



Universidade do Minho

Mestrado Integrado em Engenharia Informática

Agentes Inteligentes

AGENTES E SISTEMAS MULTIAGENTE

Gestão do fluxo de tráfego aéreo

Fase 2

Luís Gomes (A78701)

Joel Rodrigues (A79068)

Elisa Valente (A79093)

5 de Dezembro de 2018

Resumo

A necessidade da existência de sistemas escaláveis para a gestão do fluxo de tráfego aéreo tem vindo a aumentar. Estes devem ter a capacidade de evitar as possíveis colisões e controlar as chegadas aos aeroportos. Para dar resposta a este tipo de problema, surgem diferentes abordagens e entre elas apresentam-se os sistemas multiagente que permitem que cada elemento do sistema seja autónomo, especialmente quando é utilizada uma abordagem descentralizada. Neste documento é apresentada a segunda fase de um projeto que tem por base a simulação de um espaço aéreo e a sua gestão. Assim, após a conclusão da primeira etapa, onde foi definida a abordagem ao problema e respetiva modelação, pretende-se demonstrar a implementação do sistema pensado na fase anterior.

Conteúdo

1	Introdução	4
2	Implementação	5
2.1	Agentes e respetivos comportamentos (<i>Behaviours</i>)	5
2.1.1	Aeronave	5
2.1.2	Aeroporto	8
2.1.3	Interface	9
2.2	Classes de apoio	9
2.3	Cálculo de prioridades	10
2.3.1	Colisão cruzada	10
2.3.2	Colisão em frente	11
2.3.3	Aterragem	12
3	Resultado final	13
4	Conclusão	15

Figuras

1	Algoritmo do cálculo da rota	6
2	Algoritmo para aferir existência de colisão quando os vetores são paralelos.	7
3	Algoritmo para aferir existência de colisão caso os vetores não sejam paralelos.	7
4	Vetor normalizado da direção (v) representado através da componente x e y	10
5	Pesos associados aos atributos com colisão em cruzada.	11
6	Pesos associados aos atributos com colisão em frente.	11
7	Pesos associados aos atributos na fase de aterragem.	12
8	Desvio de tempestades de duas aeronaves e reajuste de rota.	13
9	Alteração de velocidade em colisão cruzada.	13
10	Desvio de rotas na iminência de colisões em frente.	14

1 Introdução

O presente documento, elaborado no âmbito da unidade curricular de Agentes Inteligentes, tem o objetivo de apresentar as decisões tomadas na realização da segunda fase do trabalho prático. Nesta etapa partiu-se para a implementação do sistema multiagente idealizado e modelado na fase anterior. A linguagem utilizada para o desenvolvimento do sistema multiagente é *Java*, sendo utilizada a *framework JADE*, uma vez que esta simplifica a implementação de sistemas multiagentes através de um *middleware* que atende às especificações *FIPA*.

De forma a explicar o processo de implementação, na secção 2.1, são apresentados os agentes que atuam no sistema e os comportamentos (*behaviours*) que cada um deles implementa. De seguida, são mostradas as classes que são utilizadas como apoio ao sistema, que não sendo agentes, têm a intenção de melhorar a sua organização e de o tornar mais realista, 2.2. Exemplos disso são classes como **Ambiente**, **Movimentacao** e **Localizacao**. Com o intuito de explicitar o que se considera mais importante na tomada de decisões, quer no que fazer na iminência de colisões, quer na gestão de aterragens, são apresentados os pesos de cada fator no cálculo de prioridades na secção 2.3.

Como finalização do trabalho, na secção 3 é exibido o resultado final, sendo apresentadas algumas imagens que demonstram o que foi implementado, dando uma visão do seu funcionamento e do tipo de problemas que o sistema resolve.

2 Implementação

Nesta secção serão apresentadas todos os aspetos essenciais para a implementação do sistema multiagente capaz de gerir o tráfego aéreo de forma eficiente. Nesta fase foram codificados todos os diagramas anteriormente desenvolvidos.

