



Optimisation du roulage sur les grands aéroports

Jean-Baptiste Gotteland

► To cite this version:

Jean-Baptiste Gotteland. Optimisation du roulage sur les grands aéroports. EDIT 2002, Colloque des doctorants de l'École Doctorale Informatique et Télécommunications, Feb 2002, Toulouse, France. hal-00938023

HAL Id: hal-00938023

<https://enac.hal.science/hal-00938023>

Submitted on 15 May 2014

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 roulage sur les grands aéroports

Jean-Baptiste Gotteland
gotteland@recherche.enac.fr

Résumé

Une partie significative des retards aériens est imputable aux opérations au sol sur les grands aéroports. Afin d'étudier les possibilités d'amélioration du trafic au roulage, un outil de simulation a été développé et appliqué à Roissy et Orly. Différentes méthodes d'optimisation utilisant des algorithmes génétiques et des algorithmes de parcours de graphe permettent de résoudre les conflits au sol et de séquencer les mouvements sur chaque piste. Ces différentes méthodes peuvent être évaluées et comparées par le délai qu'elles génèrent en fonction du nombre de mouvements.

1 Introduction

La croissance du trafic aérien et plus particulièrement le développement récent de nombreux hubs commerciaux sont à l'origine d'une nouvelle forme de congestion sur les grandes plates-formes aéroportuaires : la plupart des mouvements ont tendance à être programmés aux mêmes heures "stratégiques".

Les nombreux retards occasionnés sont la source d'une incertitude croissante sur les heures de décollage et d'atterrissage. Les retards peuvent atteindre plusieurs dizaines de minutes aux heures de pointe, ce qui est extrêmement pénalisant pour l'ensemble des acteurs du transport aérien.

Beaucoup d'outils d'analyse et d'aide à la décision ont été développés pour le contrôle en vol, très peu pour le contrôle au sol. Cet article présente une modélisation du trafic au roulage sur un aéroport. Elle est intégrée dans un simulateur comprenant un module de résolution de conflit optimisant les trajectoires des avions.

Différentes méthodes d'optimisation sont comparées par corrélation entre le délai qu'elles génèrent et le nombre de mouvements impliqués. Elles utilisent des algorithmes génétiques et des algorithmes de parcours de graphe pour trouver le meilleur chemin et/ou les meilleurs points d'attente pour chaque avion, en respectant les normes de séparation au sol, pistes comprises.

2 Modélisation

Le problème est de trouver un ensemble optimal de trajectoires admissibles pour le trafic au roulage.

Une trajectoire est définie par une heure de départ (ou d'arrivée), un **chemin** et des **points d'attente** sur ce chemin.

Le caractère optimal de l'ensemble des trajectoires peut avoir différentes définitions, et sera considéré globalement comme le minimum d'une fonction de coût précisée ci-dessous.

Les trajectoires sont admissibles si d'une part les chemins empruntés sont conformes avec les contraintes opérationnelles de l'aéroport (décrites au 2.2), et si d'autre part les avions sont séparés (comme détaillé dans les parties suivantes).

2.1 Fonction de coût

La fonction de coût évaluant un ensemble de trajectoires admissibles peut faire intervenir différents facteurs, comme par exemple la durée et/ou la longueur de chaque trajectoire. Une attente peut être jugée plus ou moins coûteuse qu'une déviation. On peut aussi privilégier les départs soumis à un créneau de décollage en rendant leurs retards plus coûteux...

Dans la version actuelle, la fonction de coût f_c retenue prend en compte, pour chaque avion i ($1 \leq i \leq N$), son temps de roulage r_i et le délai d_i dû à sa déviation par rapport au plus court chemin :

$$f_c = \sum_{i=1}^N f_{c_i} \quad \text{avec} \quad f_{c_i} = r_i + d_i$$

Les déviations sont ainsi deux fois plus coûteuses que les attentes.

2.2 L'aéroport

Afin d'attribuer à chaque mouvement un ensemble de chemins admissibles, l'aéroport est modélisé par un graphe reliant ses parkings, ses taxiways et ses pistes.

