RGSA – Revista de Gestão Social e Ambiental
ISSN: 1981-982X
Submission date: 1/3/2025
Acceptance date: 2/28/2025
DOI: https://doi.org/10.24857/rgsa.v19n4-032
Organization: Interinstitutional Scientific Committee
Chief Editor: Ana Carolina Messias de Souza Ferreira da Costa
Assessment: Double Blind Review pelo SEER/OJS

UM ESTUDO DE OTIMIZAÇÃO DOS MODELOS INDEXADOS PELO TEMPO E PELO ARCO-TEMPO E DA META-HEURÍSTICA IGA PARA O PROBLEMA DE SEQUENCIAMENTO DE AERONAVES

Lorrany Guilherme Santos[1]
Hélio Yochihiro Fuchigami[2]

RESUMO

Objetivo: Este estudo investiga estratégias de otimização para o Problema de Sequenciamento de Aeronaves, visando aumentar a eficiência das pistas e reduzir custos de combustível. Para isso, são analisadas abordagens de programação linear inteira mista (MILP) e meta-heurísticas.

Referencial Teórico: Baseia-se nos dados da Airports Council International (ACI) sobre o crescimento do tráfego aéreo e nos estudos de Yu *et al*., Mehta *et al*. e Farhadi *et al*., que destacam o impacto dos congestionamentos e a importância da otimização do sequenciamento de pousos e decolagens.

Método: A pesquisa compara modelos MILP e meta-heurísticas, utilizando solução inicial para melhorar a eficiência computacional. São analisadas 75 instâncias reais e comparados os solvers CPLEX e GUROBI em termos de soluções ótimas e gaps de otimalidade.

Resultados e Discussão: A inserção de soluções iniciais melhora a eficiência dos modelos MILP. CPLEX e GUROBI apresentam desempenhos distintos: um obtém melhores soluções ótimas, enquanto o outro reduz o gap de otimalidade.

Implicações da Pesquisa: Os achados oferecem insights para a gestão aeroportuária, auxiliando no planejamento operacional, redução de custos e aprimoramento de softwares de otimização.

Originalidade/Valor: Este estudo contribui ao comparar abordagens de otimização para o sequenciamento de aeronaves, com impacto na eficiência operacional, redução de custos e melhoria na gestão do tráfego aéreo.

Palavras-chave: Sequenciamento de Aeronaves, Gestão de Capacidade, Programação Linear Inteira Mista, Sequenciamento de Pistas.

AN OPTIMIZATION STUDY OF THE TIME-INDEX AND ARC-TIME-INDEXED MODELS AND THE IGA METAHEURISTIC FOR THE AIRCRAFT SEQUENCING PROBLEM

ABSTRACT

Objective: This study investigates optimization strategies for the Aircraft Sequencing Problem, aiming to increase runway efficiency and reduce fuel costs. To this end, mixed-integer linear programming (MILP) and metaheuristic approaches are analyzed.

Theoretical Framework: The study is based on data from the Airports Council International (ACI) on air traffic growth and the research of Yu *et al*., Mehta *et al*., and Farhadi *et al*., which highlight the impact of congestion and the importance of optimizing landing and takeoff sequencing.

Method: The research compares MILP models and metaheuristics, incorporating an initial solution to enhance computational efficiency. Seventy-five real-world instances are analyzed, and the solvers CPLEX and GUROBI are compared in terms of optimal solutions and optimality gaps.

[1] Universidade Federal de Goiás, Goiânia, Goiás, Brasil. E-mail: lorranyguilherme@ufg.br
[2] Universidade Federal de São Carlos, São Carlos, São Paulo, Brasil. E-mail: helio@dep.ufscar.br

Results and Discussion: The inclusion of initial solutions improves the efficiency of MILP models. CPLEX and GUROBI show distinct performances: one excels in obtaining optimal solutions, while the other achieves a lower optimality gap.

Research Implications: The findings provide insights for airport management, assisting in operational planning, cost reduction, and the improvement of optimization software.

Originality/Value: This study contributes by comparing optimization approaches for aircraft sequencing, impacting operational efficiency, cost reduction, and air traffic management improvement.

Keywords: Aircraft Scheduling, Capacity Management, Mixed-Integer Linear Programming, Runway Sequencing.

UN ESTUDIO DE OPTIMIZACIÓN DE LOS MODELOS INDEXADOS POR TIEMPO Y ARCOS-TIEMPO Y LA METAHEURÍSTICA IGA PARA EL PROBLEMA DE SECUENCIACIÓN DE AERONAVES

RESUMEN

Objetivo: Este estudio investiga estrategias de optimización para el Problema de Secuenciación de Aeronaves, con el objetivo de aumentar la eficiencia de las pistas y reducir los costos de combustible. Para ello, se analizan enfoques de programación lineal entera mixta (MILP) y metaheurísticas.

Marco Teórico: Se basa en los datos de Airports Council International (ACI) sobre el crecimiento del tráfico aéreo y en los estudios de Yu *et al*., Mehta *et al*. y Farhadi *et al*., que destacan el impacto de la congestión y la importancia de optimizar la secuenciación de aterrizajes y despegues.

Método: La investigación compara modelos MILP y metaheurísticas, incorporando una solución inicial para mejorar la eficiencia computacional. Se analizan 75 instancias reales y se comparan los solvers CPLEX y GUROBI en términos de soluciones óptimas y brechas de optimalidad.