2.1 Agentes e respetivos comportamentos (*Behaviours*)

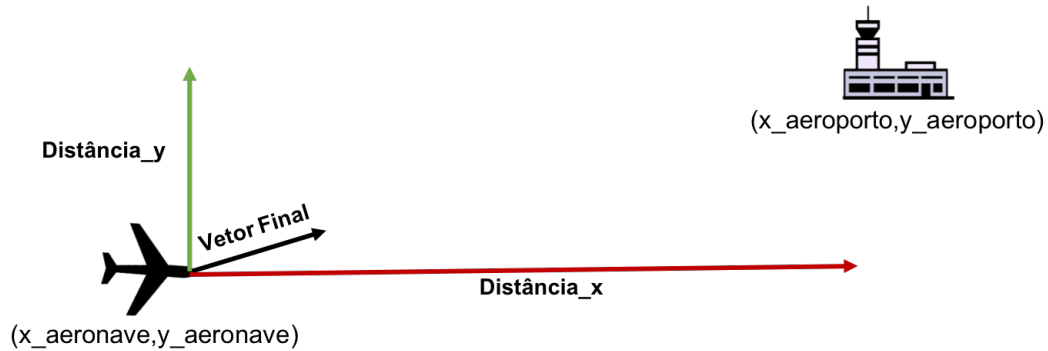
Para o correto funcionamento do sistema foram pensados e desenvolvidos 3 agentes:

- **Aeronave**, representativo dos aviões em voo;
- **Aeroporto**, representativo das estações de origem e destino dos aviões;
- **Interface**, tem por função apresentar ao utilizador uma representação do sistema a simular.

2.1.1 Aeronave

O agente Aeronave tem como principal objetivo realizar viagens completas (origem-destino) sem colidir e do modo mais eficiente possível, minimizando o tempo de chegada e o tempo de voo. Para isso, esta implementa vários comportamentos que são acionados em diversos momentos da viagem:

- **iniciaViagem** (*OneShotBehaviour*) - é iniciado quando uma aeronave é criada e permite à aeronave escolher o aeroporto de destino e calcular a rota que permite alcançá-lo. A figura 1 esquematiza o processo do cálculo da rota, que é efetuado tanto no momento de início da viagem como no recálculo da rota após a resolução de uma colisão.



$$\text{Distância}_x = x_{\text{aeroporto}} - x_{\text{aeronave}}$$

$$\text{Distância}_y = y_{\text{aeroporto}} - y_{\text{aeronave}}$$

$$\text{Módulo} = \sqrt{\text{Distância}_x^2 + \text{Distância}_y^2}$$

$$\text{Vetor Final} = \left(\text{Distância}_x / \text{Módulo}, \text{Distância}_y / \text{Módulo} \right)$$

Figura 1: Algoritmo do cálculo da rota

- **efetuaViagem** (*TickerBehaviour* - 100ms) - é iniciado quando a aeronave recebe permissão do aeroporto de origem para decolar. Neste comportamento são tratadas as ações que uma aeronave deve efetuar ao movimentar-se, incluindo a detecção e tratamento de meteorologia desfavorável na sua rota, deslocação da aeronave (atualização da posição) e verificação de chegada ao aeroporto de destino, caso em que é retirada a permissão de voo.
- **enviaLocalizacao** (*TickerBehaviour* - 100ms) - é iniciado também no momento de decolagem. Envia a localização da aeronave para cada um dos agentes registrados no *DFService* com o tipo "aeronave".
- **detetaColisao** (*CyclicBehaviour*) - é iniciado no momento de decolagem e é este que recebe a mensagem de permissão do aeroporto para decolar. É neste comportamento que os agentes efetuam a maior parte das comunicações. Sempre que o agente aeronave recebe mensagens com a localização das outras aeronaves trata de

calcular a distância a que se encontra delas para verificar a possibilidade de colisão. Caso se detete uma colisão, é desencadeada uma troca de mensagens que levarão à resolução da mesma.

O método responsável por aferir a possibilidade de colisão é baseado no cálculo da função das retas correspondentes aos vetores das direções das aeronaves como exemplificado nas figuras 2 e 3.

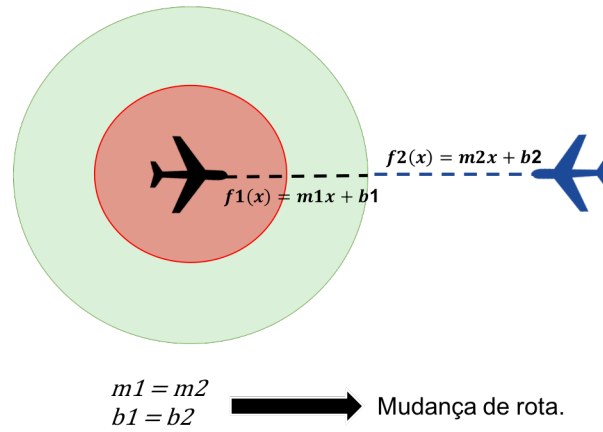


Figura 2: Algoritmo para aferir existência de colisão quando os vetores são paralelos.

