que l’algorithme polynomial est 50 à 200 fois plus rapide que le PL selon la taille des instances. Pour évaluer l’apport de l’algorithme polynomial en terme de qualité de solution, nous implémentons chaque méthode dans un algorithme de descente à voisinage variable. L’algorithme de descente recherche la meilleure séquence d’avions vis-à-vis de l’objectif et le coût d’une séquence est calculé soit par l’algorithme polynomial dans une première version de la méthode de descente, soit par le PL dans une deuxième version. A chaque version de la méthode de descente est alloué un temps maximum d’exécution relativement court. Les résultats obtenus sont comparés à une meilleure solution qui a été calculée sur un temps beaucoup plus long. La Table 1 reporte les écarts relatifs à cette meilleure solution pour différentes instances caractérisées par le nombre d’avions n et le nombre de pentes des fonctions de coût. Δ_{AP} est l’écart relatif obtenu par la descente utilisant l’algorithme polynomial et Δ_{PL} est l’écart relatif obtenu par la descente utilisant le PL. Le gain apporté par l’algorithme polynomial est assez significatif et il est d’autant plus important que le nombre de pentes de la fonction de coût des avions augmente.

n	Temps max (s.)	2 pentes		10 pentes		50 pentes		200 pentes	
		Δ_{AP} (%)	Δ_{PL} (%)						
100	5	0.1	2.58	0.66	9.3	0.0	29.23	0.0	38.72
150	5	3.08	9.86	3.98	23.27	7.8	54.27	4.24	64.14
200	5	0.8	6.5	0.22	13.22	0.11	27.71	1.53	27.66
250	5	0.98	11.12	1.87	25.07	2.53	34.85	5.18	37.72
500	10	1.05	13.5	0.8	24.23	2.35	30.66	5.64	32.35

TAB. 1 – Comparaison des méthodes utilisant l’algorithme polynomial et la programmation linéaire

Références

- [1] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, D. Abramson (2000), *Scheduling Aircraft Landings-The static case*, Transportation Science **34(2)**, 180–197.
- [2] J.E. Beasley (1996), *Obtaining test problems via Internet*, Journal of Global Optimization **8**, 429–433. Available from : <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/airlandinfo.html>
- [3] A. Faye (2015), *Solving the Aircraft Landing Problem with time discretization approach*, European Journal of Operational Research **242**, 1028–1038.
- [4] A. Faye (2018), *A quadratic time algorithm for computing the optimal landing times of a fixed sequence of planes*, European Journal of Operational Research **270**, 1148–1157.
- [5] H. Pinol, J.E. Beasley (2006), *Scatter Search and Bionomic Algorithms for the aircraft landing problem*, European Journal of Operational Research **171**, 439–462.
- [6] M. J. Soomer, G. J. Franx (2008), *Scheduling aircraft landings using airlines’ preferences*, European Journal of Operational Research **190**, 277–291.
- [7] S. Vadlamani, S. Hosseini (2014), *A novel heuristic approach for solving aircraft landing problem with single runway*, Journal of Air Transport Management **40**, 144–148.