Resultados y Discusión: La inclusión de soluciones iniciales mejora la eficiencia de los modelos MILP. CPLEX y GUROBI muestran desempeños distintos: uno obtiene mejores soluciones óptimas, mientras que el otro reduce la brecha de optimalidad.

Implicaciones de la investigación: Los hallazgos ofrecen información relevante para la gestión aeroportuaria, ayudando en la planificación operativa, la reducción de costos y la mejora del software de optimización.

Originalidad/Valor: Este estudio contribuye al comparar enfoques de optimización para la secuenciación de aeronaves, impactando en la eficiencia operativa, la reducción de costos y la mejora en la gestión del tráfico aéreo.

Palabras clave: Programación de Aeronaves, Gestión de Capacidad, Programación Lineal Entera Mixta, Secuenciación de Pistas.

1 INTRODUÇÃO

Segundo a Airports Council International (ACI, 2012), o tráfego aéreo mundial deve mais que dobrar até 2025, resultando em congestionamentos no solo e atrasos nos voos (Yu *et al.*, 2011). Esses problemas podem ser mitigados com o sequenciamento adequado das

operações de pousos e decolagens. Mehta *et al.* (2013) demonstram que otimizar esses sequenciamentos pode reduzir atrasos em até quatro horas por dia em grandes aeroportos. Farhadi *et al.* (2014) propuseram um modelo matemático e heurísticas para o sequenciamento de pousos e decolagens, que servem como base para esta pesquisa. No entanto, ainda há espaço para melhorar os tempos de execução, fator crucial para a agilidade nas decisões em ambientes dinâmicos.

Esta pesquisa implementa três formulações matemáticas e uma metaheurística em Julia, com testes computacionais no Aeroporto Internacional de Doha e dados de Farhadi *et al.* (2014). A eficiência do modelo é avaliada com base em tempo de CPU e gap de otimalidade.

O trabalho está estruturado da seguinte maneira: seção 2 apresenta a fundamentação teórica, seção 3 descreve o problema e as formulações propostas, seção 4 mostra os resultados, e seção 5 discute as contribuições e perspectivas futuras.

2 REFERENCIAL TEÓRICO

Os problemas de otimização na programação de aeronaves são classificados em problemas de pouso (ALP), de decolagem (ATP) e de sequenciamento (ASP), sendo este último o mais abrangente, pois reflete melhor a realidade dos controladores de tráfego aéreo (Artiouchine *et al.*, 2008; Boysen & Fliedner, 2011; Farhadi *et al.*, 2014; Bennell *et al.*, 2017; Riahi *et al.*, 2019). Apesar de a literatura conter mais estudos sobre ALP, os ASP apresentam potencial de pesquisa.

Os trabalhos sobre ASP visam reduzir emissões de carbono (Rosenthal & Eisenstein, 2016), minimizar ruído (Rodríguez-Díaz *et al.*, 2017), maximizar a capacidade aeroportuária e minimizar o consumo de combustível (D'ariano *et al.*, 2015; Samà *et al.*, 2013, 2014), reduzir atrasos (Faye, 2018; Pohl *et al.*, 2021; Salehipour, 2020; Zhou *et al.*, 2018), minimizar o makespan (Fernandes & Müller, 2019; Prakash *et al.*, 2018), reduzir a ociosidade de pistas (Jacquillat & Odoni, 2015; Ma *et al.*, 2019) e minimizar desvios de atrasos e antecipação (Bencheikh *et al.*, 2011; Briskorn & Stolletz, 2014; Sabar & Kendall, 2015).

Uma restrição essencial é o tempo de separação entre aeronaves para evitar turbulência, sendo abordada em Heidt *et al.* (2014), Fernandes & Müller (2019), Girish (2016), Riahi *et al.* (2019) e Rodríguez-Díaz *et al.* (2019). Esse tempo depende da classe de peso das aeronaves e pode ser assimétrico (Farhadi *et al.*, 2014). Alguns estudos, no entanto, desconsideram essa característica e assumem a separação entre pares consecutivos como suficiente (Bennell *et al.*, 2017; Faye, 2018; Lieder *et al.*, 2015; Rathinam *et al.*, 2009; Sabar & Kendall, 2015).

Entre os principais estudos sobre ASP, Farhadi *et al.* (2014) propuseram um modelo MILP e uma heurística, enquanto Riahi *et al.* (2019) exploraram soluções rápidas com heurísticas, incluindo um desvio padrão aceitável na distribuição.

3 DESCRIÇÃO DO PROBLEMA E FORMULAÇÕES

O problema tratado (ASP) pode ser definido genericamente da seguinte forma. Considere um conjunto de n aeronaves que realizam as operações de pouso e decolagem em m pistas. Não há exclusividade de pistas para as operações, ou seja, cada pista pode receber irrestritamente as aeronaves de pouso e decolagem em qualquer ordem, desde que respeitem o tempo mínimo de separação (*sij*) entre elas (sendo *i* e *j* índice de aeronaves). A matriz de separação é estabelecida ao considerar as três diferentes classes de aeronaves (pesada, intermediária e leve) e as operações possíveis (pouso e decolagem). Além disso, cada aeronave apresenta um custo de consumo de combustível (*wi*), de acordo com as classes citadas anteriormente. Esse parâmetro também é responsável por estabelecer a prioridade dada a cada uma das aeronaves, sendo que aquelas que realizam operação de pouso têm prioridade com relação às de decolagens, devido a restrições de combustível e dificuldade no controle aéreo.