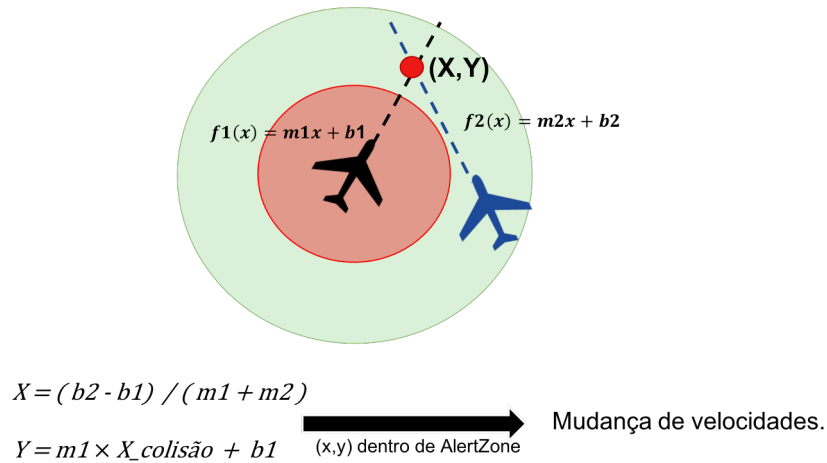


Figura 3: Algoritmo para aferir existência de colisão caso os vetores não sejam paralelos.

- **preparaAterragem** (*TickerBehaviour* - 100ms) - é também iniciado no momento da descolagem e é responsável pelo envio das mensagens durante o processo de aterragem. Este comportamento assenta essencialmente em duas condições, a primeira quando a aeronave se aproxima do aeroporto (*RequestZone*) e neste caso envia uma intenção de aterragem. A segunda quando a aeronave entra na *AlertZone* do aeroporto e nesta situação envia as suas informações para facilitar futuras possíveis tomadas de decisões por parte do aeroporto, a partir deste momento a aeronave esta constantemente a enviar o tempo de chegada à estação destino.

2.1.2 Aeroporto

O agente Aeroporto tem como principal objetivo dar permissão às aeronaves que nele se encontrem estacionadas para descolar e gerir a ordem de aterragem. Como tal, implementa dois comportamentos:

- **PermissaoDescolagem** (*TickerBehaviour* - 5000ms) - é iniciado no momento de criação do aeroporto e é o responsável por enviar as mensagens de permissão de descolagem para as aeronaves que nele se encontrem estacionadas. Estas mensagens são enviadas a cada 5 segundos para não existir a possibilidade de colisão entre aeronaves na descolagem.
- **RecebeTempoChegada** (*CyclicBehaviour*) - é também iniciado no momento de criação do aeroporto e permite a comunicação com as aeronaves na fase de aterragem. Durante essa fase, este agente atualiza as suas estruturas com os dados recebidos, de forma facilitar a tomada de decisão na necessidade de forçar uma aeronave a aguardar pela existência de recursos disponíveis. Este é considerado o comportamento do aeroporto mais importante uma vez que é através deste que se gerem as aterragens.

2.1.3 Interface

É através do agente interface que o utilizador vai interagir com as aeronaves e os aeroportos de forma a observar as tomadas de decisão.

- **recebeLocalizacoes** (*CyclicBehaviour*) - este comportamento é responsável por receber as localizações e direções enviadas pelas aeronaves. Sempre que recebe uma informação nova, este agente atualiza as estruturas internas para que possam ser transmitidas para o objeto encarregue do desenho gráfico do sistema.
- **showInfo** (*OneShotBehaviour*) - este comportamento é responsável pela inicialização do objeto encarregue de apresentação do sistema. Nesta ação são lidas comunicadas as estruturas referidas anteriormente, para que possa desenhar periodicamente as aeronaves com as diferentes localizações e direções.

2.2 Classes de apoio

Para além da criação dos agentes anteriormente apresentados, também foram criadas várias classes de apoio que, não sendo agentes, têm a intenção de melhorar a organização do sistema e de o tornar mais realista. Deste modo, são apresentadas de seguidas as três classes de apoio fundamentais:

- **Ambiente** - Esta classe representa as condições climáticas existentes no sistema. Tem a função de criar e guardar as localizações das tempestades geradas aleatoriamente. Para além disso também possui a lista dos aeroportos existentes no sistema para, por um lado, ser de mais fácil acesso para as aeronaves a escolha do aeroporto destino e por outro a interface conhecer as posições estáticas dos agentes aeroportos.
- **Movimentacao** - Esta classe representa as propriedades da movimentação de uma aeronave, isto é, o vetor normalizado da direção da aeronave em cada instante, nomeadamente a componente em x e y e também a velocidade instantânea da aeronave.

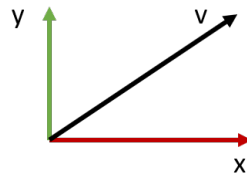


Figura 4: Vetor normalizado da direção (v) representado através da componente x e y .

- **Localizacao** - Esta classe representa as coordenadas geográficas relativas à origem.

2.3 Cálculo de prioridades

Aquando da detecção de uma colisão ocorre uma classificação da mesma, sendo que esta pode ser em cruz ou em frente. Para resolver essa colisão passa-se ao processo de decisão de qual a aeronave que deve sofrer alteração na trajetória. Para além da iminência de colisão, a aeronave também pode sofrer alterações na sua trajetória na fase de aterragem, quando por exemplo, o aeroporto não apresenta pistas de aterragem ou estacionamento disponíveis.

Estes processos de decisão de qual aeronave deve alterar a sua trajetória têm como característica a associação de pesos aos atributos da aeronave, que variam consoante a sua classificação.

2.3.1 Colisão cruzada

Quando se verifica que o tipo de colisão que as aeronaves vão sofrer é cruzada considera-se que o fator mais importantes é a velocidade, com um peso de 45%. Sendo que neste caso, a aeronave que apresentar maior velocidade deve aumentar ainda mais a sua velocidade e a que apresentar menor velocidade deve diminuir.

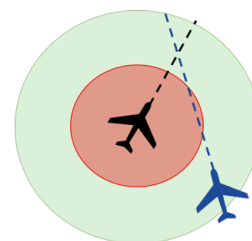
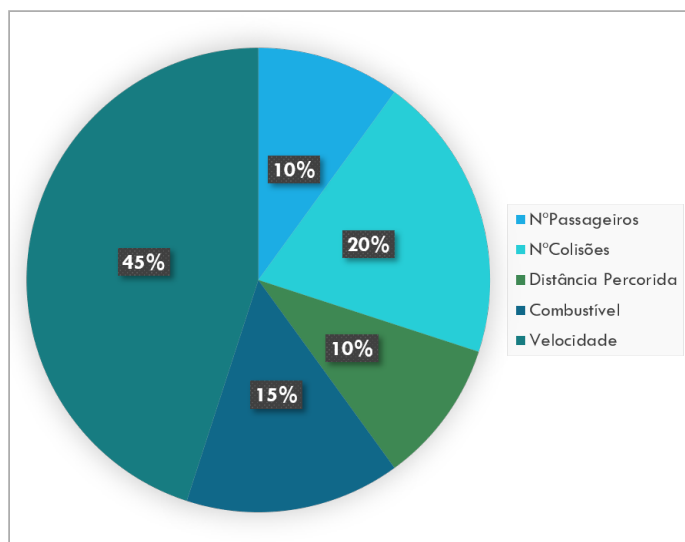


Figura 5: Pesos associados aos atributos com colisão em cruzada.

2.3.2 Colisão em frente

Quando o tipo de colisão detetada é em frente considera-se que os fatores mais importantes são a distância percorrida e o número de colisões (ambos 30%). Isto porque, pretende-se dar prioridade a aeronaves que se encontrem a lidar com outras colisões e ao conforto dos passageiros, porque como se sabe, viagens longas são mais cansativas.

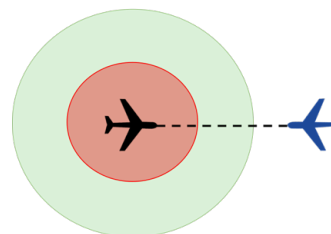
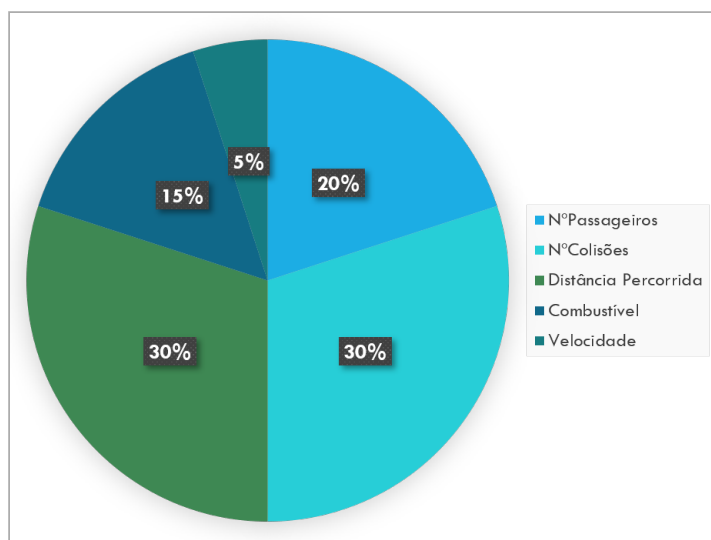


Figura 6: Pesos associados aos atributos com colisão em frente.

2.3.3 Aterragem

Quando se está no processo de aterragem, considera-se que tem prioridade de aterragem aquela aeronave que apresentar menor quantidade de combustível (sendo 40% o peso associado a este atributo).

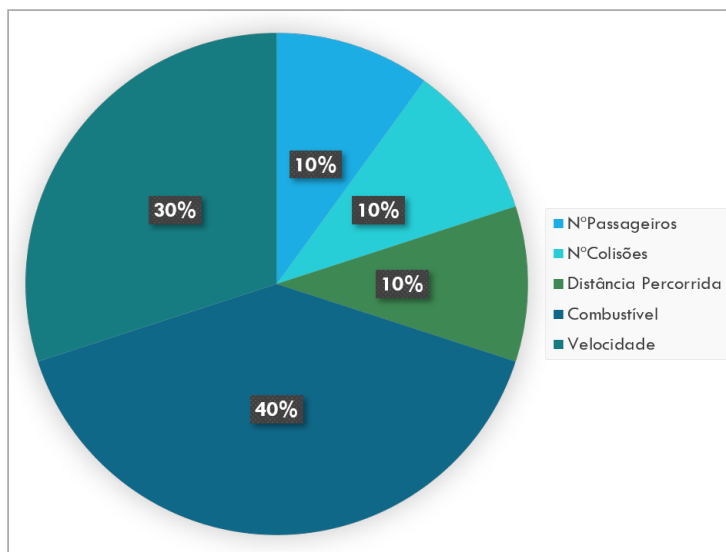


Figura 7: Pesos associados aos atributos na fase de aterragem.

3 Resultado final

Nesta secção são apresentadas algumas imagens do sistema em funcionamento onde se pretende mostrar as decisões tomadas pelas aeronaves em momentos de colisão iminente.

Na figura 8 pode-se aferir que duas aeronaves que tinham uma tempestade no caminho da sua rota, efetuaram alteração da mesma para evitar a passagem nessas zonas de meteorologia adversa.

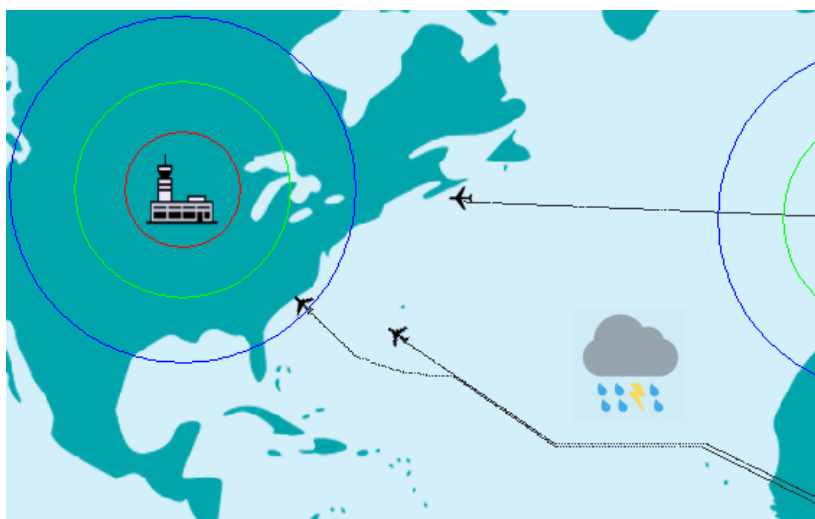


Figura 8: Desvio de tempestades de duas aeronaves e reajuste de rota.

Na figura 9 verifica-se que para evitar a colisão a aeronave 1 aumentou a sua velocidade e a aeronave 2 diminuiu.

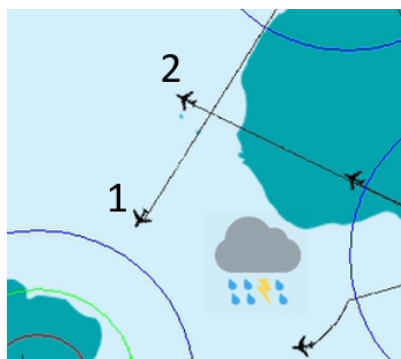


Figura 9: Alteração de velocidade em colisão cruzada.

Na figura 10 confere-se que duas aeronaves foram forçadas a alterar a rota devido a uma possível colisão em frente.

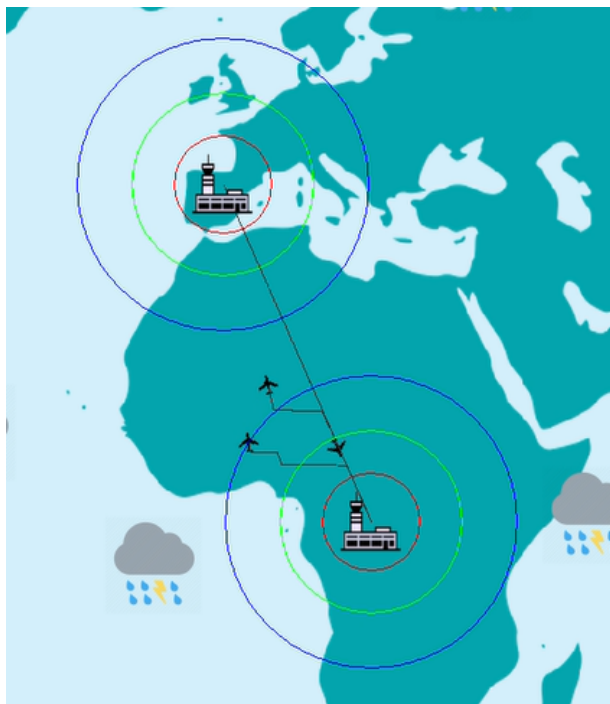


Figura 10: Desvio de rotas na iminência de colisões em frente.

4 Conclusão

Como se tem verificado, existe uma forte componente de investigação na área de sistemas multiagente e na sua aplicação a domínios da área do controlo do fluxo de tráfego aéreo. Este tipo de sistemas permite uma maior autonomia de cada um dos intervenientes aéreos do sistema, uma vez que estes não dependem de entidades de controlo que tomem as decisões de alteração de rotas.

O sistema implementado tem como base principal a modelação efetuada e, como tal, todos os modelos que foram determinados na primeira fase do projeto foram implementados com as devidas modificações que permitiram melhorar o sistema inicialmente pensado.

Apesar de todas as características serem implementadas devidamente, é reconhecida a possibilidade de otimizar muitos dos aspetos, como é o exemplo do cálculo da alteração de rota que está longe de obter uma alteração ótima quando encontra um obstáculo. De ressaltar também que não se consegue mostrar nenhuma imagem da interface que seja capaz de demonstrar a gestão de aterragens por parte dos aeroportos, apesar do seu correto funcionamento.

A partir do trabalho efetuado, seria interessante acrescentar funcionalidades ao sistema que foram pensadas mas não exequíveis no período de tempo em questão. Exemplo disto são o combustível que levaria a uma necessidade de alteração de destino durante o voo.