
METODOLOGIAS PARA RESOLUÇÃO DE CONFLITOS ENTRE AERONAVES BASEADAS EM *GAME TREE*

Lucas B. Monteiro*, Cristiano P. Garcia, Geraldo P. Rocha Filho, Li Weigang
Universidade de Brasília (UnB), Departamento de Ciências da Computação (CIC)

* **Corresponding author e-mail adress:** lucasbmonteiro@gmail.com

PAPER ID: SIT204

RESUMO

Nos últimos anos, houve um progresso significativo no desenvolvimento de modelos relacionados ao gerenciamento de tráfego aéreo (ATM, do inglês *Air Traffic Management*). Considerando vários tipos de eventos envolvidos em um voo (precisão dos sensores, incertezas climáticas e meteorológicas, mudanças nas trajetórias das aeronaves, etc.), muitos esforços ainda precisam ser empregados no desenvolvimento de métodos que possam ser aplicados ao ambiente real com máxima precisão. Todos esses métodos precisam lidar com uma grande quantidade de dados. Nesse contexto, alguns pesquisadores têm buscado diferentes metodologias de inteligência artificial para melhorar a tomada de decisão no transporte aéreo. Esta pesquisa tem como objetivo propor dois métodos diferentes de resolução de conflitos baseados em *game tree*: Busca em Profundidade (DFS, do inglês Depth-First Search) e Alfa-Beta. Serão apresentados os conceitos básicos desses métodos, bem como a modelagem necessária para sua aplicação. Por fim, os resultados das simulações envolvendo diferentes combinações de aeronaves serão analisados, como forma de indicar a possibilidade de aplicação de cada método estudado.

Palavras-chave: Gerenciamento de Tráfego Aéreo, Resolução de Conflitos, Depth-First Search, Alpha-Beta.

1. INTRODUÇÃO

Atualmente, em diversas áreas de aplicação é comum a coleta de uma grande quantidade de dados, que acabam por influenciar sobremaneira a tomada de decisão. Esse novo cenário representa uma mudança de paradigma das decisões baseadas em suposições ou modelos manuais para análises baseadas em modelos matemáticos orientados por dados. O tema *big data* exerce impacto nos mais variados aspectos da sociedade, tais como varejo, indústria, ciências, serviços financeiros e, especialmente, o transporte aéreo (Kieckbusch et al., 2019).

O transporte aéreo global, que desempenha um papel fundamental na promoção do desenvolvimento econômico e social, e que apoia, direta e indiretamente, o emprego de 58,1 milhões de pessoas, contribui com mais de US\$ 2,4 trilhões para a economia global e transporta mais de 3,3 bilhões de passageiros e US\$ 6,4 trilhões de mercadorias todos os anos, é um importante exemplo do desafio apresentado para a resolução de problemas relacionados à complexidade e incerteza de grandes volumes de dados. Desde 1977, a escala do tráfego aéreo global dobrou a cada 15 anos e provavelmente continuará a dobrar. Esse crescimento da demanda sinaliza a necessidade de mais métodos orientados a dados como forma de melhorar segurança e o desempenho dos voos (Murça et al., 2020).

Portanto, é fundamental o desenvolvimento de novas soluções de tráfego aéreo (Organization, 2013). Nesse sentido, diferentes programas entraram em execução com o objetivo de aprimorar soluções de navegação aérea, como o Single European Sky ATM Research (SESAR) na Europa, o Next Generation Air Transportation System (NextGen) nos EUA e o SIRIUS, no Brasil. Tais programas visam também a promoção da evolução do gerenciamento de tráfego aéreo, convergindo para um ambiente automatizado, integrado e interoperável (Monteiro et al., 2017).