A proposta dos modelos é elaborar uma programação que minimize o custo de consumo de combustível. Para a solução do problema, foram comparadas três formulações matemáticas: a primeira baseada em variáveis de precedência e alocação, proposta por Farhadi *et al.* (2014); a segunda, com variáveis indexadas no tempo; e a última, com variáveis indexadas por arcos e tempo. Na sequência, foi implementada a metaheurística *Iterated Greedy Algorithm* (IGA).

Para fins comparativos, foi implementado o modelo mais citado na literatura para o problema de sequenciamento de aeronaves, desenvolvido por Beasley *et al.* (2000). No entanto, a formulação MILP de Beasley *et al.* (2000) possui uma relaxação linear fraca, incapaz de solucionar os exemplares do *benchmark* de Farhadi *et al.* (2014).

Por fim, comparou-se a eficiência computacional das formulações ao inserir a sequência FIFO como solução inicial do modelo. Os parâmetros utilizados nos modelos passaram por testes preliminares, com o objetivo de reduzir a relaxação da formulação. Os parâmetros, as variáveis, os modelos e a metaheurística proposta são apresentados a seguir.

3.1 MODELO BASEADO EM VARIÁVEIS DE PRECEDÊNCIA E ALOCAÇÃO

- Índices:

i, j: índices das aeronaves $i \in I = \{1, \ldots, n\}$; $j \in J = \{1, \ldots, n\}$

k: índice da pista $k \in K = \{1, \ldots, m\}$

- Parâmetros:

w_i : custo de consumo de combustível da aeronave i

r_i: instante de liberação da aeronave

d_i : tempo limite para alocação da aeronave i

s_{ij} : tempo mínimo de separação entre as aeronaves i e j

- Variáveis de decisão:

$$X_{ik} \begin{cases} 1\ se\ \text{aeronave}\ i\ \text{está alocada na pista}\ k \\ 0, \text{caso contrario} \end{cases}$$

$$Y_{ij} \begin{cases} 1 \text{ se a aeronave } i \text{ está alocada antes da aeronave } j \\ \qquad\qquad \text{na mesma pista} \\ 0, \text{caso contrario} \end{cases}$$

Onde:

$a_i \geq 0$, instante que a aeronave *i* acessa a pista.

- Modelo matemático:

$$\text{Min Z} = \sum_{j=1}^{n} w_j(a_j - r_j) \qquad (1)$$

$$\sum_{k=1}^{m} X_{ik} = 1, \ \forall i \in I \qquad (2)$$

$$r_i \leq a_i, \forall i \in I \qquad (3)$$

$$a_i \leq d_i, \forall i \in I \qquad (4)$$

$$a_j \geq a_i + s_{ij} - B(1 - Y_{ij}), \ \forall i \in I\ ;\ \forall j \in J;\ i \neq j \qquad (5)$$

$Y_{ij}+Y_{ji} \geq X_{ik}+X_{jk}- 1,\ \forall i \in I\ ;\ \forall j \in J\ ;\ \forall i \in I\ ;\ \forall k \in K\ ;\ i < j$ (6)

$a_i \geq 0, \forall i \in I$ (7)

$X_{ik} \in \{0,1\},\ \forall i \in I\ ;\ \forall k \in K$ (8)

$Y_{ij} \in \{0,1\}, \forall i \in I\ ;\ \forall j \in J$ (9)

A função objetivo (1) minimiza a soma dos custos de atraso, ponderada pelo consumo de combustível gasto por cada aeronave. As restrições (2) garantem que cada aeronave deve ser alocada a apenas uma pista. As equações (3) e (4) asseguram os intervalos limite para cada operação, sendo o instante em que a aeronave acessa a pista deve ser sempre maior que o instante de liberação e menor que o tempo limite. As restrições (5) introduzem um tempo mínimo de separação entre cada par de aeronaves, seja consecutivo ou não, que é atribuído à mesma pista. As restrições (6) garantem que a precedência entre qualquer par de aeronaves deve ser estabelecida se elas forem atribuídas à mesma pista. Finalmente, os conjuntos de restrições (7), (8) e (9) estabelecem o domínio das variáveis de decisão. O valor de B (geralmente referenciado na literatura como big M) é considerado como, B = d_i - r_i + s_{ij}.

3.2 MODELO COM VARIÁVEIS INDEXADAS NO TEMPO

Levando em consideração a notação para a primeira formulação matemática, apresentamos a notação adicional para o segundo modelo.

- Índice:

t: instante de tempo $t \in T = \{1, \ldots, \max(d)\}$

- Variáveis de decisão:

$$X_{ijt} \begin{cases} 1, \textit{se aeronave } i \textit{ está alocada imediamente} \\ \quad \textit{antes da aeronave } j \textit{ na mesma pista}, \\ \quad \textit{com } j \textit{ acessando no instante } t \\ \quad \textit{e } i \textit{ acessando no intante } t - s_{ij}, \\ 0, \textit{caso contrario} \end{cases}$$

Onde:

$a_i \geq 0$, momento que a aeronave i acessa a pista

- Modelo matemático:

$$\sum_{k=1}^{m} \sum_{t=1}^{T} X_{ikt} = 1, \forall i \in I \quad (10)$$

$$\sum_{i=1}^{n} X_{ikt} \leq 1, \forall k \in K; \forall t \in T \quad (11)$$

$$a_i = \sum_{k=1}^{m} \sum_{t=1}^{T} t(X_{ikt}), \ \forall i \in I \quad (12)$$

$$Y_{ij} + Y_{ji} \geq X_{ikt} + X_{jkt} - 1, \ \forall i \in I ; \forall j \in J; \ \forall k \in K ; i<j \quad (13)$$

$$X_{ikt} \in \{0,1\}, \forall i \in I ; \forall k \in K; \forall t \in T; i<j \quad (14)$$

O segundo modelo considera a função objetivo (1), as restrições (3), (4), (5), (7) e (9). Além disso, as expressões (10) e (11) garantem que cada aeronave deve ser alocada a exatamente uma pista e em um instante de tempo. a a exatamente uma pista e em um instante de tempo. As equações (12) realizam o cálculo para o instante de acesso de cada aeronave. As restrições (13) realizam o acoplamento entre as variáveis e estabelecem que a precedência entre qualquer par de aeronaves atribuídos à mesma pista. Por fim, os conjuntos de restrições (14) estabelecem o domínio das variáveis de decisão.

