

Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Abordagem de Agrupamento no Planejamento de
Voo sob Tempo Severo Convectivo**

Iuri Souza Ramos Barbosa

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientador
Prof. Dr. Li Weigang

Brasília
2019

Dedicatória

Dedico aos professores que me inspiraram e que foram os maiores incentivadores da minha carreira. Às pessoas que me ensinaram a vencer obstáculos e a nunca desistir, em especial, aos meus queridos Pais.

Agradecimentos

Agradeço ao meu orientador, *Prof. Dr. Li Weigang*, pelo apoio, pela confiança e pela paciência demonstrados ao longo desses anos de trabalho. Agradeço também aos colegas do TransLab, *Lucas, Igor, Leonardo e Vitor*, obrigado pelo apoio e ajuda constantes.

Agradeço a minha amada esposa, companheira de todas as horas, *Luana*, por estar sempre me apoiando e incentivando, mesmo nos momentos mais difíceis. Obrigado, te amo! Por fim, agradeço a minha família, por ter me ensinado desde cedo a enfrentar meus desafios com coragem e disposição.

Resumo

Uma aeronave em voo detecta a presença de condições climáticas adversas em sua rota através de seu radar meteorológico de bordo, e de posse dessa informação, o piloto toma a decisão de realizar ou não um desvio do obstáculo meteorológico a sua frente. No entanto, o radar meteorológico da aeronave possui uma limitação quanto ao alcance da detecção de células convectivas (nuvens *cumulonimbus*) no espaço aéreo. Dessa forma, os procedimentos que o piloto pode realizar ficam limitados à precisão deste dispositivo. Este trabalho tem por objetivo desenvolver um sistema inteligente voltado ao *planejamento de voo num espaço aéreo sob condições meteorológicas adversas*. É proposta uma metodologia computacional utilizando duas fontes de dados: uma de rastreamento de aeronaves em voo para delimitar as aerovias do espaço aéreo e uma outra fonte para o rastreamento de ocorrências de células convectivas via radares meteorológicos terrestres para acompanhar as condições meteorológicas adversas no espaço aéreo brasileiro. Dessa forma, sabendo das localizações das células convectivas e das posições das aerovias no espaço aéreo, é possível identificar em duas etapas se ocorrerá uma interseção entre uma aeronave utilizando aerovias do espaço aéreo e as células convectivas rastreadas. Numa primeira etapa, posições de aeronaves em voo são utilizadas na identificação das fronteiras que delimitam as aerovias no espaço aéreo, e numa segunda etapa, as localizações das aerovias obtidas na etapa anterior são comparadas com as localizações das células convectivas em busca de possíveis interseções entre as mesmas. Para delimitar as fronteiras das aerovias conectando dois aeroportos, um algoritmo de interpolação linear é utilizado para fazer a transformação de posições de voo rastreadas em posições de voo normalizadas, enquanto que um algoritmo de *clustering* é utilizado para agrupar estas posições em *clusters*, que por sua vez definem as fronteiras das aerovias do espaço aéreo. Para identificar interseções entre as aerovias identificadas e as células convectivas rastreadas, as posições das células convectivas são comparadas com as posições pertencentes aos *clusters* identificados, e uma interseção é identificada caso a distância calculada seja menor ou igual ao raio de ação da célula convectiva. As interseções identificadas são, então, utilizadas na fase de planejamento de voo pelas partes interessadas. Simulações com diferentes algoritmos e parâmetros em diversos cenários de teste mostram que os melhores resultados quanto à

delimitação de aerovias são obtidos com a utilização do algoritmo de *clustering Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) fazendo uso da fórmula *Haversine* como medida de distância. Tanto a utilização do algoritmo de *clustering Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN), de uso mais conveniente que o algoritmo DBSCAN, quanto a utilização da distância euclideana, utilizada como medida de distância padrão nos algoritmos de DBSCAN e HDBSCAN, mostraram resultados aquém do esperado. Por fim, a solução proposta contribui para a otimização do fluxo de tráfego aéreo, redução do atraso de aeronaves, redução do consumo de combustível, etc., ou seja, representa um fator de impacto positivo no Gerenciamento de Tráfego Aéreo.

Palavras-chave: Planejamento de Voo, Meteorologia, Otimização, Aprendizagem Não Supervisionada, Análise de Agrupamento de Dados.

Abstract

Aircraft detect the existence of adverse weather conditions on their way with the help of their onboard weather radar. Knowing the location of the weather obstacle, the pilot then decides whether or not to change the course of the aircraft. However, the onboard weather radar has some limitations related to the detection range of convection cells (*cumulonimbus* clouds) in the airspace. Therefore, the course of actions that the pilot can perform is limited to the accuracy of this device. The proposed work builds an intelligent system for flight planning under adverse weather conditions. The proposed methodology mainly uses two data sources. On the one hand, it consumes flight tracking information to delimit the boundaries of airways while, on the other hand, consumes weather information on convection cells obtained from ground-based weather radars in the Brazilian airspace. Therefore, knowing the locations of convection cells and the boundaries of the airways in the airspace, it is possible to identify in two steps whether an aircraft is heading towards the tracked convection cells. Firstly, tracked flight positions are used to identify the boundaries that delimit the airways in the airspace. Secondly, the locations obtained for the airways in the previous step are then compared with the locations of the convection cells in order to identify possible intersections between them. To delimit the boundaries of airways connecting departure and destination airports, a linear interpolation algorithm is used to transform the tracked flight positions into normalized flight positions. Next, a clustering algorithm is used to group these new flight positions into clusters, which in turn define the boundaries of the airways. An intersection between the newly defined airways and the tracked convection cells is found if the measured distance is less than or equal to the radius of the convection cell. The identified intersections are then used for flight planning by stakeholders. Regarding the delimitation of airways, after running simulations using different algorithms and parameters in several test scenarios, the best results are obtained using the *Density-Based Spatial Clustering of Applications with Noise* algorithm along with the *Haversine* formula as its distance measure parameter. Both the usage of *Hierarchical Density-Based Spatial Clustering of Applications with Noise* algorithm, and the usage of Euclidean distance provide poor results. Finally, the proposed solution contributes to the optimization of air traffic flow, reduction of aircraft delay, reduction

of fuel consumption, etc., in other words, has a positive impact factor in Air Traffic Management.

Keywords: Flight Planning, Meteorology, Optimization, Unsupervised Learning, Cluster Analysis.

Sumário

1	Introdução	1
1.1	Planejamento de Voo sob Condições Meteorológicas Adversas	1
1.2	Definição do Problema	3
1.3	Objetivos	4
1.4	Metodologia	5
1.5	Estrutura do Trabalho	5
2	Trabalhos Relacionados	7
2.1	<i>Shortest Path Problem</i>	7
2.2	Abordagens utilizando <i>Machine Learning</i>	9
2.3	Outras Abordagens	10
2.4	Considerações Finais	10
3	Fundamentação Teórica	12
3.1	Mineração de Dados	12
3.1.1	Processo de Descoberta do Conhecimento	12
3.1.2	Padrões em Mineração de Dados	13
3.2	Agrupamento de Dados (<i>Clustering</i>)	14
3.2.1	Conceitos Básicos	14
3.2.2	Métodos de <i>Clustering</i>	16
3.2.3	Processo de Delimitação de Aerovias como um Problema de <i>Clustering</i>	16
3.2.4	Density-Based Spatial Clustering of Applications with Noise (DBSCAN)	18
3.2.5	Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)	21
4	Gerenciamento de Tráfego Aéreo	24
4.1	Sistema ATM	24
4.1.1	Navegação Aérea	24
4.1.2	Meteorologia Aeronáutica	27

4.2	Modernização do Sistema de Gerenciamento de Tráfego Aéreo	30
4.2.1	System-Wide Information Management	30
4.2.2	Programa SIRIUS	31
5	Modelagem e Implementação	34
5.1	Arquitetura Proposta	34
5.2	Fontes de Dados	35
5.2.1	Rastreamento de Aeronaves	35
5.2.2	Tempo Severo Convectivo	36
5.2.3	Outras Fontes de Dados	37
5.3	Normalização de Posições de Voo	40
5.4	Medidas de Distância entre Posições de Voo	45
5.4.1	Distância Euclideana	45
5.4.2	Fórmula de <i>Haversine</i>	45
5.5	Delimitação de Aerovias	47
5.6	Interseções entre Células Convectivas e Aerovias	51
6	Simulações e Resultados	56
6.1	Configuração dos Cenários de Teste	56
6.1.1	Método de Seleção de Cenários de Teste	57
6.1.2	Seleção de Parâmetros Gerais	58
6.1.3	Seleção de Parâmetros - DBSCAN	59
6.1.4	Seleção de Parâmetros - HDBSCAN	59
6.1.5	Métodos de Avaliação de Experimentos	59
6.2	Experimentos e Resultados	62
6.2.1	Cenário Santos Dumont - Congonhas	62
6.2.2	Cenário Congonhas - Santos Dumont	68
6.2.3	Cenário Presidente JK - Congonhas	73
6.2.4	Cenário Congonhas - Presidente JK	77
6.2.5	Cenário Pinto Martins - Guarulhos	82
6.2.6	Cenário Guarulhos - Pinto Martins	86
6.3	Discussão e Análise de Resultados	91
6.3.1	Análise do Processo de Rastreamento de Posições de Voo	91
6.3.2	Análise do Processo de Normalização de Posições de Voo	92
6.3.3	Análise do Processo de Delimitação de Aerovias	93
6.3.4	Análise do Processo de Interseções entre Células Convectivas e Aerovias	96

7 Conclusões e Trabalhos Futuros	98
7.1 Contribuições e Pesquisas Futuras	99
7.2 Considerações Finais	100
Referências	101

Lista de Figuras

1.1	Trajetória do voo GOL2026 (esquerda) e células convectivas detectadas via STSC (direita). Informações registradas em 26 de Janeiro de 2015 às 18:08 GMT-3..	2
3.1	Etapas do processo de mineração de dados.	13
3.2	<i>Clusters</i> com formatos arbitrários.	18
3.3	<i>Clusters</i> encontrados pelo algoritmo DBSCAN.	19
3.4	<i>Clusters</i> encontrados pelo algoritmo K-Means.	19
3.5	Ilustração dos conceitos de <i>density-reachability</i> e <i>density-connectivity</i>	21
3.6	<i>Clusters</i> encontrados pelo algoritmo HDBSCAN.	22
4.1	Fases de um voo operado por Instrument Flight Rules (IFR).	26
4.2	Distribuição de <i>flight levels</i> num espaço RSVM.	28
5.1	Visão geral do sistema. As informações de entrada / saída estão na cor verde, o armazenamento de dados está em amarelo, os dados recuperados / temporários estão em azul e os algoritmos propostos estão coloridos em vermelho..	35
5.2	Rastreamento de aeronaves no cotexto global pelo Flightradar24.	37
5.3	Disposição de células convectivas no espaço aéreo brasileiro.	38
5.4	Disposição de células convectivas próximas à Brasília.	38
5.5	Representação do processo de delimitação de aerovias.	48
6.1	Posições dos voos GLO1055 (esquerda) e GLO1013 (direita) em latitudes e longitudes.	63
6.2	Altitudes das posições dos voos GLO1055 (esquerda) e GLO1013 (direita) ao longo do eixo tempo.	63
6.3	Velocidades das posições dos voos GLO1055 (esquerda) e GLO1013 (direita) ao longo do eixo tempo.	64
6.4	Partição de <i>longitude</i> = $-44, 166^\circ$ dos experimentos #8 (esquerda) e #24 (direita) contendo as posições de voo normalizadas separadas em <i>clusters</i>	67

6.5	Aerovias dos experimentos #8 (esquerda) e #24 (direita) geradas durante o processo de delimitação de aerovias.	67
6.6	Posições dos voos GLO1006 (esquerda) e ONE6000 (direita) em latitudes e longitudes.	69
6.7	Altitudes das posições dos voos GLO1006 (esquerda) e ONE6000 (direita) ao longo do eixo tempo.	69
6.8	Velocidades das posições dos voos GLO1006 (esquerda) e ONE6000 (direita) ao longo do eixo tempo.	69
6.9	Aerovia do experimento #8 gerada durante o processo de delimitação de aerovias.	72
6.10	Posições do voo TAM3173 em latitudes e longitudes.	73
6.11	Altitudes das posições do voo TAM3173 ao longo do tempo.	74
6.12	Velocidades das posições do voos TAM3173 ao longo do tempo.	74
6.13	Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias.	76
6.14	Posições do voo TAM3733 em latitudes e longitudes.	78
6.15	Altitudes das posições do voo TAM3733 ao longo do tempo.	78
6.16	Velocidades das posições do voos TAM3733 ao longo do tempo.	79
6.17	Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias.	81
6.18	Posições do voo GLO1507 em latitudes e longitudes.	83
6.19	Altitudes das posições do voo GLO1507 ao longo do tempo.	83
6.20	Velocidades das posições do voos GLO1507 ao longo do tempo.	85
6.21	Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias.	86
6.22	Posições do voo TAM3326 em latitudes e longitudes.	87
6.23	Altitudes das posições do voo TAM3326 ao longo do tempo.	88
6.24	Velocidades das posições do voos TAM3326 ao longo do tempo.	88
6.25	Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias.	90
6.26	Aerovias do experimento #5 geradas durante o processo de delimitação de aerovias (onde os pontos em vermelho representam as posições de voo relativas ao cenário de teste Congonhas - Santos Dumont enquanto os pontos e azul ao cenário Santos Dumont - Congonhas).	95

6.27	Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias (onde os pontos em vermelho representam as posições de voo relativas ao cenário de teste Congonhas - Presidente JK enquanto os pontos e azul ao cenário Presidente JK - Congonhas).	96
6.28	Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias (onde os pontos em vermelho representam as posições de voo relativas ao cenário de teste Guarulhos - Pinto Martins enquanto os pontos e azul ao cenário Pinto Martins - Guarulhos).	97

Lista de Tabelas

5.1	Quantidade de números de voo (<i>callsigns</i>) por companhia aérea.	40
5.2	Trechos com maior quantidade de <i>callsigns</i> rastreados.	41
6.1	Trechos com maior quantidade de posições de voo rastreadas.	57
6.2	Processo de normalização realizado no trecho Santos Dumont - Congonhas.	62
6.3	Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Santos Dumont - Congonhas (onde $N = NORM$, $N' = NORM'$, $dist. = distance_measure$, $hav. = haversine$, $auc. = euclidean$, $pts. = min_pts$). .	65
6.4	Experimentos do processo de delimitação de aerovias utilizando HDBSCAN no trecho Santos Dumont - Congonhas (onde $N = NORM$, $N' = NORM'$, $dist. = distance_measure$, $hav. = haversine$, $auc. = euclidean$, $pts. = min_pts$).	66
6.5	Processo de interseção entre aerovias e células convectivas no trecho Santos Dumont - Congonhas (onde n.d. significa <i>não determinado</i> , $dist. = distance_measure$, $hav. = haversine$, $auc. = euclidean$, $pts. = min_pts$). .	68
6.6	Processo de normalização realizado no trecho Congonhas - Santos Dumont.	68
6.7	Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Congonhas - Santos Dumont (onde $N = NORM$, $N' = NORM'$, $dist. = distance_measure$, $hav. = haversine$, $auc. = euclidean$, $pts. = min_pts$). .	71
6.8	Experimentos do processo de delimitação de aerovias utilizando HDBSCAN no trecho Congonhas - Santos Dumont (onde $N = NORM$, $N' = NORM'$, $dist. = distance_measure$, $hav. = haversine$, $auc. = euclidean$, $pts. = min_pts$).	71
6.9	Processo de interseção entre aerovias e células convectivas no trecho Congonhas - Santos Dumont (onde n.d. significa <i>não determinado</i> , $dist. = distance_measure$, $hav. = haversine$, $auc. = euclidean$, $pts. = min_pts$). .	72
6.10	Processo de normalização realizado no trecho Presidente JK - Congonhas. .	73
6.11	Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Presidente JK - Congonhas (onde $N = NORM$, $N' = NORM'$, $dist. = distance_measure$, $hav. = haversine$, $auc. = euclidean$, $pts. = min_pts$). .	75

6.12	Experimentos do processo de delimitação de aerovias utilizando HDBSCAN no trecho Presidente JK - Congonhas (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	75
6.13	Processo de interseção entre aerovias e células convectivas no trecho Presidente JK - Congonhas (onde $dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	77
6.14	Processo de normalização realizado no trecho Congonhas - Presidente JK.	77
6.15	Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Congonhas - Presidente JK (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	80
6.16	Experimentos do processo de delimitação de aerovias utilizando HDBSCAN no trecho Congonhas - Presidente JK (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	80
6.17	Processo de interseção entre aerovias e células convectivas no trecho Congonhas - Presidente JK (onde $dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	81
6.18	Processo de normalização realizado no trecho Pinto Martins - Guarulhos.	82
6.19	Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Pinto Martins - Guarulhos (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	84
6.20	Experimentos do processo de delimitação de aerovias utilizando HDBSCAN no trecho Pinto Martins - Guarulhos (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	84
6.21	Processo de interseção entre aerovias e células convectivas no trecho Pinto Martins - Guarulhos (onde $dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	86
6.22	Processo de normalização realizado no trecho Guarulhos - Pinto Martins.	87
6.23	Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Guarulhos - Pinto Martins (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	89
6.24	Experimentos do processo de delimitação de aerovias utilizando HDBSCAN no trecho Guarulhos - Pinto Martins (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).	90

6.25	Processo de interseção entre aerovias e células convectivas no trecho Guarulhos - Pinto Martins (onde <i>dist.</i> = <i>distance_measure</i> , <i>hav.</i> = <i>haversine</i> , <i>eucl.</i> = <i>euclidean</i> , <i>pts.</i> = <i>min_pts</i>).	91
6.26	Resultados do processo de normalização em todos os cenários de teste.	93
6.27	Resultados do processo de delimitação de aerovias do experimento #5 em todos os cenários de teste.	95

Lista de Abreviaturas e Siglas

ADS-B Automatic Dependent Surveillance-Broadcast.

API Application Program Interface.

ATM Air Traffic Management.

ATS Air Traffic Service.

CLIOS Complex, Large-Scale, Integrated, Open Systems.

COMAER Comando da Aeronáutica.

DBSCAN Density-Based Clustering Based on Connected Regions with High Density.

DECEA Departamento de Controle do Espaço Aéreo.

GANP Global Air Navigation Plan.

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise.

IATA International Air Transport Association.

ICAO International Civil Aviation Organization.

IFR Instrument Flight Rules.

NextGen Next Generation Air Transportation System.

REDEMET Rede de Meteorologia do Comando da Aeronáutica.

RVSM Reduced Vertical Separation Minima.

SESAR Single European Sky ATM Research.

SISCEAB Sistema de Controle do Espaço Aéreo Brasileiro.

STSC Sistema de Tempo Severo Convectivo.

SWIM System-Wide Information Management.

VFR Visual Flight Rules.

Capítulo 1

Introdução

O presente trabalho propõe um sistema inteligente voltado ao planejamento de voo num espaço aéreo sob condições meteorológicas adversas.

1.1 Planejamento de Voo sob Condições Meteorológicas Adversas

O Gerenciamento de Tráfego Aéreo (*Air Traffic Management* - ATM) é uma atividade que depende de vários fatores para seu bom funcionamento. Um dos fatores com grande impacto no funcionamento do Sistema ATM é o fator meteorológico. Dependendo das condições meteorológicas, situações atípicas podem acontecer como: aeronaves podem ser obrigadas a desviar de sua rota em voo, aeronaves podem ser redirecionadas para pouso em outro aeroporto, voos podem ser cancelados, aeroportos podem ser fechados por um determinado período, etc. Por ser um fator de impacto recorrente no Gerenciamento de Tráfego Aéreo, condições meteorológicas são monitoradas com frequência por radares meteorológicos [1], de modo a disponibilizar as informações mais atualizadas aos *stakeholders* ATM. Assim sendo, o presente trabalho tem por objetivo construir um sistema de suporte voltado ao planejamento de voo sob condições meteorológicas adversas.

No cenário atual uma aeronave em voo detecta nuvens *cumulonimbus* (nuvens de forte desenvolvimento vertical que representam uma condição meteorológica adversa) em sua rota através de seu radar meteorológico de bordo, e de posse desta informação o piloto toma a decisão de desviar ou não desta célula convectiva. No entanto, o radar meteorológico da aeronave possui uma limitação quanto ao alcance de detecção de células convectivas no espaço aéreo. Dessa forma, a quantidade de ações e procedimentos que o piloto pode realizar fica limitada à precisão deste dispositivo.

Ao mesmo tempo, estas informações, relativas às posições de células convectivas no espaço aéreo, também podem ser obtidas através de radares meteorológicos terrestres, como o Sistema de Tempo Severo Convectivo (STSC) [2]. Estes radares por sua vez oferecem uma maior riqueza de informações sobre a localização, tamanho, abrangência e deslocamento das células convectivas detectadas pelo STSC. Dessa forma, o piloto da aeronave pode atuar de forma antecipada e preventiva como: desvio antecipado de rota, mudança de aerovias para navegação, etc.

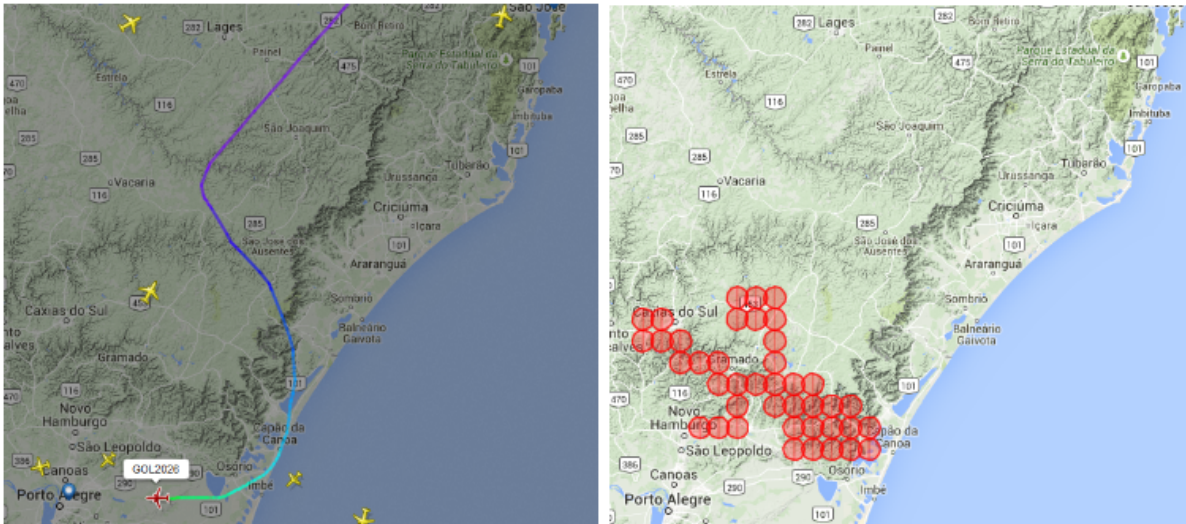


Figura 1.1: Trajetória do voo GOL2026 (esquerda) e células convectivas detectadas via STSC (direita). Informações registradas em 26 de Janeiro de 2015 às 18:08 GMT-3..

Na Figura 1.1, estão representadas a trajetória de um voo sob influência de condições meteorológicas adversas (linha roxa) e as células convectivas identificadas pelo STSC (círculos em vermelho). Como se pode observar, o piloto redirecionou a aeronave para desviar das células convectivas identificadas pelo radar meteorológico de bordo. Vale ressaltar que o piloto só tinha acesso a informações do radar meteorológico de bordo, ao passo que informações sobre células convectivas identificadas a partir de radares meteorológicos terrestres, conforme Figura 1.1 direita, não estavam disponíveis para o piloto no momento do desvio. De modo que, caso estas informações fossem utilizadas, isto resultaria numa detecção e mudança de rota antecipados, e conseqüentemente, num menor tempo de voo, redução do consumo de combustível, aumento da segurança, etc.

A solução proposta visa *prover às partes interessadas informações relevantes* quanto à posição de células convectivas monitorando se estas intersectam as aerovias do espaço aéreo. Isto é, pretende servir como interface de apoio fornecendo informações meteorológicas de forma oportuna aos *stakeholders* ATM para o processo de tomada de decisões, como antecipar e/ou evitar os procedimentos de desvio de rota. A solução proposta pode

ser subdividida em duas etapas. Na primeira etapa, dados de rastreamento de aeronaves em voo são utilizados na *identificação das fronteiras que delimitam as aerovias* no espaço aéreo com a utilização de algoritmos de *clustering*. Na segunda etapa, as fronteiras das aerovias obtidas na etapa anterior são comparadas com as localizações das células convectivas obtidas por *radares meteorológicos terrestres* em busca de *interseções* entre as mesmas. As informações sobre as interseções identificadas podem ser, então, utilizadas na fase de planejamento de voo pelas partes interessadas em ATM, contribuindo assim para o processo de tomada de decisão.

1.2 Definição do Problema

Atualmente pilotos realizam desvios nas aeronaves com base em informações geradas pelo radar meteorológico de bordo das aeronaves, que em tempo hábil alertam pilotos sobre a existência de obstáculos a frente [3]. As informações provenientes de radares de bordo limitam a tomada de decisão dos pilotos por três principais motivos:

- Devido à limitação de alcance de radares meteorológicos de bordo, informações sobre células convectivas na rota de aeronaves variam conforme a acurácia desses equipamentos, o que reduz a chance de um desvio de trajetória ótimo. No entanto, informações sobre células convectivas já estão disponíveis para consulta das aeronaves através de radares meteorológicos terrestres e do Sistema de Detecção de Descargas Atmosféricas [1].
- Não provêm informações completas sobre as condições meteorológicas da região onde a aeronave sobrevoa, de forma que, é possível que o piloto opte por um caminho que parece ser o melhor no momento, embora este se mostre uma escolha ruim mais a frente, com o aparecimento de outras células convectivas na rota da aeronave.
- Deixam a cargo exclusivamente dos pilotos a tarefa de escolher a melhor direção para o desvio de rota da aeronave, o que naturalmente pode resultar em trajetórias não ótimas. Uma solução computacional que sugerisse as melhores rotas poderia reduzir desperdício de recursos (tempo, combustível, dinheiro, pessoal, etc.).

Considerando o exposto acima, o problema a ser considerado é: como melhorar as trajetórias de aeronaves em voo num ambiente de condições meteorológicas adversas, contribuindo, assim, para a melhoria do tráfego aéreo.

1.3 Objetivos

Este trabalho tem por objetivo desenvolver um *sistema inteligente* de suporte ao planejamento de voo num espaço aéreo sob influência de condições meteorológicas adversas. Especificamente, o sistema tem por objetivo *prover mais informações* aos *stakeholders* em ATM ao *monitorar* interseções entre células convectivas e as aerovias do espaço aéreo. As informações providas visam à melhoria do *processo de tomada de decisões* dos *stakeholders* em ATM, principalmente, no que se refere à otimização de desvios de aeronaves frente a condições meteorológicas adversas. Detalhando de forma mais específica o objetivo, têm-se:

- Proposição de uma *metodologia computacional* para o problema de planejamento de voo sob condições meteorológicas adversas com a utilização de duas fontes de dados: uma de rastreamento de aeronaves em voo para delimitar aerovias do espaço aéreo e outra para o rastreamento de ocorrências de células convectivas via radares meteorológicos terrestres para acompanhar condições meteorológicas adversas no espaço aéreo brasileiro;
- Desenvolvimento de uma *arquitetura*, conforme a metodologia proposta, separada em componentes: rastreamento de aeronaves e suas posições de voo, rastreamento de ocorrências de células convectivas no espaço aéreo, transformações de posições de voo, delimitações das fronteiras das aerovias com a utilização de algoritmos de *Machine Learning* com esquema de Aprendizagem Não Supervisionada e identificação de interseções entre células convectivas e as aerovias identificadas;
- Desenvolvimento de um *sistema* seguindo a arquitetura proposta que permita a configuração do sistema utilizando diferentes modelos e parâmetros, realize simulações para diferentes combinações de modelos e parâmetros e em diversos cenários de teste, e por fim, reporte seus resultados nas diferentes métricas pré-estabelecidas;
- Identificação de *modelos e parâmetros ótimos* para os componentes da arquitetura proposta com a comparação das simulações realizadas a partir dos diversos experimentos em diferentes cenários de teste;

Em última análise, a solução proposta deve contribuir para a otimização do fluxo de tráfego aéreo, redução do atraso de aeronaves, redução do consumo de combustível, etc., ou seja, a solução deve representar um fator de impacto positivo no Gerenciamento de Tráfego Aéreo.

1.4 Metodologia

A metodologia proposta é uma solução *alternativa* à utilizada atualmente pela indústria, propõe a utilização de um serviço de rastreamento de posições de voo da aeronave e a utilização de radares meteorológicos terrestres para alertar pilotos sobre se as aeronaves encontram-se em direção a células convectivas (condições meteorológicas adversas). O trabalho será realizado utilizando as seguintes etapas de pesquisa:

- **Levantamento de dados de posições de voo:** rastreamento de informações das aeronaves e de suas posições de voo no espaço aéreo brasileiro;
- **Levantamento de dados de células convectivas:** rastreamento de ocorrências e posições geográficas de células convectivas;
- **Levantamento do estado da arte em algoritmos de *clustering*:** estudo de algoritmos de agrupamento que lidam com os requisitos computacionais do problema;
- **Definição da arquitetura:** proposta que melhor identifique as aerovias presentes no espaço aéreo e que melhor identifique as interseções entre as aerovias encontradas e as células convectivas rastreadas;
- **Implementação da arquitetura:** desenvolvimento de um sistema que implementa a arquitetura proposta e seus componentes;
- **Implementação do simulador:** implementação de um sistema de suporte para a realização de simulações utilizando diferentes modelos, parâmetros, cenários, etc.;
- **Simulação:** realização de simulações de diversos experimentos em diferentes cenários de teste;
- **Análise de resultados:** definição de diferentes métricas para avaliar e comparar os experimentos simulados e seus resultados obtidos;
- **Seleção de modelo e parâmetros:** identificação dos modelos e parâmetros ótimos segundo análise dos resultados obtidos.

1.5 Estrutura do Trabalho

Este trabalho está dividido em sete capítulos. No Capítulo 2 são apresentados trabalhos relacionados abordando: 1) soluções de problemas em Gerenciamento de Tráfego Aéreo utilizando fontes de dados similares e 2) soluções existentes para planejamento de voo sob condições meteorológicas adversas.

No Capítulo 3 são apresentados conceitos essenciais da análise de agrupamento de dados para melhor compreensão da metodologia proposta no Capítulo 5. São também detalhados os algoritmos de *clustering* e seus parâmetros utilizados na solução proposta, como os algoritmos de *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) e *Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN).

O Capítulo 4 apresenta uma fundamentação teórica dos conceitos relevantes do Gerenciamento de Tráfego Aéreo para melhor entendimento do problema em questão, tais como: Sistema ATM, Navegação Aérea, Meteorologia Aeronáutica e a modernização do Sistema ATM. Este capítulo oferece a contextualização para o problema de planejamento de voo sob condições climáticas adversas num contexto de ATM.

No Capítulo 5, é feita uma descrição detalhada das fontes de dados utilizadas no projeto: dados de rastreamento de aeronaves em tempo real e dados meteorológicos relativos ao tempo severo convectivo. Além do mais, a arquitetura e seus componentes são detalhados, assim como detalhes de implementação da solução proposta.

No Capítulo 6, estudos de casos são detalhados segundo a arquitetura e implementação propostas. São simulados uma série de experimentos utilizando diversos modelos e combinações de parâmetros e em diferentes cenários de teste. Neste capítulo, também são definidas as métricas utilizadas para se avaliar o desempenho dos experimentos, que por sua vez são usadas para comparar e discutir os resultados obtidos dos mesmos. Modelos e parâmetros ótimos também são encontrados para cada componente da solução proposta.

Por fim, no Capítulo 7 são discutidas as conclusões e resultados da pesquisa, assim como sugestões de trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Neste Capítulo é feita uma revisão dos trabalhos de planejamento de voos sob condições meteorológicas adversas. Não há muitas pesquisas com foco no problema específico de planejamento sob condições meteorológicas adversas. Mesmo que alguns destes trabalhos não estejam diretamente relacionados ao problema em questão, acredita-se que os mesmos podem servir como fonte de inspiração para a metodologia proposta.

Nas seções deste Capítulo, serão apresentadas duas abordagens para o tratamento do problema de planejamento de voos. A seção 2.1 descreve o problema de planejamento de voo sob a perspectiva de um *shortest path problem*. A seção 2.2 aborda o problema de planejamento de voo utilizando técnicas de *machine learning*. Em ambas as seções foram escolhidos trabalhos que utilizam fontes de dados similares às utilizadas pelo presente trabalho, isto é, utilizando rastreamento de aeronaves e dados meteorológicos. Na seção 2.3 outras abordagens são mencionadas. Por fim, na seção 2.4 as abordagens estudadas e a abordagem proposta são relacionadas entre si.

2.1 *Shortest Path Problem*

Schilke e Hecker (2014) propuseram um sistema de otimização de trajetórias de aeronaves sob tempo severo convectivo. As rotas das aeronaves são definidas com a utilização de informações sobre as condições meteorológicas regularmente atualizadas via radares terrestres [4].

Informações sobre células convectivas fornecidas por estes radares incluem posição, direção e velocidade de deslocamento destas células. Com base nestas informações, as posições das células convectivas presentes no espaço aéreo são estimadas, e dessa forma, é possível adaptar as rotas das aeronaves de forma eficiente. A busca pela rota mais eficiente é tratada como um problema de busca pelo menor caminho num grafo (*shortest path problem*).

Na modelagem proposta, a aeronave sobrevoa livremente o espaço aéreo (*free-flight*), possuindo máxima liberdade na seleção de sua rota, sem levar em conta aerovias pré-definidas no espaço aéreo. Sabe-se, no entanto, que a maior parte de voos comerciais operam de forma diferente, por Instrument Flight Rules (IFR) e trafegam por aerovias pré-definidas pelo Controle de Tráfego Aéreo. Neste trabalho, o espaço aéreo é representado por uma *grid de pontos* de duas dimensões.

Inicialmente a rota é calculada sem levar em consideração problemas ocasionados por células convectivas, e é utilizado como *algoritmo de busca* uma modificação dos algoritmos A^*/Theta^* . Theta^* , variação de A^* , é utilizado a fim de permitir que a aeronave se movimente em qualquer ângulo a partir de um ponto do espaço aéreo, ao contrário de A^* , que somente permitiria movimentos para os pontos vizinhos da *grid* de pontos. A função de custo é definida de forma a atribuir valores muito elevados para regiões representando células convectivas. Dessa forma, o algoritmo ao procurar pelo caminho de menor custo evita passar por tais regiões.

O cálculo da rota inicial (os pontos da *grid* entre origem e destino) é o primeiro passo para a geração da rota a ser utilizada. Depois, a partir desta previsão inicial, é estimado o tempo que a aeronave vai sobrevoar cada um dos pontos encontrados. Para cada ponto intermediário, as condições meteorológicas são novamente estimadas para avaliar se a previsão inicial continua válida. Caso contrário, a busca pela rota de menor custo é recalculada. A modelagem proposta pode exigir um elevado tempo computacional dependendo do cenário em questão, isto é, da disposição das células convectivas e seus deslocamentos.

Em outro trabalho, Cheung (2018) propõe um algoritmo de predição de trajetória (*trajectory prediction*) num ambiente de incertezas climáticas [5]. O trabalho em questão estuda o impacto da incerteza relacionada a *numerical weather prediction* (NWP) no planejamento de voos comerciais. Este trabalho tem por objetivo reduzir o *gap* entre pesquisa atmosférica e a indústria da aviação. Da mesma forma que o trabalho anterior, este estudo também é abordado como um *shortest path problem* e representa o espaço aéreo por uma *grid* de pontos, também chamados de *nós*. O estudo também se concentra na otimização de *free-flights*.

O trabalho utiliza o algoritmo A^* na busca pela rota mais eficiente de origem e destino. Ao contrário do trabalho anterior, a função de custo é definida como o tempo que a aeronave leva para percorrer dois nós de sua *grid*. É utilizada no cálculo a velocidade da aeronave relativa ao solo (*ground speed*), que é calculada utilizando parâmetros climáticos (NWP) como: *tailwind*, *crosswind*, temperatura, etc. Dessa forma, o algoritmo proposto busca pela rota mais rápida entre origem e destino considerando as incertezas relacionadas a NWP.

2.2 Abordagens utilizando *Machine Learning*

Ayhan e Samet (2016) descrevem uma abordagem estocástica para predição de trajetória de aeronaves (*trajectory prediction*) visando um planejamento de voos mais eficiente e realista levando em consideração parâmetros meteorológicos, como velocidade do vento, direção do vento, temperatura, etc. [6]. O estudo utiliza como fontes de dados o rastreamento de aeronaves e observações meteorológicas no espaço aéreo.

Os autores desenvolvem uma abordagem, onde o espaço aéreo é modelado como uma *grid* de pontos de três dimensões, na qual cada ponto representa a localização de uma observação meteorológica. Para a construção desta *grid* de pontos, o espaço aéreo é, primeiramente, separado em regiões onde os parâmetros associados à meteorologia permanecem homogêneos durante um certo intervalo de tempo. Em seguida, destas regiões homogêneas são encontrados seus centróides que, por fim, definem a *grid* de pontos. Em seguida, as posições de voo das aeronaves são combinadas com a *grid* de pontos, de forma a gerar novas posições passando pelos pontos desta *grid*, e dessa forma representar tanto os *atributos de espaço-tempo*, como latitude, longitude, altitude, *timestamp*, quanto os *atributos meteorológicos* utilizados pelo estudo, como velocidade do vento, direção do vento, temperatura, etc. Em seguida, a partir de dados históricos de rastreamento de posições de voo e observações meteorológicas, é utilizado um modelo estocástico baseado em *Hidden Markov Model* (HMM) para a predição de trajetórias que levam em consideração as incertezas meteorológicas presentes no espaço aéreo.