Le coût pour aller d'un taxiway à un autre élément est le temps nécessaire pour parcourir ce taxiway. Ce temps est calculé en fonction du rayon de virage et des procédures spécifiques liées aux pistes (point d'arrêt obligatoire, sorties lentes, normales ou rapides...) et aux parkings (vitesses restreintes, attentes après push-backs...)

Certains taxiways sont des "sens-interdits" utilisables à titre plus ou moins exceptionnel. Leur coût est alors pondéré par un coefficient représentant la tolérance de passage par ce taxiway, conformément à la pratique opérationnelle. Les différents chemins admissibles pour un mouvement peuvent alors être obtenus avec des algorithmes classiques de parcours de graphe.

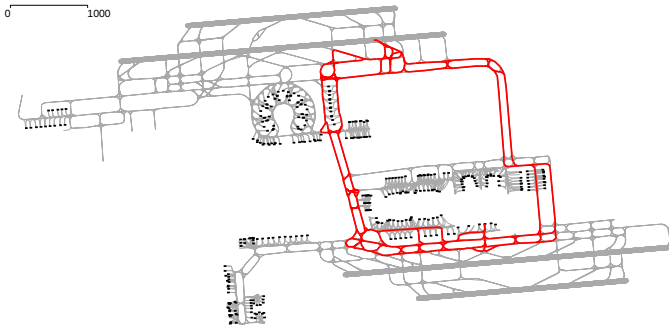


FIG. 1 – Graphe de Roissy

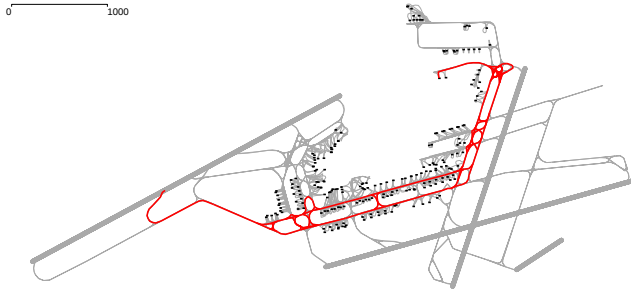


FIG. 2 – Graphe d'Orly

L'algorithme du Dijkstra [AMO93] donne tous les meilleurs chemins et les coûts minimums correspondants (i.e. les temps de parcours minimaux) pour aller d'un nœud donné à tous les autres. A partir du résultat du Dijkstra, il est possible de trouver les k_0 meilleurs chemins pour aller d'un point à un autre par Énumération Récursive [MJ96]. En répétant m fois ces deux algorithmes (Dijkstra et Énumération Récursive), et en augmentant à chaque fois le coût des nœuds empruntés par les chemins trouvés, on peut obtenir jusqu'à mk_0 chemins "suffisamment" différents les uns des autres.

Chaque mouvement se voit ainsi attribué k chemins possibles ($k \leq mk_0$) allant du parking à la piste demandée pour un départ, et de la piste d'atterrissage au parking pour une arrivée.

Les figures 1 et 2 représentent les graphes de Roissy et d'Orly et donnent un exemple de la trace des k meilleurs chemins allant d'un point à un autre.

2.3 Le trafic

2.3.1 Les plans de vol

Les mouvements sont décrits par leur plan de vol. Celui-ci fournit notamment l'heure de départ ou d'arrivée, le parking et la piste demandés, et le type avion. Le type avion permet d'appréhender la distance de décollage ou d'atterrissage, donc le(s) point(s) d'entrée ou de sortie de la piste envisageables. Il donne également la catégorie de turbulence de sillage (faible, moyenne ou élevée) utilisée pour

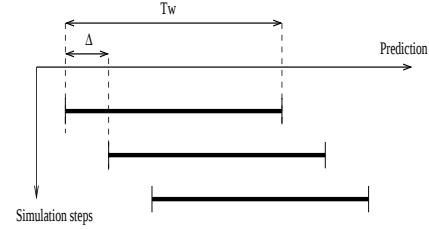


FIG. 3 – Fenêtres glissantes

les règles de séquençement de piste.