3.3 MODELO COM VARIÁVEIS INDEXADAS POR ARCOS E TEMPO

Ainda considerando a notação para a primeira formulação matemática, apresentamos a notação adicional para o terceiro modelo.

- Índice:

t: instante de tempo $t \in H = \{1, \ldots, \max(d) + \max(s)\}$

- Variáveis de decisão:

$$X_{ijt} \begin{cases} 1, se\ aeronave\ \ i\ está\ alocada\ imediamente \\ \quad antes\ da\ aeronave\ j\ na\ mesma\ pista\ , \\ \quad com\ \ j\ acessando\ \ no\ instante\ t \\ \quad e\ i\ acessando\ no\ intante\ t - s_{ij}, \\ \quad 0, caso\ contrario \end{cases}$$

Onde:

$a_i \geq 0$, momento que a aeronave *i* acessa a pista

- Modelo matemático:

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} \sum_{t=1}^{T-s_{ij}} X_{ijt} = 1 \ \forall j \in J \qquad (15)$$

$$\sum_{j=1}^{n} X_{0j1} = m \qquad (16)$$

$$\sum_{\substack{j=1 \\ i \neq j}}^{n} \sum_{t=1}^{T} X_{j0t} = m \qquad (17)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} X_{ijt} - \sum_{\substack{i=0 \\ i \neq j}}^{n} X_{ji(t+s_{ij})} = 0 \ \forall j \in J; \ \forall t \in T \qquad (18)$$

$$a_j \geq a_i + s_{ij} - B\,(1 - \sum_{t=1}^{T} X_{ijt}) \ \forall i \in I; \ \forall j \in J; \ i \neq j \qquad (19)$$

$$a_i = \sum_{t=1}^{T} \sum_{j=0}^{n} t\ (X_{ijt}) \ \ \forall i \in I \qquad (20)$$

$$X_{ijt} \in \{0,1\}, \ \forall i \in I\ ; \ \forall j \in J; \ \forall t \in T \qquad (21)$$

O terceiro modelo considera aeronaves fictícias (índice zero) e representam os instantes que as pistas ficam ociosas. A função objetivo (1), as restrições (2), (3) e (7) são equivalentes aos modelos anteriores. Além disso, as restrições (15) garantem que cada aeronave deve acessar apenas uma pista, em um único instante de tempo. As equações (16) e (17) delimitam a capacidade de pistas. As equações (16) e (17) delimitam a capacidade de pistas. Os grupos (18) e (19) garantem a não sobreposição de alocações e introduzem um tempo mínimo de separação

entre qualquer par de aeronaves, seja consecutivo ou não, que são atribuídos à mesma pista. As restrições (20) realizam o cálculo para o momento de acesso de cada aeronave. E os conjuntos de restrições (21) estabelecem o domínio das variáveis de decisão.

3.4 NÚMERO DE VARIÁVEIS E RESTRIÇÕES

O tamanho dos problemas em relação aos modelos 1, 2 e 3 em termos do número e tipo das variáveis e das restrições, é apresentado na Tabela 1.

Tabela 1

Tamanho dos modelos em termos do número de variáveis e de restrições

	Variáveis	Restrições
Modelo 1	$n(1+m+n)$	$n(3+m+nm) - \frac{n^2-n}{2}$
Modelo 2	$n(n+m+m(T+1)+1)$	$n^2(1+m)+3n+ (\sum_{i=1}^{n} d_i - r_i) * m - \frac{(n^2-n)}{2}$
Modelo 3	$n^2H + 2n$	$n^2 + n(3+T)+2$

Desta forma, o um problema-teste com n= 5 aeronaves e m = 2 pistas, uma variação entre o tempo limite máximo e o instante de liberação de 600 unidades, T= 950 e H= 1.146 terá 40 variáveis e 65 restrições para o modelo1, 9.550 variáveis e 6.080 restrições para o modelo 2 e 28.660 variáveis e 4.792 restrições para o modelo 3.

3.5 METAHEURISTICA IGA

A metaheurística *Iterated Greedy Algorithm* (IGA) foi inicialmente desenvolvida por Ruiz e Stutzie (2007) para o ambiente *flow shop*. A solução inicial foi baseada em uma adaptação da heurística construtiva NEH de Nawaz *et al*. (1983), na qual a solução inicial do problema foi a sequência FIFO. Posteriormente, essa solução passou pelas etapas de destruição, construção e pesquisa local. Os parâmetros utilizados foram calibrados para o ASP utilizando 30 arquivos de testes da base de dados de Farhadi *et al*. (2014).