O objetivo principal deste trabalho é apresentar abordagens inspiradas em algoritmos de busca baseados em árvores de jogos - *game tree*, Busca em Profundidade (DFS, do inglês

Depth First Search) e Alfa-Beta, com foco na resolução de conflitos entre aeronaves, destacando as principais características e aplicabilidade dos modelos estudados. Dentre as vantagens desta abordagem em comparação ao estado-da-arte, destacam-se: a possibilidade de tratar grandes volumes de estados; a dispensa de reprocessamento (o modelo já leva em conta eventuais conflitos que poderiam ser inseridos em virtude de uma tomada de decisão); e a identificação de soluções globais ao invés de sub-ótimos locais. Espera-se que tais metodologias auxiliem no estabelecimento de um gerenciamento de tráfego aéreo mais eficaz.

O restante do trabalho está estruturado da seguinte forma. A Seção 2. descreve os trabalhos relacionados, discute os desafios e oportunidades enfrentados por esta pesquisa e introduz os conceitos fundamentais das metodologias empregadas. A Seção 3. apresenta a solução proposta em detalhes. A seção 4. avalia os resultados obtidos nos estudos de caso definidos para validar a proposta. Por fim, a seção 5. apresenta a conclusão e sugestões para trabalhos futuros.

2. ESTADO DA ARTE

Existem diferentes estudos focados tanto em *big data*, de maneira genérica, quanto no caso aplicado do transporte aéreo. Dentre as pesquisas avaliadas, Ribeiro (2019) apresentou um framework eficiente para armazenamento e gerenciamento de trajetórias quadridimensionais (4D) usando banco de dados NoSQL, que apresenta desempenho destacado na tarefa de detecção de conflitos. Apesar do avanço no tratamento de *big data* para detecção de conflitos, o trabalho de Ribeiro (2019) não apresentou uma metodologia igualmente eficiente para resolução de conflitos.

Kuchar & Yang (2000) consolidaram 68 estudos dentre as principais abordagens da literatura recente sobre detecção e resolução de conflitos entre aeronaves (CD&R), propondo uma taxonomia que permite classificar as vantagens e desvantagens de cada estudo, focando em 6 aspectos: propagação de estado: nominal (trajetória única), pior caso ou probabilístico;

dimensões do estado: somente horizontal, somente vertical ou ambas (desejável); detecção de conflitos: explícito (desejável) ou não; resolução de conflitos: explícita (desejável) ou não; manobras de resolução: curvas, manobras verticais, mudanças de velocidade ou uma combinação dessas opções (desejável); múltiplos conflitos: pares ou globais (desejáveis).

Yang & Wei (2021) apresentam um modelo para gerenciamento de Aeronaves de Decolagem e Pouso Vertical (eVTOL) sob o pano de fundo da Mobilidade de Área Urbana (UAM). A capacidade de gerenciar essas aeronaves com segurança em uma área urbana lotada é um desafio significativo para o gerenciamento de tráfego aéreo. Para permitir operações sob demanda seguras e eficientes de voo livre autônomo em UAM, Yang & Wei (2021) apresentam uma abordagem como um Processo de Decisão de Markov (MDP) usando um algoritmo de busca baseado na abordagem Monte Carlo. No cenário estudado por Yang & Wei (2021), a aeronave tem um objetivo e decide qual direção tomar em tempo real, enquanto que no transporte aéreo as trajetórias estão sujeitas ao plano de voo previamente definido e modificado devido à restrição final e implementação das trajetórias.

Especificamente no campo de *game tree*, Gianazza & Alliot (2002) apresenta uma proposta de otimização do controle de tráfego aéreo baseada em algoritmos *tree search* e algoritmo genético. Ambas as abordagens forneceram configurações ótimas de setor, em um tempo curto o suficiente para um uso operacional.

Pesquisas envolvendo jogos nos quais humanos competem com uma máquina (computador) são estudadas há anos, e ocupam papel central no campo de estudos relacionados à tomada de decisão. Nos primeiros programas, o computador possuía limitações em termos de modelagem e/ou hardware (processamento e memória), porém os avanços nesse campo e no campo da arquitetura de computadores finalmente permitiram que os computadores comesçassem a se destacar em jogos complexos, incluindo o xadrez. Muitos algoritmos já foram desenvolvidos para encontrar o melhor próximo passo para o computador, incluindo algoritmos

sequenciais como MiniMax, NegaMax, Negascout, SSS*, B* e Alfa-Beta (Elnaggar et al., 2014).

As principais abordagens de programas de jogos usam árvores de jogo para encontrar a próxima melhor jogada, e, em geral, são modelados da seguinte forma (Elnaggar et al., 2014):

- Cada nó representa um estado de jogo.
- A raiz representa o estado atual do jogo.
- Todas as ramificações de um determinado nó representam todos os movimentos possíveis a partir desse nó.
- O nó que não possui nenhum sucessor é chamado de folha.

Uma função de avaliação é usada para determinar quando um nó folha representa uma vitória, derrota, empate ou apenas uma pontuação, caso o algoritmo tenha sido interrompido antes de qualquer jogador vencer, perder ou o jogo terminar com um empate. Os desenvolvedores costumam fazer isso porque, em jogos mais complexos, não existe um algoritmo prático que possa pesquisar em toda a árvore em um tempo razoável, mesmo que use o poder do processamento paralelo. Um exemplo para isso é o jogo de damas e xadrez onde é necessário avaliar cerca de 10^{20} e 10^{40} nós respectivamente (Elnaggar et al., 2014).

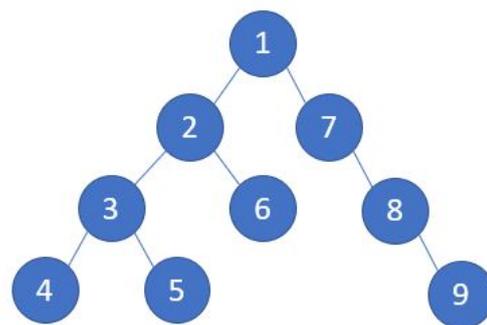


Figure 1 Abordagem baseada no DFS.

A categorização mais comum para algoritmos de busca em árvore de jogos é baseada em profundidade e amplitude. Na pesquisa em profundidade (DFS) o algoritmo começará da raiz e explorará o primeiro nó-filho de cada nível, descendo em uma única linha de nós até atingir o limite de profundidade previamente estabelecido antes de retroceder. Por sua vez, a busca em largura consiste na inspeção de todos os filhos

do nó raiz antes de inspecionar qualquer nó do nível imediatamente abaixo, conforme ilustrado na Figura 1 (Elnaggar et al., 2014).

Os algoritmos de busca em profundidade são categorizados como algoritmos de força bruta, que consiste na observação de todas as variações em uma determinada profundidade. Os principais exemplos de algoritmos de força bruta são MiniMax, NegaMax, Alfa-Beta, NegaScout e Principle-Variation. O algoritmo MiniMax é um algoritmo de árvore de jogo que é dividido logicamente em duas etapas, a primeira para o primeiro jogador que é o computador e a segunda para o segundo jogador que é o humano. O algoritmo tenta encontrar a melhor jogada permitida para o computador, mesmo que o humano faça a melhor jogada para ele. O que significa que maximiza a pontuação do computador quando escolhe o movimento do computador, enquanto minimiza essa pontuação escolhendo o melhor movimento legal para o humano quando escolhe o movimento humano (Russell & Norvig, 2009).

Já o algoritmo Alfa-Beta é uma modificação inteligente que pode ser aplicada ao algoritmo MiniMax (ou NegaMax). Esta variante baseia-se no princípio de que muitos galhos podem ser podados da árvore do jogo, o que reduz o tempo necessário para realizar uma busca na árvore, e dará o mesmo resultado que MiniMax. A ideia principal do algoritmo é cortar os ramos desinteressantes da árvore do jogo. O algoritmo Alfa-Beta usa duas variáveis (alfa e beta) para detectar esses casos, portanto, qualquer valor menor que alfa ou maior que beta será cortado automaticamente sem afetar o resultado da árvore de pesquisa, conforme ilustrado na Figura 2 (Knuth, 2000).

Neste trabalho, serão estudadas duas abordagens de *game tree*: a primeira baseada no algoritmo MiniMax, aqui denominada DFS, e a segunda baseada no Alfa-Beta (Pruning), aqui denominada ABP.

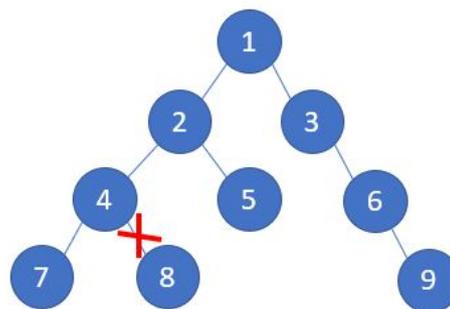


Figure 2 Abordagem baseada no Alfa-Beta.

3. MÉTODO DA PESQUISA E PROPOSTA DE SOLUÇÃO

O presente trabalho faz parte de um estudo mais completo que envolve desenvolvimento de um framework completo para detecção e resolução de conflitos, sendo a primeira parte, detecção de conflitos, objeto de outra publicação¹. O foco deste documento está na resolução de conflitos entre aeronaves, e como os algoritmos baseados em *game tree* podem auxiliar nesta tarefa. Inicialmente, é importante estabelecermos as premissas que possibilitaram a modelagem ora apresentada, e que permite a aplicação das abordagens DFS e Alfa-Beta.

Em um processo de detecção de conflitos entre aeronaves é desejável que se identifique não só quais aeronaves estão em conflito como também outras aeronaves que podem ser eventualmente impactadas por qualquer alteração nas trajetórias das aeronaves conflitantes. A quantidade de aeronaves envolvidas ou impactadas por um conflito, bem como a quantidade de alterações nas trajetórias de uma ou múltiplas aeronaves, leva a diferentes estados de combinações que podem solucionar ou não um determinado conflito. A quantidade de estados possíveis E pode ser encontrada por meio da Equação 1, onde a indica a quantidade de aeronaves envolvidas ou impactadas por um conflito e t a quantidade de trajetórias para cada aeronave.

$$E = t^a \quad (1)$$

É possível que uma quantidade mínima de aeronaves necessite de alterar sua trajetória,

¹<https://latam.ieceer9.org/index.php/transactions/article/view/6811>

ou seja, o t seria diferente de 1 apenas para uma pequena quantidade de aeronaves. Porém, para fins de considerar o cenário mais complexo possível, neste trabalho consideraremos que todas as aeronaves apresentarão t trajetórias possíveis para cada simulação.

Desta forma, no processo de detecção de conflitos, o modelo já considera tanto conflitos entre as trajetórias previstas, ou *default*, quanto trajetórias alternativas para cada aeronave, garantindo que se a escolha por uma determinada trajetória alternativa leva a um novo conflito, este segundo conflito deve fazer parte do problema a ser resolvido. Assim, ao final do processo de detecção de conflitos, todas as aeronaves impactadas, seja pela trajetória *default* ou por quaisquer uma das trajetórias alternativas apresentadas, são colocadas em um cluster de conflitos, possibilitando a identificação de todas as combinações ou estados possíveis.

Após a identificação de todos os estados possíveis, o problema passa a ser, então, encontrar uma combinação ou um estado que represente uma resolução de conflitos satisfatória, ou seja, que não exista conflito e que apresente o melhor valor de uma função de avaliação possível. Essa função de avaliação pode representar qualquer critério desejado (consumo de combustível, quantidade de passageiros impactados, performance das aeronaves, etc), e é calculada para cada nó da aeronave, sendo os valores calculados para os nós folha os valores de função de avaliação que serão usados como critérios de escolha.

Por fim, tanto o algoritmo DFS quanto o algoritmo ABP serão executados para uma mesma combinação de simulação (a aeronaves e t trajetórias), sendo computados o desempenho de cada abordagem na resolução de conflitos.

4. ANÁLISE DOS RESULTADOS

Como forma de avaliar as abordagens para resolução de conflitos, foram criados conjuntos de dados aleatórios com diferentes configurações de aeronaves e trajetórias por aeronaves, o que possibilitou a execução da proposta com combinações de maior e menor

grau de complexidade. Foram projetados cenários com 3, 5, 7, 9, 12, 15, 18, 20, 25 e 30 aeronaves e 3 trajetórias por aeronave em todos os cenários. A complexidade desses testes pode ser verificada na Tabela 1, que indica a ordem de grandeza da quantidade de estados possíveis para cada configuração testada (função E), conforme demonstrado pela Equação 1.

Table 1 Casos de teste

Aeronaves	Trajeto�rias	Estados Poss�veis
3	3	10^1
5	3	10^2
7	3	10^3
9	3	10^4
12	3	10^5
15	3	10^7
18	3	10^8
20	3	10^9
25	3	10^{11}
30	3	10^{14}

Considerando que   desej vel um tempo de resposta curto, ou seja, que o algoritmo encontre a solu o em um curto intervalo de tempo, definiu-se um limiar de 120 segundos, ou seja, foram considerados, apenas, resultados nos quais a solu o pode ser encontrada em menos de 2 minutos.

Table 2 Resultados obtidos com os algoritmos DFS e ABP

a	DFS (ms)	ABP (ms)	E ABP
3	0,33	0,27	10^1
5	0,07	0,00	10^1
7	0,63	0,10	10^1
9	7,83	0,13	10^1
12	145,27	0,40	10^2
15	3.858,57	1,40	10^2
18	106.181,87	3,57	10^2
20	-	8,90	10^2
25	-	62,13	10^2
30	-	374,97	10^2

A Tabela 2 apresenta os resultados obtidos tanto com o algoritmo DFS quanto com o algoritmo ABP. Foi definida uma fun o de avalia o gen rica, ou seja, com valores rand micos, e buscou-se pelo estado que apresentasse o maior valor da fun o de avalia o. As c lulas preenchidas com "—" indicam que a execu o do algoritmo demorou

mais do que 120 segundos, ou seja, a execução foi abortada/desconsiderada. A segunda e a terceira coluna apresentam, respectivamente, os tempos de execução dos algoritmos DFS e ABP, respectivamente, e a última coluna apresenta o espaço de busca verificado pelo algoritmo ABP, permitindo avaliar em qual medida o mecanismo de *pruning* do Alfa-Beta possibilitou reduzir esse espaço de busca.

No cenário mais complexo, que é composto por 30 aeronaves, o algoritmo ABP possibilitou uma redução na busca bastante significativa, da ordem de 10^{28} . Como o algoritmo DFS percorre toda a árvore, seu espaço de busca torna-se significativamente maior. A comparação da última coluna da Tabela 1 com a última coluna da Tabela 2 indica que a abordagem ABP reduz consideravelmente o esforço do algoritmo, justificando, assim, o resultado consideravelmente melhor.

A Figura 3 apresenta o gráfico comparando as duas abordagens, ilustrando os números apresentados na Tabela 2. A reta da função Limiar divide o gráfico na posição 120.000 milissegundos, separando os resultados que são viáveis (abaixo da reta) dos que são inviáveis (acima da reta), estes últimos por levarem mais de 2 minutos para identificação do resultado. Como pode ser observado, o algoritmo baseado na abordagem DFS consegue solucionar em tempo satisfatório configurações de conflitos mais simples, com menos aeronaves. Graficamente, esse comportamento é representado pela manutenção da função referente ao DFS manter-se abaixo do Limiar até a posição 18 (aeronaves) no eixo horizontal. Na aplicação em que o limite de execução ficou próximo do limite aceitável, isto é, com mais aeronaves (justamente a posição 18 do eixo horizontal), o algoritmo DFS encontrou a solução em 1 minuto e 46 segundos, em um contexto com 10^8 estados possíveis, situação que apesar de mais simplificada, pode ser aplicada em setores específicos do espaço aéreo.

Por sua vez, a abordagem ABP conseguiu encontrar a solução para todos os testes realizados, inclusive os mais complexos (da ordem de 10^{44} combinações possíveis), com um pico de tempo de execução ainda menor

do que o observado com a abordagem DFS, isto é, aproximadamente 47 segundos. Esse comportamento é ilustrado pela manutenção da função referente à abordagem ABP sempre abaixo do Limiar, mais precisamente sempre muito próximo do eixo horizontal, sugerindo que para todos os casos a solução foi encontrada em tempo sensivelmente inferior aos 2 minutos estabelecido.

Por fim, é importante ressaltar que, como o foco do trabalho consistiu na resolução de conflitos entre aeronaves agrupadas em clusters de conflitos, não foram identificados datasets equivalentes que pudessem ser utilizados como forma de comparar os resultados. Diante dessa dificuldade, optou-se por utilizar dados randomizados gerados a partir de um algoritmo próprio, que possibilitou, também, garantir que a abordagem proposta pudesse ser aplicada tanto em cenários simples quanto em cenários demasiadamente complexos, que, inclusive, não devem ser observados no contexto real, garantindo a eficácia da proposta.

5. CONCLUSÃO

Investigar metodologias e modelos para manipulação de grandes volumes de dados (*big data*), principalmente quando aplicadas às diferentes áreas da sociedade, é fundamental para o incremento da qualidade na tomada de decisão, e no caso do transporte aéreo não poderia ser diferente. Este trabalho teve como objetivo exercitar o tratamento de *big data* aplicado ao contexto de análise prévia de rotas para buscar e solucionar possíveis conflitos de tráfego aéreo, investigando métodos que pudessem auxiliar a tomada de decisão por órgãos de gerenciamento de fluxo e de controle de tráfego aéreo considerando situações de grande complexidade.

Foram apresentadas e estudadas duas abordagens inspiradas em algoritmos clássicos baseados em *game tree*, Busca em Profundidade (DFS) e Alfa-Beta (ABP), com foco na etapa de resolução de conflitos entre aeronaves, tendo sido possível observar que, com uma modelagem adequada, tais algoritmos podem

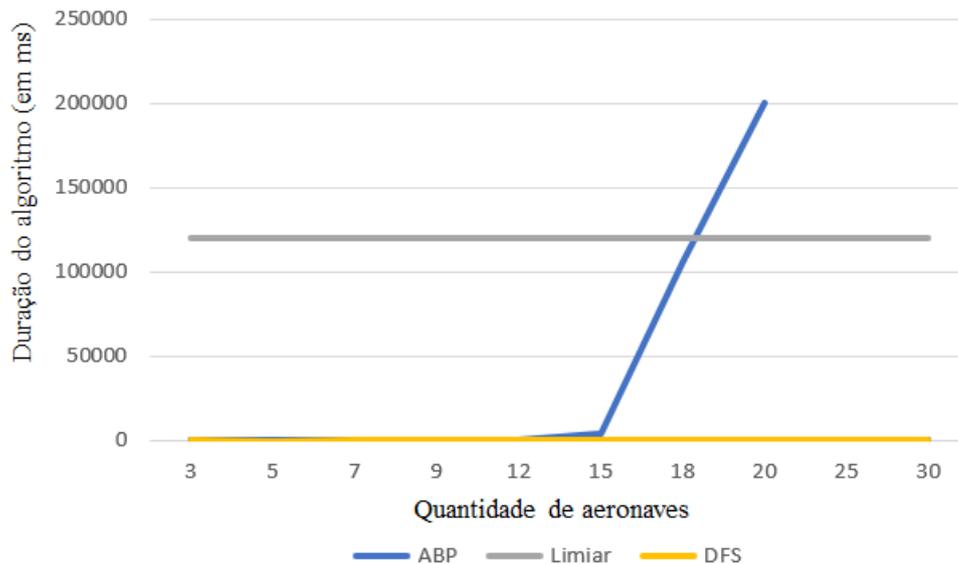


Figure 3 Comparação das abordagens DFS e ABP com o Limiar.

auxiliar na identificação da solução mais adequada, ou seja, na seleção da combinação de trajetórias que garante a segurança do voo, em especial a abordagem inspirada no Alfa-Beta. Adicionalmente, a metodologia apresentada possibilita lidar com um importante problema das abordagens de detecção e resolução de conflitos, que é provocar novos conflitos ao se tentar solucionar um conflito inicial. A ideia de se trabalhar com clusters de conflitos possibilita, em um único passo, antecipar e resolver eventuais consequências de uma mudança de trajetória necessária para se resolver um conflito explícito.

Como sugestão para trabalhos futuros, é importante amadurecer a modelagem, principalmente nas fases anteriores do voo e no estágio de detecção de conflitos, de modo a possibilitar a identificação das medidas adequadas para resolução de conflitos com disponibilidade de tempo suficiente para que as soluções propostas possam ser executadas. Adicionalmente, pode ser interessante aplicar novas abordagens baseadas em algoritmos de busca, de forma a avaliar os comportamentos comparativamente aos dois modelos aplicados neste trabalho.

AGRADECIMENTOS

Este trabalho é parcialmente apoiado por: CNPq, processo: 309545/2021-8; e FAPESP, processo: 2021/06210-3.

References

- Elnaggar, A., Gadallah, M., Mostafa, M. & Eldeeb, H. (2014). A comparative study of game tree searching methods, *International Journal of Advanced Computer Science and Applications* 5, 68–77.
- Gianazza, D. & Alliot, J.-M. (2002). Optimization of air traffic control sector configurations using tree search methods and genetic algorithms *Anais do DASC 2002, Digital Avionics Systems Conference*, Vol. 1 Irvine, United States pp. pp 2A5–1 – 2A5–8.
- Kieckbusch, D. S., Rocha Filho, G. P., Ribeiro, V. F., Fregnani, J. A., Mattos, B. S. & Weigang, L. (2019). Negotiation approach by reinforcement learning for takeoff sequencing decision in airports *Anais do 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE pp. 4477–4482.

- Knuth, D. (2000). Selected Papers on Analysis of Algorithms, Center for the Study of Language and Information, USA.
- Kuchar, J. & Yang, L. (2000). A review of conflict detection and resolution modeling methods, *IEEE Transactions on Intelligent Transportation Systems* 1(4), 179–189.
- Monteiro, L. B., Barbosa, I. S. R., Weigang, L., Fregnani, J. A., de Oliveira, I. R. & Balvedi, G. (2017). Uma proposta de arquitetura swim para o cenário brasileiro *Anais do XVI Sitraer, UFRJ*.
- Murça, M. C. R., Guterres, M. X., de Oliveira, M., Tarelho Szenczuk, J. B. & Souza, W. S. S. (2020). Characterizing the brazilian airspace structure and air traffic performance via trajectory data analytics, *Journal of Air Transport Management* 85, 101798.
- Organization, I. C. A. (2013). Doc 9750-an963. 2013-2028 global air navigation plan. fourth edition., ICAO.
- Ribeiro, V. F. (2019). A Novel Approach for Conflict Detection and Resolution for Trajectory-Based Operations in 4D-Navigation using NoSQL Databases and Local Search Algorithms PhD thesis, Universidade de Brasília.
- Russell, S. & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*, Prentice Hall Press, USA.
- Yang, X. & Wei, P. (2021). Autonomous free flight operations in urban air mobility with computational guidance and collision avoidance, *IEEE Transactions on Intelligent Transportation Systems* 22(9), 5962–5975.