2.3.2 Séparation des avions

Les règles opérationnelles de séparation des avions sont les suivantes :

- la distance entre deux avions en mouvement doit toujours être supérieure à 60 mètres ;
- il ne peut y avoir qu'un seul décollage ou atterrissage à la fois sur chaque piste ;
- après un décollage, une séparation de 1, 2 ou 3 minutes (selon la catégorie de turbulence de sillage) est nécessaire avant le décollage ou l'atterrissage suivant ;
- un avion peut circuler dans l'aire de piste (90 mètres de part et d'autre de la piste) pendant un décollage ou un atterrissage à condition qu'il soit derrière l'avion concerné.

On dira qu'il y a conflit entre deux avions si leurs trajectoires prévues ne respectent pas ces règles.

2.3.3 Incertitudes sur la vitesse

Dans la simulation, les règles de séparation déterminent l'admissibilité des trajectoires, moyennant une incertitude relative sur la vitesse de déplacement des avions. Cette incertitude transforme la position d'un avion en un segment (et non un point). Les règles de séparation sont alors appliquées à tous les points du segment.

Deux particularités doivent cependant être prises en compte :

- Dans le cas où deux avions se suivent, le deuxième pilote est supposé adapter sa vitesse à celle du premier, et seules les positions principales (sans incertitudes) sont considérées.
- Lorsqu'un avion doit attendre, le point et l'heure de fin d'attente sont supposés être respectés, ce qui permet de réduire les positions d'incertitudes de l'avion.

3 Simulation

La simulation s'effectue par fenêtres glissantes sur toute une journée de trafic : à chaque pas de simulation t (toutes les Δ minutes), la prédiction de trafic est effectuée sur un horizon temporel T_w ($T_w > \Delta$). La situation est résolue sur

cet horizon, et cette résolution donne la nouvelle situation Δ minutes plus tard (cf. figure 3).

L'horizon de prédiction (et de résolution) étant limité, des "effets d'horizon" néfastes peuvent apparaître :

- Deux avions peuvent être amenés l'un en face de l'autre, créant un blocage définitif pour les situations futures.
- Un avion peut se retrouver bloqué dans l'aire de piste alors qu'une arrivée (non prévue par les résolutions précédentes) se présente...

Une analyse des positions en fin d'horizon ($t+T_w$) est donc nécessaire pour assurer la viabilité d'une résolution : toute paire d'avion en position de face à face en fin d'horizon sera considérée comme conflictuelle, et aucun avion ne sera admis sur la piste à l'heure $t+\Delta$ s'il ne l'a pas libérée avant la fin de l'horizon.

Ces nouvelles règles "de séparation" viennent s'ajouter aux précédentes et seront en vigueur pour toutes les méthodes de résolution présentées dans les parties suivantes.

3.1 BB : Méthode 1 contre n

Dans cette stratégie, les mouvements prévus sur l'horizon sont classés et résolus l'un après l'autre : le problème est donc réduit à celui d'un seul avion devant éviter des avions dont la trajectoire a été précédemment choisie. Ce problème peut être résolu par *Branch & Bound* [HT95] sur le graphe présenté ci-dessous.

3.1.1 Définition du graphe

Sur chacun des chemins possibles de l'avion, le problème peut être résolu rapidement par une recherche meilleure en premier dans l'arbre suivant :

- Un nœud de l'arbre est une position de l'avion à une heure donnée.
- La racine de l'arbre est la position initiale de l'avion au début de l'horizon de prédiction.
- Les feuilles (nœuds terminaux) sont constituées de feuilles solution : les positions non conflictuelles de l'avion en fin d'horizon ou à la fin du chemin ; et de feuilles non solution : toute position conflictuelle de l'avion.
- Chaque nœud (non terminal) a deux fils : le premier correspond au cas où l'avion avance sur son chemin, le deuxième au cas où l'avion attend. Si le premier aboutit à une feuille solution, c'est la meilleure solution depuis ce nœud.

On peut obtenir la meilleure solution pour l'avion en itérant cette recherche meilleur en premier sur chacun de ses chemins.