4 EXPERIMENTAÇÃO COMPUTACIONAL E RESULTADOS

4.1 BENCHMARK

Para efeito de análise de desempenho dos modelos propostos, avaliaram-se na experimentação computacional 75 problemas-testes também utilizados em Farhadi *et al.* (2014). Os exemplares são formados pela combinação do número de aeronaves (n = {15,20,25,50}), número de pistas (m = {2,3,4,5}) e 5 réplicas por combinação. Além disso, constam nos exemplares as matrizes com os instantes de liberação, tempo limite máximo para realizar as operações de pouso ou decolagem, tempo mínimo de separação entre cada par de aeronaves e custo de consumo de combustível que depende da classe (pesada, intermediária e leve) e do tipo de operação (pouso ou decolagem).

4.2 IMPLEMENTAÇÃO COMPUTACIONAL

Os modelos foram implementados em linguagem Julia, escolhida para este trabalho por ter um código fácil, moderno, com IDE gratuito e eficiente para problemas de MILP. Por meio desta linguagem de modelagem, é possível realizar a execução sequencial de vários exemplares. Primeiramente, foi realizada a comparação de desempenho dos resolvedores comerciais de otimização CPLEX studio 12.9 64 bits e GUROBI 8.0.1 64 bits. Na comparação entre esses resolvedores percebeu-se que o CPLEX apresentava o melhor desempenho para exemplares que atingiam o tempo limite, e como grande parte dos exemplares se encaixam nesse caso, o CPLEX foi escolhido para finalizar toda a experimentação. Todos os testes foram conduzidos em um processador Intel®Core™i7 com 2.4 GHz, 8.0 GB de memória RAM e sistema operacional Windows 10. Os 75 problemas-testes foram resolvidos até um tempo máximo de 7200 segundos pelo algoritmo branch-and-cut incluído no CPLEX e no GUROBI. As medidas analisadas para cada um dos modelos foram o tempo médio de computação (CPU) em segundos e o gap de otimalidade.

4.3 DISCUSSÃO DOS RESULTADOS

Na Tabela 2 são apresentados os resultados numéricos da comparação entre o CPLEX e GUROBI. Para essa análise, foram utilizados os dados de saída referentes a implementação do modelo baseado em variáveis de precedência e alocação, descrito na seção III.A. Para os

exemplares solucionados na otimalidade, foram comparados os tempos de CPU. Já para aqueles que não alcançaram a otimalidade dentro do tempo limite, foram comparados os gaps de otimalidade. Os valores foram confrontados por meio do desvio percentual relativo.

Tabela 2

Comparativo da eficiência computacional dos resolvedores de otimização

		CPLEX	GUROBI
SOLUÇÕES ÓTIMAS	TEMPO DE CPU (s)	1943,06	**952,11**
	DESVIO_CPU (%)	129	**35**
	Nº de exemplares	24	23
TEMPO LIMITE	GAP	**0,44**	0,60
	DESVIO_GAP (%)	**4**	102
	Nº de exemplares	51	52

Conforme os dados da Tabela 2, o CPLEX apresenta menor eficiência computacional para solução dos exemplares que chegaram à solução ótima quando comparado com o GUROBI. Enquanto o primeiro utiliza em média 1943 segundos, o último utiliza em média 952 segundos para encontrar a melhor resposta. O desvio relativo médio do CPLEX é 129% e do GUROBI 35%. Ainda na Tabela 5 é possível perceber o comportamento inverso para exemplares que alcançaram o tempo limite: o CPLEX alcançou um gap de otimalidade médio de 0,44 e o GUROBI de 0,60. O desvio relativo médio entre eles evidencia a melhor qualidade de solução obtida pelo CPLEX nos casos citados. As Fig. 1 e 2 exibem a comparação de perfis de desempenho, proposta por Dolan e Moré (2002).

Figura 1

Perfis de desempenho do gap do CPLEX e GUROBI

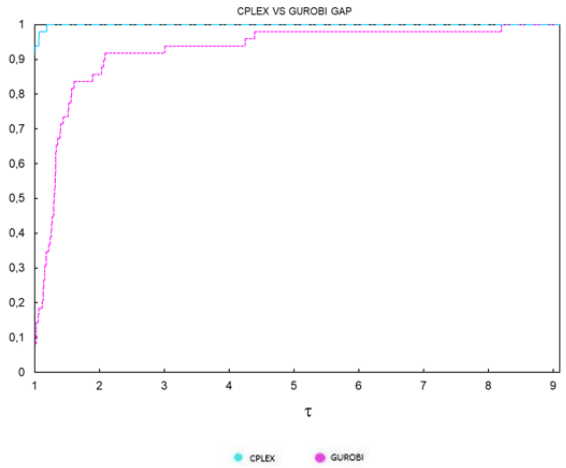
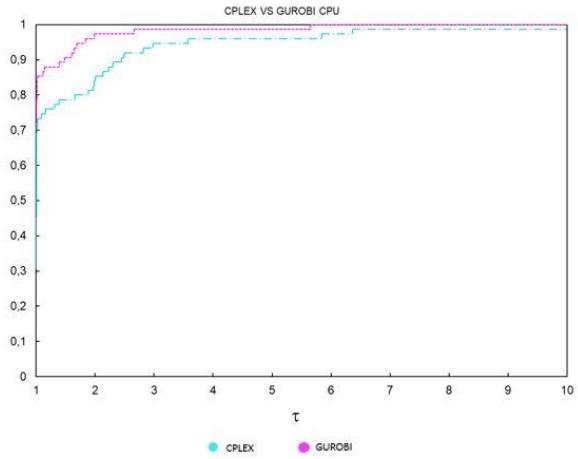


Figura 2

Comparação de perfis de desempenho entre o tempo de CPU do CPLEX e GUROBI



Para melhor compreensão dessa análise é validado dizer que os valores do eixo y indicam a probabilidade de que o método comparado, tenha o melhor desempenho dentro de determinado fator de interesse (eixo x). Portanto, os valores do eixo x=1 indicam o método mais eficiente.