Ayhan et al. (2018) apresentam um sistema de predição (*ETA Prediction System*) para o horário previsto de chegada (*estimated time of arrival* - ETA) de voos comerciais [7]. Esta é uma tarefa particularmente desafiante devido à natureza não-determinística de fatores ambientais e de tráfego aéreo.

O sistema proposto utiliza dados históricos de posições de voo e a mesma *grid* de pontos mencionada no trabalho anterior, porém, com um maior agregado de informações como: parâmetros meteorológicos (velocidade e direção do vento, temperatura, umidade e pressão atmosféricas, etc.), parâmetros de tráfego aéreo (número de aeronaves num setor do espaço aéreo, histórico de congestionamento do setor) e parâmetros aeroportuários (número de voos de chegada e partida nos aeroportos em questão, histórico de congestionamento dos aeroportos, etc.). Em seguida, é realizado um processo de *feature engineering* para a criação de *features* que apresentem maior habilidade em predizer ETA. Estas *features* são utilizadas como parâmetros de entrada num conjunto de modelos de regressão e numa *Recurrent Neural Network* (RNN). Por fim, os modelos gerados que obtiverem melhor performance na predição de ETAs, quando comparados com ETAs obtidas num sistema real em operação, são selecionados pelo sistema.

Uma observação importante é que os trabalhos apresentados acima, por utilizarem dados de rastreamento de aeronaves, não recorrem a uma abordagem *free-flight* de navegação do espaço aéreo, diferentemente das abordagens estudadas anteriormente. Dessa forma, as rotas das aeronaves ficam restritas às aerovias pré-definidas no espaço aéreo.

2.3 Outras Abordagens

Os trabalhos vistos nas seções anteriores abordam o problema de planejamento de rota como: 1) *shortest path problem* utilizando os algoritmos de *Dijkstra*, A^* ou variantes; 2) um problema de *machine learning* utilizando regressão, RNN, HNN, etc. No entanto, outros trabalhos empregam abordagens menos comuns como mencionado a seguir.

Girardet et al. (2013) abordam o mesmo problema de planejamento de rotas de aeronaves sob condições meteorológicas adversas utilizando o algoritmo *Ordered Upwind Method*, levando em consideração a interação de multiagentes na modelagem da solução [8]. Majumder e Prasad (2016) introduzem técnicas para otimizar o planejamento de trajetórias de aeronaves combinando os conceitos de planejamento de rotas global e planejamento de rotas local para encontrar a rota ideal na presença de obstáculos físicos, condições meteorológicas adversas, etc. Desta forma, os autores combinam informações de obstáculos ao longo da trajetória da aeronave (planejamento global) com dados obtidos dos sensores das aeronaves (planejamento local) [9]. Nestes dois trabalhos, é utilizada a abordagem *free-flight* de navegação no espaço aéreo.

2.4 Considerações Finais

As abordagens estudadas compartilham o objetivo comum de resolver o problema de planejamento de rota de aeronaves (*trajectory prediction*) levando em consideração informações sobre as condições meteorológicas do espaço aéreo. Em geral estas abordagens se diferenciam entre si no que se refere aos parâmetros otimizados e às informações meteorológicas utilizadas. No trabalho proposto, são utilizados dados de rastreamento de aeronaves e de posições de células convectivas no espaço aéreo. Mais especificamente, é utilizado o Sistema de Tempo Severo Convectivo (STSC) para rastreamento de células convectivas no espaço aéreo brasileiro. Uma grande vantagem dessa fonte de dados com relação às outras fontes de dados vistas é que esta pode ser acessada livremente, podendo assim, ser acessada por outros pesquisadores.

A abordagem proposta difere das abordagens da seção 2.1, modeladas como *shortest path problem*, dado que não é permitida uma navegação *free-flight*, onde aeronaves possuem elevado grau de liberdade para sobrevoar a maior parte do espaço aéreo. Na solução

proposta, aeronaves navegam o espaço aéreo utilizando aerovias pré-definidas. A abordagem também difere dos trabalhos estudados na seção 2.2 no que se refere ao objetivo da solução final. Na seção 2.2 as abordagens se preocupam em *trajectory prediction* e na previsão do horário previsto de chegada (ETA). A abordagem proposta indica a existência de interseções entre aerovias no espaço aéreo e células convectivas, servindo como um serviço auxiliar no processo de tomada de decisões ATM. Entende-se que outros fatores, como de ordem meteorológicas, controle de tráfego aéreo, etc., são tão importantes quanto tempo severo convectivo, por isso deixa-se a cargo das partes interessadas em ATM a decisão final sobre a rota mais eficiente a ser utilizada.

Capítulo 3

Fundamentação Teórica

Este capítulo descreverá os fundamentos teóricos que compreendem conceitos sobre Mineração de Dados (*Data Mining*) e Análise de Agrupamento de Dados (*Clustering*).

O processo de Descoberta do Conhecimento é utilizado como arcabouço teórico para a formulação da metodologia proposta. Processos, como limpeza, integração, seleção e transformação de dados, são constatados e desenvolvidos na prática na metodologia proposta. Neste capítulo também são apresentados os padrões em Mineração de Dados e é detalhado o padrão de Agrupamento de Dados (*Clustering*), utilizado na metodologia proposta. Mais especificamente, os algoritmos de *clustering* estudados nesta seção serão utilizados no processo de delimitação de aerovias, responsável por delimitar as fronteiras das aerovias no espaço aéreo (seção 5.5).

3.1 Mineração de Dados

Mineração de Dados (*Data Mining*) é o processo de descoberta de padrões de grandes quantidades de dados [10]. Com o crescimento explosivo do volume de dados, resultado do avanço da tecnologia e de ferramentas de coleta e armazenamento de dados, a mineração de dados visa atender a necessidade de se extrair conhecimento dessa vasta quantidade de dados disponível (também chamada de *big data*). Dessa forma, são necessárias ferramentas poderosas capazes de transformar essa enorme quantidade de dados em conhecimento a ser utilizado no processo de tomada de decisões.

3.1.1 Processo de Descoberta do Conhecimento

O processo de Descoberta do Conhecimento (*Knowledge Discovery from Data*) tem por objetivo final a obtenção de conhecimento, e segue, de forma geral, as seguintes etapas [10]:

1. **Limpeza de Dados** - ruídos e dados inconsistentes são removidos.
2. **Integração de Dados** - dados de múltiplas fontes de informação são combinados.
3. **Seleção de Dados** - dados relevantes são selecionados do banco de dados para a tarefa de análise.
4. **Transformação de Dados** - dados são transformados e manipulados para serem utilizados de forma eficaz no processo seguinte de *Data Mining*.
5. **Mineração de Dados** (*Data Mining*) - métodos / algoritmos inteligentes são aplicados para a extração de padrões de dados.
6. **Avaliação de Padrões** (*Pattern Evaluation*) - identificação de padrões relevantes para a geração de conhecimento.
7. **Apresentação de Conhecimento** - técnicas de visualização e de representação de conhecimento são utilizadas para apresentar o conhecimento obtido às partes interessadas.

As etapas de 1 a 4 fazem parte do pré-processamento de dados, onde os dados são preparados para a mineração. Nas etapas 5 e 6, padrões relevantes e conhecimento são descobertos e, posteriormente, apresentados aos usuários na etapa 7. As etapas estão ilustradas pela Figura 3.1.

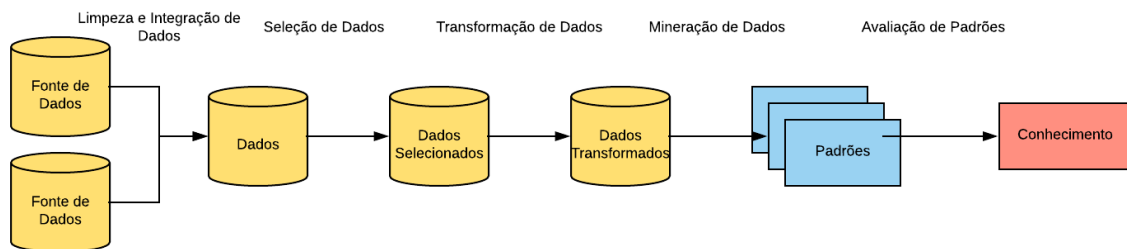


Figura 3.1: Etapas do processo de mineração de dados.

3.1.2 Padrões em Mineração de Dados

No que se refere aos tipos de padrões que podem ser extraídos no processo de mineração de dados, têm-se [10]:

1. **Padrões Frequentes, Associações e Correlações** - padrões que ocorrem com frequência nos dados resultando na descoberta de associações e correlações.

2. **Classificação e Regressão para Análise Preditiva** - processo de descoberta de um modelo (função) capaz de descrever *classes* de dados distintas.
3. **Agrupamento de Dados (*Clustering*)** - processo de descoberta de diferentes grupos *clusters* presentes na massa de dados.
4. **Análise de *Outliers* (*Outlier Analysis*)** - processo de identificação e remoção de *outliers* presentes na massa de dados (dados que representam ruído ou exceções).

Algumas considerações sobre os padrões apresentados acima são necessárias. No processo de classificação, os dados vêm categorizados em classes (*labels*) de antemão. Estes dados e suas respectivas classes são utilizados para a construção de um modelo, representando a associação entre dado e classe. Dessa forma, o modelo pode ser utilizado na predição de classe de *novos* dados cujas classes são desconhecidas. Enquanto a classificação prediz categorias (*categorical labels*), a regressão prediz valores numéricos (*numerical labels*).

Ao contrário da classificação e regressão, que analisam dados categorizados em classes (ou *labels*), o agrupamento de dados (*clustering*) analisa dados sem levar em conta suas classes, que podem existir ou não. Este processo pode ser utilizado para a geração de classes entre os diferentes grupos de dados, também chamados de *clusters*. A seguir serão estudadas as técnicas de agrupamento de dados, que serão utilizadas no desenvolvimento do presente trabalho.

3.2 Agrupamento de Dados (*Clustering*)

Agrupamento de dados (clustering) é o processo de partição de um conjunto de dados em subconjuntos [10]. Cada subconjunto de dados resultante do processo de *clustering* é chamado de grupo ou *cluster*. Diferentes métodos de *clustering* podem resultar em diferentes grupos para o mesmo conjunto de dados. *Clustering* é uma excelente ferramenta de *data mining* na descoberta de grupos previamente desconhecidos no conjunto de dados.

3.2.1 Conceitos Básicos

No processo de *clustering*, dados são agrupados com base no *princípio* de *maximizar a similaridade intra-cluster* (de dentro do grupo) e *minimizar a similaridade inter-cluster* (entre diferente grupos). Em outras palavras, *clusters* são formados para que objetos do mesmo grupo sejam similares entre si, mas para que objetos de grupos diferentes sejam dissimilares. Conseqüentemente, qualquer processo de *clustering* utiliza uma *função*

para *medir a similaridade* entre dois objetos quaisquer, também chamada de *medida de distância* (*distance measure*) entre dois objetos.

Dependendo dos requisitos da análise em questão, a escolha do algoritmo de *clustering* mais apropriado depende de uma série de fatores [10]:

1. **Escalabilidade** - alguns algoritmos funcionam somente para conjuntos de dados pequenos, outros funcionam bem para grandes conjuntos de dados.
2. **Habilidade em lidar com diferentes tipos de dados** - muitos algoritmos são implementados para agrupar somente grupos de dados numéricos, enquanto outros são implementados para lidar com outros tipos de dados (binários, categóricos, etc.).
3. **Descoberta de *clusters* de formatos arbitrários** - muitos algoritmos tendem a encontrar *clusters* esféricos com tamanhos e densidades similares. Outros algoritmos são capazes de detectar *clusters* com formatos aleatórios.
4. **Conhecimento do domínio da aplicação** - muitos algoritmos recebem parâmetros de entrada que dependem de conhecimento do problema abordado. Muitas vezes tais parâmetros são difíceis de serem determinados, exigindo um conhecimento profundo do domínio analisado.
5. **Habilidade em lidar com ruído** - alguns algoritmos são sensíveis a ruídos que, porventura, possam estar presentes no conjunto de dados e podem, conseqüentemente, produzir *clusters* de baixa qualidade.
6. **Usabilidade** - dependendo do algoritmo de *clustering* utilizado, os resultados obtidos podem ser de mais fácil interpretação / utilização por parte do usuário.

Outros aspectos importantes para a comparação entre diferentes algoritmos de *clustering* são:

1. **Crítérios de particionamento** - alguns algoritmos produzem *clusters* de mesmo nível hierárquico, enquanto outros particionam o conjunto de dados em *clusters* de diferentes níveis hierárquicos.
2. **Separação de *clusters*** - algoritmos podem separar os dados entre *clusters* mutuamente excludentes ou entre *clusters* que se intersectam uns com os outros (dados podem pertencer a mais de um grupo ao mesmo tempo).
3. **Medida de similaridade** - algoritmos podem determinar a similaridade entre dados pela distância entre os mesmos enquanto outros podem definir a similaridade dependendo da densidade e continuidade dos dados.

3.2.2 Métodos de *Clustering*

De forma geral, os algoritmos de *clustering* podem ser classificados nas seguintes categorias [10]:

1. **Métodos de particionamento** - também chamado de *partitioning*, consiste em agrupar n objetos em k *clusters*, cada qual contendo pelo menos um objeto. Como ponto de partida, é fornecido como parâmetro de entrada o número de *clusters* presentes. Estes métodos normalmente utilizam uma *técnica de realocação iterativa* no qual objetos são realocados de um grupo para o outro na tentativa de se melhorar o particionamento como um todo. O algoritmo *k-means* pertence a esta categoria.
2. **Métodos hierárquicos** - consiste em criar decomposições hierárquicas de um conjunto de dados. Podem assumir dois tipos de abordagens, *bottom-up* ou *top-down*, dependendo da forma como a estrutura hierárquica é construída, seja a partir de cada objeto individualmente até o conjunto de dados como um todo (*bottom-up*), seja a partir do conjunto de dados como um todo até cada objeto (*top-down*).
3. **Métodos baseados em densidade** - também chamados de *density-based methods* agrupam objetos em *clusters* segundo o conceito de *densidade*, isto é, métrica diretamente relacionada à quantidade de objetos presentes na vizinhança do ponto em questão. Objetos são separados em regiões de baixa e alta densidades classificadas a partir de um determinado *threshold*.

3.2.3 Processo de Delimitação de Aerovias como um Problema de *Clustering*

Como será visto no Capítulo 5, o *processo de delimitação de aerovias* visa produzir um conjunto finito de posições de voo $NORM'$ que representará as aerovias conectando aeroportos de origem e destino. Em outras palavras, as aerovias como regiões contínuas no espaço aéreo são agora representadas por um conjunto finito de posições de voo pertencentes a estas regiões. O problema de delimitação de aerovias passa a ser: como identificar o conjunto de posições de voo $NORM'$ a partir do conjunto de posições rastreadas.

O processo de delimitação de aerovias pode ser modelado como um problema de *clustering* dado que posições de voo podem ser *agrupadas* de forma a representar as fronteiras das aerovias no espaço aéreo conectando aeroportos de origem e destino. As posições de voo *não* pertencentes aos *clusters* obtidos no processo de *clustering* são recorrentes na medida em que as aeronaves desviam suas trajetórias das aerovias no espaço aéreo. Estas posições, portanto, não devem fazer parte das aerovias conectando aeroportos de origem e destino, sendo descartadas durante o processo de delimitação de aerovias / *clustering*.

No presente trabalho, os métodos de *clustering* que melhor se adaptam à solução proposta são os métodos *baseados em densidade*. Estes foram os métodos escolhidos para a formulação da solução do presente trabalho, pois:

1. Não exigem como parâmetro de entrada o número de *clusters* - como será visto no Capítulo 5, o número de *clusters* encontrados pode diferir de execução à execução no algoritmo de *clustering*. Dessa forma, seria inviável utilizar *algoritmos de particionamento*, como *k-means*, que demandam tal parâmetro de entrada, pois seria necessário fazer uma análise manual da quantidade de *clusters* para cada instância de execução do algoritmo.
2. São capazes de encontrar *clusters* em diferentes tamanhos e formatos - os *clusters* formados podem assumir quaisquer formatos, não estando limitados a formatos predeterminados, como no caso de muitos *algoritmos de particionamento* (formatos convexos).
3. Considera a presença de *outliers* - os dados analisados podem conter ruído, introduzido durante o processo de coleta / transformação de dados (processos de rastreamento / normalização de posições de voo explicados no Capítulo 5). Algoritmos baseados em densidade são capazes de fazer a separação de *outliers* na formulação de *clusters*, diferentemente de algoritmos de particionamento e algoritmos hierárquicos.
4. Não existe hierarquia entre os dados analisados - os dados analisados não se relacionam entre si hierarquicamente, como proposto por *métodos hierárquicos*. No entanto, métodos baseados em densidade podem ser combinados com métodos hierárquicos para a criação de algoritmos mais robustos, como por exemplo, *Hierarchical Density-Based Spatial Clustering of Applications with Noise* (HDBSCAN), que será visto mais a frente.

No entanto, apesar dos benefícios listados acima, surgem novos desafios para a viabilidade e implementação dos métodos baseados em densidade:

1. Exigem parâmetros que definem o conceito de densidade - apesar de não precisarem da quantidade de *clusters* como entrada, exigem outros tipos de parâmetros para definir o conceito de *densidade*, e assim separar regiões de baixa e alta densidades.
2. Parâmetros são por vezes sensíveis a pequenos ajustes - dependendo do algoritmo utilizado, podem gerar resultados distintos para pequenas variações dos parâmetros escolhidos, como por exemplo, ajustes na densidade.

Nas próximas seções serão analisados os seguintes algoritmos baseados em densidade: Density-Based Clustering Based on Connected Regions with High Density (DBSCAN) e Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN).

3.2.4 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Density-Based Clustering Based on Connected Regions with High Density (DBSCAN) faz a separação do conjunto de dados em regiões de *alta densidade* (regiões *densas*) e em regiões de *baixa densidade*. Por ser um método baseado em densidade, DBSCAN é capaz de encontrar *clusters* com formatos arbitrários.

A Figura 3.2 mostra um conjunto de pontos apresentando *clusters* de diferentes formatos. A Figura 3.3 ilustra os *clusters* identificados pelo algoritmo DBSCAN utilizando determinados valores para seus parâmetros de entrada (*min_pts* e ϵ , explicados mais a frente). Como é possível perceber, o algoritmo é capaz de identificar *clusters* de formatos variados. Enquanto a Figura 3.4 mostra o mesmo conjunto de dados sendo particionado segundo o algoritmo de *k-means*. Como pode ser visto, o algoritmo não é capaz de identificar *clusters* que não possuam formatos globulares.

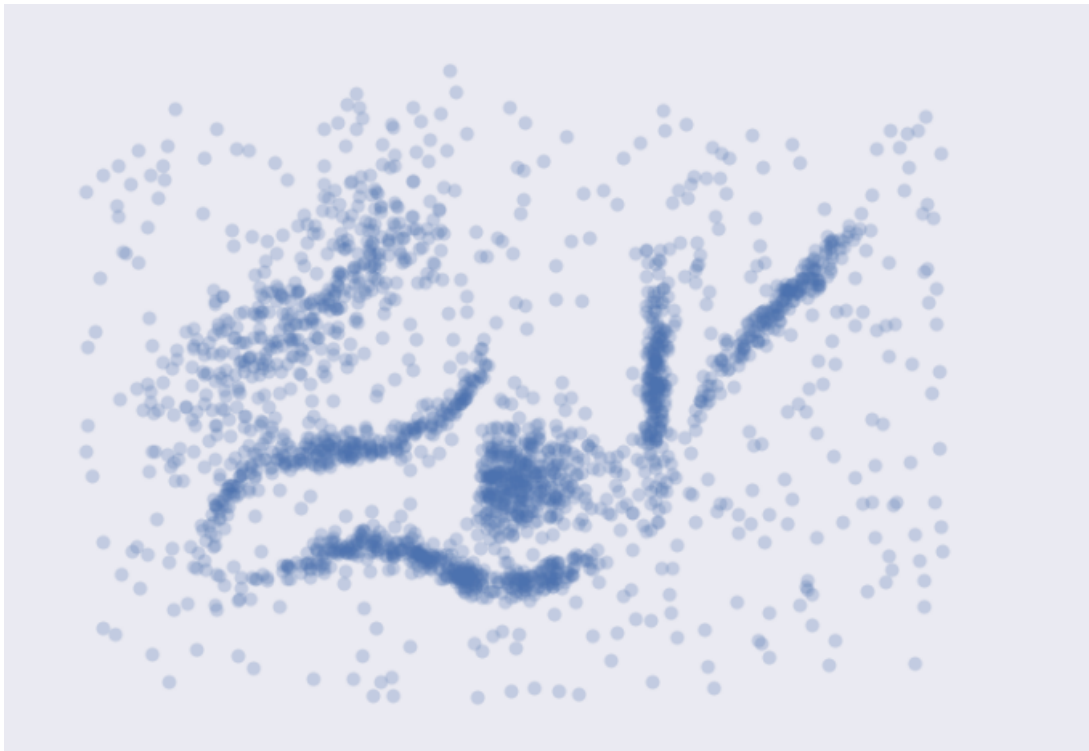


Figura 3.2: *Clusters* com formatos arbitrários (Fonte: [11]).

DBSCAN recebe dois parâmetros como entrada *min_pts* e ϵ , que estão diretamente relacionados ao conceito de *densidade* de um objeto. Enquanto o parâmetro *min_pts* define a quantidade mínima de objetos vizinhos a um dado objeto para que este seja considerado *denso*, o parâmetro ϵ define o raio da região utilizada para se encontrar os objetos vizinhos do objeto em questão. Vale ressaltar que é definido, de forma *implícita*,

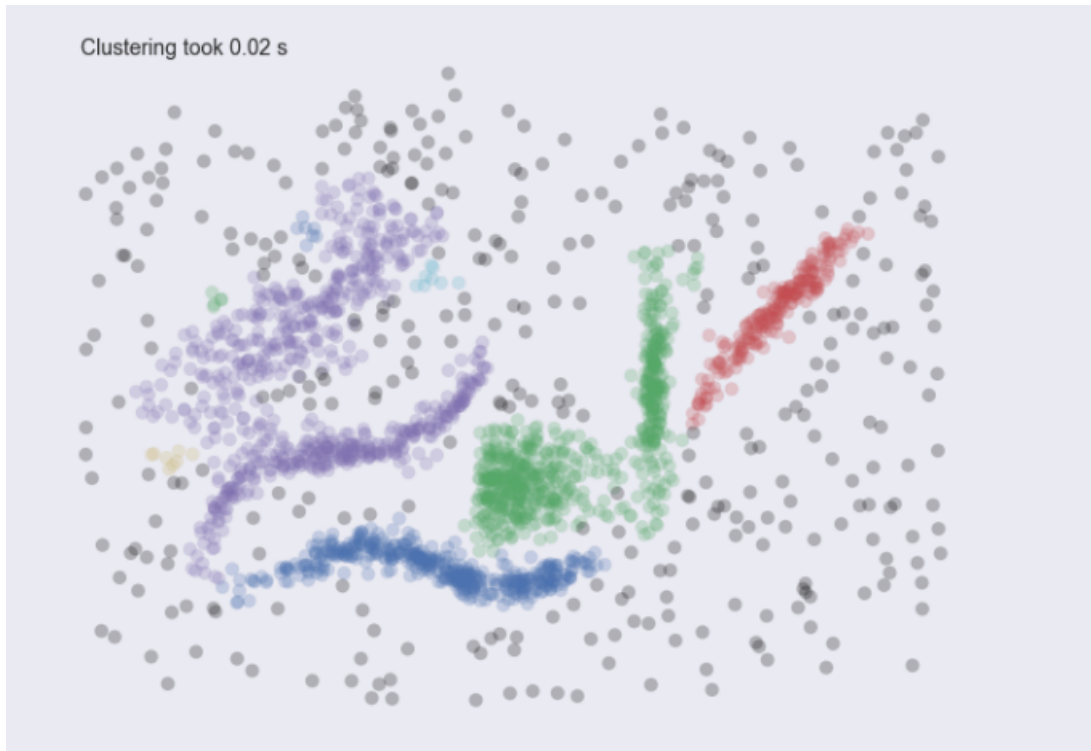


Figura 3.3: *Clusters* encontrados pelo algoritmo DBSCAN (Fonte: [11]).

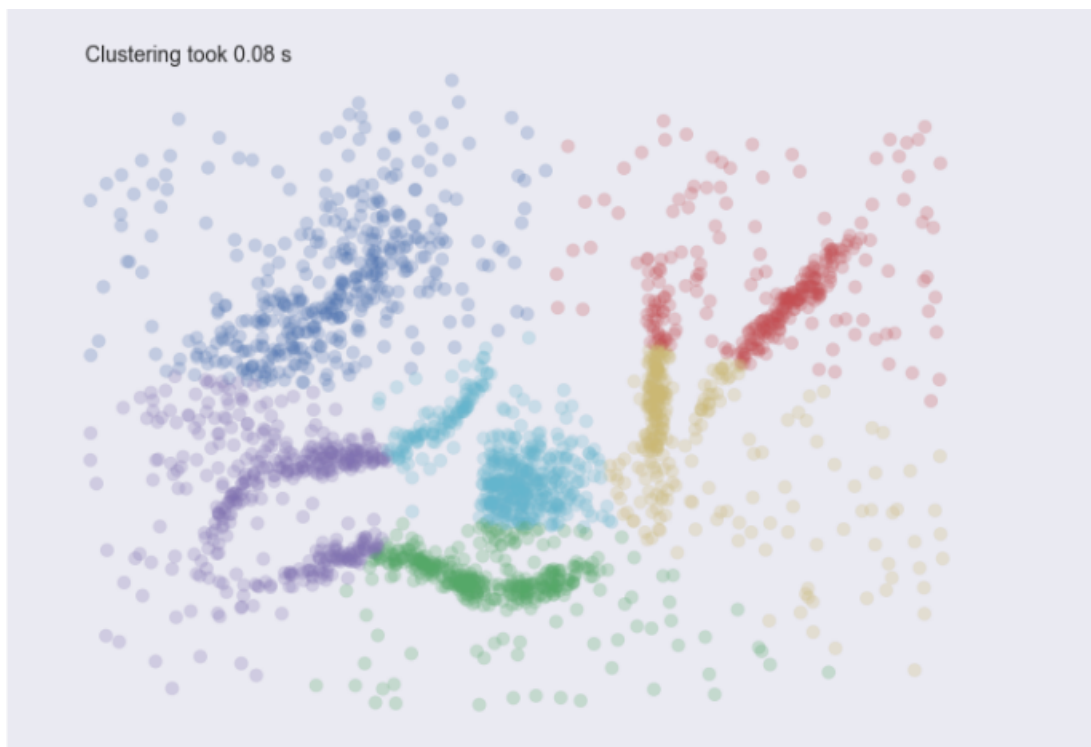


Figura 3.4: *Clusters* encontrados pelo algoritmo K-Means (Fonte: [11]).

o conceito de distância entre dois objetos. A medida de distância utilizada no DBSCAN é representada pelo parâmetro *distance_measure*, que é configurável da mesma forma que os parâmetros descritos acima. Na seção 5.4 são estudadas as diferentes medidas de distância utilizadas na solução proposta.

A vizinhança ϵ -neighborhood do objeto p consiste na região delimitada pelo círculo de centro p e raio ϵ [10]. A *densidade* de p é definida como o número de objetos na vizinhança ϵ -neighborhood do objeto. Dessa forma, objetos podem ser classificados da seguinte forma:

1. *Core Objects* - um objeto p é definido como sendo *core object* se possui pelo menos min_pts objetos em sua vizinhança ϵ -neighborhood.
2. *Non-core Objects* - um objeto p é definido como *non-core object* ou *directly density-reachable* de q se: 1) q é um *core object* e 2) p pertence à ϵ -neighborhood do objeto q [10].
3. *Outliers* - os demais objetos que não são nem *core objects* nem *non-core objects* são classificados como *outliers* (ruído).

Dados o conjunto de objetos p_1, p_2, \dots, p_n , e os parâmetros min_pts e ϵ , que definem os conceitos de densidade e vizinhança ϵ -neighborhood, é possível identificar todos os *core objects* deste conjunto de dados. Estes *core objects* são ditos estarem em regiões de alta densidade.

Um objeto p é definido como *density-reachable* de q , com relação aos parâmetros min_pts e ϵ , se existe uma sequência de objetos p_1, p_2, \dots, p_n tais que $p_1 = q$, $p_n = p$ e p_{i+1} é *directly density-reachable* de p_i para $1 \leq i \leq n$. Dois objetos p_1 e p_2 são definidos como *density-connected*, com relação aos parâmetros min_pts e ϵ , se existir um objeto q tal que ambos p_1 e p_2 são *density-reachable* de q [10]. Agrupando os objetos segundo a relação de *density-connectedness*, são encontrados os *density-based clusters* [10]. Estes conceitos estão representados na Figura 3.5, onde os objetos s e r são *density-connected* e fazem parte do mesmo *cluster*.

De forma geral o algoritmo DBSCAN pode ser executado conforme os passos numerados a seguir [10]:

1. Inicialmente, *todos os objetos* do conjunto de dados \mathcal{O} são marcados como *unvisited*.
2. O algoritmo seleciona de forma aleatória um objeto p de \mathcal{O} marcado como *unvisited*, o algoritmo marca p como *visited* e verifica se o objeto é um *core object*.
3. Se o objeto p não é *core object*, p é classificado como *outlier*. Caso contrário, são criados 1) um novo *cluster* C para p , tendo como primeiro elemento o objeto p ; e 2) um novo conjunto N contendo todos os objetos da ϵ -neighborhood de p .

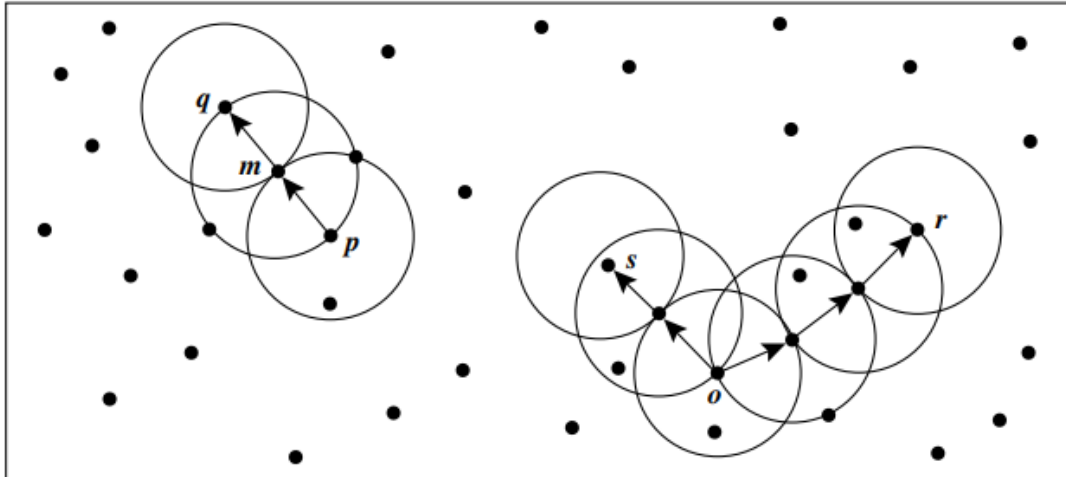


Figura 3.5: Ilustração dos conceitos de *density-reachability* e *density-connectivity* (Fonte: [10]).

4. O algoritmo, então, percorre o conjunto de objetos marcados como *unvisited* em N , marcando estes pontos como *visited* e adicionando suas ϵ -neighborhood ao conjunto N se os objetos são *core objects*. Durante o percorrimento do conjunto N , independentemente do status *visited/unvisited* dos objetos, estes são adicionando a C caso não sejam membros de algum *cluster*.
5. O algoritmo continua o processo acima até o conjunto N se tornar vazio, quando o processo retorna ao passo 2).

Se implementado da forma descrita, o DBSCAN possui complexidade computacional de tempo na ordem de $\mathcal{O}(n^2)$, onde n é a quantidade de objetos do conjunto de dados. No entanto, se implementado utilizando *spatial index*, sua complexidade é reduzida a $\mathcal{O}(n \log n)$ [10].

3.2.5 Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) é uma extensão do DBSCAN utilizada para encontrar *clusters* de densidades variadas. O algoritmo recebe somente um parâmetro de entrada: *min_pts.*, que representa a quantidade mínima de objetos para se formar um *cluster* [12]. Da mesma forma que o DBSCAN, o algoritmo de HDBSCAN define distância entre dois objetos por meio do parâmetro *distance_measure*. As medidas de distância utilizadas na solução proposta são as mesmas do DBSCAN e são estudadas na seção 5.4.

A Figura 3.6 mostra o resultado obtido pelo algoritmo HDBSCAN para o mesmo conjunto de dados da Figura 3.3.

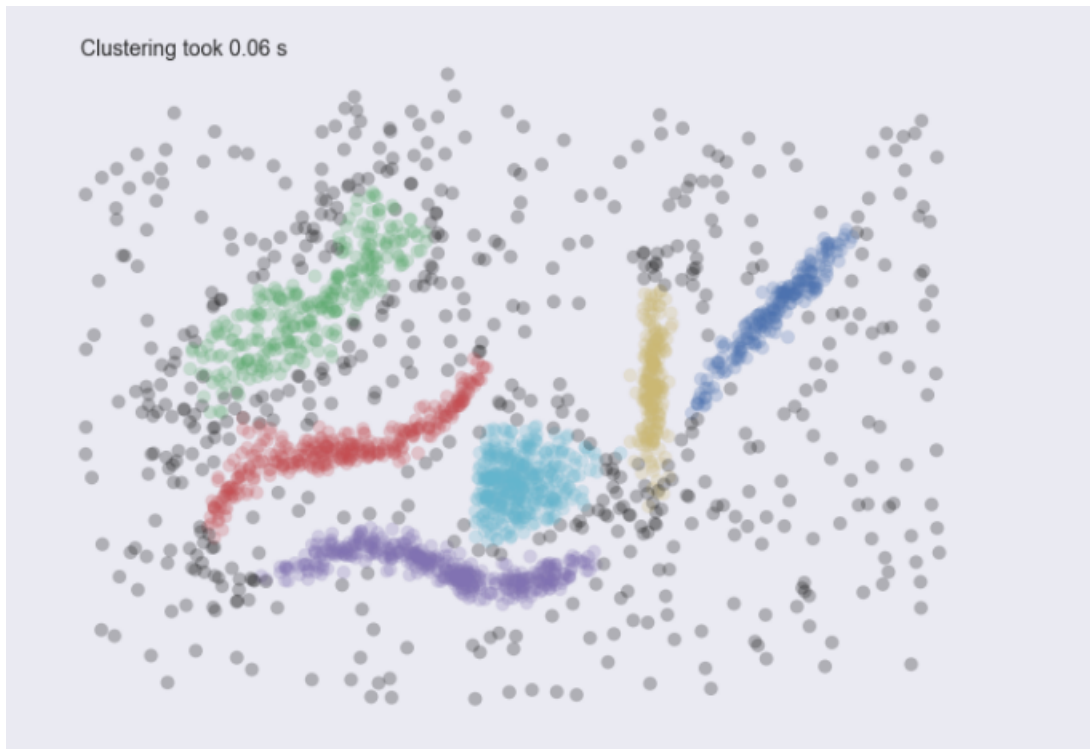


Figura 3.6: *Clusters* encontrados pelo algoritmo HDBSCAN (Fonte: [11]).

Quanto ao seu funcionamento, o algoritmo HDBSCAN utiliza conceitos similares ao do DBSCAN, e seu funcionamento é detalhado a seguir [12]:

1. *Transformação do espaço conforme densidade/dispersão* - para cada objeto p é calculada sua *core distance*, que é o menor valor para o raio com centro em p capaz de incluir min_pts . objetos na vizinhança de p . A partir dessa distância é calculada a *mutual reachability distance* entre dois objetos quaisquer p e q , que é definida como o máximo entre: *core distance* de p , *core distance* de q e distância entre p e q . Esta é a distância utilizada como referência para separar ainda mais objetos localizados em regiões de baixa densidade [12].
2. *Construção da árvore de extensão mínima* - a partir da definição de distância entre dois objetos do passo anterior, o algoritmo constrói a *árvore de extensão mínima* (ou árvore geradora mínima) para o conjunto de objetos, que agora são representados como os vértices do grafo gerado. A árvore de extensão mínima pode ser eficientemente construída utilizando o algoritmo de *Prim* [13].

3. *Construção da hierarquia de cluster* - da mesma forma que métodos hierárquicos, o algoritmo constrói uma *hierarquia de componentes conexos* a partir da árvore geradora mínima.
4. *Redução da hierarquia de cluster* - a partir do parâmetro de entrada *min_pts.*, representando o tamanho mínimo de formação de *cluster*, a hierarquia de *cluster* é transformada numa hierarquia mais simples. Para cada *cluster* da hierarquia de *cluster* obtida do passo anterior, se quaisquer dos *clusters* obtidos da divisão do *cluster* inicial possuir uma quantidade de objetos menor que *min_pts.*, este não será mais considerado como um *cluster* válido. Enquanto que se o *cluster* obtido tiver uma quantidade de objetos maior que *min_pts.*, o processo descrito será repetido recursivamente. Ao final do processo somente *clusters* que possuam uma quantidade superior ou igual a *min_pts.* serão mantidos na hierarquia de *cluster*.
5. *Extração de clusters* - a partir da hierarquia de *cluster* simplificada no passo anterior, é feita a extração de *clusters* segundo a definição de *estabilidade* [12]. O conceito de *estabilidade* é definido de forma a selecionar *clusters* que apresentam, possivelmente, diferentes densidades [12].

HDBSCAN mantém as mesmas vantagens introduzidas pelo DBSCAN, como: separação de regiões de alta e baixa densidades, identificação de *outliers*, e facilidade em sua utilização (isto é, possui parâmetros mais facilmente estimados). No entanto, o algoritmo introduz novos benefícios com relação ao DBSCAN:

1. *Clusters com diferentes densidades* - HDBSCAN é capaz de selecionar *clusters* com diferentes densidades enquanto DBSCAN seleciona apenas *clusters* de mesma densidade.
2. *Redução dos parâmetros de entrada* - Não é mais necessária a utilização do parâmetro ϵ , sendo apenas necessário o parâmetro *min_pts.*, que indica a quantidade mínima de objetos para se formar um *cluster*. Dessa forma, a utilização do algoritmo fica mais intuitiva e fácil para o usuário final.

Da mesma forma que o DBSCAN, dependendo da forma que o HDBSCAN é implementado pode apresentar uma complexidade computacional de tempo na ordem de $\mathcal{O}(n^2)$. No entanto, esta pode ser reduzida para sub- $\mathcal{O}(n^2)$ com a utilização de algoritmos mais avançados [12, 14]. Experimentos mostram que o HDBSCAN possui, em determinados cenários, uma performance superior à do DBSCAN [14].

Capítulo 4

Gerenciamento de Tráfego Aéreo

Este capítulo introduz os principais conceitos e definições relacionados ao Gerenciamento de Tráfego Aéreo para o contexto de planejamento de rotas de aeronaves.

4.1 Sistema ATM

O Sistema de Gerenciamento de Tráfego Aéreo, ou simplesmente Sistema ATM (*Air Traffic Management*), é baseado na provisão de serviços resultantes da ação conjunta de vários recursos, incluindo espaço aéreo, aeródromos, aeronaves, infraestrutura tecnológica e recursos humanos. A função principal do Sistema ATM é possibilitar o voo de um aeródromo a outro, dentro de um determinado espaço aéreo, de forma segura, obedecendo aos limites de capacidade do mesmo e, ao mesmo tempo, fazendo uso ótimo de seus recursos [15].

O Sistema ATM é um sistema de transporte que pode ser classificado como *Complex, Large-Scale, Integrated, Open Systems* (CLIOS) [16]. Nesse sentido, o Sistema ATM é complexo devido ao grande número de componentes e suas interconexões, e devido ao alto número de *stakeholders* envolvidos. Ele é considerado de larga-escala por impactar uma extensa região geográfica. É integrado devido às suas interconexões com outros sistemas de transporte (ferroviário, rodoviário, fluvial, etc.) e com outros sistemas diversos (energia, mudança climática global, etc.). E, por fim, é considerado um sistema aberto por ter seu desempenho fortemente dependente de fatores sociais, políticos e econômicos.

4.1.1 Navegação Aérea

A navegação no espaço aéreo, de forma geral, pode se dar de duas formas: através de *Visual Flight Rules* (VFR), onde cartas de navegação são seguidas de forma visual e o piloto se orienta por referências visuais externas, ou através de *Instrument Flight Rules* (IFR), onde cartas de navegação são seguidas com auxílio de instrumentos aeronáuticos

[17]. Voos IFR são o foco desta pesquisa uma vez que a maioria dos voos comerciais adota o sistema IFR como forma de navegação aérea nos dias de hoje.

Plano de Voo

O plano de voo é um documento que contém as informações específicas, relacionadas a um voo planejado, fornecidas aos órgãos que prestam serviços de tráfego aéreo (*Air Traffic Service* - ATS) [18]. Entre as informações contidas num plano de voo, podem-se destacar [18]:

- **Condições meteorológicas:** previsões meteorológicas dos aeródromos envolvidos e condições meteorológicas da rota aérea a ser utilizada;
- **Consumo de combustível:** cálculo de combustível previsto para o voo;
- **Aeródromo de decolagem e aeródromo de destino:** aeródromo/aeroporto de pouso e decolagem da aeronave;
- **Aeródromos de alternativa de pouso:** em caso de falta de combustível, falha de funcionamento da aeronave, mau tempo, ou caso se torne desaconselhável o pouso no aeródromo de destino;
- **Detalhes da rota:** aerovias, que são rotas pré-determinadas por onde as aeronaves são permitidas a trafegar, utilizadas durante o trajeto da aeronave, além do nível (altitude) correto para a rota a ser voada;
- **Tempo estimado em rota:** tempo de duração previsto do percurso entre aeroporto de origem e aeroporto de destino.

Fases de Voo

As fases operacionais de um voo IFR podem ser representadas pela Figura 4.1 e são descritas como:

- **Fase pré-voo:** consiste nas ações de planejamento do voo;
- **Taxiamento:** a aeronave recebe autorização para se posicionar na cabeceira da pista de decolagem;
- **Decolagem:** a aeronave recebe autorização para decolagem e deixa o solo;
- **Ascendência:** a aeronave sobe até a altitude prevista no plano de voo e dá entrada na aerovia indicada no plano de voo;

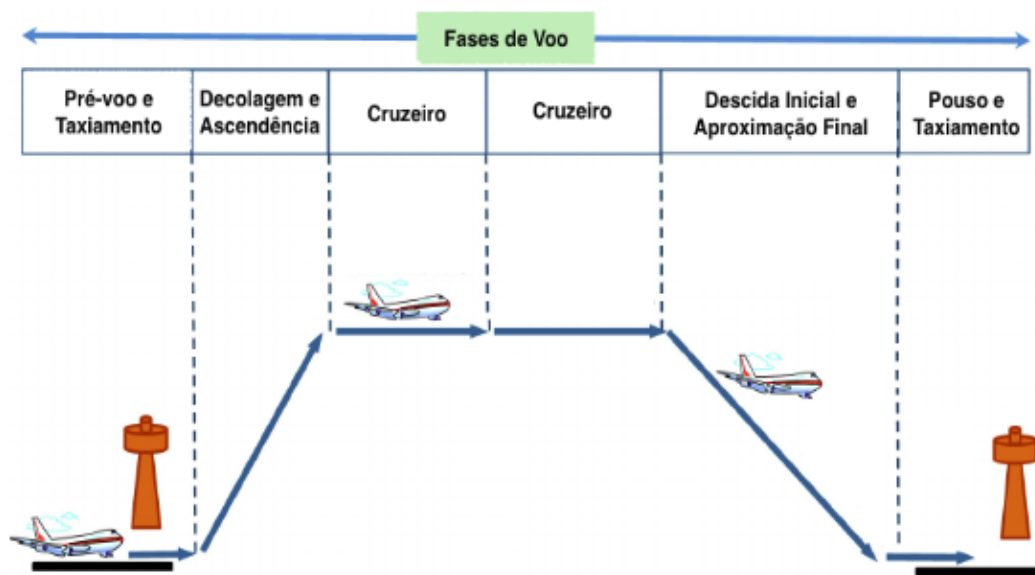


Figura 4.1: Fases de um voo operado por Instrument Flight Rules (IFR) (Fonte: [19]).

- **Cruzeiro:** corresponde ao ajuste da velocidade e altitude da aeronave nas aerovias especificadas no plano de voo;
- **Descida Inicial:** a aeronave desce gradativamente até a região próxima ao aeródromo;
- **Aproximação Final:** a aeronave efetua o procedimento de descida na pista de aterrissagem;
- **Taxiamento:** a aeronave é conduzida ao pátio até sua parada completa.

Este trabalho se concentra principalmente na fase de voo *Cruzeiro*, dado que é a fase onde a *aeronave navega em aerovias no espaço aéreo*. De forma que, o principal objeto de estudo deste trabalho é 1) a identificação dessas aerovias, assim como 2) a interseção das mesmas com células convectivas no espaço aéreo. A fase de *Cruzeiro* é também a fase onde a aeronave passa a maior parte do seu tempo em voo, e se encontra sob uma maior influência de condições meteorológicas adversas.

Aerovias no Espaço Aéreo

As aerovias são rotas aéreas pré-determinadas por onde as aeronaves são permitidas a trafegar no espaço aéreo. Estas rotas se conectam com posições geográficas específicas, chamados de *fixos*, que permitem ao piloto uma navegação segura e precisa [20]. No Brasil, a largura das aerovias pode chegar a 30 km para voos até 7,5 km de altitude

(aproximadamente), ou a 80 km para voos acima de 7,5 km de altitude [20]. Neste estudo são consideradas as aerovias do segundo grupo citado, uma vez que correspondem às utilizadas pela maioria das aeronaves comerciais (aeronaves rastreadas durante o processo de rastreamento da seção 5.2.1). As aerovias representam, portanto, uma região de volume considerável do espaço aéreo ao invés de simples retas imaginárias. São, no entanto, *corredores retos e diretos* conectando dois pontos fixos.

Uma aerovia pode ser utilizada, ao mesmo tempo, por duas ou mais aeronaves que sobrevoem o espaço aéreo seja no mesmo sentido, seja em sentidos contrários. No entanto, caso as aeronaves sobrevoem em sentidos contrários, estas não podem trafegar na mesma altitude. A separação entre aeronaves que trafegam em sentidos opostos é feita por meio da divisão vertical da aerovia em *flight levels* [21]. Estes *flight levels* mantêm uma distância vertical mínima de 1.000 pés (ou 305 metros), que garantem a segurança a bordo ao manter a separação vertical das aeronaves. Na Figura 4.2 é possível ver a distribuição dos *flight levels* e seus sentidos de voo num espaço RVSM [22]. Reduced Vertical Separation Mínima (RVSM), traduzido como Mínimo de Separação Vertical Reduzido, é o nome do protocolo desenvolvido pela ICAO de separação vertical entre aeronaves para altitudes entre o FL290 e o FL410. Vale ressaltar que o *flight level* FL290 corresponde a uma altitude de 8,8 km enquanto o *flight level* FL410 a 12,5 km, que é basicamente o alcance da aviação civil [21].

Considerando a definição de uma aerovia desenvolvida acima, a informação de *altitude* de uma dada aeronave serve apenas para especificar em qual *level flight* a aeronave se encontra. Enquanto, para saber em *qual aerovia* uma aeronave se encontra, basta conhecer suas coordenadas geográficas de *latitude* e *longitude*. Como será visto na seção 5.4, esse conhecimento será imprescindível para a definição de uma *medida de distância entre pontos de voo* de forma apropriada, que é a fórmula *Haversine* (explicada na seção 5.4.2).

4.1.2 Meteorologia Aeronáutica

No Brasil, a atividade de Meteorologia Aeronáutica é de competência do Comando da Aeronáutica (COMAER), sob a responsabilidade do Departamento de Controle do Espaço Aéreo (DECEA) [23].

A informação meteorológica é imprescindível para a segurança das operações aéreas, facilitando o planejamento de rotas mais rápidas e econômicas, e de voos regulares [24]. Com o crescimento do fluxo de tráfego aéreo e o conseqüente aumento da necessidade de um melhor aproveitamento do espaço aéreo, as informações meteorológicas passam a ser ainda mais importantes para o controle do espaço aéreo [24].

Dentre as atribuições da Meteorologia Aeronáutica estão a coleta de dados de estações e radares meteorológicos, a divulgação desses dados aos *stakeholders* ATM e a interpretação

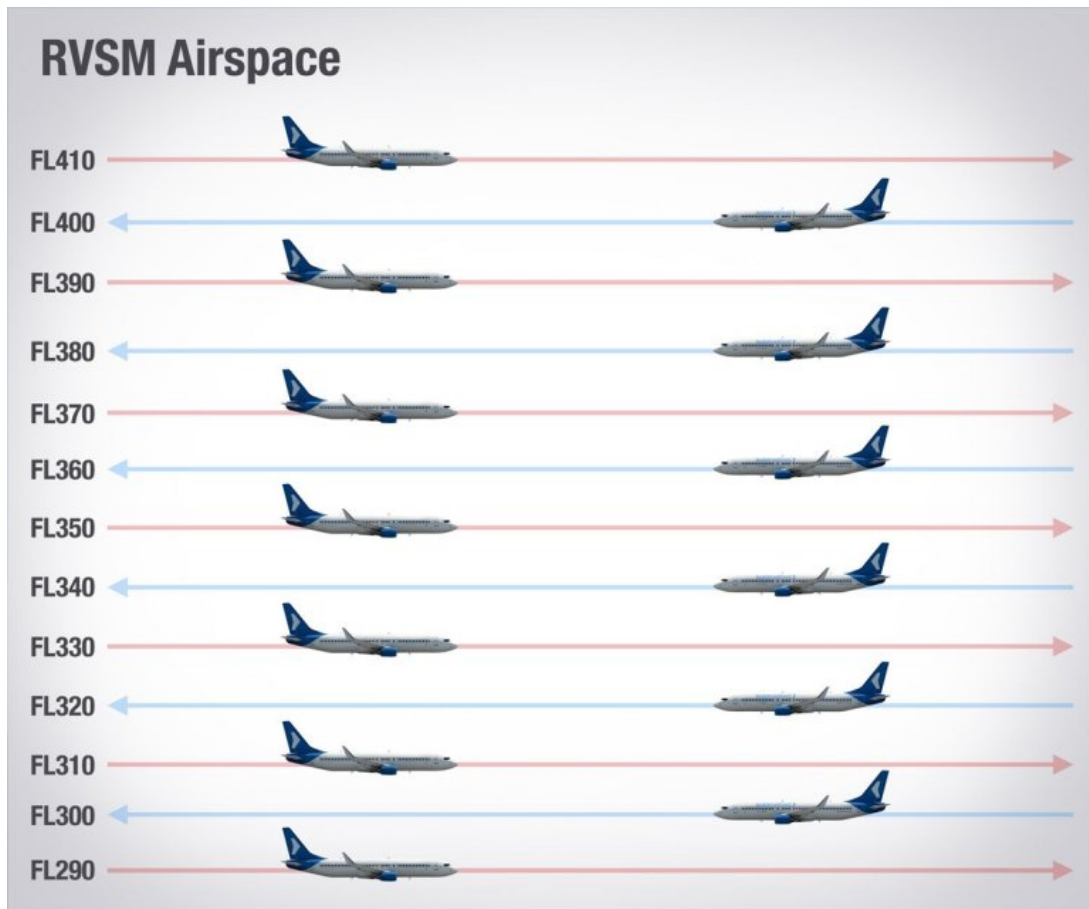


Figura 4.2: Distribuição de *flight levels* num espaço RSVM (Fonte: [22]).

dos mesmos, produzindo previsões meteorológicas específicas para regiões próximas a aeródromos e para rotas aéreas [23]. O conhecimento das condições meteorológicas nos aeródromos de partida, destino e alternativas são de suma importância para a realização ou não do voo. Ter conhecimento das condições meteorológicas em rota é também essencial para um voo seguro, econômico e confortável [23].

REDEMET e Sistema de Tempo Severo Convectivo

A Rede de Meteorologia do Comando da Aeronáutica (REDEMET) é a organização responsável por integrar os diversos produtos meteorológicos voltados à aviação e aos *stakeholders* ATM. A REDEMET e seus produtos estão subordinados ao Comando da Aeronáutica (COMAER) [25].

Entre os produtos disponíveis na REDEMET está o Sistema de Tempo Severo Convectivo (STSC) [26]. O Sistema de Tempo Severo Convectivo é destinado à detecção e ao monitoramento de células convectivas em seu estágio mais ativo: nuvens *cumulonimbus*,

responsáveis por impactar de forma considerável o espaço aéreo e a operação das aeronaves em voo [1].

As nuvens *cumulonimbus* são de forte desenvolvimento vertical, que podem atingir mais de quinze quilômetros de altura. São comumente associadas a graus severos de turbulência e correntes de ventos, sendo assim, motivo de preocupação para aeronavegantes [3]. As nuvens *cumulonimbus* são frequentes e podem ser formar em qualquer época do ano desde que existam condições atmosféricas adequadas.

O Sistema de Tempo Severo Convectivo destaca as áreas de ocorrências de tempo severo convectivo, nuvens *cumulonimbus*, com base no processamento e análise dos dados de radares meteorológicos e do Sistema de Detecção de Descargas Atmosféricas. O Sistema de Tempo Severo Convectivo é um integrador dessas informações, que por sua vez são empregadas em diversas aplicações do Sistema de Controle do Espaço Aéreo Brasileiro (SISCEAB) [2].

Desvio de Rota em Voo

As nuvens *cumulonimbus*, quando em seu estágio mais intenso, apresenta grande perigo para a aviação devido à ocorrência de pancadas de chuva, acompanhadas às vezes de granizo, além de correntes descendentes, turbulência e rajadas severas de vento [3]. Nuvens *cumulonimbus* representam um alto risco para a segurança em voo e até mesmo causar acidentes aéreos. Dessa forma, realizar o desvio de trajetória frente às nuvens *cumulonimbus* é imprescindível para a segurança em voo.

Um caso conhecido no Brasil, por exemplo, é o voo AF447 da *Air France* que partiu do Rio de Janeiro com destino à Paris no dia 1º de junho de 2009. Guiado pelo radar meteorológico de bordo, o comandante decidiu passar em meio a nuvens *cumulonimbus* a 11,5 km de altitude posicionada na rota do voo, em vez de mudar a trajetória da aeronave. As nuvens teriam causado o congelamento dos *pitots*, responsáveis por informações de velocidade do voo, seguido de uma série de erros que ocasionaram na morte das 228 pessoas a bordo [27].

As nuvens *cumulonimbus* são facilmente detectadas pelo radar meteorológico de bordo das aeronaves, em tempo hábil para permitir manobras de desvio [3]. Depois da identificação de nuvens *cumulonimbus* por parte do radar meteorológico, o desvio de rota deve ser autorizado pelo Controle de Tráfego Aéreo, devendo a aeronave fazer o retorno à aerovia de seu plano de voo [28]. Este é o procedimento atualmente empregado para a realização de desvios de obstáculos no espaço aéreo brasileiro. Pode ser realizado com a mudança de altitude da aeronave ou com a mudança de direção horizontal, sendo esta última a forma de desvio mais comum, principalmente frente a nuvens *cumulonimbus* como obstáculo.

A solução proposta tem por objetivo prover às partes interessadas informações relevantes quanto à posição de células convectivas, de forma a antecipar ou evitar os procedimentos de desvio de rotas de aeronaves em voo. Ou melhor, pretende servir como interface de apoio fornecendo informações meteorológicas de forma oportuna aos *stakeholders* ATM para a tomada de decisões.

4.2 Modernização do Sistema de Gerenciamento de Tráfego Aéreo

O aumento da demanda por tráfego aéreo é uma realidade constante encontrada em praticamente todas as partes do mundo, o que requer o desenvolvimento de novas soluções para o Sistema ATM, em especial ao gerenciamento do fluxo de tráfego aéreo, conforme descrito no *Global Air Navigation Plan* (GANP) da *International Civil Aviation Organization* (ICAO) [29]. Além disso, a comunidade ATM é diretamente dependente do recebimento de informações em tempo oportuno, relevantes, precisas e confiáveis, colaborando e orientando a tomada de decisões. Nesse contexto, o amplo compartilhamento de informações permite que a comunidade ATM conduza suas operações de forma segura e eficiente [15].

Diante desse cenário, diferentes programas ao redor do mundo entraram em execução com o objetivo de aprimorar soluções de navegação aérea, destacando-se: o *Single European Sky ATM Research* (SESAR) na Europa e o *Next Generation Air Transportation System* (NextGen) nos EUA. No Brasil, existe o Programa SIRIUS para modernização do Sistema ATM no Brasil.

Em comum, tais programas buscam promover a evolução do Gerenciamento de Tráfego Aéreo, convergindo para um ambiente automatizado, integrado e interoperável. Como consequência desta modernização do Sistema ATM, os usuários serão beneficiados, em termos de maior segurança, eficiência e economia, ao mesmo tempo em que benefícios ambientais serão gerados pela redução de queima de combustível e a consequente diminuição na emissão de gases poluentes na atmosfera.

4.2.1 System-Wide Information Management

Com o objetivo de fomentar a modernização do Sistema ATM, a *International Civil Aviation Organization* (ICAO), entidade internacional responsável pelo estabelecimento de padrões e práticas recomendadas para o setor da aviação civil em todo o mundo, propôs o conceito *System-Wide Information Management* (SWIM) por meio do qual os serviços ATM serão integrados, com uso de uma infraestrutura e de um conjunto de padrões e políticas comuns, gerando assim, maior interoperabilidade entre serviços ATM [30].

A implantação do SWIM contribui na modernização do Sistema ATM de diversas formas, destacando-se [31]:

- Melhoria na tomada de decisões dos *stakeholders* ATM durante todas as fases estratégicas e táticas do voo.
- Melhoria na gestão da informação e, conseqüentemente, do compartilhamento destas informações.
- Mais flexibilidade, simplicidade e economia nas comunicações devido a aplicação de padrões comuns de troca de informações.

4.2.2 Programa SIRIUS

No Brasil, o Departamento de Controle do Espaço Aéreo (DECEA), organização subordinada ao Comando da Aeronáutica (COMAER) que tem por finalidade planejar, gerenciar e controlar as atividades relacionadas ao controle do espaço aéreo, desenvolve o Programa SIRIUS para modernização do Gerenciamento de Tráfego Aéreo. O Programa SIRIUS desenvolve, por meio do emprego de soluções de alta tecnologia, capacitação de recursos humanos e promoção da redução dos custos operacionais, a evolução do Sistema de Gerenciamento de Tráfego Aéreo (ATM) no Brasil [32]. O programa visa os seguintes benefícios para a Comunidade ATM [33]

- **Operadoras de aeronaves e seus pilotos:** maior equidade no acesso ao espaço aéreo, maior disponibilidade de informações oportunas e pertinentes, em apoio às decisões, além da maior participação na adoção dessas decisões.
- **Provedores de serviço de Navegação Aérea:** capacidade de funcionar em um ambiente que proporcione grande volume de informação (dados em tempo real, dados sobre tendências e prognósticos do Sistema ATM), unido a uma gama de recursos automatizados para apoiar ou adotar decisões, que elevarão a capacidade dos provedores de aumentar continuamente a qualidade dos serviços prestados a todos os operadores de aeronaves e seus pilotos.
- **Autoridades reguladoras:** sistemas relacionados à segurança operacional deverão ser robustos e abertos, possibilitando que os níveis de segurança sejam medidos e supervisionados com mais facilidade, assim como comparados e integrados em escala regional e mundial.
- **Passageiros:** modernização e a implantação de novos sistemas e equipamentos, bem como a aplicação de novos procedimentos, proporcionarão aos passageiros maior

segurança, menor duração dos tempos de voo, maior confiabilidade nos horários previstos para as partidas e chegadas, entre outros.

- **Meio Ambiente:** implantação e utilização de rotas aéreas mais diretas, realização de procedimentos de subidas e descidas contínuas, utilização de altitudes de voo ideais, menores tempo de espera em pista, entre outros.
- **Instituições de pesquisa e desenvolvimento:** novos estudos a serem desenvolvidos mediante a estudos relacionados à análise de dados de interesse aeronáutico para o desenvolvimento de ferramentas específicas para o gerenciamento do tráfego aéreo, automatização ATM, entre outros.

Considerando alguns dos empreendimentos realizados pelo Programa SIRIUS nas áreas de Gerenciamento de Tráfego Aéreo e Meteorologia Aeronáutica, podem-se listar:

- **Uso flexível do espaço aéreo:** visa à otimização, à flexibilização e ao equilíbrio no uso do espaço aéreo entre seus diversos usuários por meio da coordenação entre todos os interessados [34].
- **Implementação de Gerenciamento do Fluxo de Tráfego Aéreo:** visa à implantação de ferramentas, processos e procedimentos destinados a assegurar que a capacidade do Sistema ATM seja plenamente utilizada, com o intuito de garantir um fluxo de tráfego aéreo seguro, ordenado e rápido. A aplicação de medidas de regulação do tráfego é limitada ao estabelecimento do equilíbrio entre demanda e capacidade, especialmente nos casos de redução da capacidade do sistema, devido, por exemplo, às condições meteorológicas adversas ou a problemas temporários na infraestrutura necessária ao Gerenciamento de Tráfego Aéreo (ATM) [35].
- **Evolução do Serviço de Informação de Voo:** visa à implantação de melhorias na provisão dos serviços de Informação de Voo e de Alerta no espaço aéreo brasileiro [36].
- **Coleta de dados sobre o Ambiente Meteorológico:** visa ao incremento da representatividade das informações meteorológicas no ambiente operacional aeronáutico, de modo a suportar, com qualidade, as atividades de gerenciamento e controle de tráfego aéreo, permitir o conhecimento oportuno de condições meteorológicas adversas às tripulações, o conhecimento dos níveis de voo e rotas mais favoráveis, de forma dinâmica, amigável e integrada ao Sistema de Gerenciamento de Tráfego Aéreo [37].
- **Tratamento de dados meteorológicos e integração de produtos meteorológicos ao Sistema ATM nacional:** visa à implantação de diversos sistemas

necessários ao processamento e à transmissão dos dados coletados sobre o ambiente meteorológico, objetivando a elaboração de produtos para aplicação operacional e o incremento da capacidade de detectar, medir e acompanhar fenômenos de tempo severo, de modo que os controladores de tráfego aéreo tenham capacidade de redirecionar as aeronaves, em tempo hábil, para evitar fenômenos meteorológicos perigosos, além de facilitar o estabelecimento de um fluxo do tráfego aéreo adequado nas áreas de controle terminal com alta densidade de tráfego [38].

Dos objetivos e empreendimentos listados acima, pode-se notar que todos vão ao encontro com o objetivo desta pesquisa. Atendendo a todos esses requisitos, o desenvolvimento desta e sua implantação no Sistema de Gerenciamento de Tráfego Aéreo nacional pode gerar um impacto positivo em um número considerável de *stakeholders* ATM.

Capítulo 5

Modelagem e Implementação

Neste capítulo serão apresentados conceitos gerais da solução como: definição da arquitetura geral, entradas e saídas, fontes de dados, organização dos componentes, relação entre componentes, etc. Os componentes da solução serão detalhados conforme suas modelagens e implementações.

A solução proposta pode ser subdividida em duas grandes etapas. Na primeira, dados de rastreamento de aeronaves em voo são utilizados na identificação das fronteiras que delimitam as aerovias no espaço aéreo. Na segunda etapa, as localizações das aerovias obtidas na etapa anterior são comparadas com as localizações das células convectivas em busca de possíveis interseções entre as mesmas. As interseções identificadas são, então, utilizadas na fase de planejamento de voo pelas partes interessadas.

5.1 Arquitetura Proposta

No que se refere à arquitetura da solução, o sistema obtém como entrada: informações de rastreamento de voo, informações meteorológicas *em tempo real* e os aeroportos de origem e destino; e gera como saída uma lista contendo as células convectivas que intersectam alguma aerovia ligando os aeroportos de origem e destino. A solução proposta segue os seguintes passos para medir o impacto do tempo severo convectivo nas operações ATM:

1. **Fontes de Dados:** levantamento de dados de posições de voo de aeronaves e levantamento de dados de células convectivas no espaço aéreo.
2. **Normalização de Dados:** normaliza as posições de voo de aeronaves utilizando *Interpolação Linear*.
3. **Delimitação de Aerovias:** identificação de um conjunto de posições que delimitam as fronteiras das aerovias via algoritmos *clustering* das posições de voo normalizadas.

4. **Busca por Interseções:** pesquisa por interseções entre aerovias e células convectivas através da comparação entre as posições das fronteiras das aerovias com as posições das células convectivas presentes no espaço aéreo.

A visão geral do sistema é ilustrada na Figura 5.1 e cada um dos componentes é discutido a seguir.

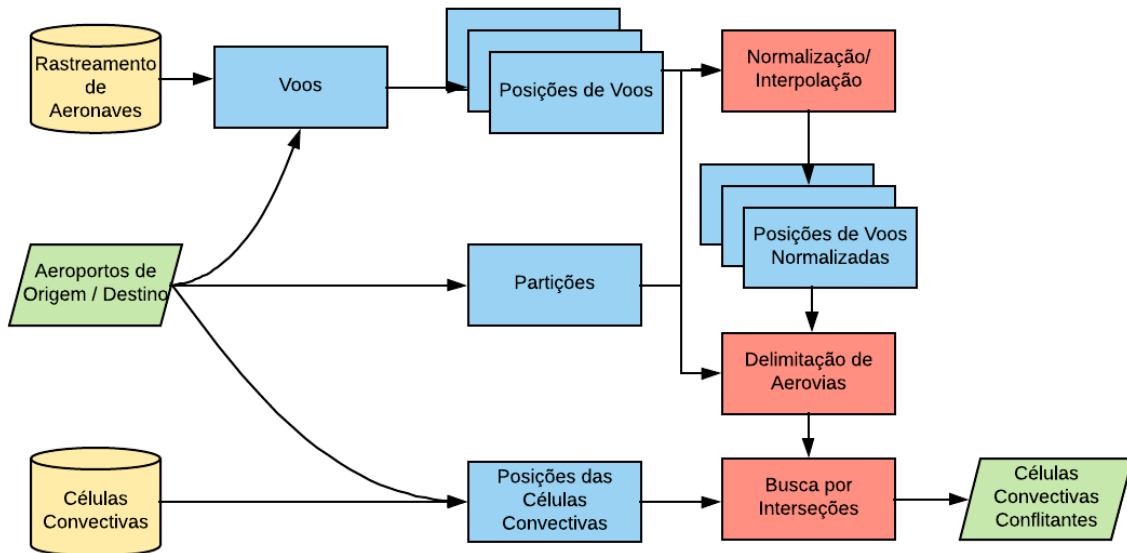


Figura 5.1: Visão geral do sistema. As informações de entrada / saída estão na cor verde, o armazenamento de dados está em amarelo, os dados recuperados / temporários estão em azul e os algoritmos propostos estão coloridos em vermelho..

5.2 Fontes de Dados

Serão utilizadas duas fontes de dados diferentes: posições de voo de aeronaves e posições de células convectivas em tempo real.

5.2.1 Rastreamento de Aeronaves

Para acompanhar trajetórias de aeronaves em tempo real, existem várias plataformas no mercado formadas de redes distribuídas e de colaboração coletiva de receptores ADS-B. *Automatic Dependent Surveillance-Broadcast* (ADS-B) é uma tecnologia de vigilância automática e cooperativa para rastreamento de aviões em tempo real. Aeronaves determinam suas posições através de navegação via satélite e emitem os dados coletados através de uma frequência de rádio. Essas informações podem, então, ser recebidas via estações

de Controle de Tráfego Aéreo ou por outras aeronaves, desde que possuam receptores ADS-B e estejam dentro de determinado alcance.

Radares e dispositivos ADS-B permitem a visualização de movimentos de aeronaves no espaço aéreo [39]. Ao rastrear a localização das aeronaves e estimar seus posicionamentos futuros, o Controle de Tráfego Aéreo pode atuar na garantia da segurança das operações no espaço aéreo. O sistema ADS-B proporciona melhorias determinantes para a vigilância aérea ao permitir um maior número de amostras e mais parâmetros sobre as aeronaves do que é convencionalmente possível obter com o radar secundário [40]. Dispositivos ADS-B são de menor custo de aquisição e manutenção, e mais eficazes em áreas de terrenos montanhosos, onde a cobertura por radar é limitada ou inexistente [40].

As informações das aeronaves em voo são enviadas constantemente para estas redes distribuídas de receptores ADS-B que trabalham de forma colaborativa. Dessa forma os dados podem ser acessados de forma livre ou não dependendo da plataforma utilizada, através de uma *Application Program Interface* (API) para consulta destas informações. Listando algumas plataformas de rastreamento de aeronaves em tempo real, têm-se: *Flightradar24* [41], *OpenSky Network* [42] e *FlightAware* [43].

O trabalho atual é implementado utilizando *OpenSky Network* uma vez que esta oferece uma API de acesso gratuito. As outras plataformas listadas acima oferecem somente serviços pagos. No entanto, as informações comuns a todas estas plataformas são:

- **Status do voo:** número do voo; companhia aérea; aeródromo/aeroporto de origem, aeródromo/aeroporto de destino; horário do voo, horário estimado de chegada, horário real de partida; tempo médio de voo; atraso médio de chegada; distância do voo; etc.;
- **Detalhes da aeronave:** identificador único da aeronave; tipo da aeronave; tempo de vida da aeronave; etc.;
- **Detalhes do voo** - posições de voo da aeronave em latitude, longitude e altitude; direção da aeronave; velocidade em relação ao solo; etc.

A Figura 5.2 mostra o rastreamento de aeronaves no cenário global, onde cada ícone amarelo representa uma aeronave em voo:

5.2.2 Tempo Severo Convectivo

Como mencionado no Capítulo 4, a Rede de Meteorologia do Comando da Aeronáutica é a organização responsável por oferecer produtos meteorológicos voltados aos *stakeholders* ATM. Um dos produtos disponíveis de acesso público é o Sistema de Tempo Severo Convectivo (STSC), que destaca as áreas de ocorrências de tempo severo convectivo,

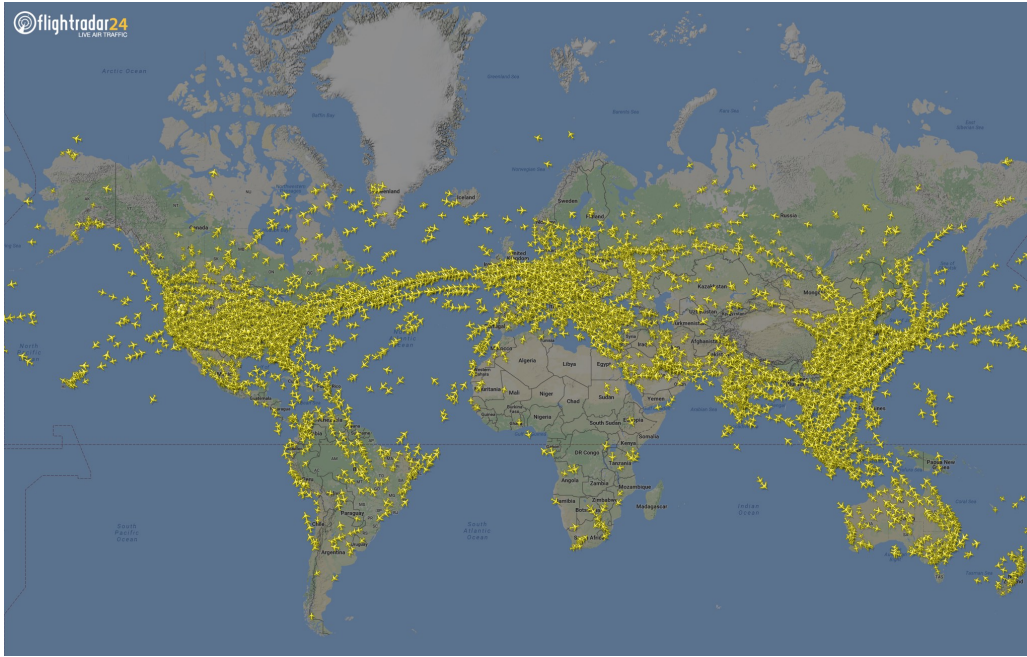


Figura 5.2: Rastreamento de aeronaves no contexto global pelo Flightradar24 (Fonte: [44]).

composto pelas nuvens *cumulonimbus*, com base no processamento e análise dos dados do Sistema de Radares Meteorológicos e do Sistema de Detecção de Descargas Atmosféricas [25].

No STSC, cada célula convectiva é representada por um círculo, sendo dessa forma, necessárias a posição de seu centro e a medida de seu raio. Dessa forma, o sistema fornece para cada célula convectiva: posição geográfica de seu centro (latitude e longitude), raio da célula convectiva, horário e fonte originadora (Radar e/ou DEA - Descargas Elétricas Atmosféricas) [1]. Pela interface web do Sistema de Tempo Severo Convectivo [2], é possível visualizar as células convectivas numa região que abrange todo o espaço aéreo brasileiro e sul americano. As células convectivas estão representadas nas Figuras 5.3 a 5.4, onde cada região vermelha corresponde a uma circunferência com diâmetro de aproximadamente 11 km.

5.2.3 Outras Fontes de Dados

São necessárias outras informações sobre voos/aeroportos para implantar a arquitetura proposta neste trabalho. É necessário conhecer as posições dos aeroportos de origem e destino para prosseguir com o processo de normalização de voo (explicado na seção 5.3).

Dessa forma, foram levantadas as informações dos 47 principais aeroportos do território brasileiro: Aeroporto Santa Maria (SBAR), Aeroporto Internacional Val de Cans/Júlio Cezar Ribeiro (SBBE), Aeroporto Pampulha - Carlos Drummond de Andrade (SBBH),

Aeroporto Internacional Presidente Juscelino Kubistschek (SBBR), Aeroporto las Brasil Cantanhede (SBBV), Aeroporto Internacional Tancredo Neves (SBCF), Aeroporto Campo Grande (SBCG), Aeroporto Bartolomeu Lysandro (SBCP), Aeroporto Afonso Pena (SBCT), Aeroporto Hugo Cantergiani Regional (SBCX), Aeroporto Marechal Rondon (SBCY), Aeroporto Internacional Eduardo Gomes (SBEG), Aeroporto Internacional Cataratas (SBFI), Aeroporto Internacional Hercílio Luz (SBFL), Aeroporto Internacional Pinto Martins (SBFZ), Aeroporto Internacional Rio Galeão – Tom Jobim (SBGL), Aeroporto Santa Genoveva (SBGO), Aeroporto Internacional Guarulhos - Governador André Franco Montoro (SBGR), Aeroporto Bahia - Jorge Amado (SBIL), Aeroporto Usiminas (SBIP), Aeroporto Francisco de Assis (SBJF), Aeroporto Internacional Presidente Castro Pinto (SBJP), Aeroporto Lauro Carneiro de Loyola (SBJV), Aeroporto Internacional Viracopos (SBKP), Aeroporto Governador José Richa (SBLO), Aeroporto Zumbi dos Palmares (SBMO), Aeroporto Alberto Alcolumbre (SBMQ), Aeroporto Internacional Ministro Victor Konder (SBNF), Aeroporto Internacional Governador Aluizio Alves (SBSG), Aeroporto Salgado Filho (SBPA), Aeroporto Prefeito Doutor João Silva Filho (SBPB), Aeroporto Governador Jorge Teixeira de Oliveira (SBPV), Aeroporto Plácido de Castro (SBRB), Aeroporto Internacional Guararapes - Gilberto Frey (SBRF), Aeroporto Santos Dumont (SBRJ), Aeroporto Leite Lopes (SBRP), Aeroporto Internacional Marechal Cunha Machado (SBSL), Aeroporto Congonhas (SBSP), Aeroporto Internacional Deputado Luiz Eduardo Magalhães (SBSV), Aeroporto Senador Petrônio Portela (SBTE), Aeroporto Ten. Cel. Aviador César Bombonato (SBUL), Aeroporto Mário de Almeida Franco (SBUR), Aeroporto Eurico de Aguiar Salles (SBVT), Aeroporto Porto Seguro (SBPS), Aeroporto Brigadeiro Lysias Rodrigues (SBPJ), Aeroporto Cabo Frio (SBCB) e Aeroporto Romeu Zema (SBAX).

Para cada aeroporto, têm-se as seguintes informações:

- **Código Aeroportuário ICAO:** composto por quatro letras usado para identificar aeroportos em todo mundo. Este código é definido pela *International Civil Aviation Organization* (ICAO), agência especializada das Nações Unidas que conta com 193 países-membros.
- **Código Aeroportuário IATA:** composto por três letras também utilizado para identificar aeroportos em todo mundo. No entanto, é definido pela *International Air Transport Association* (IATA), associação comercial de companhias aéreas que conta com 290 companhias aéreas, operando em 117 países.
- **Posição Geográfica:** latitude, longitude e altitude do aeroporto/aeródromo.

Observe que são mantidos os códigos aeroportuários tanto ICAO quanto IATA. Assim, o sistema proposto é capaz de descobrir aeroportos de origem e destino independentemente

Tabela 5.1: Quantidade de números de voo (*callsigns*) por companhia aérea.

Companhia Aérea	<i>Callsigns</i> Rastreados
GOL	618
TAM	529
AZUL	501
Avianca	162

da fonte de rastreamento de dados utilizar o primeiro e/ou segundo código de identificação de aeroportos. Observe também que são considerados apenas os voos que operam entre os aeroportos em questão, isto é, os voos devem possuir aeroportos de origem e destino entre a lista de aeroportos.

Além do mais, dependendo da fonte de dados utilizada, sequer são retornadas as informações sobre aeroportos de origem / destino (este é o caso para a fonte de dados utilizada, *OpenSky Network*). No entanto, neste caso, é possível obter aeroportos de origem / destino através do número de voo obtido pelo rastreamento de voo. Dessa forma, para oferecer uma solução robusta, que independe da fonte de dados utilizada para rastreamento de aeronaves, é mantida uma fonte de dados adicional que *associa* número de voo (chamado também de *callsign*) a aeroportos de origem / destino. Assim, são rastreados apenas os voos que possuam *callsign* presente nesta fonte de dados.

A partir do *callsign* *GLO2038*, por exemplo, é possível obter: 1) informações sobre a companhia aérea (*Gol Transportes Aéreos*) e 2) aeroportos de origem (*Rio Galeão – Aeroporto Internacional Tom Jobim*) e destino (*Aeroporto Internacional Presidente Juscelino Kubistschek*). Observe que frequentemente *mais de um voo* num dado período está associado a um *callsign*, como é o caso para o *callsign* *GLO2038*, que normalmente está agendado para todos os dias da semana num horário padrão.

Esta fonte de dados adicional conta com 1.810 *callsigns* habilitados de serem rastreados pelo sistema, sendo composta de 618 da Gol Transportes Aéreos, 529 da TAM, 501 da Azul Linhas Aéreas Brasileiras e 162 da Avianca Brasil, como mostrado na Tabela 5.1. A Tabela 5.2 mostra os 10 trechos mais comuns conectando aeroporto de origem e aeroporto de destino segundo a quantidade de *callsigns* habilitados para serem rastreados.

5.3 Normalização de Posições de Voo

Após processo de rastreamento de aeronaves, as posições de voo de aeronaves que navegaram nos *mesmos aeroportos de origem e destino* são agrupadas a fim de se identificar o conjunto de pontos que delimitam as aerovias conectando estes dois aeroportos. O *processo de normalização* (ver Figura 5.1) consiste na geração de um novo conjunto de

Tabela 5.2: Trechos com maior quantidade de *callsigns* rastreados.

Aeroporto de Origem	Aeroporto de Destino	<i>Callsigns</i> Rastreados
Congonhas (SBSP)	Santos Dumont (SBRJ)	62
Santos Dumont (SBRJ)	Congonhas (SBSP)	59
Congonhas (SBSP)	Presidente JK (SBBR)	26
Guarulhos (SBGR)	Salgado Filho (SBPA)	24
Presidente JK (SBBR)	Congonhas (SBSP)	24
Salgado Filho (SBPA)	Guarulhos (SBGR)	22
Tancredo Neves (SBCF)	Congonhas (SBSP)	22
Congonhas (SBSP)	Salgado Filho (SBPA)	21
Congonhas (SBSP)	Tancredo Neves (SBCF)	21
Salgado Filho (SBPA)	Congonhas (SBSP)	20

posições de voo que podem ser agrupadas mais facilmente, ou seja, posições de voo que facilitam a identificação do conjunto de pontos delimitando as aerovias em questão. O processo de normalização consiste, dessa forma, num *pré-processamento* realizado para o *processo de delimitação de aerovias* (seção 5.5).

Formalizando-se o conceito de aerovias, seja *AirwayPoints* o conjunto de pontos no espaço aéreo pertencentes às aerovias que conectam o aeroporto de origem A e o aeroporto de destino B . Este conjunto é representado pela Equação 5.1:

$$AirwayPoints = \bigcup_{i=1}^{\infty} Point(latitude_i, longitude_i, altitude_i) \quad (5.1)$$

Os processos de normalização e delimitação de aerovias (explicados nesta seção e na seção 5.5) têm por finalidade produzir um subconjunto finito de posições *ReducedAirwayPoints* a partir do conjunto infinito *AirwayPoints* tal que $ReducedAirwayPoints \subset AirwayPoints$. De forma que, o conjunto *ReducedAirwayPoints* será a *representação* das aerovias conectando aeroportos de origem e destino. Assim, para saber se alguma célula convectiva intersecta alguma aerovia do espaço aéreo, basta comparar a posição da célula convectiva com os pontos pertencentes a *ReducedAirwayPoints*.

Seja FS o conjunto de *todos* os voos rastreados durante o processo de rastreamento entre aeroporto de origem A e aeroporto de destino B e $TRACK$ o conjunto de *todas* as posições de voo rastreadas. Seja também um voo F_i tal que $F_i \in FLGITHS$, suas posições de voo rastreadas FP_i e uma posição de voo rastreada qualquer $F_i.position_j$.

Como visto anteriormente, as posições de voo das aeronaves correspondem às coordenadas geográficas (longitude, latitude e altitude) das aeronaves no momento em que estas foram rastreadas. O conjunto de posições de voo FP_i é representado pela Equação 5.2 e uma posição de voo qualquer de F_i é representada pela Equação 5.3:

$$FP_i = [F_i.position_0, F_i.position_1, F_i.position_2, \dots, F_i.position_m] \quad (5.2)$$

$$F_i.position_j = (latitude_{ij}, longitude_{ij}, altitude_{ij}) \quad (5.3)$$

Num cenário ideal, a posição de voo $F_i.position_j$ é utilizada para representar um elemento em *ReducedAirwayPoints* de aeroporto de origem A e aeroporto de destino B , como representado na Equação 5.4.

$$F_i.position_j \in ReducedAirwayPoints \subset AirwayPoints \quad (5.4)$$

No entanto, aeronaves *podem* desviar de suas rotas pré-estabelecidas devido à presença de células convectivas ou outros obstáculos em seus percursos, fazendo com que a Equação 5.4 nem sempre seja satisfeita.

O processo de normalização transforma um conjunto de posições de voo FP_i num conjunto mais conveniente, formado por posições de voo *normalizadas* NFP_i e representado pela Equação 5.5.

$$NFP_i = [F_i.position'_0, F_i.position'_1, F_i.position'_2, \dots, F_i.position'_n] \quad (5.5)$$

Da equação Equação 5.4, pode-se estender o mesmo raciocínio para posições de voo *normalizadas*, que não mais são que posições de voo *estimadas* da trajetória do voo em questão, resultando na Equação 5.6, válida somente em caso de não desvio de rota prevista.

$$F_i.position'_j \in ReducedAirwayPoints \subset AirwayPoints \quad (5.6)$$

Assim, no caso da não ocorrência de desvios, das Equação 5.4 e Equação 5.6, posições de voo rastreadas pertencentes a FP_i e posições de voo normalizadas pertencentes a NFP_i são fortes candidatas para comporem *ReducedAirwayPoints*, isto é, $FP_i \subset ReducedAirwayPoints$ e $NFP_i \subset ReducedAirwayPoints$.

As posições de voo normalizadas são produzidas utilizando *interpolação linear* [45], onde *uma* posição de voo *normalizada* é estimada a partir de *duas* posições de voo *rastreadas*. A interpolação é uma escolha natural como método de aproximação dada a função de estimar uma nova posição de voo em meio a outras duas posições de voo. Embora as posições de voo estejam no sistema de coordenadas geográficas, aplicar *interpolação linear* é um método válido para estimar novas posições de voo, dado que as posições de voo são rastreadas em curtos intervalos de tempo (a cada 10 segundos) [42] e, portanto, estão relativamente próximas umas das outras.

A normalização é realizada de forma que as posições de voo normalizadas *compartilhem valores de latitudes* do conjunto LAT . As Equação 5.7 e Equação 5.8 descrevem o conjunto

de latitudes LAT e a condição que posições de voo normalizadas devem satisfazer:

$$LAT(A, B) = [L_0 = A.latitude, L_1, \dots, L_j, \dots, L_n = B.latitude] \quad (5.7)$$

$$F_i.position'_j.latitude \in LAT, \quad \forall F_i \in FS, \quad \forall F_i.position'_j \in NFP_i \quad (5.8)$$

onde a diferença $L_{j+1} - L_j = LAT_DIST$ é constante para qualquer índice j .

O processo de normalização é descrito pelos Algoritmos 1 e 2. O Algoritmo 1 descreve o processo geral de normalização no qual a função recebe como parâmetro as *posições de voo rastreadas* de determinado voo, e utiliza como variável global o conjunto de latitudes LAT . Novas posições de voo, chamadas de *posições de voo normalizadas*, são criadas a partir de posições de voo obtidas no processo de rastreamento de voos. Estas posições de voo normalizadas $position'$ diferem das posições de voo rastreadas $position$ no que se refere a seus valores de latitude $position'.latitude$, que agora pertencem ao conjunto finito LAT ($position'.latitude \in LAT$) enquanto que os valores de latitude das posições de voo rastreadas $position.latitude$ podem assumir quaisquer valores entre $A.latitude$ e $B.latitude$ ($A.latitude \leq position'.latitude \leq B.latitude$). O Algoritmo 2 descreve o procedimento de interpolação linear a partir de duas posições de voo rastreadas $p1$ e $p2$, fixando o valor de latitude lat na posição de voo *normalizada*.

Algorithm 1 Normalização das posições de voo *positions* em *normalized_positions*.

```

1: procedure NORMALIZE(positions)
2:   positions'  $\leftarrow$  List() ▷ Posições de voo normalizadas
3:    $i \leftarrow 0$  ▷ Índice de positions
4:    $j \leftarrow 0$  ▷ Índice de  $LAT = [L_0, L_1, \dots, L_j, \dots]$ 
5:   while  $i + 1 < positions.length$  &  $j < LAT.length$  do
6:     if  $positions_i.latitude \leq L_j \leq positions_{i+1}.latitude$  then
7:        $position' \leftarrow$  INTERPOLATE( $positions_i, positions_{i+1}, L_j$ )
8:        $positions'.APPEND(position')$  ▷ Adiciona position' ao final de positions'
9:        $j \leftarrow j + 1$ 
10:    else if  $positions_{i+1} < L_j$  then
11:       $i \leftarrow i + 1$ 
12:    else if  $L_j < positions_i$  then
13:       $j \leftarrow j + 1$ 
14:    return positions'

```

Algumas considerações relativas ao processo de normalização são importantes de serem ressaltadas. Na análise de aeroportos de origem A e destino B que possuem valores de latitudes muito próximos um do outro, é realizada uma adaptação no procedimento de normalização. Em vez de se construir o conjunto de valores de latitudes LAT é construído um conjunto similar composto, no entanto, por valores de longitudes LON , que da mesma forma que LAT definirá os valores a serem compartilhados entre posições de voo norma-

Algorithm 2 Interpolação linear a partir de duas posições de voo sucessivas p_1 e p_2 para estimar a posição de voo normalizada p tal que $p.latitude = lat$.

- 1: **procedure** INTERPOLATE(p_1, p_2, lat) \triangleright Longitude, latitude e altitude são abreviados para lon., lat. and alt.
 - 2: $p.lat. \leftarrow lat$
 - 3: $p.lon. \leftarrow \frac{(p_2.lon. - p_1.lon.)}{(p_2.lat. - p_1.lat.)} \times (lat - p_1.lat.) + p_1.lon.$
 - 4: $p.alt. \leftarrow \frac{(p_2.alt. - p_1.alt.)}{(p_2.lat. - p_1.lat.)} \times (lat - p_1.lat.) + p_1.alt.$
 - 5: **return** p
-

lizadas. Essa alteração garante uma *maior quantidade* de valores compartilhados entre as posições de voo normalizadas, e desta forma, uma melhor distribuição para o processo de delimitação de aerovias. Na implementação da solução, a escolha entre a utilização de *LAT* ou *LON* é feita da seguinte forma: se o módulo da diferença entre as latitudes dos aeroportos de origem e destino é maior que o módulo da diferença entre as longitudes dos aeroportos em questão ($|A.latitude - B.latitude| > |A.longitude - B.longitude|$), então é utilizado o conjunto *LAT.*, caso contrário é selecionado o conjunto *LON* [46]. Sem perda de generalidade, apenas será considerado o conjunto de valores compartilhados de latitude *LAT* para facilitar o entendimento da solução proposta.

Também vale a pena mencionar que o Algoritmo 1 assume que tanto *positions* quanto *LAT* são sequências crescentes, no entanto, outra opção válida é ter ambas as sequências decrescentes. Realizando uma pequena adaptação nos Algoritmos 1 e 2 é possível levar em conta as duas situações. Como será visto na seção 6.2, essa suposição nem sempre é válida, e resultará, em alguns casos, em posições normalizadas incorretas.

Por fim, a normalização torna a tarefa de comparar rotas de diferentes voos mais simples. Sejam dois voos distintos F_i e F_j , com as mesmas *rotas previstas em plano de voo*, que no processo de rastreamento retornam as posições de voo FP_i e FP_j , respectivamente. Como FP_i e FP_j são obtidas em horários e intervalos de tempo diferentes, a comparação entre as rotas realizadas por F_i e F_j seria complexa. No entanto, normalizando as posições de voo FP_i e FP_j em NFP_i e NFP_j , pode-se definir a similaridade das *rotas realizadas* por F_i e F_j através da Equação 5.9.

$$similarity(F_i, F_j) = \frac{1}{\max_{L_k \in LAT} distance(F_i.position'_k, F_j.position'_k)} \quad (5.9)$$

onde a função *distance* mede a distância entre duas posições de voo normalizadas $F_i.position'_k$ e $F_j.position'_k$ pertencentes a F_i e F_j , respectivamente.

Da Equação 5.9, as posições de voo normalizadas $F_i.position'_k$ e $F_j.position'_k$ compartilham o mesmo valor de latitude L_k . Portanto, a condição representada pela Equação 5.10

deve ser satisfeita por *todas* as posições de voo normalizadas.

$$F_i.position'_k.latitude = F_j.position'_k.latitude = L_k, \quad \forall L_k \in LAT \quad (5.10)$$

5.4 Medidas de Distância entre Posições de Voo

Antes de prosseguir no estudo dos componentes da arquitetura proposta na seção 5.5, serão definidas algumas medidas de distância entre duas posições de voo que serão utilizadas pelos algoritmos de *clustering* estudados. Algoritmos de *clustering* utilizam, muitas vezes implicitamente, o conceito de *distância* entre dois objetos. Esta distância é representada pelo parâmetro *distance_measure* nos algoritmos de DBSCAN e HDBSCAN (seções 3.2.4 e 3.2.5). Uma escolha assertiva de uma medida de distância entre duas posições de voo é imprescindível para o bom funcionamento do algoritmo de *clustering* selecionado.

Vale ressaltar que a medida de distância utilizada não precisa ser necessariamente a distância geográfica (seção 5.4.1). No contexto de *clustering*, esta medida de distância mede a *proximidade* ou *similaridade* entre duas posições de voo. A fórmula *Haversine*, por exemplo, definida na seção 5.4.2, é a medida de distância que mais se aproxima ao conceito de aerovia definido na seção 4.1.1, e também que melhor delimita as fronteiras das aerovias conectando aeroportos de origem e destino, como será visto na seção 6.2.

5.4.1 Distância Euclideana

Como coordenadas no espaço aéreo (*longitude*, *latitude*, *altitude*) são utilizadas na representação de uma posição de voo, a medida de distância que naturalmente seria utilizada é a *distância Euclideana* (distância geográfica) [47], representada pela Equação 5.11.

$$distance(p_1, p_2) = \sqrt{(p_2.x - p_1.x)^2 + (p_2.y - p_1.y)^2 + (p_2.z - p_1.z)^2} \quad (5.11)$$

onde as coordenadas $p_1(x, y, z)$ e $p_2(x, y, z)$ representam as posições de voo p_1 e p_2 num sistema de coordenadas cartesianas [48], respectivamente.

Assim, antes do cálculo de distância entre duas posições de voo, é necessário converter suas coordenadas do sistema esférico de coordenadas para o sistema de coordenadas cartesianas [49]. O Algoritmo 3 descreve o processo de conversão para o sistema cartesiano.

5.4.2 Fórmula de *Haversine*

Como visto na seção 4.1.1, as aerovias do espaço aéreo são corredores delimitados horizontalmente e verticalmente. Dada uma determinada posição de voo, enquanto que a *latitude*

Algorithm 3 Conversão de uma posição de voo no sistema esférico de coordenadas p para o sistema de coordenadas cartesiano.

```

1: procedure CONVERTTOCARTESIANCOORDINATES( $p$ )      ▷ Longitude, latitude e
   altitude são abreviados para lon., lat. and alt.
2:    $p.x \leftarrow p.alt \times \cos(p.lat) \times \sin(p.lon)$ 
3:    $p.y \leftarrow p.alt \times \sin(p.lat)$ 
4:    $p.z \leftarrow p.alt \times \cos(p.lat) \times \cos(p.lon)$ 
5:   return  $p(x, y, z)$ 

```

e longitude determinam a presença ou não da aeronave numa aerovia, sua *altitude* define em qual *flight level* a aeronave se encontra 4.1.1. Dessa forma, duas aeronaves utilizando a mesma aerovia podem ter seus valores de latitude e longitude *relativamente próximos* um do outro enquanto possuem valores de altitudes *relativamente distantes*. Neste exemplo fictício, as duas aeronaves estão na mesma aerovia, porém estão em *flight levels* diferentes. Se for utilizada a distância euclideana (seção 5.4.1) para se medir a distância, ou melhor dizendo, a proximidade entre estas duas posições de voo, o resultado não indicaria que tais posições são similares uma da outra (estando na mesma aerovia), mas retornaria a simples distância geográfica entre as mesmas. No entanto, se forem comparadas *apenas* suas *latitudes* e *longitudes*, a medida de distância entre as duas posições seria relativamente pequena, indicando, assim, que são *similares* considerando as aerovias utilizadas.

Dessa forma, a medida utilizada para se auferir a distância entre duas posições de voo é a *fórmula de Haversine* [50]. A fórmula de *Haversine* mede a distância entre dois pontos de uma esfera a partir dos valores de *latitude* e *longitude* dos pontos em questão. Para fins de simplificação a Terra é definida como uma esfera perfeita de raio R igual a 6.371 km. O Algoritmo 4 mostra como é calculada a fórmula de *Haversine*:

Algorithm 4 Interpolação linear a partir de duas posições de voo sucessivas p_1 e p_2 para estimar a posição de voo normalizada p tal que $p.latitude = lat$.

```

1: procedure HAVERSINE( $lat_1, lon_1, lat_2, lon_2$ )
2:    $R \leftarrow 6371$                                 ▷ raio da Terra em quilômetros
3:    $\Delta lat \leftarrow radians(lat_2 - lat_1)$ 
4:    $\Delta lon \leftarrow radians(lon_2 - lon_1)$ 
5:    $lat_1 \leftarrow radians(lat_1)$ 
6:    $lat_2 \leftarrow radians(lat_2)$ 
7:    $a \leftarrow \sin(\frac{\Delta lat}{2})^2 + \cos(lat_1) \times \cos(lat_2) \times \sin(\frac{\Delta lon}{2})^2$ 
8:    $c \leftarrow 2 \times asin(\sqrt{a})$ 
9:   return  $R \times c$ 

```

Vale ressaltar, novamente, que a fórmula *Haversine* não calcula a distância real de duas posições de voo. Ela calcula a distância de duas posições geográficas como se estivessem

no mesmo nível de altitude ou na mesma superfície.

Outra observação importante é que, apesar de estar sendo analisado a distância entre duas posições de voo, essa medida de distância não se limita a este caso específico. A fórmula *Haversine* pode ser generalizada para o cálculo de quaisquer dois pontos situados na superfície terrestre. Como será visto na seção 5.6, a fórmula de *Haversine* também será utilizada para se medir a distância entre uma posição de voo (formada pelas coordenadas de latitude, longitude e altitude) e uma célula convectiva (composta apenas pelas coordenadas de latitude e longitude).

5.5 Delimitação de Aerovias

Como visto na seção 5.3, as posições de voo rastreadas são fortes candidatas para compor as aerovias no espaço aéreo (representadas pelo conjunto *ReducedAirwayPoints*) segundo a Equação 5.4. Desde que as aeronaves rastreadas não desviem de suas aerovias, as posições de voo rastreadas dessas aeronaves poderiam ser comparadas com as posições das células convectivas no espaço aéreo para identificar se existem interseções entre estas e as aerovias às quais pertencem (representadas por *ReducedAirwayPoints*). No entanto, as aeronaves sofrem desvios no espaço aéreo, e para filtrar ruído das posições de voo pertencentes a *ReducedAirwayPoints*, os processos de normalização de posições de voo e delimitação de aerovias são realizados em sequência.

Após o processo de normalização, voos com mesmos aeroportos de origem A e destino B (representados pelo conjunto FS) produzem posições de voo *normalizadas* similares umas das outras se comparadas com base em seus valores compartilhados de latitude $L_j \in LAT$. Da Equação 5.9, estes voos similares possuem posições de voo normalizadas próximas umas das outras.

Assim, é possível fazer a separação de *todas* posições de voo normalizadas entre aeroportos A e B em *grupos de posições que compartilham o mesmo valor de latitude*, também chamados de *partições*. Uma partição G_j pode, então, ser representada relativamente ao seu valor de latitude $L_j \in LAT$.

Estas *partições* por sua vez podem ser agrupadas em *subconjuntos* ou *clusters*. De forma que, cada *cluster* obtido representaria a porção de uma aerovia conectando aeroportos de origem A e destino B tal que a presença de um *cluster* indicaria a presença de uma aerovia, a presença de dois *clusters* indicaria duas aerovias, e assim sucessivamente. Em outras palavras, posições de voo normalizadas obtidas de voos que utilizaram aerovias distintas seriam agrupadas em *clusters* distintos para uma mesma partição. Por outro lado, aeronaves que sobrevoassem o espaço aéreo fora dos limites das aerovias esta-

belecionadas em plano de voo produziriam posições de voo normalizadas fora das fronteiras dos *clusters* encontrados. Estas posições são chamadas de *outliers*.

Na Figura 5.5 cada segmento de reta horizontal representa a trajetória de um voo do aeroporto de origem A ao aeroporto de destino B enquanto que os segmentos de reta verticais representam as partições obtidas. Os pontos em vermelho representam posições de voo rastreadas enquanto que os pontos em azul representam posições de voo normalizadas. As posições de voo normalizadas somente podem ser separadas em partições (segmento de reta vertical), que por sua vez, podem ser separadas em *clusters*, representados por elipses na figura. Como ilustrado na figura, a utilização de um algoritmo de *clustering* em cada partição faz a separação das posições de voo entre dois *clusters* e um *outlier*. As posições de voo pertencentes aos *clusters* são posições de voo pertencentes a *ReducedAirwayPoints*, composto pela união dessas posições de voo normalizadas.

A identificação de dois *clusters* na primeira e segunda partições indica a presença de duas aerovias conectando aeroportos A e B. A quantidade de *clusters* é interessante no que diz respeito à *avaliação* do algoritmo de *clustering* utilizado. Neste caso é esperada uma coerência na quantidade de *clusters* obtidas em cada partição, isto é, uma *sequência uniforme discreta* para a quantidade de *clusters* obtidos. Uma inconsistência na sequência obtida, como a presença de dois *clusters* em algumas partições e um *cluster* em partições subsequentes, representaria uma deficiência no algoritmo de *clustering* utilizado.

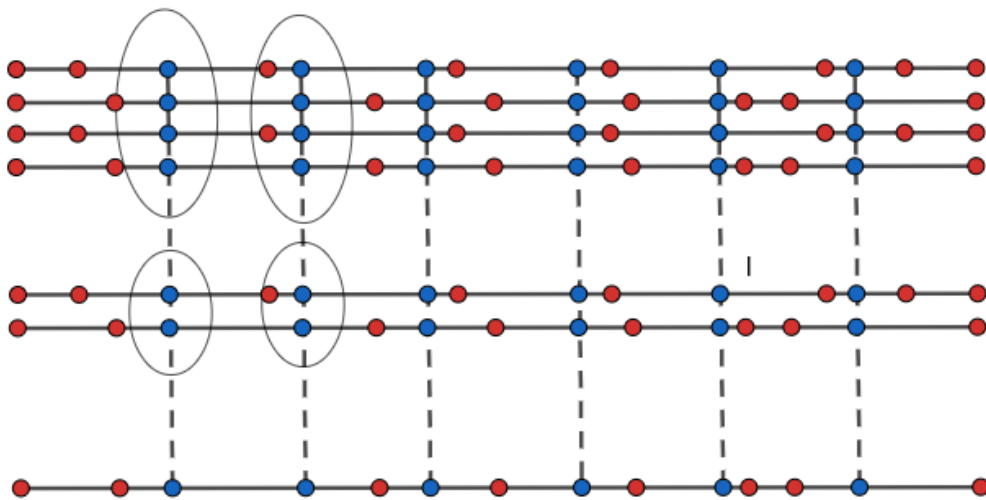


Figura 5.5: Representação do processo de delimitação de aerovias.

Estendendo a Equação 5.10, seja o conjunto G_j a *partição* formada por *todas* as posições de voo normalizadas que compartilham o mesmo valor de latitude L_j , com voos partindo do aeroporto de origem A ao aeroporto de destino B. Seja também $NORM$ o

conjunto formado por *todas* as posições de voo normalizadas obtidas a partir dos voos rastreados entre os mesmos aeroportos A e B . Estes conceitos são representados pelas Equação 5.12 e Equação 5.13.

$$G_j = \bigcup_{F_i \in FS} F_i.position'_j \quad (5.12)$$

$$NORM(A, B) = \bigcup_{F_i \in FS} \bigcup_{L_j \in LAT} F_i.position'_j \quad (5.13)$$

onde FS o conjunto dos voos com aeroporto de origem A e aeroporto de destino B , e a condição $F_i.position'_j.latitude = L_j \quad \forall L_j \in LAT$ é satisfeita.

A Equação 5.14 mostra uma outra forma de representar as Equação 5.12 e Equação 5.13, onde as posições de voo normalizadas $NORM$ podem ser agrupadas, seja utilizando o mesmo voo rastreado $F_i \in FS$, seja utilizando os valores compartilhados de latitudes $L_j \in LAT$.

$$\begin{aligned} NORM(A, B) &= \bigcup_{F_i \in FS} \bigcup_{L_j \in LAT} F_i.position'_j \\ &= \bigcup_{L_j \in LAT} \bigcup_{F_i \in FS} F_i.position'_j \\ &= \bigcup_{L_j \in LAT} G_j \end{aligned} \quad (5.14)$$

Com o objetivo de identificar os *clusters* de posições de voo normalizadas de cada partição G_j , a solução proposta executa um dos algoritmos de *clustering* vistos no Capítulo 3, dentre eles os algoritmos de DBSCAN e HDBSCAN.

Seja G_j^k um subconjunto da partição G_j e k -ésimo *cluster* de posições de voo normalizadas que compartilham os mesmos valores de latitude $L_j \in LAT$. Seja também G'_j o conjunto de posições de voo normalizadas formado pela união de todos os *clusters* G_j^k . Dessa forma, o conjunto G'_j é subconjunto da partição G_j ($G'_j \subseteq G_j$) e uma posição de voo $position'$ tal que $position' \in G_j \setminus G'_j$ é considerada um *outlier* segundo o algoritmo de *clustering* utilizado. A Equação 5.15 descreve o conjunto G'_j definido acima.

$$G'_j = \bigcup_{k=1}^{N_j} G_j^k \quad (5.15)$$

onde N_j é a quantidade de *clusters* encontrados pelo algoritmo de *clustering* no conjunto G_j .

Seja $NORM'$ o conjunto de posições de voo normalizadas formado pela união de todos os conjuntos G'_j da Equação 5.15. O conjunto de posições de voo $NORM'$ será *definido* como o conjunto *ReducedAirwayPoints* de pontos pertencentes às aerovias conectando aeroportos de origem A e destino B . Dessa forma, este será o conjunto utilizado no

processo de busca por interseções com células convectivas no espaço aéreo. Estes conceitos são representados nas Equação 5.16 e Equação 5.17.

$$NORM' = \bigcup_{L_j \in LAT} G'_j \quad (5.16)$$

$$ReducedAirwayPoints \equiv NORM' \subset NORM \quad (5.17)$$

O Algoritmo 5 descreve o procedimento de delimitação de aerovias explicado nesta seção. O algoritmo recebe como parâmetro de entrada posições de voo normalizadas e produz como saída as posições de voo normalizadas compondo o conjunto *ReducedAirwayPoints*.

Algorithm 5 Identificação das posições de voo normalizadas que fazem parte do conjunto *ReducedAirwayPoints*

```

1: procedure REDUCEAIRWAYPOINTS(positions')
2:   airway_points  $\leftarrow$  HashTable()
3:   latitude_to_positions'  $\leftarrow$  HashTable()
4:   for each position' : positions' do
5:     if not latitude_to_positions'.CONTAINSKEYS(position'.latitude) then
6:       latitude_to_positions'.INSERT(position'.latitude, List())
7:        $G_j \leftarrow$  latitude_to_positions'.GET(position'.latitude)
8:        $G_j$ .APPEND(position')
9:   for each latitude : latitude_to_positions'.KEYS() do
10:     $G_j \leftarrow$  latitude_to_positions'.GET(latitude)
11:    cluster_to_positions'  $\leftarrow$  CLUSTERALGORITHM( $G_j$ )
12:    airway_points.INSERT(latitude, cluster_to_positions')
13:   return airway_points

```

O Algoritmo 5 pode ser dividido nos seguintes passos. Inicialmente, as posições de voo normalizadas são agrupadas segundo seus valores de latitudes, de forma que posições de voo de mesma latitude pertençam à mesma *partição* G_j . O número de partições formadas é delimitado por *LAT* visto que este é conjunto de latitudes compartilhado entre posições de voo normalizadas. Assim, cada partição pode ser representada por um valor *latitude* tal que $latitude \in LAT$ na variável *airway_points*, definido como uma *hash table*.

Em seguida, *cada partição* de posições de voo normalizadas é dividida em subgrupos segundo os *clusters* obtidos pelo algoritmo de *clustering* utilizado. Cada subgrupo (*cluster*) pode ser identificado por sua respectiva *cluster label*.

O conjunto *ReducedAirwayPoints* não é retornado *diretamente* pelo Algoritmo 5, este, no entanto, é retornado de forma *indireta*. O algoritmo retorna *airway_points* que representa a estrutura de agrupamentos discutida ao longo desta seção: partições que são divididas em *clusters*, que por sua vez são compostos de posições de voos normali-

zadas. Dessa forma, contabilizando todas as posições de voo normalizadas presentes em *airway_points* tem-se o conjunto desejado *ReducedAirwayPoints*.

5.6 Interseções entre Células Convectivas e Aerovias

O processo de identificação de interseções entre células convectivas e aerovias utiliza os resultados obtidos durante o processo de delimitação de aerovias, isto é, as posições de voo pertencentes a *clusters*. Vale ressaltar que a aplicação deste processo ocorreria num cenário de tempo real, onde interseções identificadas seriam prontamente informadas às partes interessadas ATM.

As interseções entre células convectivas e aerovias são identificadas verificando se há interseções entre estas células e o conjunto de posições de voo *ReducedAirwayPoints*. Sabe-se que *ReducedAirwayPoints* representa o conjunto de posições de voo pertencentes às aerovias conectando aeroporto de origem *A* e aeroporto de destino *B*, e que estes são derivados a partir de posições de voo normalizadas durante o processo anterior de delimitação de aerovias. Se a distância entre qualquer uma dessas posições e uma célula convectiva é *inferior* à medida do raio da célula convectiva, a célula convectiva em questão *intersecta* a aerovia que possui esta posição de voo em sua composição.

Seja *CC* o conjunto de células convectivas identificadas num dado tempo *t*. Como discutido anteriormente, estas células são representadas por círculos, sendo necessários posição geográfica de seu centro e medida do seu raio de impacto. As Equação 5.18 e Equação 5.19 representam estes conceitos.

$$CC(t) = [C_0, C_1, \dots, C_i, \dots, C_n] \quad (5.18)$$

$$C_i = (latitude_i, longitude_i, radius_i) \quad (5.19)$$

onde $(latitude_i, longitude_i)$ são as coordenadas da célula e $radius_i$ é a medida do raio da célula.

Para medir a distância entre a célula convectiva e uma posição de voo normalizada pertencente a *ReducedAirwayPoints*, é utilizada a fórmula de *Haversine* (explicada na seção 5.4.2). Uma interseção entre uma célula convectiva e uma posição de voo normalizada ocorre quando a Equação 5.20 é satisfeita.

$$HAVERSINE(P_i, P_j) \leq CC_i.radius \quad (5.20)$$

onde $P_i = (CC_i.latitude, CC_i.longitude)$ e $P_j = (position'_j.latitude, position'_j.longitude)$ para qualquer $CC_i \in CC$ e qualquer $position'_j \in ReducedAirwayPoints$.

O Algoritmo 6 descreve o procedimento de busca por interseções entre células convectivas e aerovias no espaço aéreo, onde o método recebe como entrada o conjunto CC e o conjunto $airway_points$ (descrito na seção 5.5 e similar ao conjunto $ReducedAirwayPoints$), e produz como saída as células convectivas conflitantes, isto é, aquelas que intersectam as aerovias conectando aeroportos de origem e destino. No Algoritmo 6, *cada célula convectiva* é comparada a *cada posição de voo normalizada* pertencente a $airway_points$. Se a distância entre as duas for menor ou igual ao raio da célula convectiva, a célula é marcada como *conflitante* e a busca continua para a próxima célula convectiva.

Algorithm 6 Busca por células convectivas que intersectam as aerovias conectando o aeroporto de origem A ao aeroporto de destino B .

```

1: procedure FINDCONFLICTINGCONVECTIONCELLS( $cells, airway\_points$ )
2:    $conflicting\_cells \leftarrow List()$ 
3:   for each  $cell : cells$  do
4:      $conflicting \leftarrow False$ 
5:     for each  $latitude : airway\_points.KEYS()$  do
6:        $cluster\_to\_positions' \leftarrow airway\_points.GET(latitude)$ 
7:       for each  $cluster : cluster\_to\_positions'.KEYS()$  do
8:         for each  $position' : cluster\_to\_positions'.GET(cluster)$  do
9:           if  $Haversine(cell, position') \leq cell.radius$  then
10:             $conflicting\_cells.APPEND(cell)$ 
11:             $conflicting \leftarrow True$ 
12:            break
13:         if  $conflicting \equiv True$  then
14:           break
15:       if  $conflicting \equiv True$  then
16:         break
17:   return  $conflicting\_cells$ 

```

O Algoritmo 6 possui complexidade computacional de tempo da ordem de $\mathcal{O}(n \times m)$, onde n é a ordem de grandeza de $ReducedAirwayPoints$ e m é a ordem de grandeza do conjunto de células convectivas CC . Dado o número relativamente grande de posições de voo normalizadas n e de células convectivas m , a *solução apresentada é inviável*. Dessa forma, faz-se necessário buscar por uma solução *mais eficiente* para o problema de identificar interseções entre células convectivas e aerovias.

O Algoritmo 7 representa uma solução mais eficiente para o problema de busca por interseções entre células convectivas e aerovias. Primeiramente, numa etapa de pré-processamento, ordenando as células convectivas CC e o conjunto $airway_points$ por valores de latitudes, é possível reduzir o número de comparações entre células convectivas e posições de voo pertencentes a $ReducedAirwayPoints$. Em seguida, são percorridos sequencialmente os conjuntos de posições de voo agrupadas por latitudes ($airway_points$)

e de células convectivas (*cells*) com o auxílio dos ponteiros *i* e *j*, respectivamente. Se a distância entre o grupo de posições de voo e a célula convectiva em questão for menor do que o raio de atuação da célula convectiva, a célula é marcada como *conflitante* e a busca segue para a próxima célula convectiva. Se a distância encontrada é maior que o raio da célula convectiva, têm-se os seguintes cenários:

1. A latitude da célula convectiva é *maior que* a latitude do grupo de posições de voo - comparar o grupo de posições de voo atual com as células convectivas seguintes incorreria em distâncias entre ambos cada vez maiores (dado que conjunto de células convectivas *cells* está ordenado por latitudes crescentes). Dessa forma, comparações entre as mesmas se tornam desnecessárias. A próxima comparação válida é a partir do próximo grupo de posições de voo ($i \leftarrow i + 1$).
2. A latitude da célula convectiva é *menor que* a latitude do grupo de posições de voo - comparar a célula convectiva atual com os grupos de posições de voo seguintes incorreria em distâncias entre ambos cada vez maiores (dado que conjunto de grupos de posições de voo *airway_points* está ordenado por latitudes crescentes). Dessa forma, comparações entre as mesmas se tornam desnecessárias. A próxima comparação válida é a partir da próxima célula convectiva ($j \leftarrow j + 1$).

Algorithm 7 Busca *mais eficiente* por células convectivas que intersectam as aerovias conectando o aeroporto de origem *A* ao aeroporto de destino *B*.

```

1: procedure FINDCONFLICTINGCONVECTIONCELLS(cells, airway_points)
2:   airway_points.SORT(by = latitude)
3:   cells.SORT(by = latitude)
4:   i ← 0
5:   j ← 0
6:   conflicting_cells ← List()
7:   while i < airway_points.length & j < cells.length do
8:     distance = HAVERSINE*(airway_pointsi, cellsj)
9:     if distance ≤ cellj.radius then
10:      conflicting_cells.APPEND(cellj)
11:      j ← j + 1
12:     else if airway_pointsi.latitude < cellsj.latitude then
13:       i ← i + 1
14:     else if cellsj.latitude < airway_pointsi.latitude then
15:       j ← j + 1
16:   return conflicting_cells

```

Uma observação importante é a forma de se calcular a distância entre um *grupo de posições de voo* que compartilham a mesma latitude e *uma célula convectiva*. A função

*haversine** deve retornar a *menor distância* pela fórmula *haversine* entre a célula convectiva e as posições de voo em questão, como representado pela equação Equação 5.21:

$$haversine^*(airway_points_i, cells_j) = \min_{position'_k} (haversine(position'_k, cells_j)) \quad (5.21)$$

onde a posição de voo $position'_k$ é tal que $position'_k \in airway_points_i$.

O Algoritmo 7 possui complexidade computacional de tempo da ordem de $\mathcal{O}(p \log p + m \log m + (n + m) \times q)$, onde p é a ordem de grandeza de *LAT*, m é a ordem de grandeza do conjunto de células convectivas (*CC*), n é a ordem de grandeza de *ReducedAirwayPoints* e q é a ordem de grandeza do maior grupo de posições de voo em *airway_points*. No entanto, considerando que a ordem de grandeza de *ReducedAirwayPoints* é muito superior que a quantidade de células convectivas *CC* e que o conjunto *LAT* ($n \gg p$ e $n \gg m$), pode-se simplificar a complexidade computacional de tempo da solução proposta para $\mathcal{O}((n + m) \times q)$. A solução, portanto, possui um ganho computacional de performance considerável se comparada à solução anterior de complexidade $\mathcal{O}(n \times m)$.

No Algoritmo 7, além de rastrear as células convectivas consideradas *conflitantes* (que intersectam as aerovias conectando os aeroportos de origem e destino), também é possível medir o *impacto* destas células convectivas nas aerovias em questão. Dado um grupo de posições de voo que compartilham a mesma latitude *cluster_to_positions* e uma determinada célula convectiva *cell*, quanto maior a quantidade de posições de voo $position'$ de *cluster_to_positions* tal que a distância entre $position'$ e *cell* seja menor que o raio da célula convectiva, *maior o impacto* de *cell* nas aerovias conectando aeroportos de origem e destino.

O Algoritmo 8 mostra a forma em que é calculado o *impacto* de uma célula convectiva num grupo de posições de voo que compartilham a mesma latitude. O valor de impacto de uma célula convectiva *cell* num conjunto de posições de voo compartilhando mesma latitude *cluster_to_positions'* é definido como a *razão* entre a *quantidade* de posições de voo cujas distâncias da célula convectiva seja menor que o raio da célula convectiva e o *total* de posições de voo presentes no conjunto de posições de voo.

Caso a célula convectiva *cell* intersecte *todas* as posições de voo presentes na variável *cluster_to_positions*, o impacto resultante é igual a 1. Isto é, a célula convectiva *cell* impacta *todas* as aerovias conectando os aeroportos de origem e destino. No entanto, caso a célula convectiva em questão não intersecte *nenhuma* posição de voo presente em *cluster_to_positions*, o impacto resultante é igual a 0. Isto é, a célula convectiva não impacta *nenhuma* aerovia conectando os aeroportos de origem e destino.

Finalmente, o Algoritmo 8 possui mesma complexidade computacional de tempo que a solução anterior, $\mathcal{O}((n + m) \times q)$, enquanto retorna o *impacto* das células conflitantes

nos aeroportos de origem e destino.

Algorithm 8 Mede o impacto de células convectiva que intersecta as aerovias conectando o aeroporto de origem A ao aeroporto de destino B .

```
1: procedure FINDCONFLICTINGCONVECTIONCELLS(cells, airway_points)
2:   airway_points.SORT(by = latitude)
3:   cells.SORT(by = latitude)
4:    $i \leftarrow 0$ 
5:    $j \leftarrow 0$ 
6:   conflicting_cells  $\leftarrow$  List()
7:   while  $i < \text{airway\_points.length}$  &  $j < \text{cells.length}$  do
8:     distance = HAVERSINE*(airway_points $i$ , cells $j$ )
9:     if distance  $\leq$  cell $j$ .radius then
10:      impact $j$   $\leftarrow$  MEASUREIMPACT(cells $j$ , airway_points $i$ )
11:      conflicting_cells.APPEND((cells $j$ , impact $j$ ))
12:       $j \leftarrow j + 1$ 
13:     else if airway_points $i$ .latitude < cells $j$ .latitude then
14:        $i \leftarrow i + 1$ 
15:     else if cells $j$ .latitude < airway_points $i$ .latitude then
16:        $j \leftarrow j + 1$ 
17:   return conflicting_cells
18: procedure MEASUREIMPACT(cell, cluster_to_positions')
19:   all_flight_positions  $\leftarrow$  List()
20:   intersected_flight_positions  $\leftarrow$  List()
21:   for each cluster : cluster_to_positions'.KEYS() do
22:     positions'  $\leftarrow$  cluster_to_positions'.GET(cluster)
23:     for each position' : positions' do
24:       all_flight_positions.APPEND(position')
25:       if HAVERSINE(position', cell)  $\leq$  cell.radius then
26:         intersected_flight_positions.APPEND(position')
27:   impact  $\leftarrow$   $\frac{\text{intersected\_flight\_positions.length}}{\text{all\_flight\_positions.length}}$ 
28:   return impact
```

Capítulo 6

Simulações e Resultados

Para o estudo de caso, foi implementada uma aplicação seguindo os moldes da arquitetura proposta na seção 5.1, onde seus componentes podem ser revisitados pela Figura 5.1.

Na aplicação desenvolvida, OpenSky Network [42] é a fonte de dados utilizada para rastreamento de voos enquanto o STSC [1] é utilizado para rastreamento de posições de células convectivas. O software desenvolvido está disponível para acesso livre [46], onde estão implementados os componentes mencionados acima, assim como o restante da arquitetura proposta (normalização de posições de voo; delimitação de aerovias; interseções entre células convectivas e aerovias).

Neste capítulo são apresentados os experimentos utilizados para a análise e validação da arquitetura proposta. São definidos diferentes cenários de teste, formados a partir de aeroportos de origem e destino, e são definidos diversos experimentos, compostos pela combinação de determinados parâmetros. Também são discutidos os processos utilizados para a seleção de parâmetros e a avaliação dos resultados obtidos. Em seguida, para cada componente da arquitetura proposta e para cada cenário de teste, são apresentados os resultados dos experimentos selecionados. Por fim, os resultados obtidos são discutidos e analisados.

6.1 Configuração dos Cenários de Teste

Um cenário de teste é composto por um *aeroporto de origem* e um *aeroporto de destino*. Para cada cenário de teste definido, a solução proposta é aplicada em sua *totalidade* (rastreamento de posições de voo, rastreamento de células convectivas, normalização de posições de voo, delimitação de aerovias, interseção entre aerovias e células convectivas, etc.). Para a definição dos cenários de teste, foram coletadas as seguintes informações:

Tabela 6.1: Trechos com maior quantidade de posições de voo rastreadas.

Aeroporto de Origem	Aeroporto de Destino	Posições de Voo
Santos Dumont (SBRJ)	Congonhas (SBSP)	77.577
Presidente JK (SBBR)	Congonhas (SBSP)	48.346
Congonhas (SBSP)	Santos Dumont (SBRJ)	44.463
Pinto Martins (SBFZ)	Guarulhos (SBGR)	41.071
Congonhas (SBSP)	Presidente JK (SBBR)	38.563
Guararapes (SBRF)	Guarulhos (SBGR)	37.291
Santos Dumont (SBRJ)	Presidente JK (SBBR)	35.476
Guarulhos (SBGR)	Pinto Martins (SBFZ)	33.735
Tancredo Neves (SBCF)	Congonhas (SBSP)	32.547
Presidente JK (SBBR)	Santos Dumont (SBRJ)	27.908

1. Rastreamento de posições de voo das aeronaves sobrevoando o território brasileiro entre 05 de maio de 2019 e 27 de maio de 2019, registrando no total 14.108 voos e 2.041.613 posições de voo no período;
2. Ocorrências de células convectivas no território brasileiro entre 03 de maio de 2019 e 13 de maio de 2019, num total de 50.116 ocorrências de células convectivas no período.

6.1.1 Método de Seleção de Cenários de Teste

Os cenários de teste foram escolhidos a partir das informações obtidas do rastreamento de posições de voo das aeronaves no espaço aéreo brasileiro. O rastreamento de células convectivas não é utilizado na *definição dos cenários de teste*, dada a natureza aleatória das ocorrências meteorológicas, apesar de também ser utilizado como parte da solução proposta.

Inicialmente, os cenários de teste escolhidos são aqueles que obtiveram a *maior quantidade de posições de voo rastreadas*. A Tabela 6.1 ilustra os 10 trechos mais utilizados segundo as posições de voo obtidas pelo processo de rastreamento. Como é de se esperar, o trecho Rio - São Paulo (Santos Dumont-Congonhas) se encontra no topo da lista. Também vale ressaltar as posições dos trechos Brasília - São Paulo (Presidente JK - Congonhas) e Fortaleza - São Paulo (Pinto Martins - Guarulhos).

O segundo critério para a seleção de cenários de teste consiste em analisar tanto o trecho de "ida" quanto o de "volta", ao escolher analisar o trecho Rio - São Paulo (Santos Dumont-Congonhas), também é analisado o trecho de "volta" São Paulo - Rio (Congonhas - Santos Dumont). Por fim, o terceiro critério de seleção utilizado consiste em considerar o tamanho dos trechos, isto é, a distância entre aeroportos de origem e destino. Neste caso, a intenção é observar como a solução proposta lida com cenários onde os aeroportos

de origem e destino estão próximos um do outro (Rio - São Paulo | Santos Dumont-Congonhas), relativamente distantes um do outro (Brasília - São Paulo | Presidente JK - Congonhas) e muito distantes um do outro (Fortaleza - São Paulo | Pinto Martins - Guarulhos).

Considerando os critérios explanados acima, os aeroportos escolhidos para comporem os cenários de teste são listados a seguir:

1. **Rio - São Paulo** (Santos Dumont - Congonhas)
2. **São Paulo - Rio** (Congonhas - Santos Dumont)
3. **Brasília - São Paulo** (Presidente JK - Congonhas)
4. **São Paulo - Brasília** (Congonhas - Presidente JK)
5. **Fortaleza - São Paulo** (Pinto Martins -Guarulhos)
6. **São Paulo - Fortaleza** (Guarulhos - Pinto Martins)

6.1.2 Seleção de Parâmetros Gerais

Conforme visto anteriormente, é necessário definir alguns parâmetros para a configuração do processo de normalização de posições de voo (seção 5.3) e para o processo de delimitação de aerovias (seção 5.5). O parâmetro de *medida de distância* é compartilhado tanto na utilização do algoritmo DBSCAN quanto no algoritmo HDBSCAN durante o processo de *clustering*, e por este motivo, é analisado nesta seção. Os parâmetros mais específicos para cada um dos algoritmos DBSCAN e HDBSCAN serão analisados separadamente nas seções 6.1.3 e 6.1.4, respectivamente. Os parâmetros gerais são descritos a seguir:

1. *Distância entre Partições LAT_DIST* - definida como $LAT_DIST = 0,03^\circ$. A escolha deste valor deve-se ao fato deste ser da *mesma ordem de magnitude* da diferença de coordenadas entre duas posições de voo consecutivas. Um valor relativamente grande para *LAT_DIST* poderia gerar poucas posições de voo normalizadas, reduzindo assim drasticamente a quantidade de posições que passariam pelo processo de *clustering*. Enquanto um valor relativamente pequeno poderia gerar muitas posições de voo normalizadas utilizando das mesmas posições de voo rastreadas como base, e dessa forma, podendo gerar até mesmo mais posições normalizadas do que inicialmente rastreadas.
2. *Medida de Distância entre Posições de Voo - distance_measure* é uma métrica implícita nos algoritmos de *clustering* DBSCAN e HDBSCAN, como visto nas seções 3.2.4 e 3.2.5. A medida de distância entre duas posições de voo define ultimamente

o conceito de proximidade entre posições, utilizado para definir se estas posições pertencem ao mesmo *cluster*, e por conseguinte, à mesma aerovia.

A medida de distância é o único parâmetro que varia dentre os apresentados acima. Para cada cenário de teste serão realizadas simulações utilizando as medidas definidas na seção 5.4, a saber: distância euclideana (seção 5.4.1) e fórmula *Haversine* (seção 5.4.2).

6.1.3 Seleção de Parâmetros - DBSCAN

Como visto na seção 3.2.4, para utilizar o algoritmo DBSCAN são necessários dois parâmetros específicos:

1. Parâmetro *min_pts* - define a quantidade mínima de posições de voo para formação de um *cluster*. Serão utilizados os seguintes valores: 5, 25, 125.
2. Parâmetro ϵ - define o raio da ϵ - *neighborhood* da posição de voo em questão. Serão utilizados os seguintes valores para ϵ : 10, 100, 1.000 (em *metros*).

Desta forma, têm-se nove possíveis combinações para os parâmetros apresentados acima (*min_pts*, ϵ), que são: (5, 10m), (5, 100m), ..., (125, 100m), (125, 1.000m). Foram escolhidos valores de diferentes ordens de magnitudes para cada parâmetro acima a fim de descobrir a influência que cada parâmetro possui no resultado da execução do algoritmo.

6.1.4 Seleção de Parâmetros - HDBSCAN

Como visto na seção 3.2.5, somente é necessário um único parâmetro para utilizar o algoritmo HDBSCAN:

1. Parâmetro *min_pts* - define a quantidade mínima de posições de voo para formação de um *cluster*. Serão utilizados os mesmos valores utilizados no algoritmo HDBSCAN: 5, 25, 125.

Desta forma, para cada cenário de teste existem três combinações possíveis para o parâmetro *min_pts*, que por assumir os mesmos valores utilizados no algoritmo DBSCAN, facilita na comparação deste com o HDBSCAN.

6.1.5 Métodos de Avaliação de Experimentos

Enumerados os casos de teste (seção 6.1.1), os parâmetros gerais (seção 6.1.2) e os parâmetros específicos de cada algoritmo (seções 6.1.3 e 6.1.4), são definidos múltiplos experimentos de acordo com os parâmetros explicados nas seções anteriores. Por exemplo, é possível

combinar o cenário de teste *Rio - São Paulo* com os parâmetros: $distance_measure = haversine, min_pts = 5, \epsilon = 100$ (na utilização do DBSCAN); ou $distance_measure = haversine, min_pts = 25$ (na utilização do HDBSCAN).

Nesta seção são discutidos os métodos de avaliação dos resultados dos experimentos. Para uma melhor avaliação da arquitetura proposta, os experimentos serão avaliados sob a ótica de cada *processo* da arquitetura apresentada na seção 5.1: *normalização de posições de voo, delimitação de aerovias e interseções entre aerovias e células convectivas*.

Processo de Normalização de Posições de Voo

O único parâmetro configurável do processo de normalização é a distância entre partições *LAT_DIST*. Como este parâmetro é mantido constante para todos os experimentos, o resultado do processo de normalização será o mesmo para *todos experimentos num dado cenário de teste* (onde estão definidos aeroportos de origem e destino). Assim, os resultados obtidos desse procedimento serão descritos no escopo de cada cenário de teste especificamente.

Serão apresentados os resultados do processo de normalização a partir da *proporção* entre posições de voo *normalizadas* (representadas pelo conjunto *NORM*, procedimento descrito nas seções 5.3 e 5.5) e posições de voo *rastreadas* (representadas pelo conjunto *TRACK*, procedimento descrito na seções 5.2.1 e 5.3). O procedimento de normalização tem êxito quando *não* ocorre uma perda *significativa* no processo de transformação de posições de voo rastreadas para posições de voo normalizadas. O procedimento deve também gerar uma quantidade de posições de voo normalizadas *menor* do que a quantidade de posições de voo rastreadas. Não faz sentido essa transformação *gerar* mais informações do que as informações, de fato, *coletadas*.

Também serão apresentadas figuras ilustrativas do processo de normalização aplicado a voos que ocorreram no caso de teste em questão. Estas figuras têm por objetivo ajudar: 1) na compreensão da transformação realizada nas posições de voo; 2) na comparação entre posições de voo rastreadas e posições de voo normalizadas; e 3) na identificação de possíveis melhorias ao processo utilizado.

Processo de Delimitação de Aerovias

No processo de delimitação é onde é feito o uso dos demais parâmetros definidos nas seções 6.1.2, 6.1.3 e 6.1.4, pois é neste processo em que é executado o algoritmo de *clustering* selecionado.

Entre os parâmetros listados, têm-se: $distance_measure, min_pts, \epsilon$ (na utilização do DBSCAN) ou $distance_measure, min_pts$ (na utilização do HDBSCAN). Para cada

combinação de parâmetros será definido um experimento específico, que será apresentado de acordo com as seguintes métricas:

1. Proporção entre posições de voo normalizadas (representadas pelo conjunto $NORM$, definido pela Equação 5.13 na seção 5.5) e posições de voo pertencentes a *clusters* (representadas pelo conjunto $NORM'$, definido pela Equação 5.16 na seção 5.5). O algoritmo de *clustering* tem êxito quando a *proporção* entre as posições de voo normalizadas *classificadas* como ruído ($NORM \setminus NORM'$) e as posições de voo normalizadas ($NORM$) é relativamente pequena ou, em outras palavras, quando a proporção entre $NORM'$ e $NORM$ é relativamente alta.
2. Distribuição da quantidade de *clusters* encontrados em cada partição. Como visto na seção 5.5, seja a partição G_j representando as posições de voo normalizadas que compartilham latitude L_j , quantidade de *clusters* N_j (pela Equação 5.15) identificados no conjunto G_j e o conjunto G'_j representando as posições de voo normalizadas pertencentes aos *clusters* encontrados na mesma partição. A distribuição da quantidade de *clusters* em cada partição N_j deve ser *idealmente* uma *distribuição uniforme discreta* [51]. Isso se deve ao fato de uma aerovia geralmente não se bifurcar em duas outras entre aeroportos de origem e destino. Dessa forma, a quantidade de *clusters* encontrada em cada partição deve ser constante. Para analisar essa distribuição, serão utilizados: média, mediana, moda (valor mais frequente da distribuição), amplitude (diferença entre maior e menor valor da distribuição) e desvio padrão da distribuição. O algoritmo de *clustering* tem êxito se é capaz de identificar uma distribuição uniforme discreta para a sequência formada pela quantidade de *clusters* de cada partição. Neste caso, isso será refletido nas métricas obtidas, como: *média, mediana e moda próximas umas das outras; amplitude relativamente pequena; e desvio padrão reduzido.*

Processo de Interseção entre Aerovias e Células Convectivas

Por fim, são apresentados os resultados do processo de interseção entre aerovias e células convectivas. O objetivo é mostrar a *aplicação* dos resultados obtidos durante o processo de delimitação de aerovias. Será apresentada a proporção entre o número de células convectivas que intersectariam as aerovias geradas na etapa anterior e o número de células convectivas que estiveram presentes na *região delimitada pelos aeroportos de origem e destino*. Essa região é delimitada por um retângulo obtido a partir dos valores de latitude e longitude dos aeroportos de origem A e destino B , como indicado na Equação 6.1 a seguir.

Tabela 6.2: Processo de normalização realizado no trecho Santos Dumont - Congonhas.

#TRACK	#NORM	P
77.577	39.456	0,509

$$\begin{aligned}
 v_1(lat, lon) &= \min(A.lat, B.lat), \min(A.lon, B.lon) \\
 v_2(lat, lon) &= \min(A.lat, B.lat), \max(A.lon, B.lon) \\
 v_3(lat, lon) &= \max(A.lat, B.lat), \min(A.lon, B.lon) \\
 v_4(lat, lon) &= \max(A.lat, B.lat), \max(A.lon, B.lon)
 \end{aligned} \tag{6.1}$$

onde v_1, v_2, v_3, v_4 são os vértices do retângulo e lat, lon representam os valores de latitude e longitude das posições geográficas, respectivamente.

Vale ressaltar que cada processo executado depende dos processos executados anteriormente, isto é, os resultados da normalização influenciam os resultados obtidos no procedimento de delimitação, que por sua vez influenciam os resultados do procedimento de interseção. Na seção 6.3, os resultados serão analisados e discutidos sob a ótica da arquitetura proposta 5.1.

6.2 Experimentos e Resultados

Os cenários de teste foram divididos em seções de forma que: seções 6.2.1 e 6.2.2 apresentam distâncias curtas entre aeroportos (Santos Dumont - Congonhas e Congonhas - Santos Dumont); seções 6.2.3 e 6.2.4 apresentam distâncias médias entre aeroportos (Presidente JK - Congonhas e Congonhas - Presidente JK); seções 6.2.5 e 6.2.6 apresentam distâncias longas entre aeroportos (Pinto Martins - Guarulhos e Guarulhos - Pinto Martins).

Para melhor compreensão do agregado, os experimentos serão apresentados sob a ótica de cada *processo* conforme apresentados na seção 5.1.

6.2.1 Cenário Santos Dumont - Congonhas

Nesta seção são apresentados os resultados para o cenário de teste Rio de Janeiro (Santos Dumont) - São Paulo (Congonhas).

Processo de Normalização

A Tabela 6.2 detalha a quantidade de posições de voo rastreadas $\#TRACK$, a quantidade de posições de voo normalizadas $\#NORM$ e a proporção entre os dois conjuntos P .

Para ilustrar o procedimento de normalização, foram selecionados dois voos GLO1055 (em 09-05-2019 às 20:49) e GLO1013 (em 24-05-2019 às 09:55) que tiveram suas posi-

ções de voo normalizadas. As figuras a seguir mostram as posições de voo normalizadas segundo suas latitudes e longitudes (Figura 6.1), suas altitudes (Figura 6.2) e suas velocidades (Figura 6.3), onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino, os pontos em vermelho representam as posições de voo rastreadas enquanto que os pontos em azul posições de voo normalizadas.

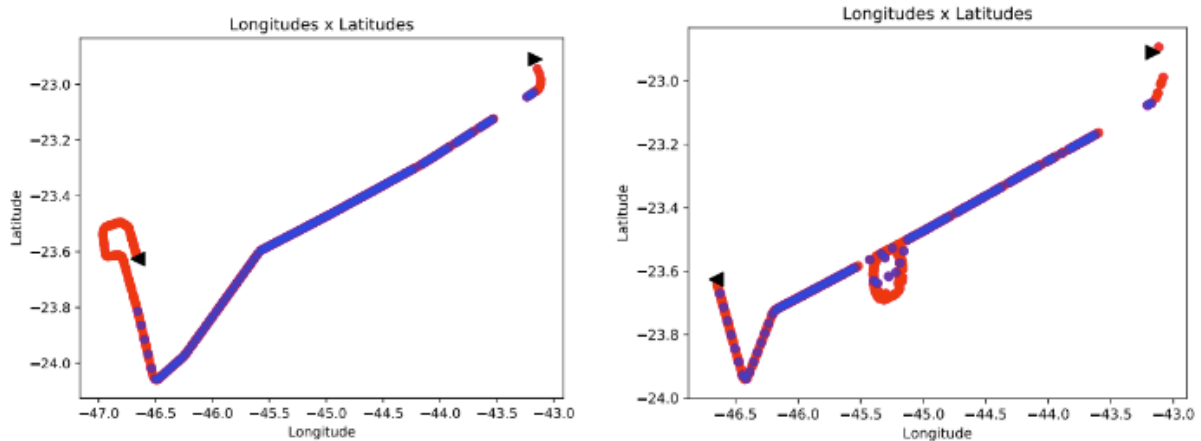


Figura 6.1: Posições dos voos GLO1055 (esquerda) e GLO1013 (direita) em latitudes e longitudes.

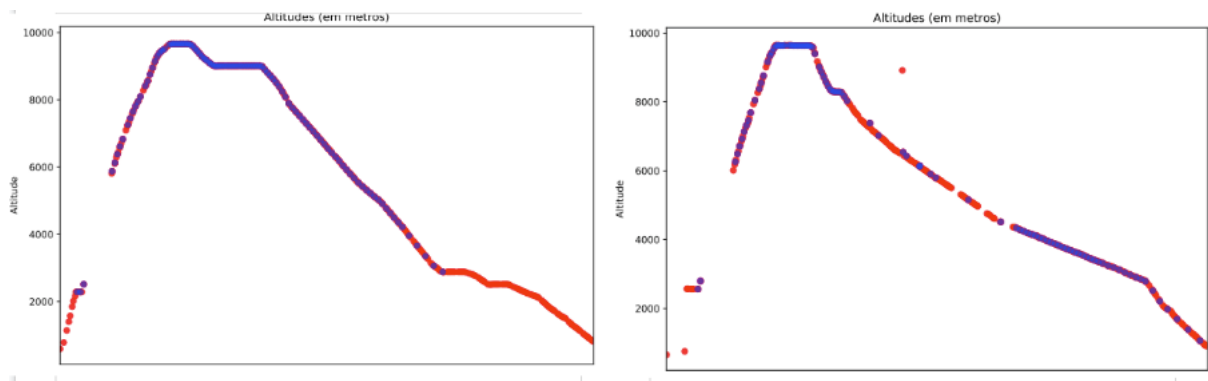


Figura 6.2: Altitudes das posições dos voos GLO1055 (esquerda) e GLO1013 (direita) ao longo do eixo tempo.

Das Figura 6.1 e Figura 6.2, é possível constatar que existem *pontos cegos* no processo de rastreamento de aeronaves, onde não foi possível determinar as posições de voo das mesmas. Isso se deve à qualidade dos dados que são fornecidos pelas plataformas formadas pelas redes distribuídas de receptores ADS-B (seção 5.2.1).

No entanto, o processo de normalização se adapta bem a esta falha no rastreamento ao *não gerar novas posições de voo* nestes pontos cegos. Como é mostrado na figura, os pontos em azul acompanham os pontos em vermelho, ou seja, não são estimadas posições

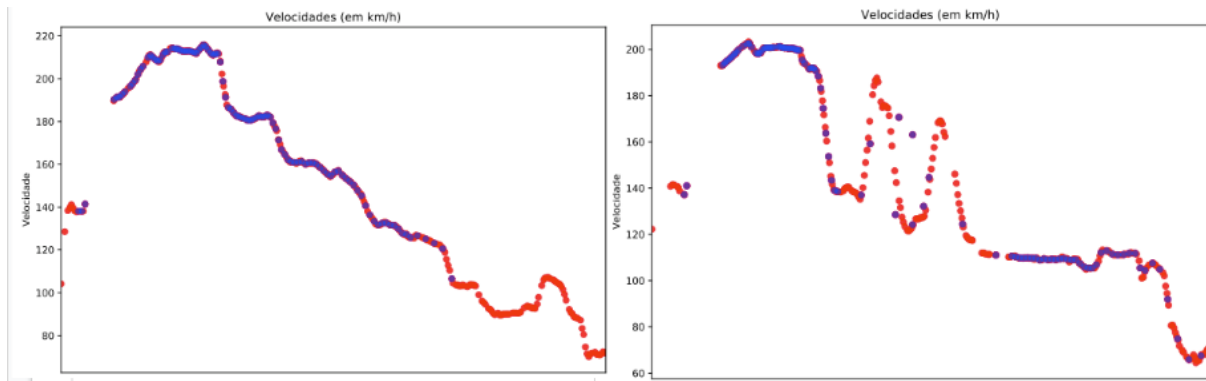


Figura 6.3: Velocidades das posições dos voos GLO1055 (esquerda) e GLO1013 (direita) ao longo do eixo tempo.

de voo normalizadas onde não existem posições de voo rastreadas próximas uma da outra. Os resultados destes pontos cegos também podem ser conferidos nos valores de velocidade das aeronaves, como mostrado na (Figura 6.3).

Das Figura 6.1, Figura 6.2 e Figura 6.3, também é possível identificar outro cenário onde o processo de normalização não funciona de forma ótima: quando a aeronave deixa de navegar em linha reta. Da Figura 6.1, por exemplo, nota-se que o voo GLO1013 realizou uma volta completa e depois retornou à mesma aerovia. O algoritmo de normalização não foi capaz de estimar com acurácia posições de voo normalizadas entre as posições rastreadas deste trecho (valendo o mesmo para as velocidades rastreadas).

Processo de Delimitação de Aerovias

No processo de delimitação de aerovias, a partição G_j , que representa as posições de voo normalizadas que compartilham latitude L_j , tem seus elementos separados em N_j *clusters*. As posições de voo normalizadas que não fazem parte de nenhum *cluster* são classificadas como *outliers*. As posições de voo normalizadas pertencentes a algum *cluster* representam o conjunto G'_j (definido na seção 5.5). Adicionalmente, a união das partições G_j representa o conjunto $NORM$ enquanto que a união dos conjuntos G'_j representa o conjunto $NORM'$ (definidos nas Equação 5.14 e Equação 5.16, respectivamente).

As Tabela 6.3 e Tabela 6.4 apresentam o resultado do processo de delimitação de aerovias para os algoritmos DBSCAN e HDBSCAN, respectivamente. Cada linha destas tabelas representa um *experimento*, isto é, a execução dos respectivos algoritmos utilizando uma combinação de parâmetros: 1) $distance_measure = haversine$ ou $euclidean$; 2) $min_pts = 5, 25, 125$; 3) $\epsilon = 10, 100, 1000$. A quantidade de posições de voo normalizadas é representada por $\#NORM$ enquanto que a quantidade de posições de voo pertencentes a algum *cluster* é representada por $\#NORM'$, e P é a proporção entre os dois conjuntos.

Tabela 6.3: Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Santos Dumont - Congonhas (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
1. <i>dist. = hav., pts. = 5, $\epsilon = 10$</i>	39456	29390	0,745	2,414	2	2	13	1,571
2. <i>dist. = hav., pts. = 5, $\epsilon = 100$</i>	39456	33350	0,845	3,328	2	2	12	2,542
3. <i>dist. = hav., pts. = 5, $\epsilon = 1000$</i>	39456	37201	0,943	2,690	2	2	7	1,262
4. <i>dist. = hav., pts. = 25, $\epsilon = 10$</i>	39456	25064	0,635	0,991	1	1	2	0,405
5. <i>dist. = hav., pts. = 25, $\epsilon = 100$</i>	39456	29433	0,746	1,017	1	1	3	0,394
6. <i>dist. = hav., pts. = 25, $\epsilon = 1000$</i>	39456	33402	0,847	1,198	1	1	4	0,529
7. <i>dist. = hav., pts. = 125, $\epsilon = 10$</i>	39456	7247	0,184	0,353	0	0	1	0,478
8. <i>dist. = hav., pts. = 125, $\epsilon = 100$</i>	39456	28088	0,712	0,879	1	1	2	0,351
9. <i>dist. = hav., pts. = 125, $\epsilon = 1000$</i>	39456	30974	0,785	0,914	1	1	2	0,361
10. <i>dist. = euc., pts. = 5, $\epsilon = 10$</i>	39456	7772	0,197	3,448	3	0	11	2,568
11. <i>dist. = euc., pts. = 5, $\epsilon = 100$</i>	39456	29036	0,736	4,448	4	3	9	2,279
12. <i>dist. = euc., pts. = 5, $\epsilon = 1000$</i>	39456	36314	0,920	2,853	2	2	8	1,516
13. <i>dist. = euc., pts. = 25, $\epsilon = 10$</i>	39456	424	0,011	0,034	0	0	1	0,182
14. <i>dist. = euc., pts. = 25, $\epsilon = 100$</i>	39456	21166	0,536	1,828	2	2	4	0,949
15. <i>dist. = euc., pts. = 25, $\epsilon = 1000$</i>	39456	32595	0,826	1,078	1	1	2	0,397
16. <i>dist. = euc., pts. = 125, $\epsilon = 10$</i>	39456	0	0,0	0,0	0	0	0	0,0
17. <i>dist. = euc., pts. = 125, $\epsilon = 100$</i>	39456	6452	0,164	0,414	0	0	2	0,732
18. <i>dist. = euc., pts. = 125, $\epsilon = 1000$</i>	39456	30450	0,772	0,914	1	1	2	0,361

Nas Tabela 6.3 e Tabela 6.4 também são apresentadas a média μ , mediana **Md**, moda **Mo**, amplitude **R** e desvio padrão σ da distribuição da quantidade de *clusters* N_j de cada partição G_j .

Para a maioria dos experimentos das Tabela 6.3 e Tabela 6.4, a proporção entre $NORM'$ e $NORM$ é um valor relativamente alto, acima de 0,7, mostrando que grande parte das posições de voo não são classificadas como *outliers*. Ou seja, a maioria dos parâmetros utilizados é capaz de encontrar *clusters* nos conjuntos de posições normalizadas (ou em outras palavras, nas partições geradas).

Considerando que o objetivo é encontrar uma *distribuição uniforme discreta* para a sequência formada pelas quantidades de *clusters* de cada partição, são calculadas a média μ , mediana **Md**, moda **Mo**, amplitude **R** e desvio padrão σ desta distribuição. No entanto, a maioria dessas métricas não representam um bom desempenho, pois:

1. *Amplitude R* - valores altos de amplitude não indicam uma distribuição uniforme discreta. Conforme mostrado nas Tabela 6.3 e Tabela 6.4, seus valores são relativamente altos, como os experimentos #1, #2, #3, #6, #10, #11, #12, #14, #19, #20, #22 e #23. Estes experimentos, então, devem ser descartados.

Tabela 6.4: Experimentos do processo de delimitação de aerovias utilizando HDBSCAN no trecho Santos Dumont - Congonhas (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
19. <i>dist. = hav., pts. = 5</i>	39456	29247	0,741	19,009	21	21	30	6,346
20. <i>dist. = hav., pts. = 25</i>	39456	28077	0,712	2,569	2	2	6	1,108
21. <i>dist. = hav., pts. = 125</i>	39456	1033	0,026	0,052	0	0	2	0,317
22. <i>dist. = euc., pts. = 5</i>	39456	32537	0,825	8,017	8	8	15	3,149
23. <i>dist. = euc., pts. = 25</i>	39456	30038	0,761	2,767	3	3	5	1.02
24. <i>dist. = euc., pts. = 125</i>	39456	15113	0,383	1,034	2	2	2	0,999

2. *Média μ , mediana Md, moda Mo* - uma outra forma de analisar a qualidade da distribuição uniforme discreta dos experimentos é comparar os valores de média, mediana e moda (seção 6.1.5). Experimentos com estes três valores muito diferentes uns dos outros devem também ser descartados da análise, pois não refletem uma distribuição uniforme discreta, como os experimentos #2, #10, #11, #19, #24. Neste caso, os experimento #7, #13, #16, #17 e #21 são falsos negativos. Este experimento possui valores iguais para média, mediana e moda, no entanto, nenhum *cluster* foi gerado em qualquer partição, isto é, o experimento não foi capaz de identificar uma aerovia sequer.

Por fim, os melhores experimentos identificados pela análise acima são #4, #5, #8, #9, #15, #18 (todos estes utilizando o algoritmo DBSCAN; marcados em negrito na Tabela 6.3). O experimento #24, apesar de não satisfazer todos os requisitos levantados na seção 6.1.5, é o melhor resultado obtido para o algoritmo HDBSCAN (em negrito na Tabela 6.4). Também é possível perceber que, das Tabela 6.3 e Tabela 6.4, os parâmetros utilizando o algoritmo DBSCAN com *distance_measure = haversine* obtiveram os *melhores resultados*.

Para ilustrar o procedimento de delimitação de aerovias, foram selecionados os experimentos #8 e #24 que obtiveram os resultados mais expressivos utilizando os algoritmos de DBSCAN e HDBSCAN, respectivamente. A Figura 6.4 representa uma partição aleatória do experimento #8 e a mesma partição obtida pelo experimento #24, onde diferentes cores representam diferentes *clusters* enquanto que a cor roxa representa *outliers*. A Figura 6.5 ilustra as aerovias geradas pelo processo de delimitação, representadas pelo conjunto $NORM'$ de cada experimento, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino e os pontos em vermelho representam as posições de voo pertencentes ao conjunto $NORM'$.

Das Figura 6.4 e Figura 6.5, é possível perceber que o experimento #8 ficou próximo em definir as fronteiras da aerovia conectando aeroportos de origem e destino (seção 4.1.1).

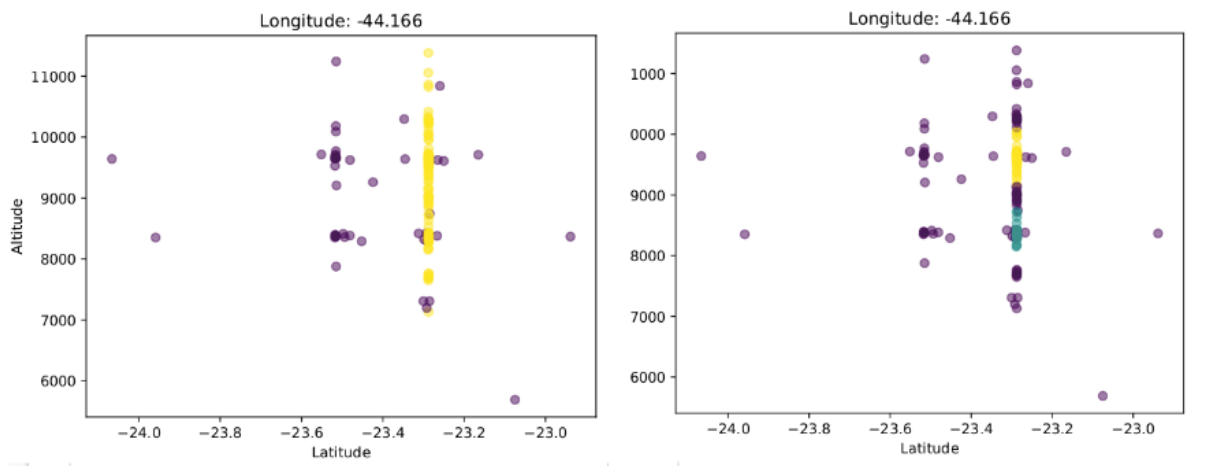


Figura 6.4: Partição de $longitude = -44,166^\circ$ dos experimentos #8 (esquerda) e #24 (direita) contendo as posições de voo normalizadas separadas em *clusters*.

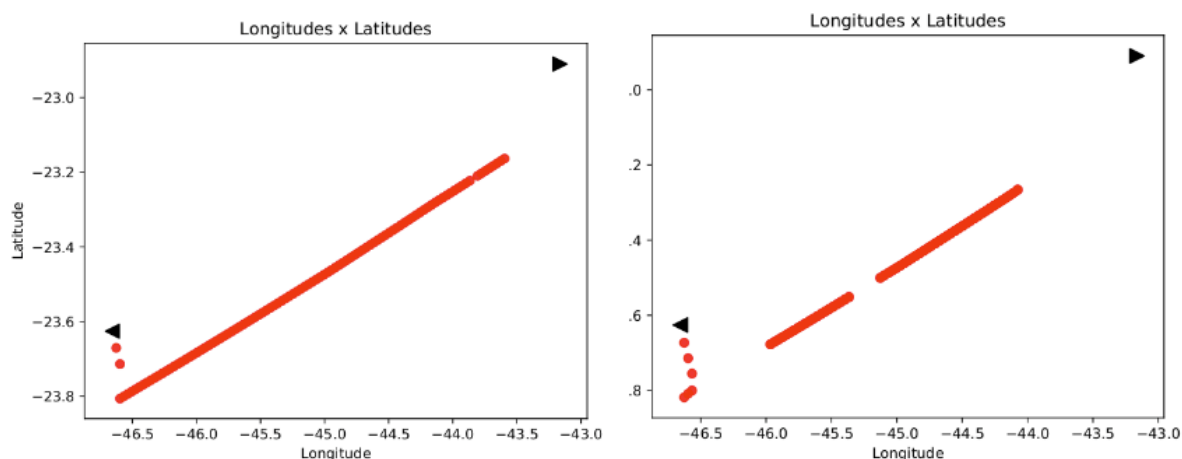


Figura 6.5: Aerovias dos experimentos #8 (esquerda) e #24 (direita) geradas durante o processo de delimitação de aerovias.

O processo de delimitação de aerovias em seguida é utilizado como suporte para o processo de interseção entre aerovias e células convectivas.

Processo de Interseção entre Aerovias e Células Convectivas

A Tabela 6.5 apresenta a quantidade de ocorrências de células convectivas na região delimitada entre aeroportos de origem e destino durante o período de rastreamento #CC (vide seção 6.1.5). A tabela apresenta também a quantidade de células convectivas #CC' que intersectaram com as aerovias obtidas do processo de delimitação de aerovias explicado anteriormente. Apenas os experimentos que obtiveram os melhores resultados estão listados na Tabela 6.5.

Tabela 6.5: Processo de interseção entre aerovias e células convectivas no trecho Santos Dumont - Congonhas (onde n.d. significa *não determinado*, *dist.* = *distance_measure*, *hav.* = *haversine*, *auc.* = *euclidean*, *pts.* = *min_pts*).

Experimento	#CC	#CC'	P
5. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, $\epsilon = 100$	0	0	n.d.
8. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 125, $\epsilon = 100$	0	0	n.d.
9. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 125, $\epsilon = 1000$	0	0	n.d.
18. <i>dist.</i> = <i>auc.</i> , <i>pts.</i> = 125, $\epsilon = 1000$	0	0	n.d.
24. <i>dist.</i> = <i>auc.</i> , <i>pts.</i> = 125	0	0	n.d.

Tabela 6.6: Processo de normalização realizado no trecho Congonhas - Santos Dumont.

#TRACK	#NORM	P
44.463	30.247	0,680

Como pode se ver, não foram *registradas nenhuma* célula convectiva na região que delimita os aeroportos de Santos Dumont e Congonhas. Consequentemente, também não teve ocorrências de células que intersectaram com as aerovias em questão.

Vale ressaltar que a aplicação do processo de interseção só faria sentido num cenário *online*, onde interseções identificadas pelo sistema em operação seriam prontamente informadas às partes interessadas. Apesar dos experimentos não terem sido realizados em tempo real, a simulação desse componente serve, no entanto, para mostrar seu correto funcionamento mesmo que num cenário *offline*.

6.2.2 Cenário Congonhas - Santos Dumont

Nesta seção são apresentados os resultados para o cenário de teste São Paulo (Congonhas) - Rio de Janeiro (Santos Dumont).

Processo de Normalização

A Tabela 6.6 detalha a quantidade de posições de voo rastreadas *#TRACK*, a quantidade de posições de voo normalizadas *#NORM* e a proporção entre os dois conjuntos *P*.

Para ilustrar novamente o procedimento de normalização, foram selecionados dois voos GLO1006 (em 16-05-2019 às 09:28) e ONE6000 (em 21-05-2019 às 06:55) que tiveram suas posições de voo normalizadas. As figuras a seguir mostram as posições de voo normalizadas segundo suas latitudes e longitudes (Figura 6.6), suas altitudes (Figura 6.7) e suas velocidades (Figura 6.8), onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino, os pontos em vermelho representam as posições de voo rastreadas enquanto que os pontos em azul posições de voo normalizadas.

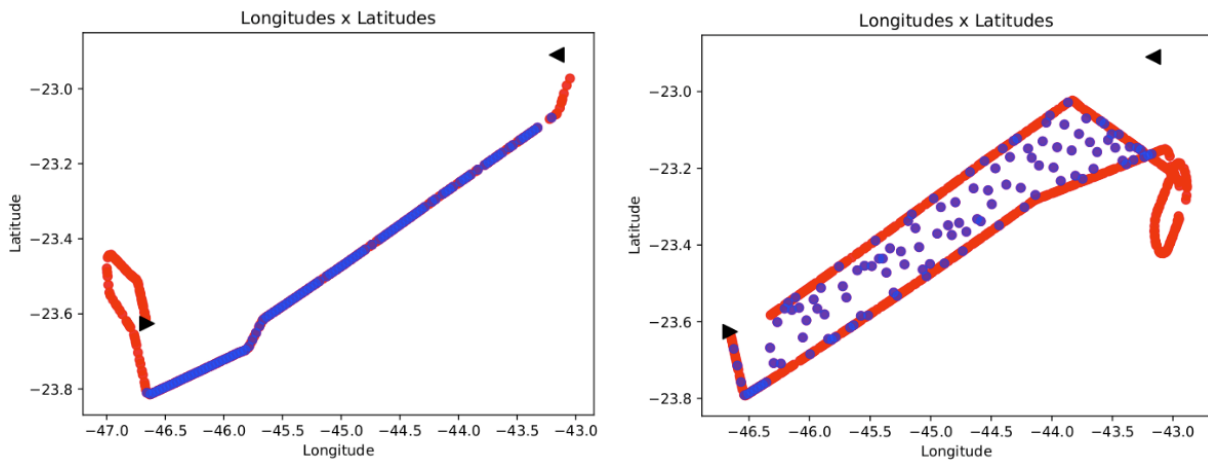


Figura 6.6: Posições dos voos GLO1006 (esquerda) e ONE6000 (direita) em latitudes e longitudes.

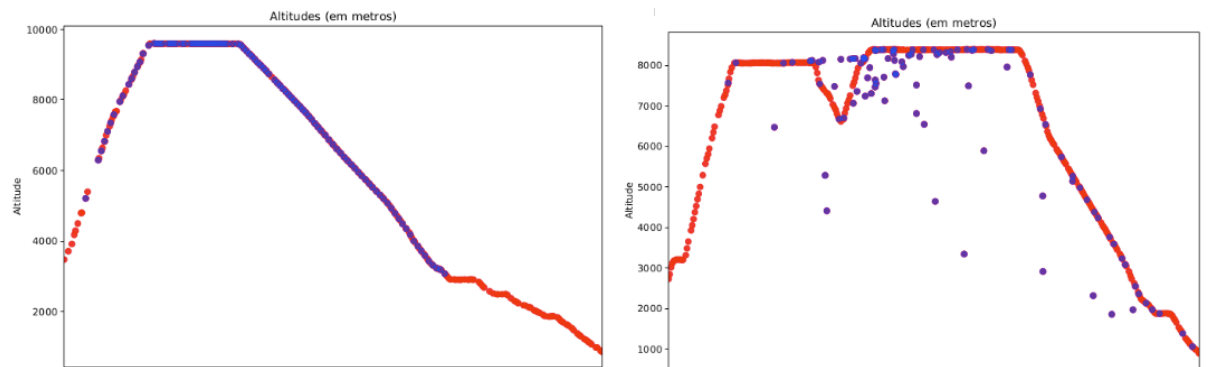


Figura 6.7: Altitudes das posições dos voos GLO1006 (esquerda) e ONE6000 (direita) ao longo do eixo tempo.

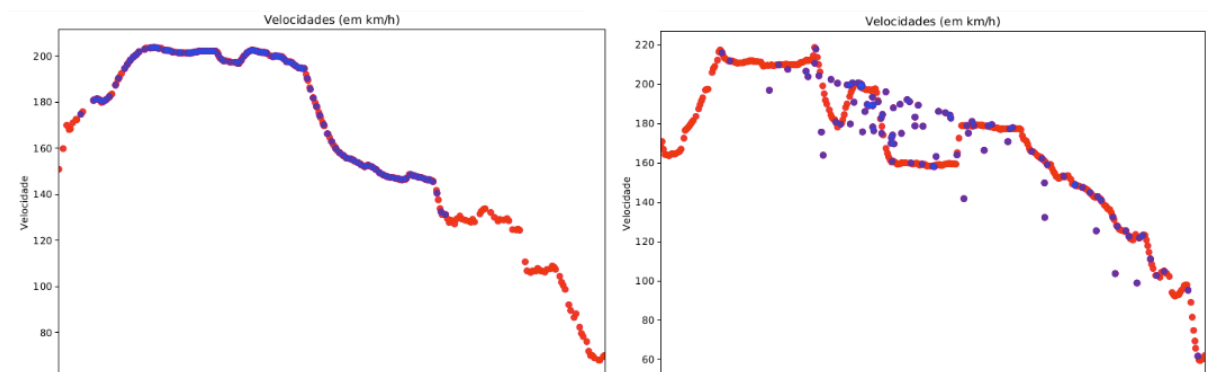


Figura 6.8: Velocidades das posições dos voos GLO1006 (esquerda) e ONE6000 (direita) ao longo do eixo tempo.

Das Figura 6.6 e Figura 6.7, é possível constatar que, da mesma forma que na seção 6.2.1, existem pontos cegos no processo de rastreamento de aeronaves. Das Figura 6.6, Figura 6.7 e Figura 6.8, também é possível identificar outro cenário onde o processo de normalização não funciona corretamente: quando o voo ONE6000 retorna ao mesmo aeroporto de origem. Este e outros casos confirmam a necessidade em se utilizar algoritmos de *clustering* que consideram ruídos no momento da separação de objetos em *clusters*, como os algoritmos de DBSCAN e HDBSCAN.

Processo de Delimitação de Aerovias

As Tabela 6.7 e Tabela 6.8 apresentam o resultado do processo de delimitação de aerovias para os algoritmos DBSCAN e HDBSCAN, respectivamente. Da mesma forma que na seção 6.2.1, cada linha destas tabelas representa um *experimento* (a execução dos respectivos algoritmos utilizando uma combinação de parâmetros). A quantidade de posições de voo normalizadas é representada por $\#NORM$ enquanto que a quantidade de posições de voo pertencentes a algum *cluster* é representada por $\#NORM'$, e P é a proporção entre os dois conjuntos. Nas Tabela 6.7 e Tabela 6.8 também são apresentadas a média μ , mediana **Md**, moda **Mo**, amplitude **R** e desvio padrão σ da distribuição da quantidade de *clusters* N_j de cada partição G_j .

Utilizando os mesmos critérios da seção 6.1.5, os mesmos utilizados na análise do cenário de teste da seção 6.2.1, chega-se à conclusão que os experimentos que mais se aproximariam de uma distribuição uniforme discreta para a sequência formada pelas quantidades de *clusters* de cada partição, são #4, #5, #6, #8, #9, #15 e #18 (todos estes utilizando o algoritmo DBSCAN e marcados em negrito na Tabela 6.7). De forma geral, da Tabela 6.7, os experimentos utilizando o algoritmo DBSCAN com *distance_measure = haversine* obtiveram os melhores resultados. Nenhum experimento utilizando HDBSCAN teve uma performance significativa o suficiente para o processo de delimitação de aerovias.

É importante ressaltar também que, basicamente, o mesmo conjunto de experimentos bem sucedidos no cenário de teste da seção 6.2.1 é o conjunto que obteve sucesso no presente cenário de teste (à exceção do experimento #6, que foi incluído no conjunto atual). Em outras palavras, os parâmetros escolhidos como os melhores para as aerovias conectando o aeroporto A ao aeroporto B são os mesmos para as aerovias conectando o aeroporto B ao A . Este fato colabora para a identificação de um algoritmo de caráter genérico que utilize um valores constante para os parâmetros escolhidos.

A Figura 6.9 ilustra a aerovia gerada pelo processo de delimitação para o experimento #8 no cenário de teste Congonhas - Santos Dumont, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino e os pontos em vermelho representam as posições de voo pertencentes ao conjunto $NORM'$.

Tabela 6.7: Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Congonhas - Santos Dumont (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
1. $dist. = hav., pts. = 5, \epsilon = 10$	30247	24841	0,821	1,121	1	1	6	1,123
2. $dist. = hav., pts. = 5, \epsilon = 100$	30247	26337	0,871	1,164	1	1	7	0,991
3. $dist. = hav., pts. = 5, \epsilon = 1000$	30247	28250	0,934	1,336	1	1	4	0,955
4. $dist. = hav., pts. = 25, \epsilon = 10$	30247	21875	0,723	0,733	1	1	2	0,563
5. $dist. = hav., pts. = 25, \epsilon = 100$	30247	25614	0,847	0,75	1	1	2	0,471
6. $dist. = hav., pts. = 25, \epsilon = 1000$	30247	27234	0,9	0,802	1	1	2	0,495
7. $dist. = hav., pts. = 125, \epsilon = 10$	30247	0	0,0	0,0	0	0	0	0,0
8. $dist. = hav., pts. = 125, \epsilon = 100$	30247	24693	0,816	0,664	1	1	1	0,472
9. $dist. = hav., pts. = 125, \epsilon = 1000$	30247	25980	0,859	0,664	1	1	1	0,472
10. $dist. = euc., pts. = 5, \epsilon = 10$	30247	3932	0,13	2,086	0	0	14	3,145
11. $dist. = euc., pts. = 5, \epsilon = 100$	30247	24315	0,804	2,026	2	0	6	1,637
12. $dist. = euc., pts. = 5, \epsilon = 1000$	30247	27755	0,918	1,267	1	1	3	0,855
13. $dist. = euc., pts. = 25, \epsilon = 10$	30247	0	0,0	0,0	0	0	0	0,0
14. $dist. = euc., pts. = 25, \epsilon = 100$	30247	18502	0,612	1,207	1	2	3	0,961
15. $dist. = euc., pts. = 25, \epsilon = 1000$	30247	27040	0,894	0,802	1	1	2	0,495
16. $dist. = euc., pts. = 125, \epsilon = 10$	30247	0	0,0	0,0	0	0	0	0,0
17. $dist. = euc., pts. = 125, \epsilon = 100$	30247	3740	0,124	0,198	0	0	1	0,399
18. $dist. = euc., pts. = 125, \epsilon = 1000$	30247	25598	0,846	0,664	1	1	1	0,472

Tabela 6.8: Experimentos do processo de delimitação de aerovias utilizando HDBSCAN no trecho Congonhas - Santos Dumont (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
19. $dist. = hav., pts. = 5, \epsilon = 1000$	30247	21085	0,697	14,319	19	0	28	9,966
20. $dist. = hav., pts. = 25, \epsilon = 1000$	30247	16551	0,547	2,078	2	0	5	1,538
21. $dist. = hav., pts. = 125, \epsilon = 1000$	30247	0	0,0	0,0	0	0	0	0,0
22. $dist. = euc., pts. = 5, \epsilon = 1000$	30247	24091	0,796	3,879	3	0	15	3,424
23. $dist. = euc., pts. = 25, \epsilon = 1000$	30247	18306	0,605	1,552	2	2	4	1,093
24. $dist. = euc., pts. = 125, \epsilon = 1000$	30247	14	0,0	0,017	0	0	2	0,185

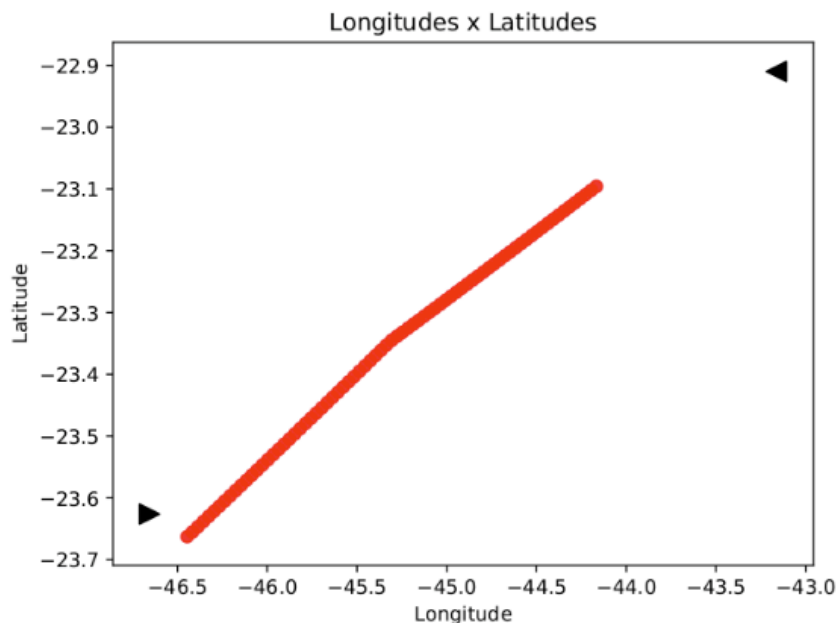


Figura 6.9: Aerovia do experimento #8 gerada durante o processo de delimitação de aerovias.

Processo de Interseção entre Aerovias e Células Convectivas

A Tabela 6.9 apresenta a quantidade de ocorrências de células convectivas na região delimitada entre aeroportos de origem e destino durante o período de rastreamento #CC (vide seção 6.1.5). A tabela apresenta também a quantidade de células convectivas #CC' que intersectaram com as aerovias obtidas do processo de delimitação de aerovias explicado anteriormente. Apenas os experimentos que obtiveram os melhores resultados estão listados na Tabela 6.9.

Como pode se ver, não foi registrada nenhuma célula convectiva na região que delimita os aeroportos de Congonhas e Santos Dumont.

Tabela 6.9: Processo de interseção entre aerovias e células convectivas no trecho Congonhas - Santos Dumont (onde n.d. significa *não determinado*, *dist.* = *distance_measure*, *hav.* = *haversine*, *auc.* = *euclidean*, *pts.* = *min_pts*).

Experimento	#CC	#CC'	P
4. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, ϵ = 10	0	0	n.d.
5. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, ϵ = 100	0	0	n.d.
6. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, ϵ = 1000	0	0	n.d.
8. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 125, ϵ = 100	0	0	n.d.
9. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 125, ϵ = 1000	0	0	n.d.
15. <i>dist.</i> = <i>auc.</i> , <i>pts.</i> = 25, ϵ = 1000	0	0	n.d.
18. <i>dist.</i> = <i>auc.</i> , <i>pts.</i> = 125, ϵ = 1000	0	0	n.d.

Tabela 6.10: Processo de normalização realizado no trecho Presidente JK - Congonhas.

#TRACK	#NORM	P
48.346	28.809	0,596

6.2.3 Cenário Presidente JK - Congonhas

Nesta seção são apresentados os resultados para o cenário de teste Brasília (Presidente JK) - São Paulo (Congonhas).

Processo de Normalização

A Tabela 6.10 detalha a quantidade de posições de voo rastreadas $\#TRACK$, a quantidade de posições de voo normalizadas $\#NORM$ e a proporção entre os dois conjuntos P .

As Figura 6.10, Figura 6.11 e Figura 6.12 mostram as posições de voo normalizadas do voo TAM3173 (em 21-05-2019 às 06:14) segundo suas latitudes e longitudes, suas altitudes e suas velocidades, respectivamente, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino, os pontos em vermelho representam as posições de voo rastreadas enquanto que os pontos em azul posições de voo normalizadas.

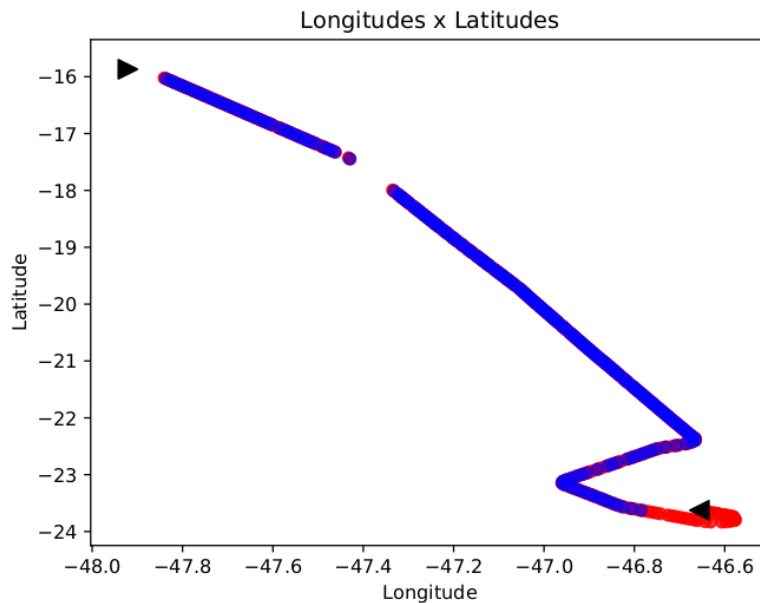


Figura 6.10: Posições do voo TAM3173 em latitudes e longitudes.

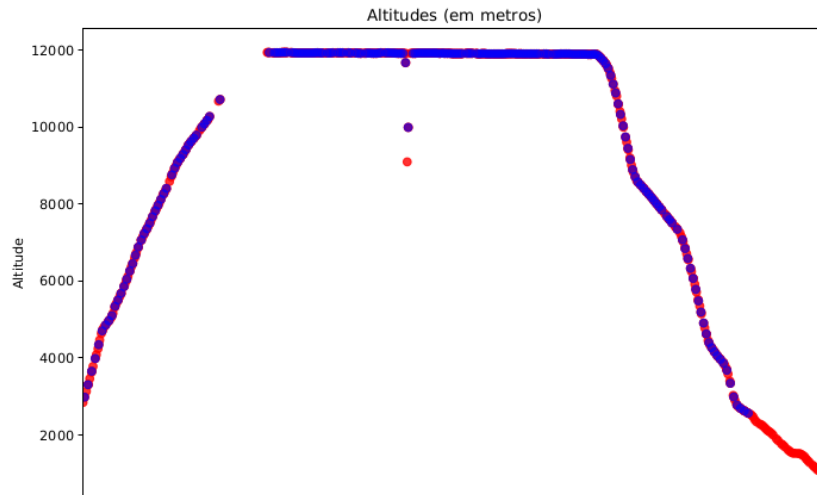


Figura 6.11: Altitudes das posições do voo TAM3173 ao longo do tempo.

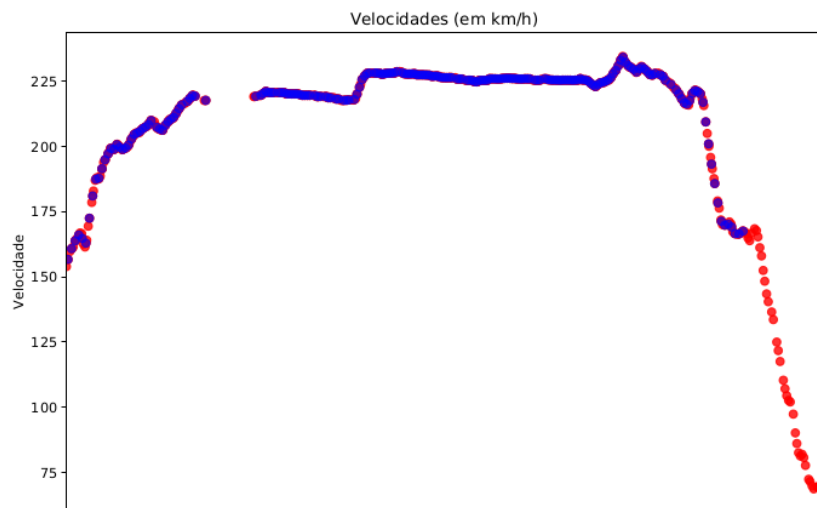


Figura 6.12: Velocidades das posições do voos TAM3173 ao longo do tempo.

Processo de Delimitação de Aerovias

As Tabela 6.11 e Tabela 6.12 apresentam o resultado do processo de delimitação de aerovias para os algoritmos DBSCAN e HDBSCAN, respectivamente. Da mesma forma que na seção 6.2.1, cada linha destas tabelas representa um *experimento* (combinação de parâmetros). A quantidade de posições de voo normalizadas é representada por $\#NORM$ enquanto que a quantidade de posições de voo pertencentes a algum *cluster* é representada por $\#NORM'$, e P é a proporção entre os dois conjuntos. Nas Tabela 6.11 e Tabela 6.12 também são apresentadas a média μ , mediana \mathbf{Md} , moda \mathbf{Mo} , amplitude \mathbf{R} e desvio padrão σ da distribuição da quantidade de *clusters* N_j de cada partição G_j .

Tabela 6.11: Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Presidente JK - Congonhas (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
1. <i>dist. = hav., pts. = 5, $\epsilon = 10$</i>	28809	18993	0,659	1,116	1	1	7	1,048
2. <i>dist. = hav., pts. = 5, $\epsilon = 100$</i>	28809	22247	0,772	1,558	1	1	6	1,297
3. <i>dist. = hav., pts. = 5, $\epsilon = 1000$</i>	28809	26333	0,914	1,602	1	1	6	1,086
4. <i>dist. = hav., pts. = 25, $\epsilon = 10$</i>	28809	13780	0,478	0,637	1	1	2	0,579
5. <i>dist. = hav., pts. = 25, $\epsilon = 100$</i>	28809	19892	0,69	0,709	1	1	2	0,496
6. <i>dist. = hav., pts. = 25, $\epsilon = 1000$</i>	28809	22725	0,789	0,876	1	1	3	0,532
7. <i>dist. = hav., pts. = 125, $\epsilon = 10$</i>	28809	0	0,0	0,0	0	0	0	0,0
8. <i>dist. = hav., pts. = 125, $\epsilon = 100$</i>	28809	7898	0,274	0,243	0	0	1	0,429
9. <i>dist. = hav., pts. = 125, $\epsilon = 1000$</i>	28809	12425	0,431	0,378	0	0	1	0,485
10. <i>dist. = euc., pts. = 5, $\epsilon = 10$</i>	28809	3874	0,134	0,972	0	0	5	1,264
11. <i>dist. = euc., pts. = 5, $\epsilon = 100$</i>	28809	18937	0,657	2,355	2	4	5	1,626
12. <i>dist. = euc., pts. = 5, $\epsilon = 1000$</i>	28809	25729	0,893	1,574	1	1	6	1,139
13. <i>dist. = euc., pts. = 25, $\epsilon = 10$</i>	28809	210	0,007	0,012	0	0	1	0,109
14. <i>dist. = euc., pts. = 25, $\epsilon = 100$</i>	28809	13457	0,467	1,016	1	2	2	0,888
15. <i>dist. = euc., pts. = 25, $\epsilon = 1000$</i>	28809	21859	0,759	0,845	1	1	3	0,532
16. <i>dist. = euc., pts. = 125, $\epsilon = 10$</i>	28809	0	0,0	0,0	0	0	0	0,0
17. <i>dist. = euc., pts. = 125, $\epsilon = 100$</i>	28809	0	0,0	0,0	0	0	0	0,0
18. <i>dist. = euc., pts. = 125, $\epsilon = 1000$</i>	28809	1866	0,065	0,056	0	0	1	0,229

Tabela 6.12: Experimentos do processo de delimitação de aerovias utilizando HDSCAN no trecho Presidente JK - Congonhas (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
19. <i>dist. = hav., pts. = 5, $\epsilon = 1000$</i>	28809	20719	0,719	5,378	6	8	11	2,815
20. <i>dist. = hav., pts. = 25, $\epsilon = 1000$</i>	28809	8469	0,294	0,789	0	0	4	1,006
21. <i>dist. = hav., pts. = 125, $\epsilon = 1000$</i>	28809	0	0,0	0,0	0	0	0	0,0
22. <i>dist. = euc., pts. = 5, $\epsilon = 1000$</i>	28809	23739	0,824	3,402	4	4	7	1,534
23. <i>dist. = euc., pts. = 25, $\epsilon = 1000$</i>	28809	14655	0,509	1,167	2	2	4	1,008
24. <i>dist. = euc., pts. = 125, $\epsilon = 1000$</i>	28809	0	0,0	0,0	0	0	0	0,0

Utilizando os mesmos critérios da seção 6.1.5, os mesmos utilizados na análise do cenário de teste da seção 6.2.1, chega-se à conclusão que os experimentos que mais se aproximariam de uma distribuição uniforme discreta para a sequência formada pelas quantidades de *clusters* de cada partição, são #4, #5, #6 e #15 (todos estes utilizando o algoritmo DBSCAN e marcados em negrito na Tabela 6.11). De forma geral, da Tabela 6.11, os experimentos utilizando o algoritmo DBSCAN com $distance_measure = haversine$ obtiveram os melhores resultados. Nenhum experimento utilizando HDBSCAN teve uma performance significativa o suficiente para o processo de delimitação de aerovias.

Da mesma que nas seções 6.2.1 e 6.2.2, foi possível encontrar interseções expressivas entre os conjuntos de experimentos com melhores resultados. O conjunto de experimentos #4, #5, #6 e #15 é comum aos três cenários de teste até agora analisados.

A Figura 6.13 ilustra a aerovia gerada pelo processo de delimitação para o experimento #5 no cenário de teste Presidente JK - Congonhas, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino e os pontos em vermelho representam as posições de voo pertencentes ao conjunto *NORM'*.

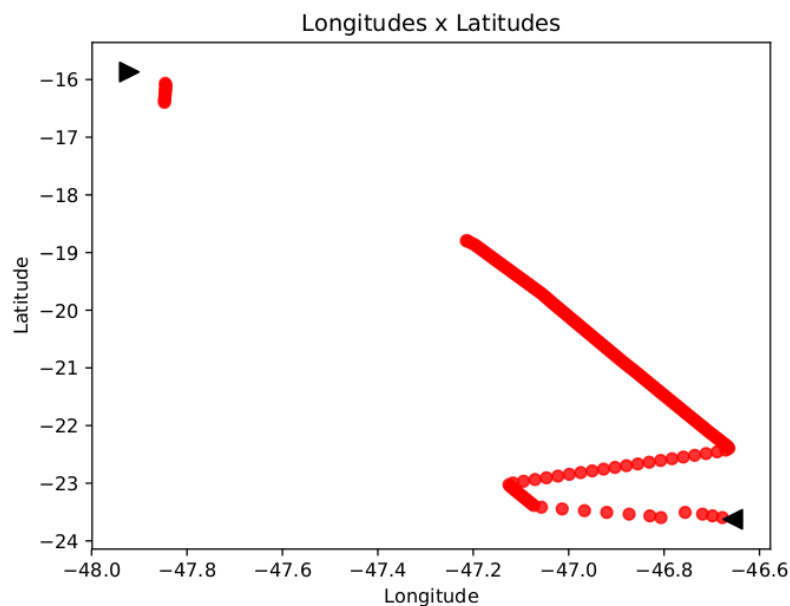


Figura 6.13: Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias.

Processo de Interseção entre Aerovias e Células Convectivas

A Tabela 6.13 apresenta a quantidade de ocorrências de células convectivas na região delimitada entre aeroportos de origem e destino durante o período de rastreamento #CC

Tabela 6.13: Processo de interseção entre aerovias e células convectivas no trecho Presidente JK - Congonhas (onde $dist.$ = $distance_measure$, $hav.$ = $havarsine$, $auc.$ = $euclidean$, $pts.$ = min_pts).

Experimento	#CC	#CC'	P
4. $dist. = hav., pts. = 25, \epsilon = 10$	69	0	0,0
5. $dist. = hav., pts. = 25, \epsilon = 100$	69	0	0,0
6. $dist. = hav., pts. = 25, \epsilon = 1000$	69	0	0,0
15. $dist. = auc., pts. = 25, \epsilon = 1000$	69	0	0,0

Tabela 6.14: Processo de normalização realizado no trecho Congonhas - Presidente JK.

#TRACK	#NORM	P
38.563	24.077	0,624

(vide seção 6.1.5). A tabela apresenta também a quantidade de células convectivas $\#CC'$ que intersectaram com as aerovias obtidas do processo de delimitação de aerovias explicado anteriormente. Apenas os experimentos que obtiveram os melhores resultados estão listados na Tabela 6.13.

Como pode se ver, ao contrário das seções 6.2.1 e 6.2.2, neste cenário de teste foram registradas 69 células convectivas na região que delimita os aeroportos Presidente JK e Congonhas. No entanto, as ocorrências das células convectivas foram afastadas das aerovias geradas nos experimentos selecionados acima.

6.2.4 Cenário Congonhas - Presidente JK

Nesta seção são apresentados os resultados para o cenário de teste São Paulo (Congonhas) - Brasília (Presidente JK).

Processo de Normalização

A Tabela 6.14 detalha a quantidade de posições de voo rastreadas $\#TRACK$, a quantidade de posições de voo normalizadas $\#NORM$ e a proporção entre os dois conjuntos P .

As Figura 6.14, Figura 6.15 e Figura 6.16 mostram as posições de voo normalizadas do voo TAM3733 (em 14-05-2019 às 22:11) segundo suas latitudes e longitudes, suas altitudes e suas velocidades, respectivamente, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino, os pontos em vermelho representam as posições de voo rastreadas enquanto que os pontos em azul posições de voo normalizadas.

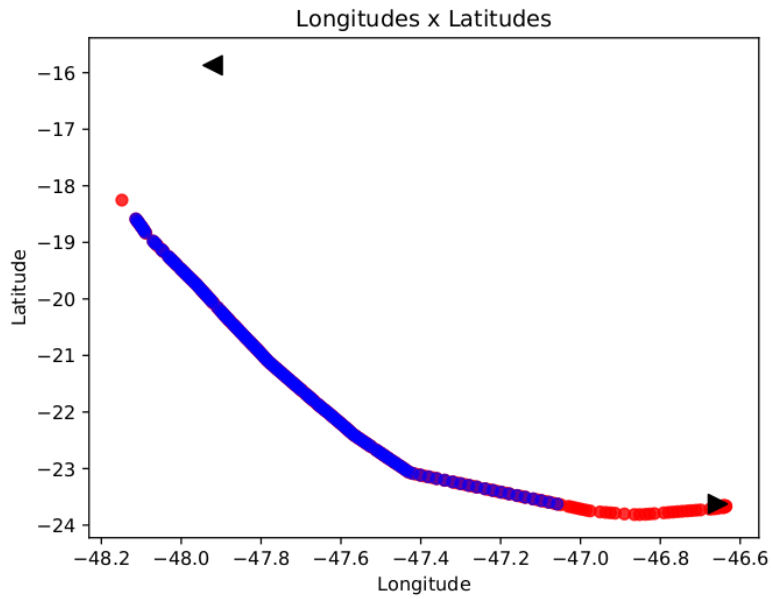


Figura 6.14: Posições do voo TAM3733 em latitudes e longitudes.

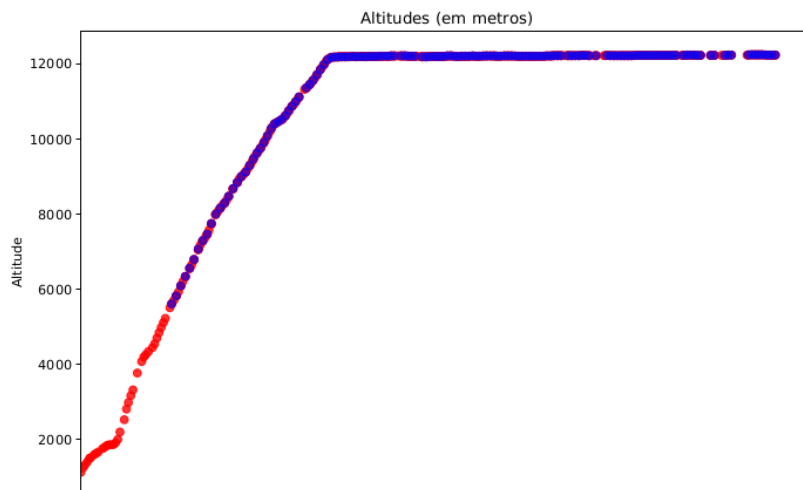


Figura 6.15: Altitudes das posições do voo TAM3733 ao longo do tempo.

Processo de Delimitação de Aerovias

As Tabela 6.15 e Tabela 6.16 apresentam o resultado do processo de delimitação de aerovias para os algoritmos DBSCAN e HDBSCAN, respectivamente. Da mesma forma que na seção 6.2.1, cada linha destas tabelas representa um *experimento* (combinação de parâmetros). A quantidade de posições de voo normalizadas é representada por $\#NORM$ enquanto que a quantidade de posições de voo pertencentes a algum *cluster* é representada por $\#NORM'$, e P é a proporção entre os dois conjuntos. Nas Tabela 6.15 e Tabela 6.16

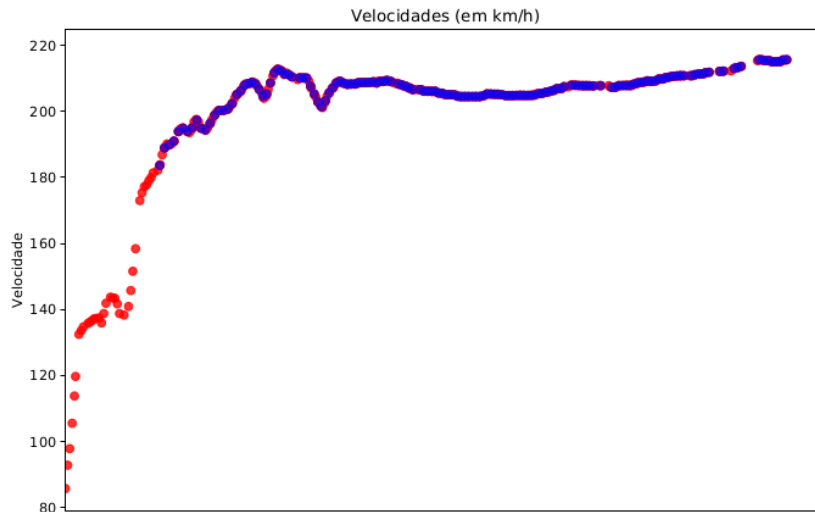


Figura 6.16: Velocidades das posições do voos TAM3733 ao longo do tempo.

também são apresentadas a média μ , mediana **Md**, moda **Mo**, amplitude **R** e desvio padrão σ da distribuição da quantidade de *clusters* N_j de cada partição G_j .

Utilizando os mesmos critérios da seção 6.1.5, os mesmos utilizados na análise do cenário de teste da seção 6.2.1, chega-se à conclusão que os experimentos que mais se aproximariam de uma distribuição uniforme discreta para a sequência formada pelas quantidades de *clusters* de cada partição, são #4, #5, #6 e #15 (todos estes utilizando o algoritmo DBSCAN e marcados em negrito na Tabela 6.15). De mesma forma que nos cenários de teste anteriores, da Tabela 6.15, os experimentos utilizando o algoritmo DBSCAN com *distance_measure = haversine* obtiveram os melhores resultados. Nenhum experimento utilizando HDBSCAN teve uma performance significativa o suficiente para o processo de delimitação de aerovias.

Novamente o mesmo conjunto de experimentos bem sucedidos no cenário de teste da seção 6.2.3 é o conjunto que obteve sucesso no presente cenário de teste. Dessa forma, os parâmetros escolhidos como os melhores para as aerovias conectando o aeroporto A ao aeroporto B são os mesmos para as aerovias conectando o aeroporto B ao A , colaborando para a identificação de um algoritmo de caráter genérico utilizando valores constantes para seus parâmetros.

A Figura 6.17 ilustra a aerovia gerada pelo processo de delimitação para o experimento #5 no cenário de teste Congonhas - Presidente JK, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino e os pontos em vermelho representam as posições de voo pertencentes ao conjunto $NORM'$.

Tabela 6.15: Experimentos do processo de delimitação de aerovias utilizando DBSCAN no trecho Congonhas - Presidente JK (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
1. <i>dist. = hav., pts. = 5, $\epsilon = 10$</i>	24077	18276	0,759	1,06	1	1	6	0,854
2. <i>dist. = hav., pts. = 5, $\epsilon = 100$</i>	24077	19884	0,826	1,022	1	1	4	0,757
3. <i>dist. = hav., pts. = 5, $\epsilon = 1000$</i>	24077	22349	0,928	1,345	1	1	7	1,183
4. <i>dist. = hav., pts. = 25, $\epsilon = 10$</i>	24077	14913	0,619	0,655	1	1	1	0,475
5. <i>dist. = hav., pts. = 25, $\epsilon = 100$</i>	24077	19275	0,801	0,741	1	1	2	0,476
6. <i>dist. = hav., pts. = 25, $\epsilon = 1000$</i>	24077	20373	0,846	0,759	1	1	2	0,502
7. <i>dist. = hav., pts. = 125, $\epsilon = 10$</i>	24077	0	0,0	0,0	0	0	0	0,0
8. <i>dist. = hav., pts. = 125, $\epsilon = 100$</i>	24077	10500	0,436	0,332	0	0	1	0,471
9. <i>dist. = hav., pts. = 125, $\epsilon = 1000$</i>	24077	12179	0,506	0,384	0	0	1	0,486
10. <i>dist. = euc., pts. = 5, $\epsilon = 10$</i>	24077	3938	0,164	1,552	1	0	5	1,719
11. <i>dist. = euc., pts. = 5, $\epsilon = 100$</i>	24077	18112	0,752	2,845	4	4	7	1,867
12. <i>dist. = euc., pts. = 5, $\epsilon = 1000$</i>	24077	21467	0,892	1,237	1	1	6	1,034
13. <i>dist. = euc., pts. = 25, $\epsilon = 10$</i>	24077	0	0,0	0,0	0	0	0	0,0
14. <i>dist. = euc., pts. = 25, $\epsilon = 100$</i>	24077	11294	0,469	1,013	1	2	2	0,958
15. <i>dist. = euc., pts. = 25, $\epsilon = 1000$</i>	24077	19800	0,822	0,737	1	1	2	0,469
16. <i>dist. = euc., pts. = 125, $\epsilon = 10$</i>	24077	0	0,0	0,0	0	0	0	0,0
17. <i>dist. = euc., pts. = 125, $\epsilon = 100$</i>	24077	0	0,0	0,0	0	0	0	0,0
18. <i>dist. = euc., pts. = 125, $\epsilon = 1000$</i>	24077	5447	0,226	0,177	0	0	1	0,381

Tabela 6.16: Experimentos do processo de delimitação de aerovias utilizando HDSCAN no trecho Congonhas - Presidente JK (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
19. <i>dist. = hav., pts. = 5, $\epsilon = 1000$</i>	24077	16705	0,694	4,44	4	0	11	3,456
20. <i>dist. = hav., pts. = 25, $\epsilon = 1000$</i>	24077	4020	0,167	0,487	0	0	3	0,881
21. <i>dist. = hav., pts. = 125, $\epsilon = 1000$</i>	24077	0	0,0	0,0	0	0	0	0,0
22. <i>dist. = euc., pts. = 5, $\epsilon = 1000$</i>	24077	20508	0,852	3,026	4	4	9	2,049
23. <i>dist. = euc., pts. = 25, $\epsilon = 1000$</i>	24077	14075	0,585	1,22	2	2	3	0,982
24. <i>dist. = euc., pts. = 125, $\epsilon = 1000$</i>	24077	0	0,0	0,0	0	0	0	0,0

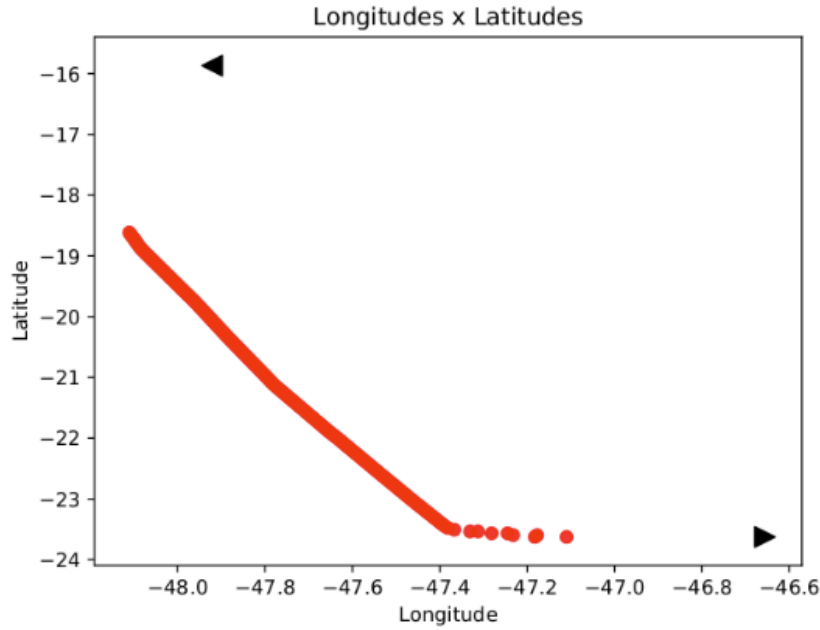


Figura 6.17: Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias.

Tabela 6.17: Processo de interseção entre aerovias e células convectivas no trecho Congonhas - Presidente JK (onde *dist.* = *distance_measure*, *hav.* = *haversine*, *euc.* = *euclidean*, *pts.* = *min_pts*).

Experimento	#CC	#CC'	P
4. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, $\epsilon = 10$	69	5	0,072
5. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, $\epsilon = 100$	69	5	0,072
6. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, $\epsilon = 1000$	69	5	0,072
15. <i>dist.</i> = <i>euc.</i> , <i>pts.</i> = 25, $\epsilon = 1000$	69	5	0,072

Processo de Interseção entre Aerovias e Células Convectivas

A Tabela 6.17 apresenta a quantidade de ocorrências de células convectivas na região delimitada entre aeroportos de origem e destino durante o período de rastreamento #CC (vide seção 6.1.5). A tabela apresenta também a quantidade de células convectivas #CC' que intersectaram com as aerovias obtidas do processo de delimitação de aerovias explicado anteriormente. Apenas os experimentos que obtiveram os melhores resultados estão listados na Tabela 6.17.

Da Tabela 6.17, as mesmas 69 células convectivas CC encontram presentes na região que delimita os aeroportos de Congonhas e Presidente JK. No entanto, desta vez, houve interseções entre as células convectivas rastreadas e as aerovias geradas nos experimentos #4, #5, #6 e #15. Para todos os experimentos foram encontradas cinco ocorrências de células convectivas intersectando as aerovias geradas.

Tabela 6.18: Processo de normalização realizado no trecho Pinto Martins - Guarulhos.

#TRACK	#NORM	P
41.071	26.207	0,638

Vale ressaltar novamente que a aplicação do processo de interseção só faria sentido num cenário *online*, onde interseções identificadas pelo sistema em operação seriam prontamente informadas às partes interessadas. Apesar dos experimentos não terem sido realizados em tempo real, a simulação desse componente serve, no entanto, para mostrar seu correto funcionamento mesmo que num cenário *offline*.

6.2.5 Cenário Pinto Martins - Guarulhos

Nesta seção são apresentados os resultados para o cenário de teste Fortaleza (Pinto Martins) - São Paulo (Guarulhos).

Processo de Normalização

A Tabela 6.18 detalha a quantidade de posições de voo rastreadas *#TRACK*, a quantidade de posições de voo normalizadas *#NORM* e a proporção entre os dois conjuntos *P*.

As Figura 6.18, Figura 6.19 e Figura 6.20 mostram as posições de voo normalizadas do voo GLO1507 (em 21-05-2019 às 18:21) segundo suas latitudes e longitudes, suas altitudes e suas velocidades, respectivamente, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino, os pontos em vermelho representam as posições de voo rastreadas enquanto que os pontos em azul posições de voo normalizadas.

Neste cenário de teste, percebe-se uma influência mais direta dos pontos cegos (mencionados na seção 6.2.1), isto é, de regiões onde as aeronaves não foram rastreadas pelo serviço de rastreamento utilizado. Conseqüentemente, o processo de normalização também fica comprometido pela redução de posições de voo rastreadas. Uma possível solução para esse problema seria substituir a utilização de uma plataforma gratuita [42] para uma plataforma paga [41].

Processo de Delimitação de Aerovias

As Tabela 6.19 e Tabela 6.20 apresentam o resultado do processo de delimitação de aerovias para os algoritmos DBSCAN e HDBSCAN, respectivamente. Da mesma forma que na seção 6.2.1, cada linha destas tabelas representa um *experimento* (combinação de parâmetros). A quantidade de posições de voo normalizadas é representada por *#NORM*

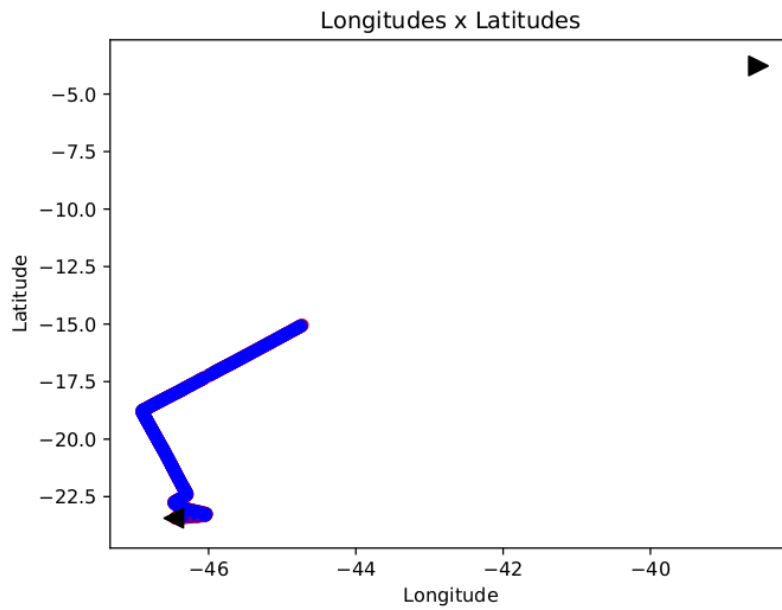


Figura 6.18: Posições do voo GLO1507 em latitudes e longitudes.

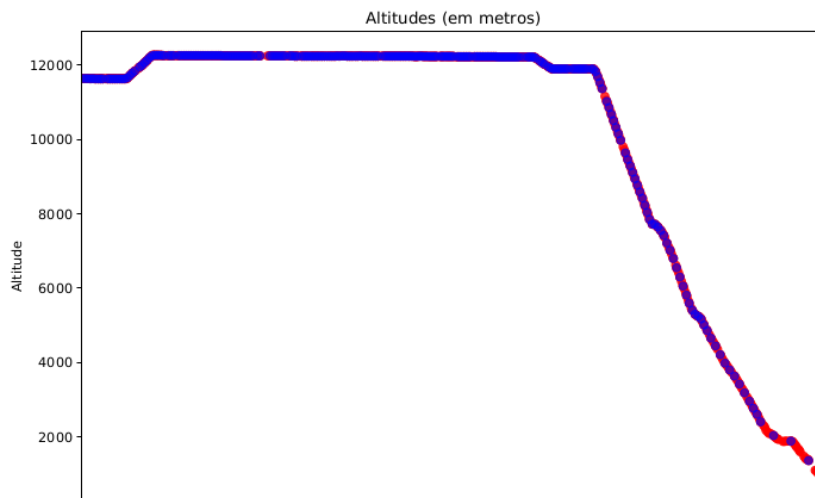


Figura 6.19: Altitudes das posições do voo GLO1507 ao longo do tempo.

enquanto que a quantidade de posições de voo pertencentes a algum *cluster* é representada por $\#NORM'$, e P é a proporção entre os dois conjuntos. Nas Tabela 6.19 e Tabela 6.20 também são apresentadas a média μ , mediana **Md**, moda **Mo**, amplitude **R** e desvio padrão σ da distribuição da quantidade de *clusters* N_j de cada partição G_j .

Utilizando os mesmos critérios da seção 6.1.5, os mesmos utilizados na análise do cenário de teste da seção 6.2.1, chega-se à conclusão que os experimentos que mais se aproximariam de uma distribuição uniforme discreta para a sequência formada pelas quantidades

Tabela 6.19: Experimentos do processo de delimitação de aerovias utilizando DBS-CAN no trecho Pinto Martins - Guarulhos (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
1. <i>dist. = hav., pts. = 5, $\epsilon = 10$</i>	26207	9053	0,345	1,125	1	1	5	0,944
2. <i>dist. = hav., pts. = 5, $\epsilon = 100$</i>	26207	12429	0,474	1,163	1	1	5	0,752
3. <i>dist. = hav., pts. = 5, $\epsilon = 1000$</i>	26207	16444	0,627	1,819	2	1	6	1,203
4. <i>dist. = hav., pts. = 25, $\epsilon = 10$</i>	26207	29	0,001	0,002	0	0	1	0,047
5. <i>dist. = hav., pts. = 25, $\epsilon = 100$</i>	26207	9007	0,344	0,522	1	1	1	0,5
6. <i>dist. = hav., pts. = 25, $\epsilon = 1000$</i>	26207	10843	0,414	0,621	1	1	1	0,485
7. <i>dist. = hav., pts. = 125, $\epsilon = 10$</i>	26207	0	0,0	0,0	0	0	0	0,0
8. <i>dist. = hav., pts. = 125, $\epsilon = 100$</i>	26207	0	0,0	0,0	0	0	0	0,0
9. <i>dist. = hav., pts. = 125, $\epsilon = 1000$</i>	26207	0	0,0	0,0	0	0	0	0,0
10. <i>dist. = euc., pts. = 5, $\epsilon = 10$</i>	26207	169	0,006	0,065	0	0	2	0,264
11. <i>dist. = euc., pts. = 5, $\epsilon = 100$</i>	26207	8572	0,327	1,926	2	0	5	1,601
12. <i>dist. = euc., pts. = 5, $\epsilon = 1000$</i>	26207	15017	0,573	1,518	1	1	5	0,901
13. <i>dist. = euc., pts. = 25, $\epsilon = 10$</i>	26207	0	0,0	0,0	0	0	0	0,0
14. <i>dist. = euc., pts. = 25, $\epsilon = 100$</i>	26207	159	0,006	0,011	0	0	1	0,105
15. <i>dist. = euc., pts. = 25, $\epsilon = 1000$</i>	26207	8682	0,331	0,471	0	0	1	0,499
16. <i>dist. = euc., pts. = 125, $\epsilon = 10$</i>	26207	0	0,0	0,0	0	0	0	0,0
17. <i>dist. = euc., pts. = 125, $\epsilon = 100$</i>	26207	0	0,0	0,0	0	0	0	0,0
18. <i>dist. = euc., pts. = 125, $\epsilon = 1000$</i>	26207	0	0,0	0,0	0	0	0	0,0

Tabela 6.20: Experimentos do processo de delimitação de aerovias utilizando HDBS-CAN no trecho Pinto Martins - Guarulhos (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
19. <i>dist. = hav., pts. = 5, $\epsilon = 1000$</i>	26207	21101	0,805	2,768	3	3	7	1,638
20. <i>dist. = hav., pts. = 25, $\epsilon = 1000$</i>	26207	5020	0,192	0,46	0	0	2	0,842
21. <i>dist. = hav., pts. = 125, $\epsilon = 1000$</i>	26207	0	0,0	0,0	0	0	0	0,0
22. <i>dist. = euc., pts. = 5, $\epsilon = 1000$</i>	26207	20485	0,782	3,362	4	4	8	1,833
23. <i>dist. = euc., pts. = 25, $\epsilon = 1000$</i>	26207	4881	0,186	0,438	0	0	2	0,827
24. <i>dist. = euc., pts. = 125, $\epsilon = 1000$</i>	26207	0	0,0	0,0	0	0	0	0,0

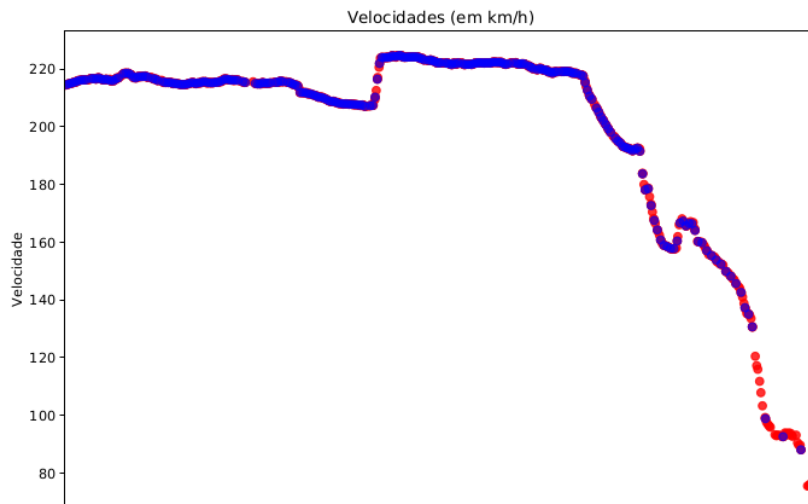


Figura 6.20: Velocidades das posições do voos GLO1507 ao longo do tempo.

de *clusters* de cada partição, são #5 e #6 (todos estes utilizando o algoritmo DBSCAN com *distance_measure = haversine* e marcados em negrito na Tabela 6.19).

Neste cenário de teste, o conjunto de experimentos otimizados é reduzido, no entanto, ainda é um subconjunto dos cenários de teste das seções anteriores. Os experimentos #5 e #6 aparecem em todos os conjuntos de parâmetros bem sucedidos, sugerindo, assim, a existência de valores constantes para os parâmetros que otimizariam qualquer cenário de teste.

A Figura 6.21 ilustra a aerovia gerada pelo processo de delimitação para o experimento #5 no cenário de teste Pinto Martins - São Paulo, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino e os pontos em vermelho representam as posições de voo pertencentes ao conjunto *NORM'*. Da Figura 6.21, pode-se constatar a influência do processo de rastreamento possui no restante dos processos. O processo foi indiretamente prejudicado pela falha no rastreamento.

Processo de Interseção entre Aerovias e Células Convectivas

A Tabela 6.21 apresenta a quantidade de ocorrências de células convectivas na região delimitada entre aeroportos de origem e destino durante o período de rastreamento #*CC* (vide seção 6.1.5). A tabela apresenta também a quantidade de células convectivas #*CC'* que intersectaram com as aerovias obtidas do processo de delimitação de aerovias explicado anteriormente. Apenas os experimentos que obtiveram os melhores resultados estão listados na Tabela 6.21.

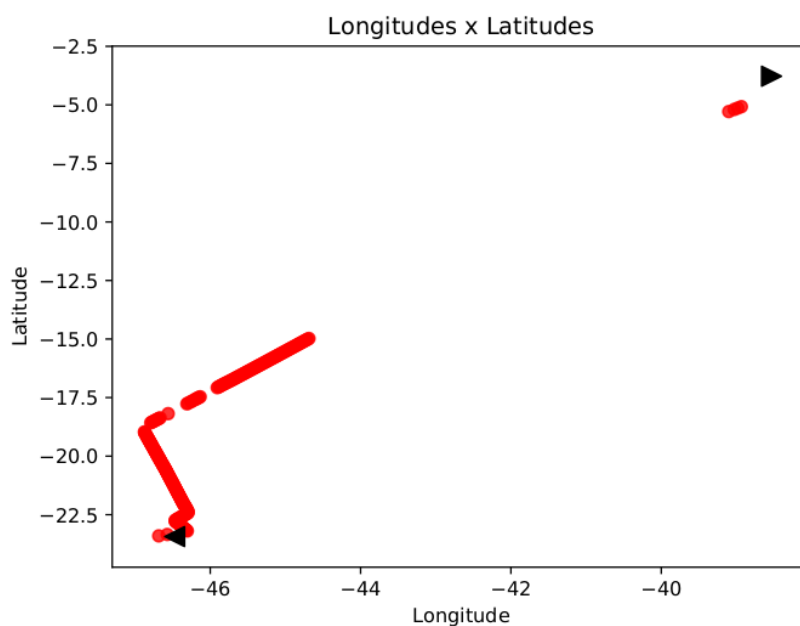


Figura 6.21: Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias.

Tabela 6.21: Processo de interseção entre aerovias e células convectivas no trecho Pinto Martins - Guarulhos (onde *dist.* = *distance_measure*, *hav.* = *haversine*, *auc.* = *euclidean*, *pts.* = *min_pts*).

Experimento	#CC	#CC'	P
5. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, ϵ = 100	876	0	0,0
6. <i>dist.</i> = <i>hav.</i> , <i>pts.</i> = 25, ϵ = 100	876	0	0,0

Neste cenário de teste foram *registradas* 876 células convectivas na região que delimita os aeroportos de Pinto Martins e Guarulhos. Vale ressaltar que esse número é naturalmente maior que os anteriores uma vez que há uma região mais extensa entre os aeroportos Pinto Martins e Guarulhos. No entanto, devido, tanto da falha em rastreamento quanto da posição das células convectivas rastreadas, não foi possível localizar nenhuma interseção entre células convectivas e aerovias geradas para os experimentos selecionados.

6.2.6 Cenário Guarulhos - Pinto Martins

Nesta seção são apresentados os resultados para o cenário de teste São Paulo (Guarulhos) - Fortaleza (Pinto Martins).

Tabela 6.22: Processo de normalização realizado no trecho Guarulhos - Pinto Martins.

#TRACK	#NORM	P
33.735	22.815	0,676

Processo de Normalização

A Tabela 6.22 detalha a quantidade de posições de voo rastreadas $\#TRACK$, a quantidade de posições de voo normalizadas $\#NORM$ e a proporção entre os dois conjuntos P .

As Figura 6.22, Figura 6.23 e Figura 6.24 mostram as posições de voo normalizadas do voo TAM3326 (em 26-05-2019 às 19:05) segundo suas latitudes e longitudes, suas altitudes e suas velocidades, respectivamente, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino, os pontos em vermelho representam as posições de voo rastreadas enquanto que os pontos em azul posições de voo normalizadas.

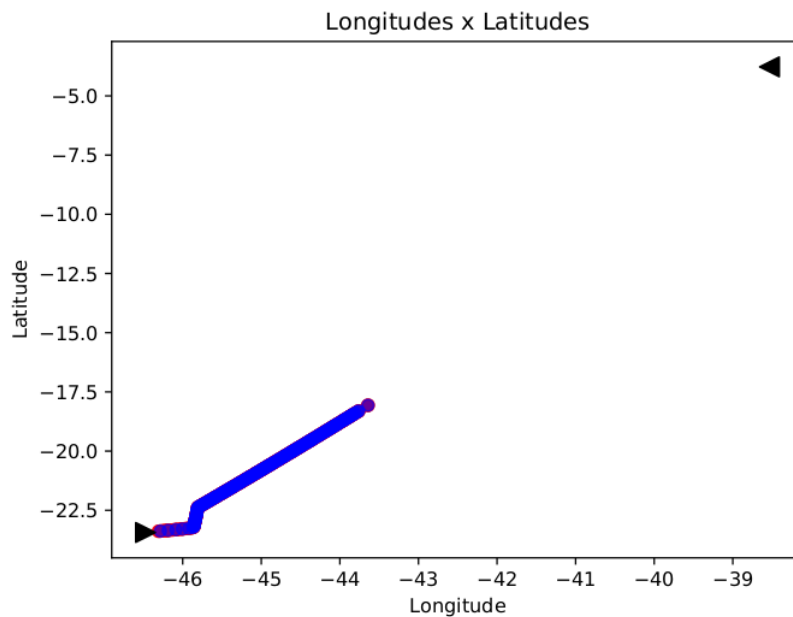


Figura 6.22: Posições do voo TAM3326 em latitudes e longitudes.

Os comentários sobre o processo de normalização aplicados a este cenário de teste seguem a mesma linha de raciocínio dos da seção 6.2.5.

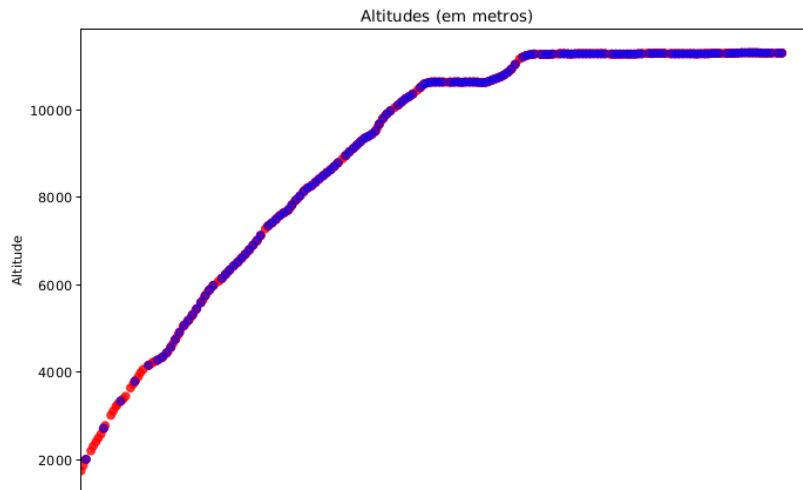


Figura 6.23: Altitudes das posições do voo TAM3326 ao longo do tempo.

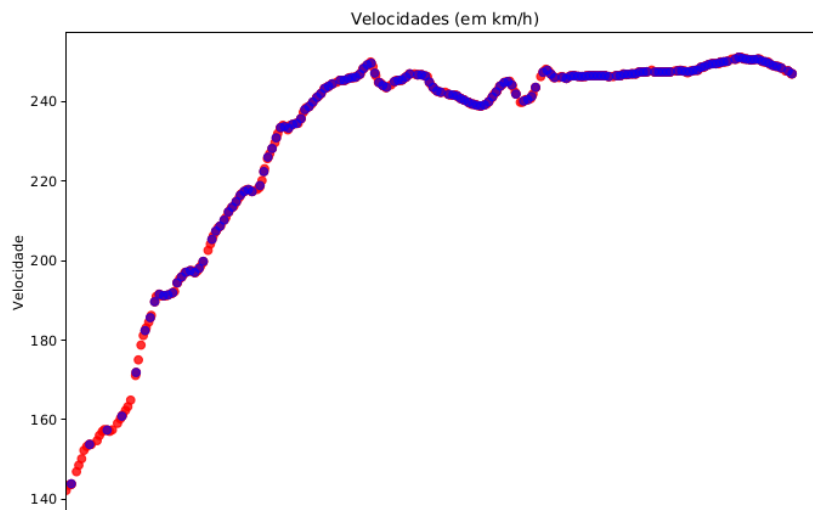


Figura 6.24: Velocidades das posições dos voos TAM3326 ao longo do tempo.

Processo de Delimitação de Aerovias

As Tabela 6.23 e Tabela 6.24 apresentam o resultado do processo de delimitação de aerovias para os algoritmos DBSCAN e HDBSCAN, respectivamente. Da mesma forma que na seção 6.2.1, cada linha destas tabelas representa um *experimento* (combinação de parâmetros). A quantidade de posições de voo normalizadas é representada por $\#NORM$ enquanto que a quantidade de posições de voo pertencentes a algum *cluster* é representada por $\#NORM'$, e P é a proporção entre os dois conjuntos. Nas Tabela 6.23 e Tabela 6.24 também são apresentadas a média μ , mediana **Md**, moda **Mo**, amplitude **R** e desvio

Tabela 6.23: Experimentos do processo de delimitação de aerovias utilizando DBS-CAN no trecho Guarulhos - Pinto Martins (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
1. <i>dist. = hav., pts. = 5, $\epsilon = 10$</i>	22815	4258	0,187	0,924	1	0	4	0,889
2. <i>dist. = hav., pts. = 5, $\epsilon = 100$</i>	22815	8502	0,373	1,624	1	1	7	1,452
3. <i>dist. = hav., pts. = 5, $\epsilon = 1000$</i>	22815	17499	0,767	2,392	2	2	7	1,494
4. <i>dist. = hav., pts. = 25, $\epsilon = 10$</i>	22815	0	0,0	0,0	0	0	0	0,0
5. <i>dist. = hav., pts. = 25, $\epsilon = 100$</i>	22815	2644	0,116	0,221	0	0	1	0,415
6. <i>dist. = hav., pts. = 25, $\epsilon = 1000$</i>	22815	7003	0,307	0,52	1	0	2	0,531
7. <i>dist. = hav., pts. = 125, $\epsilon = 10$</i>	22815	0	0,0	0,0	0	0	0	0,0
8. <i>dist. = hav., pts. = 125, $\epsilon = 100$</i>	22815	0	0,0	0,0	0	0	0	0,0
9. <i>dist. = hav., pts. = 125, $\epsilon = 1000$</i>	22815	0	0,0	0,0	0	0	0	0,0
10. <i>dist. = euc., pts. = 5, $\epsilon = 10$</i>	22815	26	0,001	0,014	0	0	1	0,116
11. <i>dist. = euc., pts. = 5, $\epsilon = 100$</i>	22815	3809	0,167	1,406	1	0	5	1,362
12. <i>dist. = euc., pts. = 5, $\epsilon = 1000$</i>	22815	15157	0,664	1,921	2	2	5	1,144
13. <i>dist. = euc., pts. = 25, $\epsilon = 10$</i>	22815	0	0,0	0,0	0	0	0	0,0
14. <i>dist. = euc., pts. = 25, $\epsilon = 100$</i>	22815	0	0,0	0,0	0	0	0	0,0
15. <i>dist. = euc., pts. = 25, $\epsilon = 1000$</i>	22815	5138	0,225	0,379	0	0	2	0,491
16. <i>dist. = euc., pts. = 125, $\epsilon = 10$</i>	22815	0	0,0	0,0	0	0	0	0,0
17. <i>dist. = euc., pts. = 125, $\epsilon = 100$</i>	22815	0	0,0	0,0	0	0	0	0,0
18. <i>dist. = euc., pts. = 125, $\epsilon = 1000$</i>	22815	0	0,0	0,0	0	0	0	0,0

padrão σ da distribuição da quantidade de *clusters* N_j de cada partição G_j .

Nenhum dos critérios da seção 6.1.5 para os experimentos dessa seção, de forma, que *nenhum experimento* se aproximaria de uma distribuição uniforme discreta para a sequência formada pelas quantidades de *clusters* de cada partição. Novamente, este fato se deve pela qualidade das posições de voo rastreadas entre aeroportos de Guarulhos e Pinto Martins. A presença de pontos cegos no rastreamento influencia diretamente o processo de normalização, que por sua vez influencia indiretamente o processo de delimitação. Essas regiões onde são possíveis rastrear as aeronaves, prejudicam ultimamente na obtenção de *clusters* que fazem parte das aerovias, e dessa forma, também prejudicam na obtenção de uma distribuição uniforme discreta para a sequência de quantidades de *clusters* presentes em cada partição (com a introdução de muitos *zeros* na sequência, onde existem estes pontos cegos).

No entanto, do conjunto de experimentos da Tabela 6.23, dois experimentos #5 e #6 (marcados em negrito) possuem resultados melhores que os demais das Tabela 6.23 e Tabela 6.24. Tendo em vista os resultados das seções anteriores, isto é, que os experimentos #5 e #6 estão entre os experimentos otimizados dos cenários de teste estudados, estes resultados atuais sugerem a existência de valores constantes para os parâmetros que

Tabela 6.24: Experimentos do processo de delimitação de aerovias utilizando HDBS-CAN no trecho Guarulhos - Pinto Martins (onde $N = NORM, N' = NORM', dist. = distance_measure, hav. = haversine, euc. = euclidean, pts. = min_pts$).

Experimento	#N	#N'	P	μ	Md	Mo	R	σ
19. $dist. = hav., pts. = 5, \epsilon = 1000$	22815	17886	0,784	3,027	3	4	8	1,814
20. $dist. = hav., pts. = 25, \epsilon = 1000$	22815	8663	0,38	0,757	0	0	2	0,97
21. $dist. = hav., pts. = 125, \epsilon = 1000$	22815	0	0,0	0,0	0	0	0	0,0
22. $dist. = euc., pts. = 5, \epsilon = 1000$	22815	17335	0,76	3,0	3	4	7	1,908
23. $dist. = euc., pts. = 25, \epsilon = 1000$	22815	8449	0,37	0,73	0	0	2	0,963
24. $dist. = euc., pts. = 125, \epsilon = 1000$	22815	0	0,0	0,0	0	0	0	0,0

otimizariam qualquer cenário de teste.

A Figura 6.25 ilustra a aerovia gerada pelo processo de delimitação para o experimento #5 no cenário de teste São Paulo - Pinto Martins, onde a seta apontando à direita representa o aeroporto de origem, a seta à esquerda o aeroporto de destino e os pontos em vermelho representam as posições de voo pertencentes ao conjunto $NORM'$. Da Figura 6.25, pode-se constatar a influência do processo de rastreamento possui no restante dos processos.

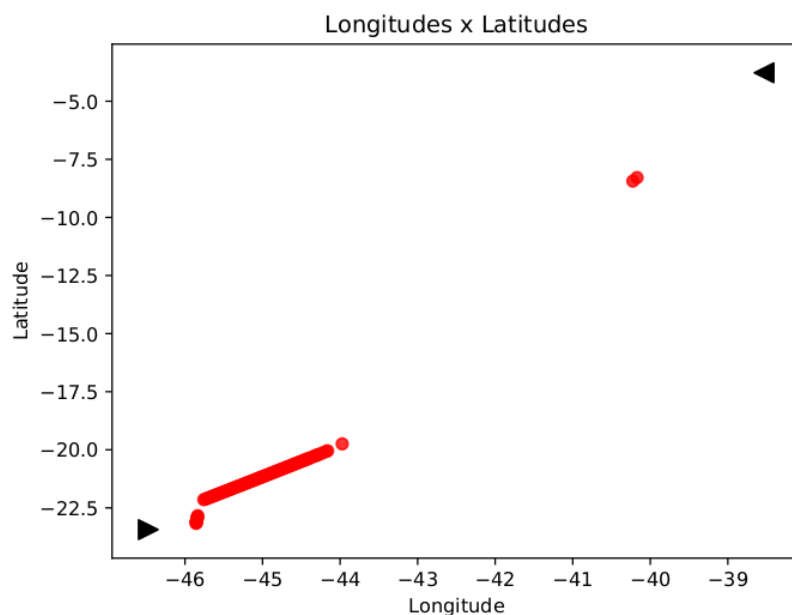


Figura 6.25: Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias.

Tabela 6.25: Processo de interseção entre aerovias e células convectivas no trecho Guarulhos - Pinto Martins (onde $dist.$ = $distance_measure$, $hav.$ = $haversine$, $auc.$ = $euclidean$, $pts.$ = min_pts).

Experimento	#CC	#CC'	P
5. $dist. = hav., pts. = 25, \epsilon = 100$	876	1	0,001
6. $dist. = hav., pts. = 25, \epsilon = 1000$	876	2	0,002

Processo de Interseção entre Aerovias e Células Convectivas

A Tabela 6.25 apresenta a quantidade de ocorrências de células convectivas na região delimitada entre aeroportos de origem e destino durante o período de rastreamento $\#CC$ (vide seção 6.1.5). A tabela apresenta também a quantidade de células convectivas $\#CC'$ que intersectaram com as aerovias obtidas do processo de delimitação de aerovias explicado anteriormente. Apenas os experimentos que obtiveram os melhores resultados estão listados na Tabela 6.25.

Neste cenário de teste foram *registradas* 876 células convectivas na região que delimita os aeroportos de Guarulhos e Pinto Martins (da mesma forma que na seção 6.2.5). No entanto, foram detectados uma interseção entre aerovias e células convectivas no experimento $\#5$ enquanto que duas interseções para o experimento $\#6$. Os mesmos comentários sobre o a falha no rastreamento de posições de voo do cenário da seção 6.2.5 se aplicam para este cenário de teste.

6.3 Discussão e Análise de Resultados

Dos resultados da seção 6.2, é possível traçar algumas conclusões gerais sobre os componentes da solução proposta: processo de rastreamento de posições de voo, normalização de posições de voo, delimitação de aerovias e interseções entre aerovias e células convectivas.

6.3.1 Análise do Processo de Rastreamento de Posições de Voo

O processo de rastreamento de posições de voo é o processo que, de forma geral, influencia todos os outros processos da arquitetura proposta (seção 5.1). Das posições de voo rastreadas, é realizado o processo de normalização de posições de voo, e em seguida a delimitação de aerovias. Dessa forma, o serviço de rastreamento de voo é de suma importância para a qualidade dos demais serviços.

A plataforma utilizada para rastreamento de aeronaves é o OpenSky Network [42]. A plataforma se mostra como uma boa opção por ser de acesso gratuito e possuir uma API de suporte para o rastreamento de aeronaves. Além do mais, por se tratar de um

rede distribuída colaborativa de receptores ADS-B, possui uma boa região de cobertura ao *redor do mundo*.

No entanto, a plataforma por vezes apresenta falhas no rastreamento em determinadas regiões do espaço aéreo brasileiro, mais especificamente na região delimitada entre os aeroportos Pinto Martins e Guarulhos (seções 6.2.5 e 6.2.6). Nestas regiões não são geradas informações sobre as posições de voo das aeronaves, estas regiões são também chamadas no presente trabalho de *pontos cegos*. Isto provavelmente é devido à falta de receptores ADS-B nestas regiões específicas recebendo e compartilhando informações. Dessa forma, a solução proposta não é capaz de identificar as partes das aerovias que por ventura cruzem estas regiões.

Esse problema de falha no rastreamento pode ser percebido claramente nos cenários de teste Pinto Martins - Guarulhos e Guarulhos - Pinto Martins (nas Figura 6.18 e Figura 6.22), onde a maior parte dos trajetos das aeronaves não é rastreado. Isto, em última análise, acaba influenciando o processo de delimitação de aerovias destes cenários de teste, como mostrado na Figura 6.21. Também é possível identificar este problema de forma relativamente significativa nos cenários de teste Brasília - Congonhas e Congonhas - Brasília (nas Figura 6.10, Figura 6.14, Figura 6.13 e Figura 6.17).

6.3.2 Análise do Processo de Normalização de Posições de Voo

O processo de normalização se adapta relativamente bem a eventuais problemas durante o processo de rastreamento de posições de voo. Não é necessário, por exemplo, a presença de *todas* as posições de voo de um dado voo para que o processo de normalização seja bem sucedido. Isto porque, após todas as posições de voo serem normalizadas, no processo de delimitação de aerovias, é executado um algoritmo de *clustering* para cada partição de posições de voo, de forma, que a falta de posições em uma dada partição não deve influenciar de forma significativa a separação de *clusters*. A falta de posições de voo passa a ser um problema maior quando, em vez de ser exceção, passa a ser regra para uma dada partição, de forma que influencia a separação de posições de voo em *clusters*. Daí, segue o problema dos pontos cegos no processo de rastreamento, comentado na seção 6.3.1.

Um cenário onde o processo de normalização não funciona de forma satisfatória ocorre quando a aeronave não segue seu *curso natural e direto* entre aeroportos de origem e destino. Isto é, quando a aerovia realiza uma volta / retorno ao longo da aerovia em que se encontra. Esta situação pode ser melhor percebida no voo ONE6000 entre Congonhas - Santos Dumont na Figura 6.6, onde a aeronave possivelmente retornou ao seu aeroporto de origem após algum tempo de percurso. Outra situação onde esse problema pode ser visto, dessa vez de forma mais amena, é no voo GLO1013 entre Santos Dumont - Congonhas na Figura 6.1.

Tabela 6.26: Resultados do processo de normalização em todos os cenários de teste.

Cenário de teste	#TRACK	#NORM	P
1. Santos Dumont - Congonhas	77.577	39.456	0,509
2. Congonhas - Santos Dumont	44.463	30.247	0,680
3. Presidente JK - Congonhas	48.346	28.809	0,596
4. Congonhas - Presidente JK	38.563	24.077	0,624
5. Pinto Martins - Guarulhos	41.071	26.207	0,638
6. Guarulhos - Pinto Martins	33.735	22.815	0,676

No processo de normalização (Algoritmo 1 da seção 5.3), assume-se que as posições de voo da aeronave sejam sequências *crescentes* ou *decrecentes* para garantir o correto funcionamento do algoritmo de interpolação linear entre duas posições de voo (Algoritmo 2 da seção 5.3). Essa é uma suposição *válida* uma vez que também leva em consideração, implicitamente, o fato de que as aerovias conectando aeroportos de origem e destino são *corredores retos e diretos* (seção 4.1.1). Ou seja, é de se esperar que tanto as latitudes quanto as longitudes das posições de voo pertencentes às aerovias sejam *crescentes* ou *decrecentes*.

Dessa forma, se aeronaves, seja por qual motivo, navegam fora do padrão esperado, o processo de normalização gerará posições de voo normalizadas na forma de ruído (como pode ser visto nas Figura 6.6 e Figura 6.1). No entanto, o ruído gerado no processo de normalização será tratado no processo de delimitação de aerovias na seção 6.3.3 a seguir.

Por fim, o parâmetro *LAT_DIST* utilizado para construir o conjunto de partições *LAT* se mostrou satisfatório. A Tabela 6.26 mostra as proporções obtidas para cada cenário de teste escolhido. Como se pode constatar, a proporção de posições de voo normalizadas ficou entre 51% e 68%. O processo de transformação para posições *normalizadas* não reduziu significativamente os dados coletados originalmente de posições de voo *rastreadas*.

6.3.3 Análise do Processo de Delimitação de Aerovias

No processo de delimitação de aerovias, para cada partição entre aeroportos de origem e destino, as posições de voo normalizadas são separadas em *clusters* e ruído. As posições de voo normalizadas pertencentes a *clusters* são as posições de voo utilizadas para se *delimitar / definir as fronteiras das aerovias entre os aeroportos em questão*.

Os processos de rastreamento e normalização, por vezes, inserem / geram posições de voo na forma de ruído. Dessa forma, o algoritmo de *clustering* utilizado no processo de normalização (Algoritmo 5 da seção 5.5) deve ser capaz de fazer a filtragem de ruído do conjunto de posições de voo iniciais. Dessa forma, o processo de delimitação corrige

eventuais problemas decorrentes dos processos de rastreamento e normalização através da utilização de algoritmos de *clustering* específicos. Algoritmos baseados em densidade, como o DBSCAN e HDBSCAN, foram os escolhidos para realizar este procedimento.

Para cada partição entre aeroportos de origem e destino, o algoritmo de *clustering* é executado e gera, como resultado, a separação das posições de voo em *clusters*. Pela definição de aerovias (corredores retos e diretos no espaço aéreo), é esperado que a quantidade de *clusters* obtida para cada partição seja uma *distribuição uniforme discreta*. É objetivando esta sequência uniforme discreta que os experimentos escolhidos (combinação de algoritmos e seus parâmetros num dado cenário de teste) são avaliados.

Vale ressaltar que dificilmente a sequência de quantidade de *clusters* em cada partição será perfeita. Isso se deve a falhas inerentes do processo de rastreamento e normalização, conforme explicado nas seções 6.3.1 e 6.3.2). No entanto, ainda assim, o processo de delimitação de cada um destes experimentos pode ser comparado a uma distribuição uniforme discreta para: 1) *medir o grau de proximidade* entre o resultado real e o resultado ideal dos experimentos; e 2) *comparar e ordenar os experimentos* uns aos outros.

Segundo os critérios de avaliação da seção 6.1.5, o experimento que obteve os melhores resultados considerando todos os cenários de teste da seção 6.2 é o experimento #5. O experimento #5 é definido utilizando DBSCAN como algoritmo de *clustering* (seção 3.2.4); fórmula *Haversine* como parâmetro para medida de distância $distance_measure = haversine$ (seção 5.4.2); $min_pts = 25$ e $\epsilon = 100$ como parâmetros do algoritmo DBSCAN (seção 6.1.3). Na seção 6.2, experimentos combinando DBSCAN com parâmetro de medida de distância a fórmula *Haversine* obtiveram os resultados mais satisfatórios entre as simulações executadas. Este modelo utilizado é uma das principais contribuições / descobertas deste presente trabalho.

A Tabela 6.27 mostra os resultados do processo de delimitação de aerovias obtidos pelo experimento # 5 para cada cenário de teste analisado. Para a *maioria* dos cenários de teste, o experimento #5 obteve resultados satisfatórios:

1. *Proporção* entre posições de voo pertencentes a *clusters* e posições de voo normalizadas P é relativamente elevada;
2. Valores de *média*, *mediana* e *moda* estão próximos uns dos outros;
3. *Amplitude* (diferença entre o maior e o menor valor da sequência) é relativamente pequena;
4. *Desvio padrão* da sequência também é relativamente pequeno.

Por fim, as Figura 6.26, Figura 6.27 e Figura 6.28 ilustram as aerovias geradas pelo processo de delimitação para o experimento #5 para os seguintes pares de cenários de

Tabela 6.27: Resultados do processo de delimitação de aerovias do experimento #5 em todos os cenários de teste.

Cenário de teste	#N	#N'	P	μ	Md	Mo	R	σ
1. Santos Dumont - Congonhas	39456	29433	0,746	1,017	1	1	3	0,394
2. Congonhas - Santos Dumont	30247	25614	0,847	0,75	1	1	2	0,471
3. Presidente JK - Congonhas	28809	19892	0,69	0,709	1	1	2	0,496
4. Congonhas - Presidente JK	24077	19275	0,801	0,741	1	1	2	0,476
5. Pinto Martins - Guarulhos	26207	9007	0,344	0,522	1	1	1	0,5
6. Guarulhos - Pinto Martins	22815	2644	0,116	0,221	0	0	1	0,415

teste: Congonhas - Santos Dumont e Santos Dumont - Congonhas; Congonhas - Presidente JK e Presidente JK - Congonhas; Guarulhos - Pinto Martins e Pinto Martins - Guarulhos, respectivamente.

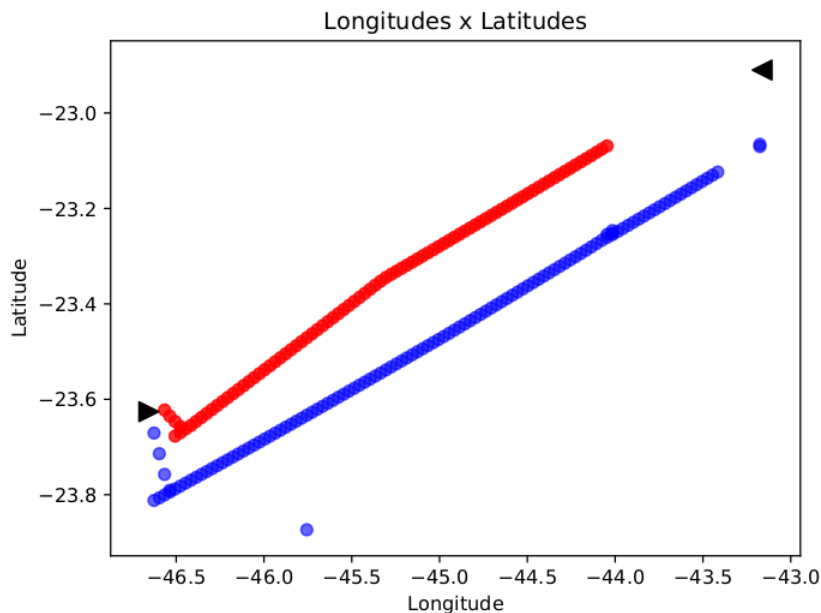


Figura 6.26: Aerovias do experimento #5 geradas durante o processo de delimitação de aerovias (onde os pontos em vermelho representam as posições de voo relativas ao cenário de teste Congonhas - Santos Dumont enquanto os pontos e azul ao cenário Santos Dumont - Congonhas).

Por fim, uma importante observação sobre o processo de delimitação de aerovias é que sua *finalidade* é servir de suporte ao processo de interseções entre células convectivas e aerovias. De posse de uma quantidade suficiente de posições de voo rastreadas pelo processo de rastreamento, as fronteiras que definem as aerovias conectando aeroportos de origem e destino tendem a mudar pouco com o *aumento da quantidade de novas posições de voo rastreadas*. Em outras palavras, uma vez realizado o processo de delimitação de

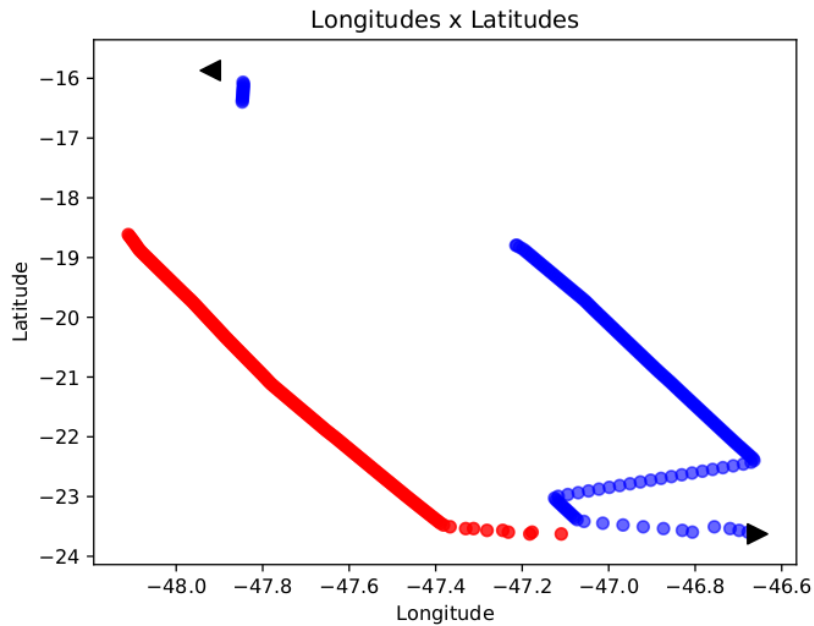


Figura 6.27: Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias (onde os pontos em vermelho representam as posições de voo relativas ao cenário de teste Congonhas - Presidente JK enquanto os pontos e azul ao cenário Presidente JK - Congonhas).

aerovias utilizando um número suficiente de posições de voo, as *aerovias tomam contornos definidos*. Dessa forma, os resultados obtidos nessa etapa, isto é, as posições de voo pertencentes a *clusters*, podem ser deixados em *cache* e reutilizados no procedimento de interseções entre células convectivas e aerovias.

6.3.4 Análise do Processo de Interseções entre Células Convectivas e Aerovias

O processo de interseção entre células convectivas e aerovias é a última etapa da solução proposta, e utiliza diretamente os resultados do processo de delimitação de aerovias e indiretamente os resultados dos processos anteriores. Dessa forma, por ser o último procedimento realizado, está mais suscetível às falhas dos processos anteriores como já foi mostrado na seção 6.2.

Como mencionado na seção 6.3.3, esta etapa pode utilizar como parâmetro de entrada resultados em *cache* das posições de voo pertencentes a *clusters*, economizando assim recursos computacionais. Isto é imprescindível já que o objetivo principal do processo de interseção é funcionar de forma *online*, acompanhando as mudanças das posições das células convectivas em tempo real para verificar possíveis interseções entre estas e as

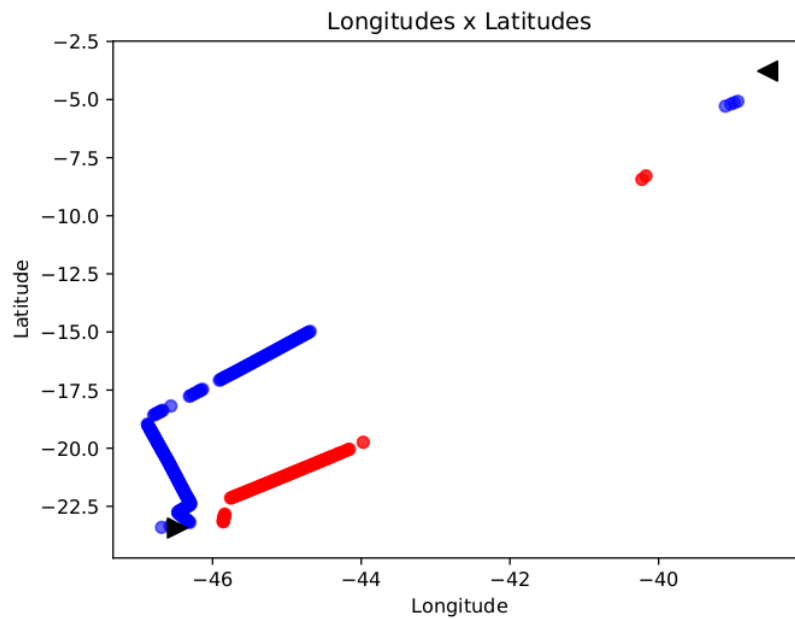


Figura 6.28: Aerovia do experimento #5 gerada durante o processo de delimitação de aerovias (onde os pontos em vermelho representam as posições de voo relativas ao cenário de teste Guarulhos - Pinto Martins enquanto os pontos e azul ao cenário Pinto Martins - Guarulhos).

aerovias do espaço aéreo de forma suficientemente rápida, tendo como finalidade a pronta informação às partes interessadas em ATM.

Pelo mesmo motivo, para funcionar de forma otimizada, os Algoritmos 6, 7 e 8 na seção 5.6 foram sucessivamente refatorados. O Algoritmo 8 na seção 5.6, por exemplo, apresenta uma complexidade computacional de tempo *independente* da quantidade de células convectivas presentes no espaço aéreo no momento da execução do algoritmo. O algoritmo escala igualmente bem tanto para poucas quanto para muitas células convectivas no espaço aéreo.

Capítulo 7

Conclusões e Trabalhos Futuros

No âmbito do Gerenciamento de Tráfego Aéreo, é imperativo o estabelecimento de processos e o desenvolvimento de sistemas que auxiliem *stakeholders* a adotar ações otimizadas visando economizar recursos. Dessa forma, neste trabalho é proposta uma solução para o problema de planejamento de voo sob condições meteorológicas adversas, que se encarrega em auxiliar pilotos na tomada de decisões em navegação aeronáutica. Como consequência direta, a solução aumenta a percepção sobre o espaço aéreo e suas condições climáticas, que em última análise, pode gerar redução de custos (tempo de voo, combustível, etc.) às companhias aéreas.

Neste contexto, foi apresentada uma solução para o problema de planejamento de voo sob condições meteorológicas adversas com a utilização de duas fontes de dados: uma de rastreamento de aeronaves em voo para delimitar aerovias do espaço aéreo e outra para o rastreamento de ocorrências de células convectivas via radares meteorológicos terrestres para acompanhar condições meteorológicas adversas no espaço aéreo brasileiro. A solução proposta tem por objetivo funcionar como um procedimento alternativo à identificação de interseções entre aerovias e células convectivas atualmente realizada pelos pilotos, de forma a ampliar o acesso à informação, e assim, dar um maior suporte à tomada de decisões dos pilotos.

Os resultados obtidos no estudo de caso são promissores, na medida em que foi possível: 1) delimitar as fronteiras das aerovias conectando determinados aeroportos através de transformações feitas em posições de voo inicialmente rastreadas (processos de normalização de posições de voo e de delimitação de aerovias); e 2) encontrar interseções entre células convectivas no espaço aéreo e as aerovias geradas nas etapas anteriores. Dessa forma, com o desenvolvimento do projeto, foi possível aplicar algoritmos de *clustering* de Inteligência Artificial na busca de soluções para o Transporte Aéreo.

No entanto, algumas suposições foram estabelecidas com o intuito de viabilizar a solução proposta. Entre elas, por exemplo, o fato que o processo de normalização de posições

de voo leva em consideração que as posições de voo rastreadas seguem valores de latitude e longitude crescentes ou decrescentes (que provou-se nem sempre ser verdade, como no caso de rastreamento de aeronaves que realizam voltas em suas aerovias). Neste caso, foram utilizados algoritmos que fazem separação de *clusters* e ruídos no processo seguinte de delimitação de aerovias para tratar eventuais problemas decorrentes do processo de normalização. Além do mais, foram percebidas limitações do processo de rastreamento utilizado, onde os resultados finais dependem diretamente da qualidade das posições de voo rastreadas. Assim sendo, a efetiva utilização do sistema concebido requer o aperfeiçoamento da solução proposta, como a utilização de uma melhor fonte para rastreamento de dados, a revisão das limitações encontradas na arquitetura proposta, etc., de forma à auxiliar à tomada de decisões no planejamento de voo sob condições meteorológicas adversas.

7.1 Contribuições e Pesquisas Futuras

Após a conclusão do presente trabalho, é possível listar de forma objetiva as contribuições do estudo realizado, dentre elas estão:

1. Proposição de uma metodologia computacional para o problema de planejamento de voo sob condições meteorológicas adversas;
2. Desenvolvimento de uma arquitetura segundo a metodologia proposta, que possui como componentes: rastreamento de posições de voo, normalização de posições de voo, delimitação de aerovias através da utilização de algoritmos de *clustering*, rastreamento de células convectivas, e interseção entre aerovias e células convectivas;
3. Desenvolvimento de um sistema seguindo a arquitetura proposta com a implementação dos componentes projetados; sistema configurável segundo os modelos e parâmetros utilizados; sistema capaz de realizar simulações utilizando diferentes modelos e parâmetros em diversos cenários de teste; e capaz de reportar os resultados obtidos pelas simulações em métricas pré-definidas;
4. Identificação de modelos e parâmetros ótimos para os componentes da arquitetura proposta através da comparação das simulações realizadas a partir dos diversos experimentos em diferentes cenários de teste, onde se chegou à conclusão, por exemplo, que o modelo de delimitação de aerovias obtém os melhores resultados com a utilização do algoritmo de *clustering* DBSCAN em conjunto com a fórmula *Haversine* como parâmetro para medida de distância;

Para a efetivação das contribuições acima citadas, foram desenvolvidas as seguintes atividades:

1. Rastreamento de aeronaves e suas posições de voo no espaço aéreo brasileiro;
2. Rastreamento de ocorrências e posições geográficas de células convectivas;
3. Levantamento do estado da arte de algoritmos de *clustering*;
4. Definição da arquitetura para identificar aerovias no espaço aéreo e suas interseções com células convectivas rastreadas;
5. Implementação de um sistema utilizando a arquitetura proposta;
6. Implementação de um sistema de suporte para simulação de experimentos;
7. Simulação de diversos experimentos em diferentes cenários de teste;
8. Definição de diferentes métricas para avaliar e comparar experimentos através da análise dos resultados obtidos;
9. Seleção do modelo e parâmetros ótimos segundo análise dos resultados obtidos dos experimentos.

A partir das contribuições apresentadas, sugere-se para pesquisas futuras:

1. A utilização de uma melhor fonte de dados para rastreamento de aeronaves e suas posições de voo; isto pode incorrer em custos adicionais para se ter acesso a diferentes serviços de rastreamento de voos;
2. O desenvolvimento de um processo de normalização de posições de voo capaz de levar em consideração mudanças no trajeto das aeronaves em voo;
3. A utilização de outros algoritmos de *clustering* no processo de delimitação de aerovias no espaço aéreo.

7.2 Considerações Finais

Por fim, pretende-se utilizar este trabalho para auxiliar na tomada de decisões dos *stakeholders* no que se refere ao planejamento de voo sob condições meteorológicas adversas. A implantação do sistema desenvolvido representa o maior desafio para prosseguimento do presente trabalho, ficando a cargo, principalmente, do interesse de empresas do setor de Aviação.

Referências

- [1] REDEMETS: *Sistema de Tempo Severo Convectivo*. <https://www.redemet.aer.mil.br/index.php?i=produtos&p=descargas-atmosfericas>, acesso em 2019-06-07. 1, 3, 29, 37, 56
- [2] REDEMETS: *Sistema de Tempo Severo Convectivo*. <https://www.redemet.aer.mil.br/stsc/public/produto#>, acesso em 2019-06-07. 2, 29, 37, 38
- [3] DECEA: *CB - Cúmulo-Nimbo*. <https://www.decea.gov.br/sirius/index.php/2011/06/14/cb-cumulo-nimbo/>, acesso em 2019-06-07. 3, 29
- [4] SESARWPE: *Dynamic Route Optimization Based on Adverse Weather Data*. https://www.sesarju.eu/sites/default/files/SID_2014-13.pdf, acesso em 2019-06-07. 7
- [5] Cheung, Jacob C. H.: *Flight planning: node-based trajectory prediction and turbulence avoidance*. *Meteorological Applications*, 25(1):78–85, 2018, ISSN 1469-8080. <http://dx.doi.org/10.1002/met.1671>. 8
- [6] Ayhan, Samet e Hanan Samet: *Aircraft trajectory prediction made easy with predictive analytics*. Em *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, páginas 21–30, New York, NY, USA, 2016. ACM, ISBN 978-1-4503-4232-2. <http://doi.acm.org/10.1145/2939672.2939694>. 9
- [7] Ayhan, Samet, Pablo Costas e Hanan Samet: *Predicting estimated time of arrival for commercial flights*. Em *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, páginas 33–42, New York, NY, USA, 2018. ACM, ISBN 978-1-4503-5552-0. <http://doi.acm.org/10.1145/3219819.3219874>. 9
- [8] Girardet, Brunilde, Laurent Lapasset, Daniel Delahaye, Christophe Rabut e Yohann Brenier: *Generating optimal aircraft trajectories with respect to weather conditions*. Em *ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, Toulouse, France, julho 2013. <https://hal-enac.archives-ouvertes.fr/hal-00867818>. 10
- [9] Majumder, S. e M. S. Prasad: *Flight path optimization based on obstacles and weather updates*. Em *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, páginas 425–430, Feb 2016. 10

- [10] Han, Jiawei, Micheline Kamber e Jian Pei: *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edição, 2011, ISBN 0123814790, 9780123814791. 12, 13, 14, 15, 16, 20, 21
- [11] *Comparing python clustering algorithms*. https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html, acesso em 2019-06-07. 18, 19, 22
- [12] *How hdbscan works*. https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html, acesso em 2019-06-07. 21, 22, 23
- [13] Martel, Chip: *The expected complexity of prim's minimum spanning tree algorithm*. Inf. Process. Lett., 81(4):197–201, fevereiro 2002, ISSN 0020-0190. [http://dx.doi.org/10.1016/S0020-0190\(01\)00220-4](http://dx.doi.org/10.1016/S0020-0190(01)00220-4). 22
- [14] *Benchmarking performance and scaling of python clustering algorithms*. https://hdbscan.readthedocs.io/en/latest/performance_and_scalability.html, acesso em 2019-06-07. 23
- [15] Departamento de Controle do Espaço Aéreo: *Concepção Operacional ATM Nacional*. <https://publicacoes.decea.gov.br/?i=publicacao&id=3678>, acesso em 2019-06-07. 24, 30
- [16] Dodder, Rebecca S., Joseph M. Sussman e Joshua B. Mcconnell: *The concept of the “cliros process”: Integrating the study of physical and policy systems using mexico city as an example*, 2004. 24
- [17] Departamento de Controle do Espaço Aéreo: *Requisitos para Operação VFR ou IFR em Aeródromos*. <https://publicacoes.decea.gov.br/?i=publicacao&id=4592>, acesso em 2019-06-07. 25
- [18] Departamento de Controle do Espaço Aéreo: *Plano de Voo*. <https://publicacoes.decea.gov.br/?i=publicacao&id=4589>, acesso em 2019-06-07. 25
- [19] Canino-Rodríguez, José M., Jesús García Herrero, Juan A. Besada Portas, Antonio G. Ravelo-García, Carlos Manuel Travieso-González e Jesús Bernardino Alonso Hernández: *Human computer interactions in next-generation of aircraft smart navigation management systems: Task analysis and architecture under an agent-oriented methodological approach*. Sensors, 15(3):5228–5250, 2015. <https://doi.org/10.3390/s150305228>. 26
- [20] *Highways do céu: por dentro do fantástico mundo das aerovias - decea*. <https://www.decea.gov.br/blog/?p=319>, acesso em 2019-06-07. 26, 27
- [21] *Qual distância mínima um avião deve manter ao voar acima de outro? - decea*. <https://www.decea.gov.br/blog/?tag=aerovia>, acesso em 2019-06-07. 27
- [22] *Reduced vertical separation minima (rvsm) - skybrary*. [https://www.skybrary.aero/index.php/Reduced_Vertical_Separation_Minima_\(RVSM\)](https://www.skybrary.aero/index.php/Reduced_Vertical_Separation_Minima_(RVSM)), acesso em 2019-06-07. 27, 28

- [23] DECEA: *Concepção de Emprego Operacional de Recursos do DECEA pela Força Aérea Brasileira*. <https://publicacoes.decea.gov.br/?i=publicacao&id=4451>, acesso em 2019-06-07. 27, 28
- [24] Departamento de Controle do Espaço Aéreo: *Meteorologia Aeronáutica*. <https://www.decea.gov.br/?i=atividades&p=meteorologia-aeronautica>, acesso em 2019-06-07. 27
- [25] REDEMET: *REDEMET - Rede de Meteorologia do Comando da Aeronáutica*. <https://www.redemet.aer.mil.br/>, acesso em 2019-06-07. 28, 37
- [26] REDEMET: *STSC - Sistema de Tempo Severo Convectivo*. <https://www.redemet.aer.mil.br/stsc/public/produto>, acesso em 2019-06-07. 28
- [27] *Tudo de pitot e o acidente da air france*. <https://www.monolitonimbus.com.br/tubo-de-pitot-e-acidente-air-france/>, acesso em 2019-06-07. 29
- [28] DECEA: *Procedimentos Especiais para Contingências em Voo na FIR Atlântico*. <https://publicacoes.decea.gov.br/?i=publicacao&id=3377>, acesso em 2019-06-07. 29
- [29] ICAO: *Global Air Navigation Plan*, 2013. https://www.icao.int/publications/Documents/9750_4ed_en.pdf, acesso em 2019-06-07. 30
- [30] ICAO: *Manual on System Wide Information Management (SWIM) Concept, (Doc 10039)*, 2015. <https://www.icao.int/airnavigation/IMP/Documents/SWIM%20Concept%20V2%20Draft%20with%20DISCLAIMER.pdf>, acesso em 2019-06-07. 30
- [31] SESAR: *SWIM Concept of Operations*, 2013. <http://www.eurocontrol.int/publications/sesar-swim-concept-operations>, acesso em 2019-06-07. 31
- [32] DECEA: *Programa SIRIUS*. <https://www.decea.gov.br/sirius/index.php/o-sirius/o-que-e-2/>, acesso em 2019-06-07. 31
- [33] DECEA: *Programa SIRIUS - Benefícios*. <https://www.decea.gov.br/sirius/index.php/o-sirius/beneficios/>, acesso em 2019-06-07. 31
- [34] DECEA: *Uso Flexível do Espaço Aéreo*. <https://www.decea.gov.br/sirius/index.php/2014/02/27/uso-flexivel-do-espaco-aereo-fua-2/>, acesso em 2019-06-07. 32
- [35] DECEA: *Implantação do Gerenciamento do Fluxo do Tráfego Aéreo*. <https://www.decea.gov.br/sirius/index.php/2014/02/27/implementacao-de-gerenciamento-do-fluxo-de-trafego-aereo-atfm/>, acesso em 2019-06-07. 32
- [36] DECEA: *Evolução do Serviço de Informação do Voo*. <https://www.decea.gov.br/sirius/index.php/2014/02/27/evolucao-do-servico-de-informacao-de-voo-nas-fir-re-bs-cw-e-az-e-nas-tma-rj-e-sp>, acesso em 2019-06-07. 32

- [37] DECEA: *Coleta de Dados sobre o Ambiente Meteorológico*. <https://www.decea.gov.br/sirius/index.php/2014/02/27/coleta-de-dados-sobre-o-ambiente-meteorologico/>, acesso em 2019-06-07. 32
- [38] DECEA: *Tratamento de Dados Meteorológicos e Integração de Produtos Meteorológicos ao Sistema ATM Nacional*. <https://www.decea.gov.br/sirius/index.php/2014/02/27/tratamento-de-dados-meteorologicos/>, acesso em 2019-06-07. 33
- [39] DECEA: *DECEA - Vigilância Aérea*. <https://www.decea.gov.br/?i=atividades&p=vigilancia-aerea>, acesso em 2019-06-07. 36
- [40] DECEA: *Entenda os benefícios do ADS-B, sistema de vigilância aérea a ser utilizado em breve no País*. <https://www.decea.gov.br/blog/?p=418>, acesso em 2019-06-07. 36
- [41] Flightradar24: *Flightradar24*. <http://www.flightradar24.com/>, acesso em 2019-06-07. 36, 82
- [42] OpenSky Network: *OpenSky Network*. <https://opensky-network.org/>, acesso em 2019-06-07. 36, 42, 56, 82, 91
- [43] FlightAware: *FlightAware*. <https://flightaware.com/>, acesso em 2019-06-07. 36
- [44] Flightradar24: *Flightradar24 - Global Coverage*. <https://www.flightradar24.com/static/images/pages/flightradar24-global-coverage.jpg>, acesso em 2019-06-07. 37
- [45] Phillips, George M.: *Interpolation and Approximation by Polynomials*. Springer-Verlag New York, 2003. 42
- [46] *iuriramos/dunnotheway*. <https://github.com/iuriramos/DunnoTheWay/>, acesso em 2019-06-07. 44, 56
- [47] *Distance - from wolfram mathworld*. <http://mathworld.wolfram.com/Distance.html>, acesso em 2019-06-07. 45
- [48] *Cartesian coordinates - from wolfram mathworld*. <http://mathworld.wolfram.com/CartesianCoordinates.html>, acesso em 2019-06-07. 45
- [49] *Spherical coordinates - from wolfram mathworld*. <http://mathworld.wolfram.com/SphericalCoordinates.html>, acesso em 2019-06-07. 45
- [50] Chris Veness: *Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript*. <https://www.movable-type.co.uk/scripts/latlong.html>, acesso em 2019-06-07. 46
- [51] *Discrete uniform distribution - from wolfram mathworld*. <http://mathworld.wolfram.com/DiscreteUniformDistribution.html>, acesso em 2019-06-07. 61