Cependant, il est possible de connaître pour chaque nœud de l'arbre relatif à l'un des chemins, le retard pris par l'avion jusqu'à ce nœud. Ceci permet de borner la recherche par le minimum des retards pris par l'avion sur les chemins déjà explorés : si le retard dépasse cette borne, la recherche sur ce chemin sera abandonnée.

L'algorithme devient ainsi un *Branch & Bound* avec stratégie d'exploration "meilleur en premier".

3.1.2 Classement des mouvements

Le classement des mouvements est bien sur un point crucial puisque les derniers avions sont extrêmement pénalisés, la plupart des classements pouvant même s'avérer sans solution.

L'analyse du problème permet cependant de dégager deux principales contraintes que doit respecter le classement :

- Les avions prévus à l'atterrissage ne peuvent pas être retardés avant d'être sortis de la piste, et doivent donc disposer de la piste libre : ils doivent être classés avant les décollages sur cette piste.
- Les avions au décollage faisant la queue pour la piste ne peuvent être classés que dans l'ordre donné par la queue, c'est-à-dire en fonction de la distance restante avant la piste.

En dehors de ces contraintes, un classement "réaliste" des mouvements peut être obtenu par comparaison des heures d'activation du plan de vol.

3.2 GA et GA+BB : algorithmes génétiques

Deux stratégies de résolution sont développées et utilisent des algorithmes génétiques classiques [Gol89, Mic92].

La première stratégie recherche un chemin et éventuellement une attente par mouvement. La deuxième recherche un chemin et un ordre de classement par mouvement, en utilisant un algorithme BB (cf. 3.1) pour développer et évaluer les trajectoires correspondantes.

3.2.1 Codage des données

Dans la première stratégie, la trajectoire d'un avion est décrite par 3 paramètres : le numéro n du chemin suivi, la position p d'attente, et l'heure t de fin d'attente (si p est atteinte après l'heure t , l'avion n'attend pas). Les éléments de la population (ou chromosomes) sont donc constitués de $3N$ variables pour un problème à N avions.

Dans la deuxième stratégie, la trajectoire d'un avion est donnée par 2 paramètres : le numéro n du chemin suivi et l'ordre de classement o de l'avion. Les chromosomes sont donc constitués de $2N$ variables.

3.2.2 Fonction d'adaptation

Dans les deux stratégies, les trajectoires décrites par un chromosome peuvent ne pas être admissibles. La fonction d'adaptation (ou fitness) F à maximiser, comprise entre 0 et 1, doit assurer qu'un chromosome représentant une solution admissible est toujours mieux évalué qu'un chromosome décrivant une situation avec conflits.

Pour cela, la fitness d'une solution admissible sera toujours supérieure à $\frac{1}{2}$ et la fitness des éléments contenant des conflits inférieure à $\frac{1}{2}$.

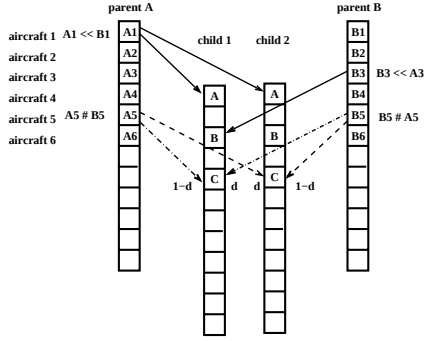


FIG. 4 – Croisement

Si n_c est le nombre de conflits et f_c la fonction de coût décrite au 2.1, la fitness sera donnée par :

$$\begin{aligned} \text{si } n_c > 0, \quad F &= \frac{1}{1+n_c} \\ \text{si } n_c &= 0, \quad F = \frac{1}{2} + \frac{1}{1+f_c} \end{aligned}$$

3.2.3 Croisement et Mutation

Le caractère partiellement séparable du problème de résolution de conflits [DA98, DAN96] permet d'obtenir des opérateurs de croisement et de mutation plus adaptés (avec une probabilité plus grande d'obtenir des fils plus adaptés que leurs parents).