Através dessas das Fig. 1 e 2 é possível perceber que para qualquer fator de interesse (τ) o CPLEX apresenta a maior fração de exemplares com um melhor gap e o GUROBI a maior fração de exemplares com o menor tempo de CPU, respectivamente, confirmando as análises anteriormente descritas.

Como pode ser visto na Fig. 1, para τ =1, o CPLEX é melhor em 92% dos exemplares e o GUROBI consequentemente em 8% dos exemplares. Analogamente, conforme a Fig. 2, o eixo x = 1 aponta que 72% dos exemplares o GUROBI resolve o modelo com menor tempo de CPU. Nas Fig. 3 e 4 são ilustrados o comportamento do gap de otimalidade com o decorrer do tempo de resolução.

Figura 3

Evolução do gap de otimalidade em exemplares que chegaram à solução ótima

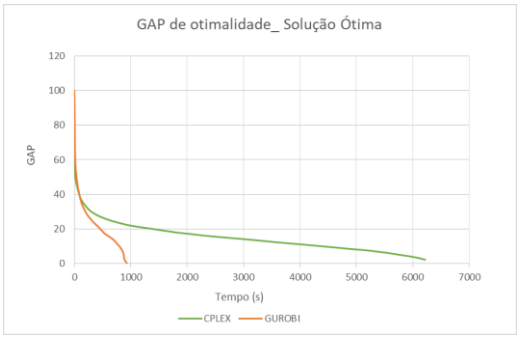
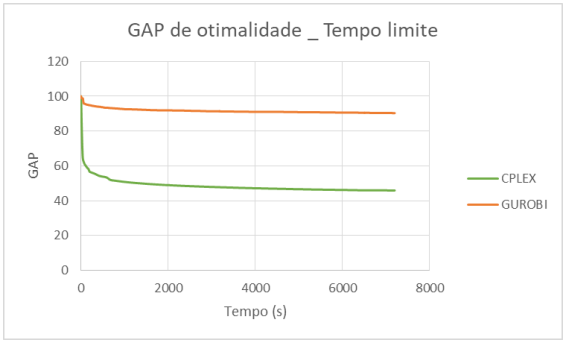


Figura 4

Evolução do GAP de otimalidade em exemplares que alcançaram o tempo limite



Para construção da Fig. 3 é considerado um exemplar que alcançou a solução ótima e a Fig. 4 foi construída com um exemplar que executou até tempo limite. Nas Fig. 3 e 4 são apresentados os valores para os desvios médios relativo. Mais uma vez, para problemas com solução ótima, o GUROBI demonstra superioridade ao necessitar de menos tempo para concluir a execução (atingir o gap zero). Enquanto para os problemas que alcançaram o tempo limite, o CPLEX demonstra superioridade ao conseguir um rápido decaimento no gap logo no início da execução e manter essa vantagem durante todo o tempo estipulado.

Na Tabela 3 são demonstrados a comparação das formulações através das relaxações lineares.

Tabela 3

Comparação da eficiência das relaxações lineares

	Objetivo	Tempo CPU(s)
MIP1	0	0,19
MIP2	0	6,94
MIP3	630	337,28

Nota-se que as formulações MIP1 e MIP2 fornecem valores ótimos iguais a zero em todos os 75 exemplares, e além disso o tempo necessário para a execução é de pouquíssimos segundos (0,19 segundos em média para o primeiro e 6,94 segundos em média para o segundo). Já a formulação indexada por arcos e tempo (MIP3) resolve os 55 primeiros exemplares na otimalidade e os demais chegam ao tempo máximo estipulado de 7.200 segundos sem encontrar a solução ótima. Apesar de demandar um tempo maior, essa última formulação apresenta o melhor limite inferior para o problema inteiro. Na Tabela 4 são comparados os desempenhos computacionais das três formulações matemáticas e da respectiva execução com a regra FIFO como solução inicial do modelo.

Tabela 4

Comparação entre as formulações matemáticas

	GAP	CPU(s)	Soluções factíveis	Sem solução factível	nº de soluções ótimas	% de soluções ótimas
MIP 1	44%	2025,97	75	0	24	32%
MIP 1_FIFO	43%	1362,09	75	0	22	29%
MIP 2	81%	661,02	75	0	1	1%
MIP 2_FIFO	82%	496,14	75	0	1	1%
MIP 3	28%	5226,21	15	60	1	1%
MIP 3_FIFO	26%	-	15	60	0	0%

A inclusão da regra FIFO como solução inicial dos problemas causa uma melhoria de 1% na primeira formulação (MIP1) e uma melhoria de 33% no tempo de CPU (variação de 2025,97s para 1362,09s). Na segunda formulação (MIP2) não existiu melhoria estatística no gap, no entanto o tempo de CPU apresentou uma redução de 25% em seu valor (variação de 661,02s para 496,14s). Por fim, na terceira formulação com variáveis indexadas por arcos e tempo (MIP3 e MIP3_FIFO) apenas 15 exemplares obtiveram soluções factíveis devido à maior complexidade dessa formulação. Para esse último grupo, a tabela indica uma melhoria de 2% no gap ao inserir a sequência FIFO como solução inicial. A análise do tempo de CPU para a terceira formulação fica comprometida, pois o MIP3_FIFO não encontrou nenhuma solução ótima e portanto, não expõe o tempo de CPU.

De forma geral, todas as formulações obtiveram uma melhoria individual ao inserir a sequência FIFO como solução inicial do problema. Mesmo a segunda formulação desenvolvida pelos autores (que apresenta variação positiva no gap) ao inserir a sequência FIFO demonstra evolução na eficiência computacional ao analisar o tempo de CPU. A análise de perfis de desempenho que comparara a eficiência da inclusão da solução inicial FIFO no modelo foi aplicada apenas no MIP1 e pode ser observada nas Fig. 5 e 6.

Figura 5

Comparação de perfis de desempenho entre o GAP do MIP1 e MIP1_FIFO

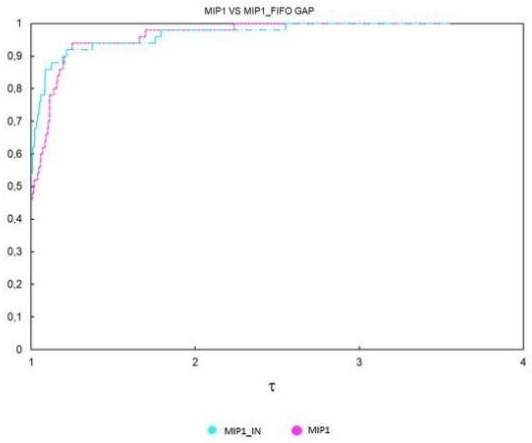
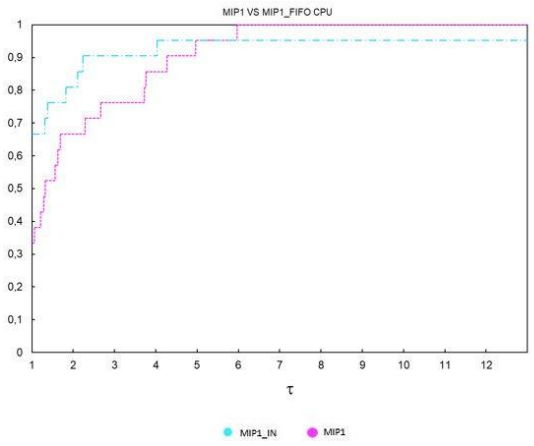


Figura 6

Comparação de perfis de desempenho entre o tempo de CPU do MIP1 e MIP1_FIFO

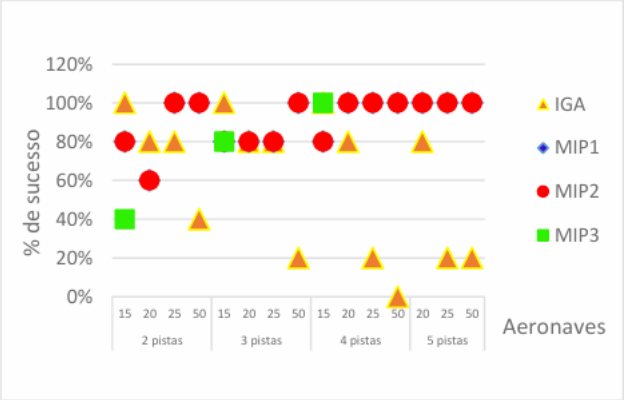


Conforme a Fig. 5 o modelo MIP1_FIFO é melhor em 55% dos exemplares. Da mesma forma, a Fig. 6 ilustra que o tempo de CPU aponta que o MIP1_FIFO é mais eficiente em 65% dos exemplares e o MIP1 consequentemente em 35% dos exemplares.

Com relação a metaheurística IGA, a comparação de eficácia da função objetivo mostrou que esse algoritmo alcançou uma taxa de sucesso geral de 60% em uma média de 14,76 segundos. O gap da solução com relação a melhor solução encontrada foi de 2% e ao considerar apenas aquelas com desvio positivo esse resultado é de 4%. A Fig. 7 ilustrada a taxa de sucesso de todas as soluções observando a distribuição de pistas e aeronaves.

Figura 7

Comparação de perfis de desempenho entre o tempo de CPU do MIP1 e MIP1_FIFO



Conforme a Fig. 7 a metaheurística IGA apresenta taxa de sucesso de 100% nos exemplares com 15 aeronaves e 2/3/4 pistas. Além disso, a taxa de sucesso para a IGA é superior ou iguais aos dos modelos nos exemplares com 2 pistas e 20 aeronaves, 3 pistas e 20/25 aeronaves.

5 CONCLUSÃO

Este trabalho aborda um problema relevante tanto academicamente quanto comercialmente, relacionado aos atrasos em pousos e decolagens nos aeroportos, causados por problemas de sequenciamento nas pistas. O modelo proposto visa representar o cenário real, considerando restrições de separação entre aeronaves e variáveis como porte, número de tripulantes e combustível.

Os resultados indicaram que o solver CPLEX foi superior em problemas que atingiram o tempo máximo de execução, enquanto o GUROBI foi mais eficaz nos casos de solução ótima. Além disso, foram apresentadas duas novas formulações, que, apesar de apresentarem maior gap e tempos de CPU devido ao tamanho do problema, proporcionaram bons limites inferiores e orientaram algoritmos de aproximação.

O estudo também avaliou o impacto de soluções iniciais, utilizando a regra FIFO, o que melhorou a eficiência das soluções. Outra contribuição importante foi o desenvolvimento da metaheurística IGA, que obteve uma taxa de sucesso de 60% nos exemplares avaliados.

REFERÊNCIAS

ACI. (2012). Airports Council International. http://www.aci.aero/

Artiouchine, K., Baptiste, P., & Dürr, C. (2008). Runway sequencing with holding patterns. European Journal of Operational Research, 189(3), 1254-1266.

Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., & Abramson, D. (2000). Scheduling aircraft landings—The static case. Transportation Science, 34(2), 180-197.

Bencheikh, G., Boukachour, J., & Alaoui, A. E. H. (2011). Improved ant colony algorithm to solve the aircraft-landing problem. International Journal of Computer Theory and Engineering, 3(2), 224.

Bennell, J. A., Mesgarpour, M., & Potts, C. N. (2017). Dynamic scheduling of aircraft landings. European Journal of Operational Research, 258(1), 315-327.

Boysen, N., & Fliedner, M. (2011). Scheduling aircraft landings to balance workload of ground staff. Computers & Industrial Engineering, 60(2), 206-217.

Briskorn, D., & Stolletz, R. (2014). Aircraft landing problems with aircraft classes. Journal of Scheduling, 17(1), 31-45.

D'Ariano, A., Pacciarelli, D., Pistelli, M., & Pranzo, M. (2015). Real-time scheduling of aircraft arrivals and departures in a terminal maneuvering area. Networks, 65(3), 212-227.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. Mathematical Programming, 91, 201-213.

Farhadi, F., Ghoniem, A., & Al-Salem, M. (2014). Runway capacity management—An empirical study with application to Doha International Airport. Transportation Research Part E: Logistics and Transportation Review, 68, 53-63.

Faye, A. (2018). A quadratic time algorithm for computing the optimal landing times of a fixed sequence of planes. European Journal of Operational Research, 270(3), 1148-1157.

Fernandes, H. F., & Müller, C. (2019). Optimization of the waiting time and makespan in aircraft departures: A real-time non-iterative sequencing model. Journal of Air Transport Management, 79, 101686.

Girish, B. S. (2016). An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft-landing problem. Applied Soft Computing, 44, 200-221.

Heidt, A., Helmke, H., Liers, F., & Martin, A. (2014). Robust runway scheduling using a time-indexed model. Fourth SESAR Innovation Days, 1-8.

Jacquillat, A., & Odoni, A. R. (2015). An integrated scheduling and operations approach to airport congestion mitigation. Operations Research, 63(6), 1390-1410.

Lieder, A., Briskorn, D., & Stolletz, R. (2015). A dynamic programming approach for the aircraft-landing problem with aircraft classes. European Journal of Operational Research, 243(1), 61-69.

Ma, J., Delahaye, D., Sbihi, M., Scala, P., & Mota, M. A. M. (2019). Integrated optimization of terminal maneuvering area and airport at the macroscopic level. Transportation Research Part C: Emerging Technologies, 98, 338-357.

Mehta, V., Reynolds, T., Ishutkina, M., Joachim, D., Glina, Y., Troxel, S., & Evans, J. (2013). Airport surface traffic management decision support: Perspectives based on tower flight data manager prototype. The National Technical Information Service, product code PB2013108463.

Nawaz, M., Enscore, J. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega, 11(1), 91-95.

Pohl, M., Kolisch, R., & Schiffer, M. (2021). Runway scheduling during winter operations. Omega, 102, 102325.

Prakash, R., Piplani, R., & Desai, J. (2018). An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput. Transportation Research Part C: Emerging Technologies, 95, 570-581.

Rathinam, S., Wood, Z., Sridhar, B., & Jung, Y. (2009). A generalized dynamic programming approach for a departure scheduling problem. AIAA Guidance, Navigation, and Control Conference, 6250.

Riahi, V., Newton, M. H., Polash, M. M. A., Su, K., & Sattar, A. (2019). Constraint guided search for aircraft sequencing. Expert Systems with Applications, 118, 440-458.

Rodríguez-Díaz, A., Adenso-Díaz, B., & González-Torre, P. L. (2017). Minimizing deviation from scheduled times in a single mixed-operation runway. Computers & Operations Research, 78, 193-202.

Rodríguez-Díaz, A., Adenso-Díaz, B., & González-Torre, P. L. (2019). Improving aircraft approach operations taking into account noise and fuel consumption. Journal of Air Transport Management, 77, 46-56.

Rosenthal, E. C., & Eisenstein, E. M. (2016). A rescheduling and cost allocation mechanism for delayed arrivals. Computers & Operations Research, 66, 20-28.

Sabar, N. R., & Kendall, G. (2015). An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem. Omega, 56, 88-98.

Salehipour, A. (2020). An algorithm for single-and multiple-runway aircraft landing problem. Mathematics and Computers in Simulation, 175, 179-191.

Samà, M., D'Ariano, A., D'Ariano, P., & Pacciarelli, D. (2014). Comparing centralized and rolling horizon approaches for optimal aircraft traffic control in terminal areas. Transportation Research Record, 2449(1), 45-52.

Samà, M., D'Ariano, A., & Pacciarelli, D. (2013). Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports. Procedia-Social and Behavioral Sciences, 80, 531-552.

Yu, S. P., Cao, X. B., & Zhang, J. (2011). A real-time schedule method for aircraft landing scheduling problem based on cellular automation. Applied Soft Computing, 11(4), 3485-3493.

Zhou, G., Wang, R., & Zhou, Y. (2018). Flower pollination algorithm with runway balance strategy for the aircraft landing-scheduling problem. Cluster Computing, 21(3), 1543-1560.