Une fitness locale F_i est calculée pour chaque avion i , en fonction du nombre de conflits n_{c_i} impliquant cet avion et de sa participation f_{c_i} à la fonction de coût (cf. 2.1) :

$$\text{si } n_{c_i} > 0, F_i = K n_{c_i} \quad \text{sinon, } F_i = f_{c_i}$$

(K étant une constante telle que $K \gg f_c$)

Le croisement de deux parents A et B est représenté figure 4. Les fitness locales A_i et B_i de chaque avion i sont comparées : Si $A_i \ll B_i$, les fils hériteront des paramètres de A pour l'avion i . Si $B_i \ll A_i$, les fils hériteront des paramètres de B pour l'avion i . Si $A_i \equiv B_i$, les fils hériteront soit des paramètres A , soit de ceux de B , soit d'un croisement aléatoire entre les paramètres de A et B pour l'avion i .

Pour chaque candidat à la mutation, un avion ayant une des pires fitness locales est choisi, et ses paramètres modifiés aléatoirement.

Ces opérateurs de croisement et de mutation sont relativement déterministes tant que les éléments de population décrivent des situations non résolues. Ils deviennent plus aléatoires lorsque la population est constituée d'un grand nombre de solutions admissibles.

3.2.4 Sharing

Le problème est combinatoire et présente de nombreux optima locaux. Le sharing décrit dans Yin et Germa [YG93] permet d'empêcher la population de s'homogénéiser trop rapidement sur les optima locaux.

Il nécessite l'introduction d'une distance entre deux éléments A et B de la population, cette distance étant utilisée pour dissocier différents groupes (clusters) dans la population. La distance définie est la suivante :

$$D(A, B) = \frac{\sum_{i=1}^N |l_{A_i} - l_{B_i}|}{N}$$

l_{A_i} (resp l_{B_i}) étant la longueur du chemin de l'avion i dans le chromosome A (resp B).

3.2.5 Critère d'arrêt

Le critère d'arrêt de l'algorithme génétique est défini par un nombre maximal absolu de 50 générations, et un nombre maximal de 20 générations sans conflit (i.e. à meilleur élément décrivant une solution admissible).

3.2.6 Résolutions par Clusters

Pour diminuer autant que possible la complexité du problème, une fermeture transitive est appliquée aux paires d'avions en conflit dans la situation initiale (sans attente ni déviation). Les ensembles (ou clusters) d'avions ainsi formés sont résolus séparément.

A chaque fois que la résolution séparée de deux clusters provoque des conflits entre leurs avions, les deux clusters sont réunis et une nouvelle optimisation est effectuée.

4 Résultats

4.1 Simulations

Les simulations sont effectuées avec les plans de vol réels d'une journée de forte activité (18/06/1999) de Roissy et d'Orly. Les résultats obtenus pour Roissy et Orly étant similaires, seuls ceux de Roissy sont présentés et commentés dans cet article.

Les trois stratégies (BB, GA et GA+BB) sont comparées avec les paramètres suivants :

- Nombre de chemins par avion : $k = 30$
- Sens uniques : Oui
- Horizon : $T_w = 5mn$
- Pas de résolution : $\Delta = 2mn$
- Incertitude sur les vitesses : $\delta = 10\%$

4.2 Comparaison des stratégies

La figure 5 donne la corrélation entre le nombre de mouvements actifs et le délai généré pour chacune des stratégies de résolution.

Pour les situations à faible trafic (moins de 10 mouvements actifs), la meilleure méthode est GA : elle ne classe pas les avions et peut ainsi trouver des solutions plus proches de l'optimum global.

Lorsque le nombre de mouvements augmente, c'est la méthode GA+BB qui génère le moins de retards : ceci signifie

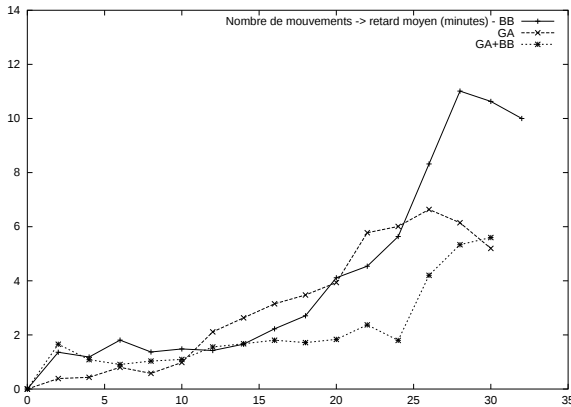


FIG. 5 – Retard moyen en fonction du nombre de mouvements

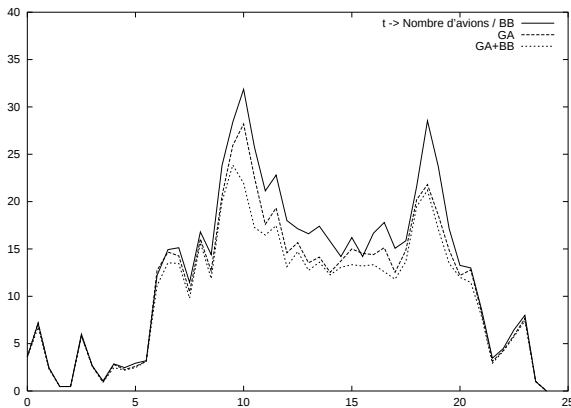


FIG. 6 – Nombre de mouvements actifs sur la journée

que le classement des avions est beaucoup moins pénalisant à forte densité de trafic, et qu'à partir d'un certain nombre de mouvements, le problème devient trop complexe pour que la méthode GA s'approche suffisamment de l'optimum global.

La méthode BB obtient de moins bons résultats que GA+BB, car elle ne remet pas suffisamment en cause le classement initial des mouvements.

Le classement s'avère donc une méthode efficace pour la résolution de conflits impliquant un grand nombre de mouvements, à condition de revoir régulièrement ce classement pour "l'adapter" aux nouvelles situations.

La figure 6 donne le nombre de mouvements actifs à chaque pas de simulation. La méthode GA+BB parvient souvent à des situations moins denses que celles des autres méthodes, et ce surtout pendant les périodes de pointe.

Ceci fait apparaître un phénomène important concernant les aéroports : une bonne résolution de la situation réduit d'une part les délais dans l'immédiat, et amène d'autre part des meilleures situations futures (impliquant moins de mouvements).

5 Conclusion

Ces premiers résultats font apparaître que le retard moyen du trafic au roulage sur des aéroports comme Roissy ou Orly peut-être réduit de plusieurs minutes en fonction de la stratégie d'optimisation employée. Ceci montre l'intérêt potentiel du développement d'outils d'aide au contrôle au sol (affectation de chemins et de points d'attente).

Les algorithmes génétiques semblent bien adaptés pour traiter ce problème combinatoire car leurs solutions s'approchent plus facilement de l'optimum global, tandis que des algorithmes déterministes (méthode 1 contre n) se bornent à des optima locaux créant des retards plus importants.

On peut noter également que le modèle a pu être amélioré facilement (nouvelles pistes à Roissy, incertitudes sur les vitesses, prise en compte des sens uniques...) sans modifier pour autant les algorithmes d'optimisation. Les simulations peuvent ainsi servir à évaluer la mise en place de nouvelles structures ou de nouvelles procédures sur un aéroport. Le critère d'optimisation peut encore être amélioré pour prendre en compte les mouvements ayant un créneau de départ impératif (suite à la régulation européenne du trafic), et les contraintes des secteurs d'approche.

Références

- [AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows, Theory, Algorithms and Applications*. Prentice Hall, 1993.
- [DA98] N. Durand and J. M. Alliot. Genetic crossover operator for partially separable functions. In *Proceedings of the third annual Genetic Programming Conference*, 1998.
- [DAN96] Nicolas Durand, Jean-Marc Alliot, and Joseph Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the Symposium on Applied Computing, Philadelphia*. ACM, 1996.
- [Gol89] D.E Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA Addison Wesley, 1989.
- [HT95] Reiner Horst and Hoang Tuy. *Global Optimization, Deterministic Approaches*. Springer, 1995.
- [Mic92] Z Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-verlag, 1992.
- [MJ96] Victor M. Jimenez and Andres Marzal. *Computing the K Shortest Paths : A New Algorithm and an Experimental Comparison*.
- [YG93] Xiaodong Yin and Noel Germain. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization.