



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Aplicação do sistema formiga para gerência do sequenciamento de taxiamento em aeroportos

Kamila Borges Nogueira
Paulo Henrique Cabral Aguiar

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientador
Prof. Dr. Li Weigang

Brasília
2013

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Computação — Licenciatura

Coordenador: Prof. Dr. Flávio Vidal

Banca examinadora composta por:

Prof. Dr. Li Weigang (Orientador) — CIC/UnB
Prof.^a Dr.^a Alba Cristina Magalhães Alves de Melo — CIC/UnB
Prof.^a Dr.^a Carla Denise Castanho — CIC/UnB

CIP — Catalogação Internacional na Publicação

Nogueira, Kamila Borges.

Aplicação do sistema formiga para gerência do sequenciamento de taxiamento em aeroportos / Kamila Borges Nogueira, Paulo Henrique Cabral Aguiar. Brasília : UnB, 2013.

113 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2013.

1. algoritmo formiga, 2. otimização, 3. taxiamento, 4. gerenciamento de tráfego aéreo

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

Dedicatória

Dedicamos este trabalho a todas as pessoas que contribuem de alguma forma para o avanço tecnológico do nosso país.

Agradecimentos

Agradeço a minha família pelo apoio dado durante a minha jornada acadêmica: meus pais Carlos e Dália, meus irmãos Karla e Rodrigo e meu sobrinho Calebe. Aos meus amigos e a todos que já me ajudaram nas dificuldades encontradas durante este longo trajeto. Agradeço em especial ao meu namorado Lucas, pelo amor e grande companheirismo durante estas últimas etapas na UnB. Por fim, agradeço à todos os meus professores, os bons e os ruins, pois de alguma forma contribuíram para a minha formação.

Kamila

Agradeço a Deus, aos meus amigos e à minha família por me auxiliarem nessa conquista.

Paulo

Ademais, agradecemos ao nosso orientador Prof. Li Weigang pela oportunidade de realização deste projeto e aos membros da banca, Prof.^a Alba e Prof.^a Carla, pela dedicação à leitura do nosso trabalho e pelas recomendações propostas.

Resumo

Na última década, o fluxo de aeronaves nos aeroportos experimentou um aumento nunca antes visto. Vários estudos e melhorias têm sido feitos no gerenciamento do tráfego aéreo no sentido de aumentar a capacidade do tráfego de aeronaves em rota, e pouco tem sido feito no âmbito da otimização de movimentações em solo. O taxiamento de aeronaves é um processo dispendioso, responsável por causar atrasos e problemas tanto para passageiros, quanto para companhias aéreas. Além disso, por depender principalmente da visibilidade do controlador, pode se tornar perigoso em dias de pouca visibilidade.

Este trabalho apresenta uma solução para sequenciamento do táxi de aeronaves utilizando o algoritmo da colônia de formigas, um método de busca e otimização inspirado no comportamento real de formigas na natureza. As simulações apresentadas foram baseadas em dados reais do Aeroporto Internacional de Brasília, como mapa do aeroporto e planos de voos. Contudo, poderá ser utilizado qualquer aeroporto para cálculos de melhores rotas de taxiamento, sendo necessário somente alterar os arquivos de entrada, que indicam a estrutura aeroportuária e os horários dos voos. O modelo apresentado se mostrou eficiente quanto à otimização do tempo de táxi e ao cumprimento das restrições de segurança.

Palavras-chave: algoritmo formiga, otimização, taxiamento, gerenciamento de tráfego aéreo

Abstract

Over the last decade, the flow of aircraft at airports experienced an increase never seen before. Whereas several improvements have been achieved in enlarging the en-route traffic capacity, little has been done in order to decrease congestion on the airport surface. Taxiing is an expensive process, responsible for causing delays and problems both for passengers and for airlines. Furthermore, because it relies mainly on the air traffic controller's visibility, it may turn into a dangerous process in the days with low visibility.

This paper presents an optimization solution for aircraft taxi-scheduling problem using ant colony, a method for search and optimization inspired by the behavior of real ants in nature. The simulations presented were based on real data from Brasilia International Airport. However, any airport can be used for optimal routes searching, being only necessary to change the input files, which indicate the airport structure and flight schedules. The presented model has shown to be efficient in achieving minimization of aircraft taxi time and complies with security constraints at airports.

Keywords: ant colony algorithm, optimization, taxi, air traffic management

Sumário

1	Introdução	1
1.1	Objetivos	2
1.1.1	Objetivos específicos	2
1.2	Metodologia	2
1.3	Organização do trabalho	3
2	Taxiamento em aeroportos	4
2.1	A importância do transporte aéreo	4
2.2	Conceitos em taxiamento	5
2.3	Problemas relacionados ao taxiamento em aeroportos	8
2.3.1	Procedimentos de táxi	8
2.3.2	Consumo de combustível e poluição	9
2.3.3	Atrasos e grandes eventos	12
2.4	Justificativa do projeto	12
3	Fundamentação teórica	14
3.1	Inspiração biológica	14
3.1.1	Estigmergia	15
3.1.2	O comportamento das formigas	15
3.2	O método de Otimização com colônia de formigas	16
3.3	Subdivisões do ACO	18
3.3.1	Sistema Formiga (AS)	18
3.3.2	Sistema da Colônia de Formigas (ACS)	18
3.4	Aplicações do ACO	19
3.4.1	Problema do Caixeiro Viajante	19
3.4.2	Problema de roteamento de veículos	21
3.4.3	Serviços de comunicação em rede	21
3.4.4	Outras aplicações	22
4	Estado da arte	23
4.1	Utilização do sistema formiga no problema do táxi	23
4.1.1	Descrição do algoritmo	23
4.1.2	Detecção de conflitos	24
4.2	Sistema formiga para previsão de congestionamento no trânsito	25
4.3	Otimização do táxi através de algoritmo genético	27
4.3.1	Codificação de cromossomo	27
4.3.2	Seleção, <i>crossover</i> e mutação	27

5	Solução, modelagem e implementação	29
5.1	Solução proposta	29
5.2	Modelagem e arquitetura	31
5.3	Implementação	34
5.3.1	Otimização de entrada de dados	34
5.3.2	Cálculo de rota para cada avião	34
5.3.3	Detecção e resolução de conflitos	35
6	Simulações e resultados	36
6.1	Planejamento de estudo	36
6.1.1	Caso 1: Fluxo baixo	37
6.1.2	Caso 2: Fluxo médio	38
6.1.3	Caso 3: Fluxo alto	38
6.2	Modificação de parâmetros	38
7	Conclusão	44
	Referências	46

Lista de Figuras

2.1	Rotas mais movimentadas do mundo em número de passageiros.[17]	5
2.2	Processo de decolagem de uma aeronave [24].	7
2.3	Emissões de dióxido de carbono por setor.[25]	10
2.4	Tráfego aéreo e emissão de CO_2 . [27]	11
3.1	Formigas reais escolhendo o caminho mais curto [14]	16
3.2	Formigas artificiais escolhendo o caminho mais curto [14]	17
3.3	Efeito do algoritmo ACO para a constituição do circuito preferencial.	20
4.1	Fluxo de previsão de congestionamento [3]	26
5.1	Mapa do Aeroporto Internacional de Brasília	29
5.2	Arquitetura básica do algoritmo	32
5.3	Busca pela solução ótima	33
6.1	<i>Output</i> da execução do código para fluxo médio de aeronaves	36
6.2	Tempos de taxiamento em horário de baixo fluxo	37
6.3	Tempos de taxiamento em horário de médio fluxo	38
6.4	Tempos de taxiamento em horário de alto fluxo	39
6.5	Alterações no tempo de táxi provocadas por α	39
6.6	Alterações no tempo de táxi provocadas por β	40
6.7	Alterações no tempo de táxi provocadas por Q	41
6.8	Alterações no tempo de táxi provocadas por τ_0	41
6.9	Alterações no tempo de táxi provocadas por ϕ	42
6.10	Alterações no tempo de táxi provocadas por ρ	42

Lista de Tabelas

5.1	Ambiente de implementação e testes	34
6.1	Valores para testes	37

Capítulo 1

Introdução

A otimização da colônia de formigas é definida como uma técnica de busca baseada na metáfora do comportamento organizado de formigas, que constroem a melhor rota entre a fonte de comida e o formigueiro, através do depósito gradual de feromônio no caminho mais curto. É uma técnica heurística de otimização global, que apresenta vantagens com relação a outras técnicas, como o algoritmo genético, por ser mais dinâmico e se adaptar às mudanças em tempo real [15].

O desenvolvimento de sistemas que utilizam o algoritmo formiga tem avançado de forma significativa nos últimos anos, especificamente na área de transportes. Com o aumento da demanda nos aeroportos brasileiros, além da aproximação de um grande evento, que é a copa de 2014, é de grande interesse para a comunidade científica desenvolver soluções para os problemas encontrados nos aeroportos.

De acordo com dados da Agência Nacional de Aviação Civil (ANAC) [26], a aviação brasileira obteve um grande crescimento na última década. Esse crescimento foi de aproximadamente 194%, sendo que somente em 2011 esse setor cresceu quase 8%. Devido a esse crescimento, a expectativa é que ocorreria uma melhoria dos serviços oferecidos, porém não é isso que está sendo observado. O aeroporto de Brasília possui uma média de 8,5% de vôos atrasados [22]. Isso gera prejuízos para as empresas aéreas e a insatisfação de seus usuários.

Os problemas dos usuários foram abordados por especialistas que observaram que as reclamações são frequentes, principalmente quanto a atrasos nos voos, furto e extravio de bagagem. Também foi apontado o aumento de ações na justiça, evidenciando que o atendimento das empresas não tem sido suficiente.

Como solução, foi defendido o aperfeiçoamento de mecanismos administrativos para solução de falhas nos voos, reduzindo conflitos entre empresas e passageiros e evitando que o consumidor tenha que recorrer à justiça para ter seus direitos atendidos, como muitas vezes acontece.

Grande parte dos atrasos nos aeroportos ocorre durante o taximento, definido como a movimentação da aeronave no solo, pela sua própria potência, excluindo-se o processo de pouso e decolagem.

1.1 Objetivos

O objetivo deste trabalho é propor e implementar uma solução para o problema de controle de movimentação de aeronaves no solo dos aeroportos, gerando uma melhora no processo de taxiamento através do algoritmo da otimização da colônia de formigas.

Para cada aeronave será necessário criar uma programação do caminho a ser percorrido do *gate* até a *runway*, de forma a evitar acidentes entre aeronaves e diminuir o tempo de execução do taxiamento.

1.1.1 Objetivos específicos

Os objetivos detalhados podem ser descritos conforme a seguir:

- Desenvolver uma solução através da aplicação do algoritmo formiga visando melhorar o tempo gasto durante o processo de táxi.
- Definir critérios para evitar a ocorrência dos acidentes na pista de táxi.
- Reajustar parâmetros do modelo computacional de forma a aumentar a eficácia do algoritmo.
- Aplicar as soluções para atender melhor à crescente demanda de aeroportos, promovendo serviços que atendam às exigências de segurança.

1.2 Metodologia

A metodologia utilizada para a realização deste trabalho será composta pelas seguintes atividades:

- **Estudo da meta-heurística:** O sistema formiga consiste em um eficiente método de otimização baseado em probabilidade, criado para a solução de problemas que, dentre outros, envolvem a procura de caminhos. É inspirado pelo comportamento real de formigas na natureza, que possuem a habilidade de encontrar o menor caminho entre o ninho e o alimento, através do depósito de feromônio em sua rota [12]. Dessa forma, será de grande valia para resolução dos problemas de taxiamento propostos. Para isso será feito um estudo aprofundado dos princípios que norteiam o desenvolvimento deste sistema, com apresentação de problemas práticos nos quais a aplicação do sistema formiga trouxe bons resultados.
- **Desenvolvimento de pesquisa:** Será realizada uma pesquisa sobre o funcionamento dos aeroportos. Para o desenvolvimento de métodos que visam a melhoria do funcionamento de um aeroporto é necessário conhecer o aeroporto, conhecer como funciona o controle de solo e verificar onde estão os pontos fracos e fortes. Será feita uma coleta de dados referentes a grandes aeroportos para servir de base para o estudo.

Além disso, será feita uma consulta a pesquisas similares que possam auxiliar no trabalho. Como muitos trabalhos que apresentam melhorias para o processo de táxi já foram desenvolvidos, é necessário estudar esses trabalhos para entender o

que falta ser melhorado e em que ponto o algoritmo formiga pode apresentar mais vantagens para este propósito.

- **Implementação:** Por fim, será realizada a implementação e teste de uma aplicação que faça o controle de táxi de maneira automatizada. A linguagem utilizada será C.

1.3 Organização do trabalho

O trabalho está organizado de forma que primeiramente é apresentado o problema a ser tratado, em seguida uma contextualização teórica a respeito do algoritmo utilizado e por fim os resultados encontrados. A divisão em capítulos será da seguinte forma:

- Capítulo 2: Explicação a respeito dos procedimentos para o taxiamento de aeronaves, bem como os problemas encontrados no processo.
- Capítulo 3: Fundamentação teórica a respeito do algoritmo formiga, com explicação a respeito da inspiração biológica e possíveis aplicações do algoritmo.
- Capítulo 4: Pesquisas realizadas até o momento na área de transportes aéreos, utilizando tanto o algoritmo formiga quanto outras metodologias.
- Capítulo 5: Descreve a solução criada, o modelo do projeto e a implementação da solução.
- Capítulo 6: Apresenta os dados que foram utilizados para testar o sistema e os testes que foram realizados.
- Capítulo 7: Trata da conclusão deste trabalho bem como a apresentação de possíveis trabalhos futuros.

Capítulo 2

Taxiamento em aeroportos

Neste capítulo trataremos a respeito dos conceitos que envolvem a aviação, especialmente os tópicos problemáticos que são alvos deste trabalho. Além de uma introdução sobre definições de taxiamento, apresentando os problemas que giram em torno deste processo, serão expostos de forma detalhada os procedimentos que ocorrem na pista de taxiamento (chamada de *taxiway*), as vantagens e a importância da utilização do transporte aéreo, dados a respeito do consumo de combustível e poluição e, finalmente, trataremos a respeito de atrasos e grandes eventos.

2.1 A importância do transporte aéreo

A partir da Segunda Guerra Mundial houve a intensificação do uso de aviões como meio de transporte, pois possibilitam uma viagem rápida, independente do lugar onde se pretende chegar, tendo em vista que, antes realizadas em navios, as viagens duravam meses. Os aviões não desempenham funções unicamente voltadas para o transporte internacional, sendo grande também o fluxo aéreo no interior dos países.

Na década de 50 vieram os jatos que permitiram aumentar ainda mais velocidade e a confiabilidade dos equipamentos e o bom desempenho aliado a preços competitivos e que apresentavam um bom custo benefício com as viagens marítimas, além de um marketing popular, confirmaram o transporte aéreo como um excelente desenvolvedor de turismo, além de máquina propulsora da globalização.

Todos os dias, cerca de 4 milhões de pessoas viajam em aviões ao redor do mundo. Isto leva a aproximadamente 1,7 bilhão de pessoas viajando em 25 milhões de voos por ano. O número de viagens aéreas em 2012 foi maior que em 2011 em todas as regiões do mundo, com significativo aumento de 9% na Ásia e 6% na América Latina. [17]

A Figura 2.1 mostra as 10 rotas aéreas mais movimentadas do mundo, em volume de passageiros.



Figura 2.1: Rotas mais movimentadas do mundo em número de passageiros.[17]

Dentre as vantagens da utilização do transporte aéreo, podemos citar a rapidez para transportar passageiros a médias e grandes distâncias, a grande liberdade de movimentos, oferecendo segurança e comodidade para os passageiros, e o fato de ser o meio mais adequado para o transporte de mercadorias de alto valor e de mercadorias perecíveis.

Ademais, a aviação é de fundamental importância para a economia, conforme demonstrado por alguns dados da *International Air Transport Association (IATA)* [18]:

- 5,3 trilhões de mercadorias são transportadas em aviões - o que corresponde a 35% de todo o comércio mundial, por valor.
- 3,5% da economia global depende da aviação.
- 57 milhões de empregos são gerados pelo setor de aviação.

Contudo, o uso do transporte aéreo também apresenta desvantagens, tais como o impacto ambiental causado pela emissão de dióxido de carbono, poluição sonora nos arredores dos aeroportos, dispendiosa construção de infraestrutura e elevado consumo de combustível, problemas tratados com maior profundidade na seção 2.3

2.2 Conceitos em taxiamento

No procedimento de gerenciamento de tráfego aéreo, as fases de um voo podem ser divididas em três principais estados, de acordo com a movimentação da aeronave:

1. aeronave em solo/decolagem;
2. aeronave em voo;
3. aeronave em aterrissagem.

Para descrição do taxiamento, objeto de análise deste trabalho, enfatizaremos as fases 1 e 3, nas quais acontecem dois níveis primários de controle: torre de controle e controle de aproximação (APP) [24]. Os voos em processo de decolagem são transferidos do nível de torre de controle para o nível de controle de aproximação, já os voos em processo de aterrissagem são transferidos de maneira inversa, do controle de aproximação para a torre de controle.

Os controladores de torre são responsáveis por todo o gerenciamento do tráfego no aeroporto. Esta tarefa é bastante complexa, sendo então dividida em três frentes de controle: alocação (sequenciamento), taxiamento e controle de pista, e funciona da seguinte maneira [1]:

1. Antes de embarcar, após ter feito o plano de voo, o piloto se comunica com a torre pela primeira vez, diretamente da cabine. Ele é atendido pelo controlador de *clearance*, que checa os detalhes do plano de voo, comunica eventuais alterações e passa o código de transponder da aeronave - uma espécie de identificador no espaço aéreo.
2. Já com o código de transponder, o piloto fala com o controlador de solo, que também fica na torre do aeroporto. Ele observa a pista para verificar se o caminho está livre e autoriza o *pushback*: um trator reboca o avião, deixando-o na direção da pista.
3. O piloto liga o motor, segue até uma linha na cabeceira da pista, para e entra em contato com um terceiro controlador na torre para pedir autorização para decolagem. Este controlador precisa garantir que entre cada pouso e decolagem haja um tempo mínimo de segurança, entre 100 e 120 segundos.
4. A cerca de 10 quilômetros do aeroporto, o avião perde contato visual com a torre e passa a ser controlado pelo radar do Controle de Aproximação (APP). A função dos controladores do APP é garantir uma distância mínima entre os aviões nas proximidades do aeroporto.

As tarefas para gerenciamento de pista visam o cumprimento eficiente da agenda das decolagens, sendo imprescindível a elaboração de um plano para cada aeronave. Este plano de decolagem é composto essencialmente pelo planejamento de *gate* (atribuição de posição operacional em solo e início e duração do embarque de passageiros ou carga), pelo plano de taxiamento para o deslocamento da aeronave até a pista de rolagem e pelo plano de *runway*, que efetivamente conduz a aeronave até a cabeceira da pista, de onde ela deixa o solo.

Destas três tarefas de planejamento, destaca-se o plano de *runway*, do qual deriva-se o planejamento de *gate* e o planejamento de taxiamento. Aqui, é feita a especificação de qual pista de rolagem cada aeronave utilizará para decolar, e em que momento. Um dos elementos a se considerar nesta especificação é a esteira de turbulência gerada por aeronaves em operações anteriores de pouso ou decolagem.

O processo de rolagem da aeronave no trajeto compreendido entre a plataforma de embarque/carregamento e a pista de decolagem é justamente o que chamamos de táxi. Portanto, a aeronave está em processo de taxiamento enquanto a mesma se desloca pelo solo entre partes distintas do aeródromo. O processo de taxiamento ocorre tanto após a

aterisagem (*taxi-in*) quanto antes da decolagem (*taxi-out*). O escopo deste trabalho é tanto a partida das aeronaves como o pouso.

A pista de *taxiway* é a via que liga as partes do aeroporto, destinada ao taxiamento de aeronaves. Ela é dividida entre pista de acesso ao estacionamento de aeronaves, pista de táxi no pátio e pista de táxi de saída rápida. O controle deste processo está nas mãos do *Ground Controller*, que é responsável por definir a rota de táxi para cada aeronave, porém na maioria dos aeroportos já existem rotas de táxis pré-definidas.

O processo de táxi durante a aterisagem pode ser resumido da seguinte forma:

1. Uma aeronave que está chegando pede permissão de pouso ao *Runway Controller*.
2. O tempo de aterisagem é estimado como o *Estimated Time of Arrival* (ETA).
3. O *Actual Time of Arrival* (ATA) é definido como o tempo real em que a aeronave toca a pista.
4. A aeronave sai da pista e seu controle é passado para o *Ground Controller*. A aeronave parte para sua plataforma.

O processo de decolagem é feito de forma similar. A aeronave precisa de autorização antes de ligar seus motores, quando ocorre a definição do *Target Time for Departure* (TTD), que é o horário desejado para decolagem e o *Off-Block Time* (OBT), que é o horário para o *pushback* da aeronave, ou seja, o momento em que a mesma abandona sua posição de embarque e inicia a movimentação em solo. A Figura 2.2 mostra os estágios do processo de decolagem de uma aeronave.

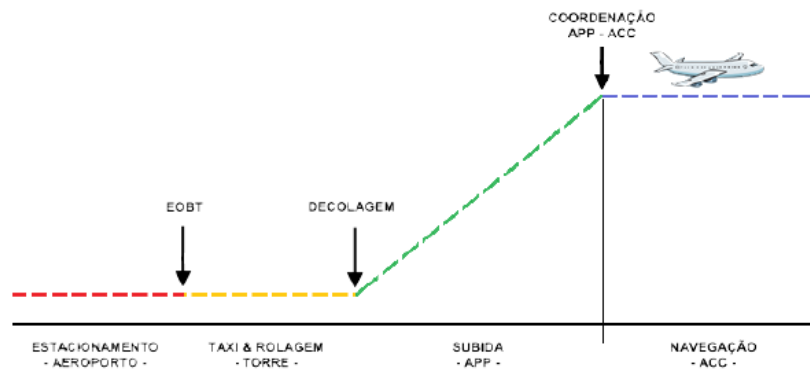


Figura 2.2: Processo de decolagem de uma aeronave [24].

O controlador de solo é responsável pelo tráfego da aeronave neste espaço de tempo, sendo sua função atender às exigências de segurança, evitando colisões que podem acontecer das seguintes formas:

1. Duas aeronaves utilizam rotas diferentes no mesmo instante que se cruzam;
2. Duas aeronaves no mesmo sentido, sendo que a aeronave de trás está numa velocidade maior que a da frente.
3. Duas aeronaves em sentidos opostos, indo uma em direção à outra.

O gargalo do tráfego aéreo está na capacidade dos aeroportos, por isto os aeroportos sempre procuram meios de otimizar os serviços relacionados, como aterrissagem, decolagem, alocação de plataformas e outras. É fundamental encontrar meios de otimizar o táxi, não só melhorando o tempo de espera, mas também visando melhorar o consumo de combustível e a emissão de gases poluentes. As seções seguintes apresentam algumas das muitas pesquisas já realizadas nesta área.

2.3 Problemas relacionados ao taxiamento em aeroportos

Esta seção apresenta os diversos problemas encontrados no taxiamento, como os atrasos provocados por este processo, a poluição vinda dos aeroportos, que exerce um considerável impacto ambiental, o dispendioso consumo de combustível, e a aproximação de grandes eventos no Brasil, que aumentará ainda mais a demanda nos aeroportos. Todos estes fatores serviram de motivação para o desenvolvimento deste trabalho.

2.3.1 Procedimentos de táxi

O taxiamento é o processo de rolagem de uma aeronave com sua própria potência pela superfície do aeródromo, excluindo o pouso e a decolagem [8]. Este processo pode acontecer antes da decolagem quando a aeronave segue da plataforma para a pista de decolagem, ou após a aterrissagem, em que a aeronave segue no sentido inverso.

O atraso nesse processo gera grandes desconfortos para os passageiros e companhias aéreas. Em debate realizado em fevereiro de 2012 [9] entre senadores e representantes do direito ao consumidor, em oposição a representantes de companhias aéreas, foram evidenciadas as deficiências de infraestrutura que impedem os aeroportos de atenderem o avanço da demanda. Desde 2005, o número de passageiros tem aumentado 15% ao ano. Outro assunto em pauta foi a má qualidade da comunicação entre companhia aérea e os usuários. Nos últimos anos, a aviação aérea brasileira tem apresentado diversos problemas, como por exemplo:

- Um índice consideravelmente alto de atrasos dos vôos.
- Pressão em cima dos controladores de tráfego aéreo.
- Crescimento da demanda por vôos e do tráfego aéreo, mas sem uma melhoria proporcional dos serviços e infra-estrutura.

Por ser um sistema extremamente crítico, qualquer erro no controle do tráfego pode gerar grandes atrasos ou até mesmo acidentes graves. O tempo de espera dentro da aeronave pode ser muito grande dependendo de engarrafamentos na pista e nem sempre o controlador de solo toma as melhores decisões para movimentação das aeronaves, gerando desconforto para os passageiros e para a tripulação.

Nos últimos anos, o tráfego aéreo experimentou um crescimento gigantesco. Para acomodar este crescimento e, ao mesmo tempo, atender as exigências de segurança, várias melhorias devem ser feitas. Recentemente, têm sido observado um grande aumento no crescimento da capacidade dos aeroportos para atender a nova demanda.

Este desenvolvimento força aeroportos, linhas aéreas e serviços de controle de tráfego aéreo a modernizar processos individuais do aeroporto, como embarque, desembarque, espera no portão de embarque etc, mas também dedicar esforços para a integração de todos estes processos e sistemas, configurando, assim, uma tomada de decisão colaborativa.

Atualmente, o controle de movimentos de aeronaves no chão (táxi) é feito, em sua maior parte, visualmente, sem qualquer auxílio computacional à tomada de decisão quanto à ordem de partida e tempos de separação entre pousos e decolagens, o que é altamente perigoso em condições de baixa visibilidade. Nesta situação, o manejo do táxi representa uma restrição à capacidade do aeroporto.

Além disso, outra preocupação que vem crescendo cada vez mais é a questão ambiental. O taxiamento aéreo aumenta o nível de emissão de poluentes. A quantidade da emissão desses poluentes é grande, e diminuir o tempo desnecessário de espera, fazendo com que as aeronaves gastem menos combustível significa um maior aproveitamento de combustível e um menor tempo com o motor ligado, diminuindo assim a emissão de gases poluentes.

Uma solução seria aumentar a capacidade do aeroporto, construindo novas pistas de decolagem/aterrisagem, por exemplo. Porém, é preciso considerar que muitas vezes não existe espaço disponível para isso e leva-se em média 3 anos para construir uma nova pista. A rede que representa o solo do aeroporto é dividida em três partes, a parte dos *gates*, a *taxiway* (pista onde ocorre o táxi) e as pistas de decolagem/aterrisagem (*runway*).

O gargalo está justamente no acesso aos *runways* dos aeroportos. É nesta etapa que precisa ocorrer uma melhora no processo de rolagem, pois é necessário melhorar o uso das *runways*, organizando o tráfego nas pistas de táxi.

2.3.2 Consumo de combustível e poluição

Combustíveis para aviões são caros e utilizados em quantidade maior que outros combustíveis que não apresentam tanta criticidade com relação a aquecimento e transporte, além de conter mais aditivos que reduzem o risco de congelar em temperaturas muito baixas, ou explodir em temperaturas muito altas.

Um avião como um Boeing 747, por exemplo, usa aproximadamente 4 litros de combustível a cada segundo. Durante um voo de 10 horas, pode usar 150 mil litros. De acordo com o site da Boeing [16], um 747 queima cerca de 12 litros por quilômetro.

Um dos maiores percentuais nos custos operacionais das empresas aéreas consiste no consumo de combustível. Estima-se que a participação do consumo de combustível atinja 20% ou mais sobre os custos operacionais de qualquer empresa aérea. Parte deste combustível é desperdiçado em aeronaves em espera de saída para táxi, ou em pontos de espera no aguardo para decolagem, ou ainda no transporte de passageiros em uma situação de *gate* afastado da região de embarque ou desembarque.

Uma das grandes preocupações no desenvolvimento de aeronaves modernas é justamente a economia de combustíveis, e novas tecnologias têm permitido que esta meta seja alcançada. Porém, o gasto em solo ainda existirá de forma significativa enquanto não forem aplicadas técnicas que dinamizem o táxi. Na Europa, por exemplo, estima-se um gasto de 10-30% de tempo de voo fazendo táxi, e que aviões de médio porte como A320 gaste entre 5-10% do seu combustível no solo [2].

A maioria das pesquisas se preocupa somente em descobrir meios de diminuir o tempo total de táxi, aumentando a eficiência das operações realizadas no aeroporto. Mas exis-

tem pesquisas que vão além e procuram também maneiras de diminuir o consumo de combustível durante o processo de táxi. Para diminuir este consumo, não é suficiente apenas diminuir o tempo total do táxi, também é necessário se observar a relação entre o consumo de combustível e a velocidade de uma aeronave.

Um grupo de pesquisadores da University of Lincoln [6] procurou analisar estes dois aspectos do táxi: combustível e tempo gastos durante o processo. Através de estudos que envolviam testes de aceleração das aeronaves, foi possível concluir que um menor tempo de táxi não implica em um menor gasto de combustível, pelo contrário, um menor tempo geralmente é conseguido as custas de um excesso de aceleração e desaceleração, processo que faz com que o consumo de combustível cresça ainda mais.

Com o desenvolvimento dos grandes centros urbanos verificou-se que os sistemas de transportes colaboram muito com a degradação da qualidade do ar. Nessa linha está inserido o meio de transporte e a logística necessária para a implementação e realização de cada tipo de transporte.

Apesar de os aeroportos serem construídos em regiões afastadas dos centros das cidades na maioria das vezes, devem ser considerados como uma fonte fixa de poluente atmosférico, e como tal devem ser feitas avaliações de estratégias de controle e gestão da qualidade do ar, bem como de diminuição de ruídos e quando necessário estudos sobre poluição de águas próximas ao aeroporto.

Com os motores ligados em solo, a queima de combustível de diversas aeronaves colabora com o aumento da poluição nas regiões próximas ao aeroporto. Segundo a IATA [18], o transporte aéreo contribui com 2% das emissões de CO_2 feitas por seres humanos, com previsão de 3% até 2050, um valor muito alto para apenas um seguimento. A Figura 2.3 mostra as parcelas que cada setor econômico contribui para emissões de carbono. Na aviação, só de 2010 para 2011 as emissões aumentaram 3.3%, o que significa um aumento de 20 milhões de toneladas. Apesar deste valor ter sido menor que o aumento do tráfego (que foi de 5,3%), ainda é um valor bem alto.

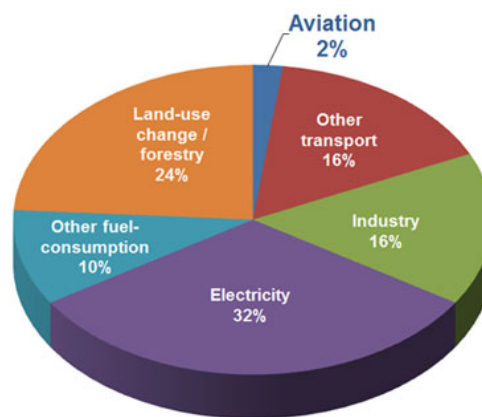


Figura 2.3: Emissões de dióxido de carbono por setor.[25]

Devido a estes valores críticos que ameaçam o meio ambiente, mudanças serão realizadas. Ainda segundo a IATA, o setor de aviação irá gastar 1.5 trilhão de dólares em novas aeronaves até 2020. Cerca de 5.500 aeronaves serão substituídas até 2020, o que corresponde a 27% do total da frota. É esperado que a tecnologia permita uma redução de 21% nas emissões de CO_2 , comparado com as emissões atualmente. A Figura 2.4

mostra a relação entre o tráfego aéreo e as emissões de CO_2 , além de estimativas para os próximos anos.

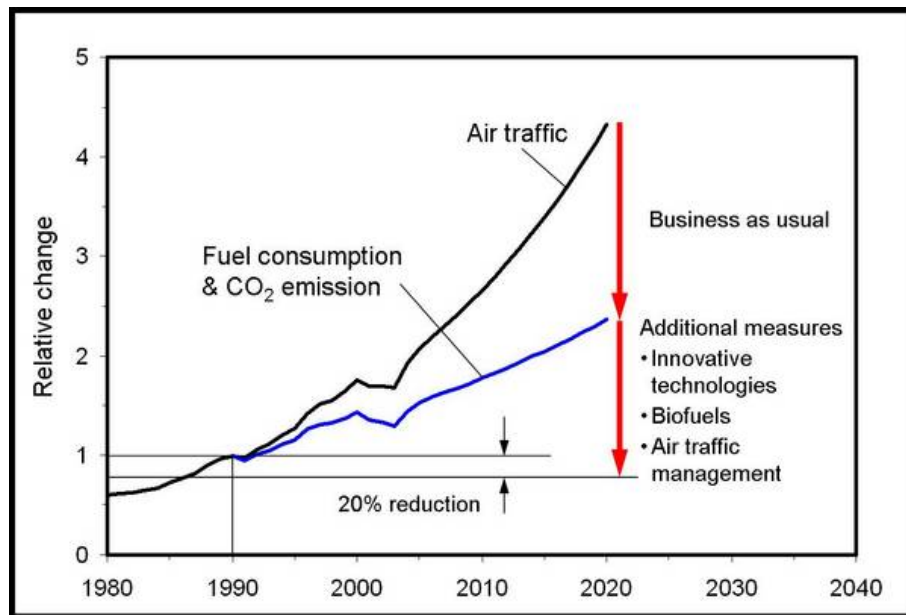


Figura 2.4: Tráfego aéreo e emissão de CO_2 . [27]

Nos Estados Unidos, jatos comerciais anualmente emitem na atmosfera 6 bilhões de quilogramas de dióxido de carbono, 45 milhões de quilogramas de monóxido de carbono, 8 milhões de quilogramas de óxidos nitrogenados e 4 milhões de quilogramas de hidrocarbonetos por ano. Em estudos conduzidos no aeroporto internacional de Boston Logan pelo Instituto de Tecnologia de Massachusetts (MIT), foi demonstrado que fazer os aviões esperarem menos que 5 minutos extras pode ser um jeito simples de reduzir consideravelmente essas emissões.

Existem algumas propostas de como se pode reduzir a quantidade de emissões de poluentes das aeronaves, aplicadas ao aeroporto de Londres:

- Proposta 1: Utilizar somente um dos motores da aeronave durante o processo de táxi. A velocidade durante este processo é a mesma que utilizando dois motores.
- Proposta 2: Rebocar a aeronave para que durante o processo de táxi, os motores da aeronave possam estar desligados.
- Proposta 3: Criar um novo terminal próximo da pista principal, isto reduziria a distância e tempo de táxi em aproximadamente 20%. Este terminal iria receber 75% do total de aeronaves, enquanto o restante iria continuar utilizar o terminal atual.
- Proposta 4: Esta proposta sugere um inovador sistema automático de reboque de aeronaves entre as posições e a pista. Este sistema é formado por um carril fixo no solo, no qual um reboque automático se move. Desta forma, as aeronaves só precisariam ligar seus motores pouco antes da decolagem.

Chegou-se a conclusão de que as propostas 2 e 4 resultariam em uma maior diminuição do consumo de combustível e com isto, uma menor emissão de poluentes, porém ambas propostas aumentam o tempo de táxi e também seria necessário um grande investimento para serem realizadas.

Um grupo do MIT realizou um estudo que analisa a influência que o gerenciamento de filas nos aeroportos possui na emissão de poluentes e consumo de combustível. Eles chegaram a conclusão de que se utilizassem um sistema em que somente um avião pudesse ser liberado para táxi por vez, isto reduziria a emissão de poluentes entre 25% e 65%, porém isto aumentaria o tempo de atraso e diminuiria a vazão dos aeroportos, mas aliar esta estratégia a um gerenciamento adequado de filas pode melhorar tanto o tempo de atraso, quanto a emissão de poluentes.

2.3.3 Atrasos e grandes eventos

Os atrasos aéreos são causados por inúmeros motivos entre eles problemas naturais, como condição de pista por motivo de chuva, neve que podem impactar na frenagem das aeronaves ou na visibilidade da torre de comando, através de falhas humanas, seja por erro em manobras por parte do piloto da aeronave ou por escolhas que nem sempre podem ser as melhores por parte dos controladores.

Devido ao aumento do número de aeronaves em solo o tempo necessário para o controlador tomar decisões de táxi vem crescendo consideravelmente, pois o mesmo encontra situações de pistas saturadas, *gates* lotados e diversas aeronaves em trânsito simultaneamente.

Os atrasos aéreos influenciam diretamente as companhias aéreas que perdem nos Estados Unidos cerca 1,42 dólares [4] por minuto atrasado no preço das passagens aéreas, assim como em processos e na perda de confiança dos clientes. Os passageiros por sua vez podem sofrer danos indiretos como atrasos ou perdas de reuniões de negócios, encontros familiares, eventos e outros que podem gerar perdas tão ou mais significativas que das companhias aéreas.

Com o advento de grandes eventos internacionais no Brasil, fruto de um crescimento e uma maior maturidade econômica e social, se faz necessário uma grande reforma estrutural no país para receber jogos como os da Copa do Mundo de Futebol em 2014 e os Jogos Olímpicos de 2016 no Rio de Janeiro. Estes eventos podem representar uma grande vitrine para o Brasil aos olhos do mundo e pode determinar uma mudança profunda no turismo e no interesse de investimentos estrangeiros no país.

Para obter sucesso nesses eventos e causar uma boa impressão, o Brasil terá de investir muitos recursos em infraestrutura de transporte, no desenvolvimento de novos meios de transporte rápido a curtas distâncias, como os veículos leves sobre trilhos (VLT), seja na reforma de estradas e ruas que possibilitem o deslocamento mais seguro e eficiente de carros e ônibus ou ainda através de uma reforma há muito tempo necessária nos grandes aeroportos do país.

2.4 Justificativa do projeto

Num estudo de caso do Aeroporto Juscelino Kubitschek de Brasília, a Universidade de Brasília propõe a modelagem de um sistema de otimização e sequenciamento de *ta-*

xiway utilizando algoritmo formiga. Este projeto foi motivado pelas falhas e pontos fracos apresentados pelos aeroportos na parte de táxi, e será integrado ao projeto FINEPE/ATECH/UNB. O sistema proposto será de caráter geral, ou seja, poderá ser adotado por qualquer aeroporto. Outros trabalhos têm sido realizados neste âmbito, com a utilização de diferentes algoritmos e abordagens para resolução do problema. Na Universidade de Brasília foram feitos trabalhos com algoritmos genéticos (citado anteriormente) e com a Teoria de Jogos [24] para resolver o problema de sequenciamento de táxi.

Este projeto tem por objetivo o estudo e desenvolvimento do protótipo de um Sistema de Gerenciamento de Partidas em Aeroportos - SiGePA [24], para prover informações em um nível tático e otimizar o fluxo de aeronaves, para uso eficiente e seguro dos recursos em solo e das pistas de pouso e decolagem nos aeroportos. O desenvolvimento do SiGePa se dará dentro do conceito de decisões colaborativas e dotado de um framework de suporte à integração, via protocolo IP, aos sistemas já existentes e a novos sistemas que ainda poderão ser desenvolvidos ou implantados.

Através deste projeto, será desenvolvida uma ferramenta para o acompanhamento em tempo real e apoio à decisão tática sobre a utilização dos recursos em solo (*Gates*, pátios, equipamentos de movimentação) e das pistas de pouso e decolagem nos aeroportos. A partir do agrupamento de informações da estrutura física do aeroporto e do andamento dos vôos, o sistema buscará a sequência ótima de operações das aeronaves a cada instante e sugerirá ações a serem tomadas para manter o fluxo eficiente das decolagens e aterrissagens.

Capítulo 3

Fundamentação teórica

O sistema formiga é uma metodologia de otimização utilizada para resolver problemas diversos dentro da computação, como aqueles que envolvem a necessidade de encontrar o menor caminho [12]. Neste capítulo, apresentaremos a inspiração biológica desse sistema, ou seja, discutiremos a respeito do comportamento real de formigas, além de uma explicação a respeito do funcionamento do algoritmo e alguns exemplos de aplicações.

3.1 Inspiração biológica

Nos últimos anos, pesquisadores de diferentes áreas da ciência têm buscado inspiração para suas pesquisas em fenômenos naturais, como por exemplo no comportamento de animais, com o objetivo de encontrar métodos que auxiliem na busca por soluções dos mais variados problemas encontrados no cotidiano. Entre essas fontes de inspiração podemos citar, por exemplo, o comportamento social de colônias de insetos, tais como formigas, abelhas e cupins, ou até mesmo outros animais como aves e peixes.

Esses métodos são definidos por Dorigo et al. [13] como *Swarm Intelligence* (Inteligência de Enxame), que são métodos inspirados no comportamento social de insetos e outros animais com o objetivo de resolver problemas. Este termo foi introduzido no fim da década de 1980 e referia-se a sistemas robóticos compostos por uma coleção de agentes simples que interagem um com o outro e com o ambiente.

O termo "enxame" é utilizado como uma alusão a qualquer coleção estruturada de agentes capazes de interagir. Esta terminologia inclui qualquer projeção de algoritmos inspirados no comportamento coletivo de insetos sociais e outras sociedades animais.

Um típico sistema que utiliza Inteligência de Enxame tem as seguintes propriedades:

- É composto por uma grande quantidade de indivíduos;
- Os indivíduos são relativamente homogêneos;
- As interações entre os indivíduos é baseada em simples regras de comportamento que exploram somente informações locais que os indivíduos trocam entre si ou pelo ambiente (estigmergia).
- O comportamento global do sistema é resultado de interações entre os indivíduos e destes com o ambiente, isto é, o grupo se auto-organiza.

A propriedade mais característica da Inteligência de Enxame é a habilidade de agir de uma forma organizada mesmo sem a presença de um coordenador. Dentre as vantagens vistas dentro deste método, podemos citar as seguintes:

- Robustez: mesmo quando um ou mais indivíduos falham, o grupo ou colônia continua a executar a tarefa dada.
- Flexibilidade: o grupo ou colônia tem a capacidade de se adaptar rapidamente a mudanças externas e internas.
- Auto-organização: o grupo ou colônia toma suas próprias decisões, ou seja, requer pouca ou nenhuma supervisão ou controle.

Neste trabalho vamos analisar em particular o comportamento social de formigas, em especial a capacidade desses insetos em encontrar o menor caminho entre seu ninho e a fonte de alimento com o auxílio do mecanismo natural das trilhas de feromônio espalhadas ao longo do percurso percorrido. Pesquisadores criaram algoritmos que simulassem tal comportamento com o objetivo de encontrar boas soluções para problemas que se adequem a esse tipo de característica.

3.1.1 Estigmergia

Em meados do século XX, o biólogo Pierre-Paul Grassé observou que algumas espécies de cupins reagiam ao que ele chamou de "estímulo significativo" [13]. Ele observou que os efeitos destas reações podem agir como novos estímulos tanto para o inseto que os produziu quanto para os outros insetos na colônia.

Grassé usou o termo *estigmergia* para descrever este tipo particular de comunicação no qual "trabalhadores são estimulados pela performance que eles atingiram".

Estigmergia é uma forma indireta e não-simbólica de comunicação mediada pelo ambiente - os insetos trocam informação através da modificação do seu ambiente. Além disso, informações estigmérgicas são locais, ou seja, só podem ser acessadas pelos insetos que visitaram o local no qual elas foram liberadas.

Um exemplo de estigmergia pode ser observado em colônias de formigas. Em muitas espécies, formigas caminham para o ninho e para a fonte de comida depositando no chão uma substância chamada feromônio. As outras formigas percebem a presença de feromônio e tendem a seguir o caminho onde a sua concentração é maior. Através deste mecanismo, formigas são capazes de transportar comida para o ninho de uma forma incrivelmente eficaz, configurando uma surpreendente habilidade de encontrar o que cientistas da computação chamam de menor caminho.

3.1.2 O comportamento das formigas

As formigas exibem um complexo comportamento social que há tempos chama a atenção de nós, humanos.

Deneubourg et al. [10] investigaram o feromônio acompanhando o comportamento de formigas. Em um experimento que ficou conhecido como "experimento da ponte dupla" (*double-bridge experiment*), o ninho de uma colônia de formigas foi conectado à fonte de comida por duas pontes de comprimentos iguais.

Em tal cenário, as formigas começam a explorar os arredores do ninho e eventualmente alcançam a fonte de comida. No caminho entre a fonte de comida e o ninho, as formigas depositam feromônio. Inicialmente, cada formiga aleatoriamente escolhe uma das duas pontes.

Contudo, devido a flutuações randômicas, depois de algum tempo uma das duas pontes apresenta uma maior concentração de feromônio do que a outra e, portanto, atrai mais formigas. Isto traz uma quantidade ainda maior de feromônio para aquela ponte, fazendo com que ela fique cada vez mais atrativa. Como resultado, a colônia inteira de formigas converte para o uso da mesma ponte.

Este comportamento a nível de colônia, baseado na exploração de feedback positivo, pode ser usado pelas formigas para encontrar o caminho mais curto entre a fonte de comida e o ninho.

Outros estudos consideraram a variante do experimento da ponte dupla no qual uma ponte é significativamente maior que a outra. Neste caso, as flutuações randômicas na escolha inicial de uma ponte é muito reduzida e um segundo mecanismo realiza um papel importante: as formigas que escolhem ao acaso a ponte menor são as primeiras a chegar no ninho. A ponte menor recebe, portanto, feromônio antes que a ponte maior e isto aumenta a probabilidade de outras formigas selecionarem essa ponte ao invés da maior.

3.2 O método de Otimização com colônia de formigas

O método colônia de formigas (ACO, do inglês *Ant Colony Optimization*), formulado na década 1990 por Marco Dorigo em sua tese de PhD, é uma heurística inspirada no comportamento real de formigas [15], com relação às suas habilidades em encontrar o caminho mais curto entre o ninho e o alimento. A Figura 3.1 representa o comportamento da colônia de formigas diante de um obstáculo colocado em seu caminho.

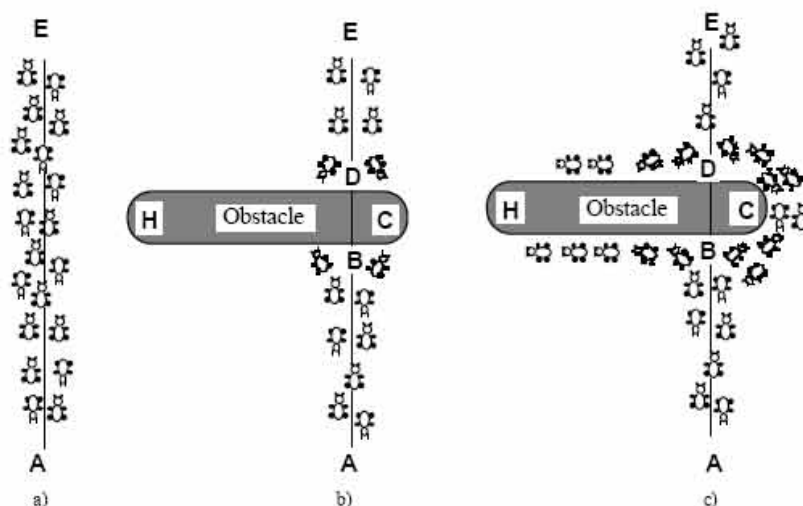


Figura 3.1: Formigas reais escolhendo o caminho mais curto [14]

a) As formigas seguem o caminho entre os pontos A e E.

- b) Um obstáculo é colocado, as formigas podem escolher ultrapassá-lo por qualquer um dos dois diferentes caminhos, com igual probabilidade.
- c) No caminho mais curto é colocado maior quantidade de feromônio.

Esta busca é efetuada através da exploração das trilhas de feromônio, substância química depositada pelas formigas durante seu percurso. É usado para problemas computacionais que envolvam percorrimento de caminhos em grafos.

No ACO, um conjunto de agentes de software chamado de *formigas artificiais* busca boas soluções para um dado problema de otimização. Para aplicar o ACO, o problema de otimização é transformado no problema de encontrar o melhor caminho em um grafo ponderado. De forma incremental, as formigas constroem soluções se movendo no grafo.

O processo de construção de solução é randômica e é baseado em um modelo de feromônios, isto é, um conjunto de parâmetros associados aos componentes dos grafos (nós ou pontas), cujos valores são modificados em tempo de execução pelas formigas [12]. Este mecanismo é representado na Figura 3.2.

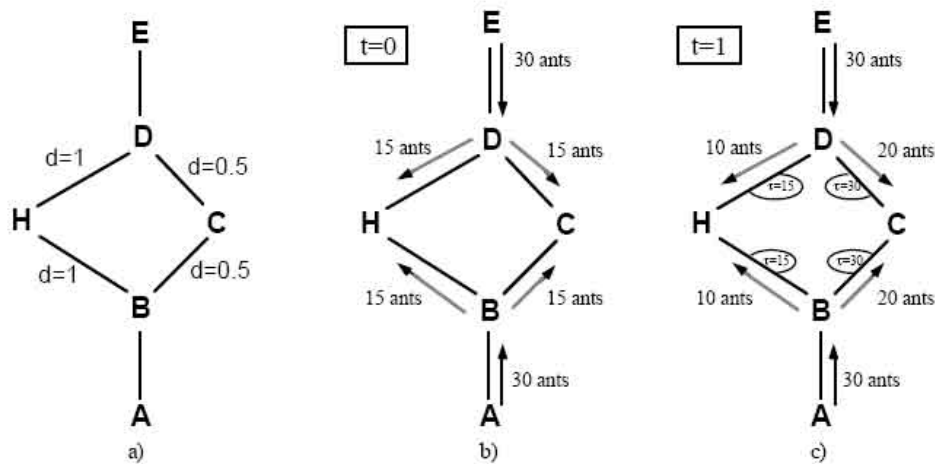


Figura 3.2: Formigas artificiais escolhendo o caminho mais curto [14]

- a) Grafo inicial com as respectivas distâncias.
- b) No tempo $t=0$ não há nenhum tipo de rastro nas arestas do grafo, logo, as formigas escolhem ir para a direita ou esquerda com igual probabilidade.
- c) No tempo $t=1$ o rastro é mais forte nas arestas menores, que se tornam, portanto, preferidas pelas formigas.

Desde a proposta do algoritmo formiga, o primeiro algoritmo ACO, foram obtidos vários resultados significantes em pesquisas. Estas contribuições focaram no desenvolvimento de variantes de algoritmos de alta-performance, aplicações bem sucedidas de algoritmos ACO a uma vasta gama de complexos problemas computacionais e a compreensão teórica de propriedades de algoritmos ACO.

A principal característica deste modelo é o *feedback* positivo, computação distribuída e o uso de uma heurística construtiva. [14].

3.3 Subdivisões do ACO

Várias vertentes do ACO tem sido propostas na literatura, sendo as principais o Sistema Formiga e o Sistema de Colônia de Formigas [13]. Abaixo apresentaremos cada um deles. Considera-se que a execução acontece sobre um grafo $G = (V, E)$, onde V representa um nó (ou vértice) e E é o caminho que liga dois nós adjacentes.

3.3.1 Sistema Formiga (AS)

O Sistema Formiga foi o primeiro ACO proposto. Sua principal característica é que, a cada iteração, os valores de feromônio são atualizados por todas as m formigas que construíram a solução na própria iteração. O feromônio τ_{ij} , associado aos nós (i, j) , é atualizado conforme a Equação 3.1

$$\tau_{ij} = (1 - \rho) * \tau_{ij} + \sum_{k=1}^m \tau_{ij}^k \quad (3.1)$$

onde ρ é a taxa de evaporação, m o número de formigas e $\Delta\tau_{ij}^k$ a quantidade de feromônio no vértice (i, j) depositado pela formiga k :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{se a } k\text{-ésima formiga utiliza a aresta } (i, j) \text{ no seu caminho;} \\ 0 & \text{caso contrário.} \end{cases} \quad (3.2)$$

onde Q é uma constante que influencia na velocidade de convergência do algoritmo e L_k mostra a distância total caminhada pela k -ésima formiga neste ciclo.

No momento em que uma formiga se encontra em um nó i , e tendo já construído uma solução parcial s_p , a probabilidade de ir para o nó j é a seguinte:

$$P_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{C_{il} \in N(s_p)} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \quad (3.3)$$

onde $N(s_p)$ é um conjunto de soluções possíveis, isto é, vértices (i, l) , onde l é um nó ainda não visitado pela formiga k . α e β são constantes e controlam a importância relativa do feromônio e do comprimento da rota, respectivamente. A informação heurística η_{ij} é dada por:

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (3.4)$$

onde d_{ij} é a distância entre i e j , o que demonstra que quanto menor o caminho, maior é sua concentração de feromônio.

3.3.2 Sistema da Colônia de Formigas (ACS)

A maior inovação do ACS foi a introdução da *atualização local de feromônio*, somada à atualização executada no final do processo de construção (chamado atualização *offline*

de feromônio). Este processo é realizado por todas as formigas a cada passo da iteração. Cada formiga aplica a atualização somente ao último vértice percorrido:

$$\tau_{ij} = (1 - \phi) * \tau_{ij} + \phi * \tau_0 \quad (3.5)$$

onde $\phi \in (0, 1]$ é o coeficiente de decaimento do feromônio, e τ_0 é o valor inicial de feromônio.

O principal objetivo da atualização local é diversificar a busca realizada por formigas subsequentes durante uma iteração: diminuindo a concentração de feromônio no vértice percorrido, as formigas encoragem as seguintes a escolher outras rotas e, portanto, a produzir soluções diversificadas. Com isto, é menos provável que várias formigas encontrem soluções idênticas durante uma iteração.

A atualização *offline* de feromônio é aplicada no final de cada iteração por apenas uma formiga, que pode ser tanto a melhor da iteração ou a melhor até o momento. A fórmula de atualização é a seguinte:

$$\tau_{ij} \leftarrow \begin{cases} (1 - \rho) * \tau_{ij} + \rho * \Delta\tau_{ij} & \text{se } (i, j) \text{ pertence à melhor rota,} \\ \tau_{ij} & \text{caso contrário.} \end{cases} \quad (3.6)$$

e a quantidade de feromônio depositada no vértice (i, j) é dada por:

$$\Delta\tau_{ij} = \frac{1}{L_b} \quad (3.7)$$

onde L_b é o tamanho da melhor rota encontrada.

Similarmente ao Sistema Formiga, a probabilidade de escolha de uma rota no Sistema da Colônia de Formigas é dada pela Equação 3.3.

3.4 Aplicações do ACO

Existem diversas implementações bem-sucedidas da metaheurística ACO, aplicadas a diferentes problemas que envolvem otimização. A seguir são citadas algumas destas aplicações.

3.4.1 Problema do Caixeiro Viajante

Neste clássico problema de otimização, onde um caixeiro deve encontrar a rota mais curta onde ele possa visitar um dado número de cidades, cada cidade exatamente uma vez [11]. É um problema muito estudado por cientistas, matemáticos e investigadores de diversas áreas, tais como logística, genética e produção, entre outros. O ACO desempenha um importante papel nas resoluções do Problema do Caixeiro Viajante (PCV), por diversas razões, entre as quais:

- É um problema para o qual os algoritmos ACO são facilmente aplicados;
- É um problema de otimização considerado, dentro da teoria da complexidade computacional, como NP-difícil;
- O PCV serve de teste para aplicação de novos algoritmos;

Nos algoritmos ACO, as formigas são simples agentes que, no caso do PCV, constroem circuitos através do movimento entre cidades no grafo do problema. A solução construída pelas formigas é elaborada por trilhos de feromônios (artificiais) e pela disponibilidade de informação heurística. Quando o algoritmo ACO é aplicado, entra em cena a chamada "força do feromônio", representada por $\tau_{ij}(t)$, que é uma informação numérica modificada ao longo da execução do algoritmo, sendo t o contador das iterações. Para aplicação no PCV, o algoritmo ACO caracteriza-se por $\tau_{ij}(t) = \tau_{ji}(t)$.

Inicialmente, cada uma das m formigas é colocada numa cidade, de forma aleatória, aplicando-se depois, de modo iterativo, uma regra de transição de estado para cada uma das cidades.

Uma formiga constrói um circuito conforme explicado a seguir. Na cidade i , a formiga escolhe uma cidade j que ainda não tenha visitado. Esta escolha é feita probabilisticamente segundo a força do feromônio $\tau_{ij}(t)$ no arco entre as cidades i e j , e a informação heurística disponível localmente, que é função do comprimento do arco. As formigas, de um modo probabilístico, preferem as cidades que estejam mais próximas e ligadas por arcos com grande força de feromônio.

Para construir uma solução executável, cada formiga possui uma forma de memória limitada chamada *tabu list*, onde é guardado o circuito percorrido até determinado momento. A memória é utilizada para determinar, a cada passo da construção, o conjunto de cidades que têm ainda que ser visitadas e para garantir que é elaborada uma solução executável. Adicionalmente, permite à formiga refazer o seu circuito, assim que esteja completo.

Depois de todas as formigas terem construído um circuito, os feromônios são atualizados. Isto é tipicamente elaborado através da descida da força dos trilhos dos feromônios, através de um fator constante, e depois é dada liberdade para que as formigas depositem os seus rastros nos arcos que visitaram. A atualização dos trilhos é feita de tal forma que os arcos mais curtos, ou visitados por muitas formigas, recebem quantidades maiores de feromônio e por isso são escolhidas com uma maior probabilidade nas iterações posteriores.

Dessa forma, a quantidade de feromônio $\tau_{ij}(t)$ representa o desejo direcionado de escolher a próxima cidade j quando a formiga estiver na cidade i .

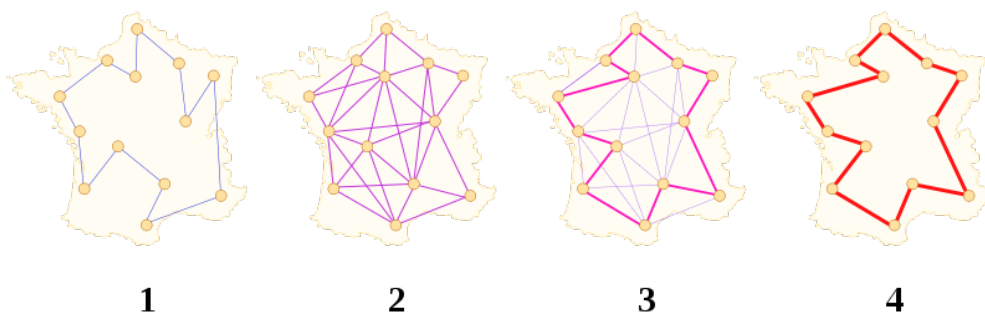


Figura 3.3: Efeito do algoritmo ACO para a constituição do circuito preferencial.

Em suma, a aplicação dos algoritmos ACO seguem um padrão de, após de iniciados o rastro de feromônios e alguns parâmetros, repetir-se o caminho pelo circuito principal até

que a condição seja conhecida e rescindida. No circuito principal, primeiro as formigas constroem circuitos executáveis, em seguida os circuitos produzidos são melhorados pela aplicação de uma procura local e, por fim, os rastros de feromônios são atualizados.

3.4.2 Problema de roteamento de veículos

É um dos problemas mais estudados na área de otimização combinatória, em que um conjunto de clientes devem ser atendidos por intermédio de uma frota de veículos, que partem de pontos chamados depósitos [28]. A premissa principal do PRV é que cada veículo v possui uma capacidade Cv e o somatório de todas as demandas dos consumidores atendidos por um veículo não pode ultrapassar sua capacidade. O problema de roteamento de veículos tem sido objeto de pesquisa por vários cientistas nas últimas décadas, pois seu foco baseia-se na necessidade da diminuição de gastos em todo o processo que engloba desde a produção de uma mercadoria até sua venda. Estima-se que o valor gasto na distribuição de uma mercadoria represente de 10% a 15% do valor final da mercadoria, um quantitativo considerável.

O objetivo da aplicação do ACO é encontrar rotas de menor custo, de forma que:

- Cada cliente é visitado apenas uma vez por apenas um veículo;
- Para cada veículo, a demanda total não exceda a capacidade do veículo;
- Cada veículo começa e termina sua viagem na mesma posição.

Dessa forma, a metaheurística da colônia de formigas é uma das formas de resolver este clássico problema de otimização. Os contextos de aplicação do modelo do roteamento de veículos e todas as suas derivações é muito vasto, pois em qualquer seguimento de comércio há a necessidade da diminuição de custos de produção e de valor final de venda, sem que para isso tenha-se que diminuir a qualidade do produto. Bons resultados na aplicação do algoritmo pode significar uma importante elevação no poder de negociação e de venda da empresa, aumentando sua competitividade.

3.4.3 Serviços de comunicação em rede

A comunicação em rede é um mais uma possível aplicação do algoritmo formiga. Na comunicação orientada à conexão, todos os pacotes da mesma sessão seguem o mesmo caminho selecionados por uma fase preliminar de configuração. Nesta fase, o cliente e o servidor (que estão em diferentes sistemas finais) enviam pacotes de controle um para o outro antes de remeterem os dados reais, procedimento que alerta o cliente e o servidor, fazendo com que se eles preparem para o recebimento de pacotes. Neste tipo de comunicação, os dispositivos nos pontos extremos usam um protocolo para estabelecer uma conexão física ou lógica *end-to-end* antes de qualquer dado ser enviado.

Na comunicação não-orientada à conexão, pacotes da mesma sessão podem seguir caminhos diferentes, devido ao fato de cada pacote conter um cabeçalho de informações com o endereço de destino que permite a entrega independente de cada pacote, dentro da rede (Internet). Em outras palavras, não existe a fase de apresentação entre os sistemas finais. Quando um dos lados de uma aplicação quer enviar pacotes ao outro, ela simplesmente

os envia. Esta técnica permite que os pacotes sejam remetidos mais rapidamente, porém não há confirmações de entrega.

Os agentes (formigas) automaticamente atravessam a rede e coletivamente constroem uma ação distribuída de roteamento. As vantagens deste tipo de algoritmo de roteamento é que eles não requerem um conhecimento prévio da rede, como as capacidades ou demandas do tráfego, utilizam uma estrutura de controle descentralizada e, além disso, possuem a habilidade de se adaptar às mudanças nos parâmetros da rede e demandas de tráfego [19].

3.4.4 Outras aplicações

Outros possíveis problemas computacionais que podem ser resolvidos através do algoritmo de otimização de formigas são os seguintes:

- Problema da coloração de grafos: O problema é encontrar uma coloração de grafos tal que não haja dois vértices adjacentes da mesma cor (coloração de vértices) e nem duas arestas adjacentes da mesma cor (coloração de arestas) e o número de cores usadas deve ser mínimo [20]. A coloração de grafos tem muitas utilidades práticas, bem como desafios teóricos. Além da forma como a cor pode ser atribuída, outras limitações também podem ser definidas no grafo. Estas técnicas chegaram a se tornar popular através da série de quebra-cabeças Sudoku.
- Problema de sequenciamento: Consiste em encontrar o caminho de menor peso em um grafo ponderado de Hamilton. Em um grafo ponderado, um peso ou conjunto de pesos é associado a cada aresta, representado da forma $w(i, j)$, ou seja, $w(1, 2)$ é o peso associado a aresta que une os nós 1 e 2. Já em um grafo com atributos A , definido por $G = (V, E, A)$, os valores são associados aos nós de G . Um grafo de Hamilton é aquele que contém cada nó do grafo exatamente uma vez. [5]
- Problema quadrático de atribuição: Ocorre quando, para designar objetos a locais, deve-se considerar as distâncias entre os pares de locais e os fluxos de algum tipo de demanda entre os pares de objetos, com o objetivo de encontrar uma alocação de custo mínimo dos objetos aos locais [28].
- Problema da supersequência comum mínima: Neste problema, um dado um conjunto de strings sobre um alfabeto, deve ser encontrado uma sequência de comprimento mínimo que seja uma supersequência de cada string do conjunto (uma supersequência S da string A pode ser obtida de A inserindo zero ou mais caracteres em A) [28]. Em outras palavras, dadas as sequências s_1, \dots, s_n (finitas) de símbolos em um alfabeto, deve ser encontrada uma supersequência de tamanho mínimo de cada s_i , para $i = 1, \dots, n$.

Capítulo 4

Estado da arte

Este capítulo tratará a respeito de estudos realizados com metodologias de otimização para resolver problemas de transportes, em princípio, o gerenciamento de táxi, utilizando o sistema formiga e algoritmos genéticos. Será apresentado também a utilização do sistema formiga em problemas de congestionamento do trânsito.

4.1 Utilização do sistema formiga no problema do táxi

A otimização de operações em solo num aeroporto apresenta complexidade NP-difícil [21] devido ao desafio de gerenciar com eficiência o tráfego no aeroporto e reduzir atrasos ou evitar conflitos nestas operações. As operações em solo são geralmente classificadas como gerenciamento de decolagem e aterrissagem, planejamento e agendamento de táxi, e resolvidas por vários métodos como algoritmos de Dijkstra e algoritmos genéticos, que são usados para obter o roteamento ótimo de táxi.

Em 2011, Changyou Liu e Yongcheng Wang [21], da Universidade de Aviação Civil da China, propuseram um modelo pioneiro que aplica o algoritmo formiga para o problema de otimização do táxi. Neste modelo, é utilizado o primeiro método proposto com colônia de formigas, o Sistema Formiga.

De acordo com os horários dos voos, é necessário designar rotas a todos os pousos e decolagens, de modo a garantir que o taxiamento ocorra utilizando-se do menor tempo ou distância possível percorrida, sob determinadas condições de segurança. As rotas de táxi são representadas por um grafo $G = (V, E)$, onde V é o conjunto de vértices (incluindo garagens) e E o conjunto de arestas que ligam dois nós. O comprimento do vértice entre um nó i e um nó j é representado como S_{ijk} , onde k representa a aeronave.

Neste modelo, cada formiga artificial tem uma quantidade constante de feromônio. Quando a formiga vai do ponto inicial para seu destino, ou seja, percorre um caminho entre os vértices S_1 e S_2 , será depositado feromônio na rota. Se esta rota for uma boa escolha, a distribuição de feromônio será maior. Se, por outro lado, a rota é ruim, a distribuição do elemento feromônio será menor. Os rastros nesta rota (ruim) vão se reduzindo até que nenhuma formiga escolha este caminho.

4.1.1 Descrição do algoritmo

Os passos para utilização do algoritmo são descritos a seguir:

1. Criar a colônia de formigas X_i , onde $i = (1, 2, \dots, M)$ e o número máximo de nós que cada formiga pode visitar é E . É estabelecida uma *tabu list*, onde cada formiga pode registrar sua viagem.
2. As formigas são colocadas no nó inicial B e começam a busca pelo melhor caminho.
3. Para a formiga X_k , onde $k = (0, 1, \dots, m)$, se o final E for encontrado, terminar a busca e selecionar a formiga $X_{(k+1)}$. A formiga escolhe o próximo vértice j com base na probabilidade P_{ij} :

$$P_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h \in S} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta} \quad (4.1)$$

onde α e β são constantes e controlam, respectivamente, a importância relativa do feromônio e da visibilidade. Se j já estiver presente na lista de vértices, um novo vértice deve ser escolhido da mesma maneira. Se a formiga não puder atingir o destino, ela se move para o vértice j e o grava na lista de vértices.

4. Calcular o tamanho dos caminhos que as formigas percorreram, gravando a melhor solução. Para a formiga k , onde $k = (0, 1, \dots, m)$, o tamanho do caminho L_k percorrido é calculado de acordo com a lista de vértices. Dado um caminho (S_i, S_j) , usa-se a seguinte fórmula para calcular a quantidade de feromônio:

$$\tau(t+n) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

onde ρ é o coeficiente de volatilidade do feromônio.

5. Para o caminho (S_i, S_j) a seguinte fórmula é usada em conjunto com a fórmula acima para atualizar o valor do feromônio:

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \tau_{ij}^k(t)$$

onde m é o número de formigas e $\Delta\tau_{ij}(t)$ mostra a quantidade de feromônio na aresta (i, j) .

6. Se $N \geq Nmax$, o algoritmo é interrompido e é exibido na saída o caminho mais otimizado encontrado na busca.

No algoritmo acima, aplicado ao problema de táxi, as formigas representariam os aviões e o caminho a rota otimizada encontrada para o problema.

4.1.2 Detecção de conflitos

Durante o processo de taxiamento, algumas restrições de segurança devem ser cumpridas, de forma a evitar conflitos entre as aeronaves. Os tipos de conflitos que podem ocorrer são descritos abaixo:

- O conflito do cruzamento: O avião que chega primeiro no ponto de cruzamento poderá passar seguramente por este ponto, enquanto um outro avião que queira passar no mesmo ponto espera a uma distância segura para evitar conflitos. Se este caso ocorrer, o tempo de espera será registrado no tempo operacional do avião em espera.
- O conflito do avião frontal: O avião que chegar primeiro ao caminho composto por dois vértices adjacentes ocupará este caminho primeiramente, e o outro avião na direção oposta irá esperar, a uma distância segura deste caminho, até que este seja completamente liberado. Se este caso ocorrer, o tempo de espera será registrado no tempo operacional do avião em espera.

O conflito do avião frontal pode ser causado por uma série de reações em cadeia ou por uma mistura de conflitos de cruzamento ou avião frontal. Estes conflitos devem ser proibidos para evitar uma situação de *deadlock*.

- O conflito do próximo avião: O avião que chega primeiro ao caminho composto por dois vértices adjacentes irá ocupar este caminho primeiramente, e o outro avião que vier na mesma direção manterá uma distância segura antes deste ponto, até que o caminho seja completamente liberado. Se este cenário ocorrer, o tempo de espera será registrado no tempo operacional do avião em espera.

4.2 Sistema formiga para previsão de congestionamento no trânsito

O crescimento da inteligência de sistemas de transporte (ITS) tem sido extremamente promissor, e vários estudos em ITS tem sido feitos do pontos de vista de inteligência artificial. Contudo, ainda existem muitos problemas que precisam ser resolvidos, e aliviar a congestão de tráfego é um dos muitos problemas. A redução de congestionamentos é uma questão urgente pois o gasto de dinheiro devido a congestionamento é alto. Em Tóquio, por exemplo, 1 km de engarrafamento chega a custar o equivalente a 8,5 milhões de reais por ano.

Neste contexto, foi proposto [3] um sistema de previsão de congestionamento que pode reagir a mudanças no tráfego de forma dinâmica, baseado em um mecanismo que utiliza um modelo chamado de comunicação de feromônio.

A tecnologia de previsão de congestionamentos é um dos elementos principais da ITS. Vários métodos foram propostos até o momento, dois deles citados abaixo.

- Previsão a longo-prazo: um método de analisar estatisticamente dados passados relacionados ao tráfego e descobrir um padrão que leva à ocorrência de congestionamento.
- Previsão a curto-prazo: um método de prever o engarrafamento alguns minutos antes, utilizando informações em tempo real.

Comunicações por feromônio são baseadas no comportamento social de formigas e são aplicadas como um modelo que é usado para responder adaptativamente a mudanças dinâmicas no ambiente em várias aplicações. Este método é uma forma eficiente de

prever congestionamento no qual cada agente (via) gera informação sobre feromônio em sua própria via e arredores e troca esta informação com seus agentes vizinhos.

Primeiramente, define-se o ambiente como se segue:

- Uma via é uma seção entre duas interseções conectadas. Cada via consiste de várias faixas, geralmente em ambas as direções.
- O número de carros passando por uma interseção é contado por um sensor instalado em cada interseção, e este número é enviado para cada agente (via) instalado em computadores servidores na beira da estrada em intervalos regulares.
- O agente instalado em cada servidor na beira da estrada calcula e prevê o congestionamento do trânsito.

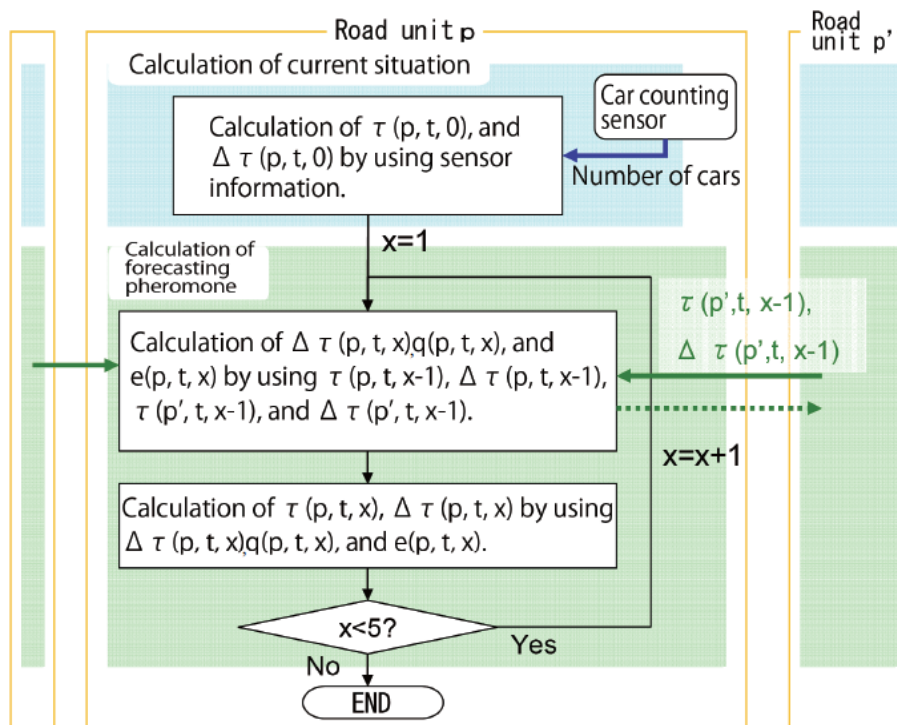


Figura 4.1: Fluxo de previsão de congestionamento [3]

Em seguida, formula-se o fluxo de densidade do tráfego usando um fator "feromônio de densidade do tráfego $\Delta \tau$ ", e o crescimento da fila usando um "feromônio de difusão do congestionamento q ". Para que as previsões fiquem mais precisas, é introduzida a taxa de evaporação e , que indica mudança na densidade do engarrafamento (de criação para dissolução). Cada agente prevê congestionamentos em intervalos de um minuto, como mostrado na figura 4.1, onde $t(p, t, x)$ é a densidade prevista de tráfego em uma via s no tempo t e $\Delta \tau(p, t, x)$ é a transição prevista no congestionamento da via s no tempo t .

4.3 Otimização do táxi através de algoritmo genético

Em 2012, um aluno do Departamento de Computação da Universidade de Brasília, em seu trabalho de conclusão de curso [26], modelou e implementou uma solução para o problema de taxiamento utilizando algoritmos genéticos.

Algoritmos genéticos são, assim como o algoritmo de otimização da colônia de formigas, técnicas heurísticas de otimização global inspiradas na natureza. Eles são definidos como uma técnica de busca baseada no processo biológico da evolução natural e da genética.

Na execução deste algoritmo, inicialmente, é gerada uma população formada por um conjunto aleatório de indivíduos que podem ser vistos como possíveis soluções do problema. Durante o processo evolutivo, esta população é avaliada: para cada indivíduo é dada uma nota, ou índice, refletindo sua habilidade de adaptação a determinado ambiente. No contexto do algoritmo genético, este índice é chamado de *fitness*.

Uma porcentagem dos mais adaptados são mantidos, enquanto os outros são descartados, característica esta que remonta ao processo de seleção natural descrito por Darwin. Os membros mantidos pela seleção podem sofrer modificações em suas características fundamentais através de mutações e cruzamento (*crossover*) ou recombinação genética gerando descendentes para a próxima geração. Este processo, chamado de reprodução, é repetido até que uma solução satisfatória seja encontrada [26]. Trazendo esta metáfora para a aplicação na melhor rota de taxiamento, a recombinação genética representa tentativas de determinação de caminhos mais otimizados.

Embora possam parecer simplistas do ponto de vista biológico, estes algoritmos são complexos o suficiente para fornecer poderosos e robustos mecanismos de busca.

4.3.1 Codificação de cromossomo

Com as informações do tipo de voo e horário, a ordem do voo e o ponto de início podem ser determinados. Em geral, haverá L rotas viáveis ao longo do conjunto de vértices da malha de possibilidades, do ponto inicial ao final. Para garantir a execução deste algoritmo, o gene cromossomo é uma das séries de números selecionados aleatoriamente dentre L rotas de taxiamento viáveis para o avião k . A seleção é feita para cada gene do cromossomo e o comprimento do cromossomo é o número total de voos. Por exemplo, a *string* de cromossomo 2 5 2 4 7 5 6 5 8 apresenta o número de trajetórias para 9 aviões. Importante notar que L é o número total de rotas viáveis de taxiamento, e o número total de voos N_f é o comprimento do cromossomo, enquanto que uma possível trajetória para o avião k é uma posição de um gene neste algoritmo.

4.3.2 Seleção, *crossover* e mutação

A função de *fitness*, citada anteriormente, mede a qualidade de cada geração da população apresentada. Ela é utilizada para alocar traços reprodutivos aos indivíduos na população em questão e então age como uma medida de qualidade que deve ser aumentada, ou seja, indivíduos com um maior valor de *fitness* terão uma maior probabilidade de serem selecionados como candidatos para exame posterior.

O valor *fitness* é escolhido em cada geração e cruzado um por um com certa probabilidade p_c . Por exemplo, os cruzamentos começam a partir do terceiro gene para dois

cromossomos quaisquer como *chromo1* e *chromo2*. O resultado após o cruzamento é mostrado abaixo:

Chrom1 : 2 5 2 4 7 5 6 5 8 → *Chrom3* : 2 5 2 4 7 5 6 5 9
Chrom2 : 1 5 3 4 7 5 6 5 9 → *Chrom4* : 1 5 3 4 7 5 6 5 8

O operador de mutação é necessário com uma certa probabilidade p_m para evitar que a procura do algoritmo leve à solução ótima local. A operação é realizada através de uma seleção aleatório no gene. Como em *chromo1*, o terceiro gene muda de 2 para 4 e a nova string é 2 5 4 4 7 5 6 5 8.

Chrom1 : 2 5 2 4 7 5 6 5 8 → *Chrom2* : 2 5 4 4 7 5 6 5 9

Como é difícil determinar o escopo da solução ótima, o número máximo de soluções é adotado para finalizar o algoritmo.

A aplicação do algoritmo foi esquematizada conforme mostrado a seguir:

1. Receber os horários de voo, bem como os pontos de início e pontos finais de decolagem ou aterrissagem.
2. A partir dos pontos de início e fim será selecionado um conjunto de rotas possíveis para cada aeronave.
3. É criada uma população inicial com auxílio da *schedule*, respeitando todas as restrições.
4. São utilizadas as funções genéticas criadas, em Java, para essa solução.
5. Encontra-se o melhor cromossomo, ou seja, aquele que apresenta a melhor solução dentre todas as soluções encontradas de forma não-determinística.
6. É apresentado o gene que representa a trajetória ótima para aquele conjunto de aeronaves na *schedule* determinada.

O trabalho realizado com algoritmos genéticos adaptado ao sistema de rotas e aeronaves mostrou-se como uma solução viável para aeroportos congestionados, desde que se tenha uma *schedule* pré-definida, recursos computacionais apropriados e tempo computacional.

Capítulo 5

Solução, modelagem e implementação

Neste capítulo será apresentada a proposta de solução para o problema, utilizando o Algoritmo de Otimização de Colônia de Formigas. Será apresentada também a modelagem e arquitetura da solução, além de detalhes da implementação.

5.1 Solução proposta

Este sistema é inspirado pelo trabalho de Changyou Liu e Yongcheng Wang [21], que utilizaram o Sistema Formiga para construção de sua solução. Utilizaremos, porém, uma outra vertente do ACO, que é o Sistema de Colônia de Formigas.

Primeiramente, o operador deverá submeter ao sistema um arquivo-texto contendo as informações pertinentes à estrutura do aeroporto, como os nomes de cada nó e a distância entre nós adjacentes. O mapa do aeroporto que é utilizado para construção do grafo foi projetado através de uma ferramenta interativa [23] que utiliza o Google Maps como base e pode ser visto na Figura 5.1



Figura 5.1: Mapa do Aeroporto Internacional de Brasília

Além disso, o controlador deverá submeter o *schedule* dos vôos, informando a posição inicial de cada aeronave e os horários de decolagem ou pouso, informações sobre as quais serão calculadas as rotas ótimas de taxiamento.

Modelo matemático

Neste modelo, de acordo com os horários dos voos, é necessário designar rotas a todos os pousos e aterrisagens, o que deve garantir que o taxiamento ocorrerá utilizando-se uma solução ótima que vise o menor tempo ou distância possível percorrida, sob determinadas condições de segurança. As rotas de táxi são representadas por um grafo $G = (V, E)$, onde V é o conjunto de vértices ou nós (incluindo garagens) e E o conjunto de arestas que ligam dois nós. O comprimento da aresta entre um nó i e um nó j é representado como d_{ijk} , onde k representa a aeronave.

A função objetivo utilizada para encontrar o menor caminho de taxiamento ou a melhor trajetória é definida por:

$$\min f = \sum_{k \in N_f} \sum_{i \in (N \cup Q)} \sum_{j \in (N \cup Q)} x_{ijk} t_{ijk} \quad (5.1)$$

onde N_f é o número total de voos, t_{ijk} o tempo de táxi da aeronave k do nó i para o nó j , e x_{ijk} é uma variável booleana. Se a aeronave k utiliza os nós i e j em sua rota, então $x_{ijk}=1$, caso contrário, $x_{ijk}=0$.

Cálculo do feromônio

O Sistema da Colônia de Formigas traz dois tipos de atualização de feromônio:

- Atualização local de feromônio: é realizada por todas as formigas a cada passo da construção, aplicada apenas ao último caminho percorrido:

$$\tau_{ij} = (1 - \phi) * \tau_{ij} + \phi * \tau_0 \quad (5.2)$$

onde $\phi \in (0, 1]$ é o coeficiente de decaimento do feromônio, e τ_0 é o valor inicial de feromônio.

- Atualização *offline* de feromônio: é realizada quando uma formiga chega a um resultado no final de uma iteração. A fórmula de atualização é a seguinte:

$$\tau_{ij} \leftarrow \begin{cases} (1 - \rho) * \tau_{ij} + \rho * \Delta\tau_{ij} & \text{se } (i, j) \text{ pertence à melhor rota,} \\ \tau_{ij} & \text{caso contrário.} \end{cases} \quad (5.3)$$

A quantidade $\Delta\tau_{ij}$ de feromônio depositada no vértice (i, j) é:

$$\Delta\tau_{ij} = \frac{Q}{L_b} \quad (5.4)$$

onde Q é uma constante de convergência e L_b é o tamanho da melhor rota encontrada desde o começo do algoritmo (*best-so-far*). Esta fórmula mostra que quanto menor a rota, maior será o depósito de feromônio nela.

Função de probabilidade de escolha

A probabilidade de uma formiga se mover do nó i para o nó j é definida pela seguinte fórmula:

$$P_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{C_{il} \in N(s_p)} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \quad (5.5)$$

onde $N(s_p)$ é um conjunto de soluções possíveis, isto é, vértices (i, l) . α e β são constantes e $\alpha \geq 0$ e $\beta \geq 0$. α é chamado fator feromônio e representa a importância relativa que o feromônio depositado do trecho (i, j) exerce sobre a escolha de uma rota. β é chamado fator inspiração e representa a importância relativa de visibilidade, ou seja, a importância que o comprimento da rota terá sobre a escolha da formiga.

O fator η_{ij} é dado por:

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (5.6)$$

onde d_{ij} é a distância entre os vértices i e j .

5.2 Modelagem e arquitetura

Para a utilização do algoritmo da colônia de formigas no problema de gerenciamento do táxi foi feita a modulação do problema de modo que a trajetória de um avião da garagem até a pista de decolagem seja representada como um grafo ponderado, onde o peso é calculado em tempo.

O funcionamento básico do sistema é representado pelo diagrama na Figura 5.2

Este fluxograma pode ser descrito da seguinte forma:

1. Primeiramente, é lido o arquivo de entrada que representa o aeroporto, com o nome de cada nó (representados no mapa da Figura 5.1) e a distância entre os nós adjacentes. A partir desta leitura, é criado o grafo. O arquivo que contém os voos a serem escalados é recebido, com o horário e os nós de origem e destino de cada aeronave.
2. Os valores de feromônio são inicializados para cada rota, de acordo com o valor de $\tau_{inicial}$. O voo a ser escalado é selecionado e as formigas são criadas e iniciam a busca através de várias iterações. Ao atingir o número máximo de iterações, a melhor solução é gravada. Este processo é repetido para cada uma das aeronaves escaladas.
3. Com as melhores rotas já calculadas para todas as aeronaves, os conflitos são tratados. Caso duas aeronaves estejam a uma distância menor do que 150 metros uma da outra, a aeronave que veio depois terá que aguardar para prosseguir seu caminho de forma segura. Neste caso, o tempo de permanência da aeronave no nó anterior ao conflito é aumentado.
4. Após todos os conflitos serem tratados, as soluções são gravadas em um arquivo de saída, contendo as rotas de taxiamento de cada uma das aeronaves.

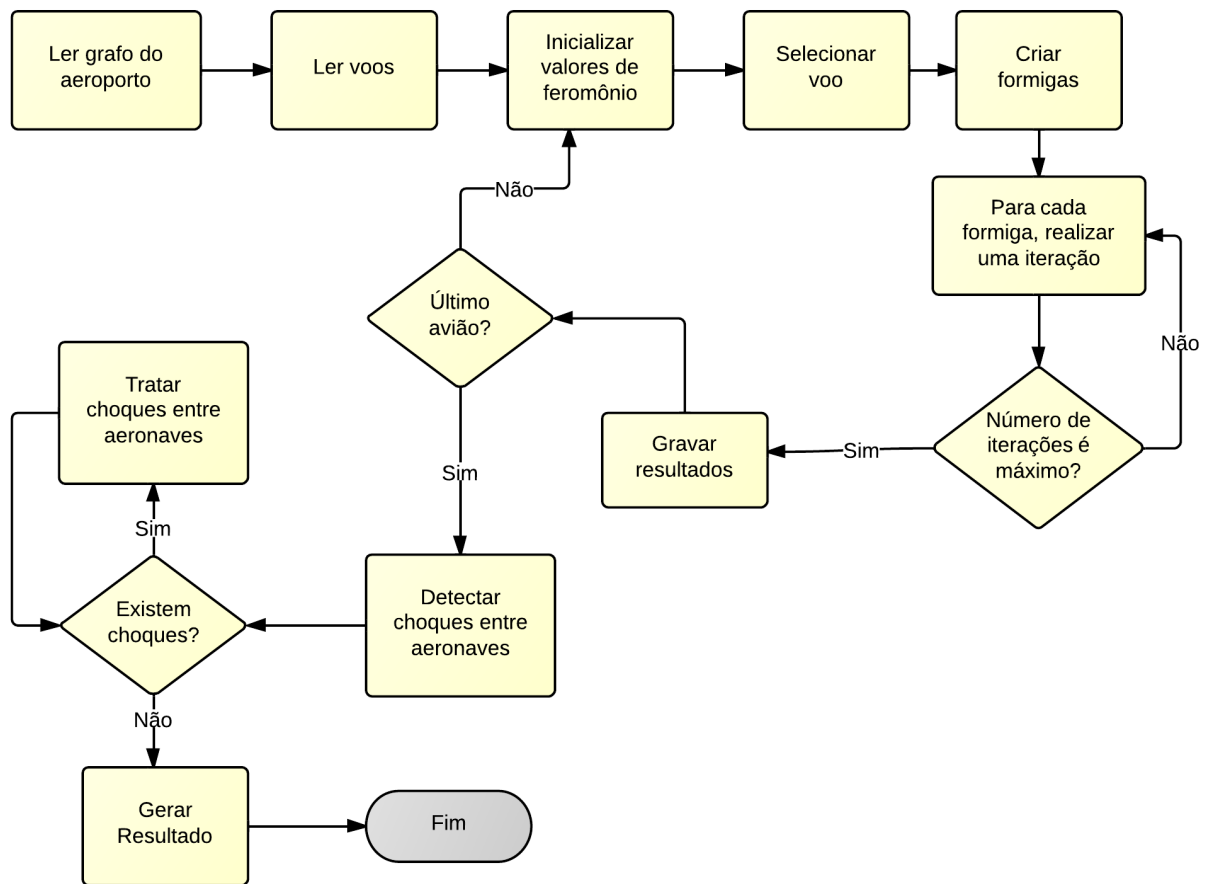


Figura 5.2: Arquitetura básica do algoritmo

O processo que ocorre dentro de cada iteração é descrito no fluxograma da Figura 5.3. Nesta fase é onde ocorrerá atualização de feromônio e cálculos de probabilidade para escolha de rotas.

Este fluxograma pode ser descrito da seguinte forma:

1. Ao iniciar uma iteração, cada formiga da colônia, formada por 60 formigas, verifica a origem e o destino do avião que será escalado. Caso não tenha chegado ao destino, a formiga verifica quais são os possíveis caminhos para seguir a partir de sua posição atual (seguir para o passo 2). Caso a formiga tenha chegado ao seu destino, seguir para o passo 5.
2. A probabilidade P_{ij} de escolha é analisada para cada uma das possíveis rotas que a formiga pode seguir, ou seja, as rotas ainda não percorridas que são possíveis de escolha a partir da posição atual. Caso não exista mais caminhos possíveis, a formiga reinicializa o percorrimento.
3. Calculada a probabilidade, a formiga se move para o próximo nó. Após este passo, o feromônio na rota percorrida é decaído de acordo com a Equação 5.2.

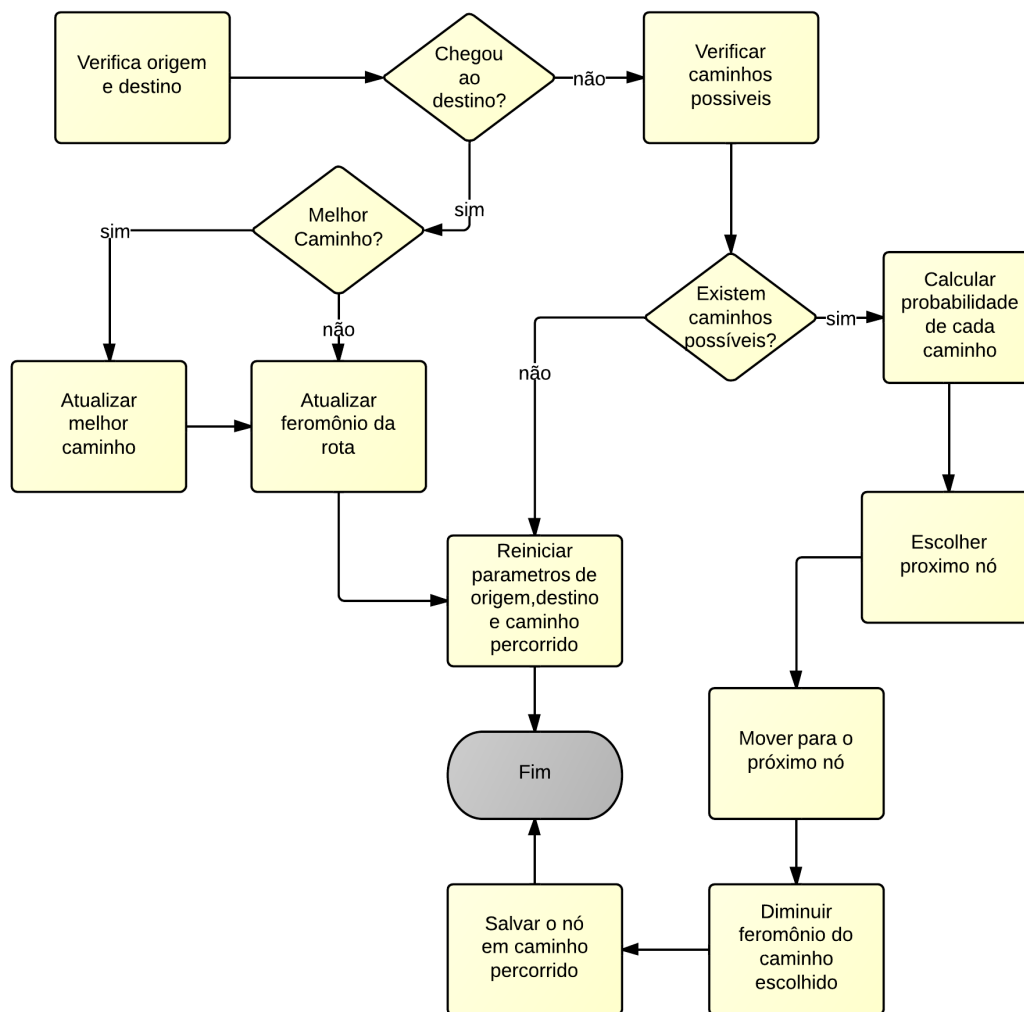


Figura 5.3: Busca pela solução ótima

4. O caminho percorrido é gravado para cada uma das formigas e a iteração é finalizada.
5. Se o caminho percorrido pela formiga for o melhor caminho dentre todas soluções encontradas pela colônia, a solução gravada como melhor caminho é atualizada.
6. Caso o destino tenha sido atingido, os valores de feromônio em todos os trechos percorridos pela formiga, e que pertencem à melhor solução gravada, são atualizados conforme a Equação 5.3
7. Os parâmetros de origem, destino e caminho percorrido são reinicializados e a iteração é finalizada.

5.3 Implementação

Nesta seção, discorreremos sobre como implementamos a parte de leitura e otimização dos dados, execução do algoritmo formiga, detecção de conflitos e resultados. O programa principal é o *Formiga.exe*, que chama todos os programas auxiliares para a execução completa do escalonamento das rotas para um grupo de aviões em um aeroporto.

A solução foi desenvolvida utilizando-se a linguagem C e o ambiente utilizado para o desenvolvimento e a realização dos testes é descrito na Tabela 5.1

Hardware	MacBook 2.16GHz Intel Core 2 Duo, 2GB 667MHz DDR2 SDRAM (PC2-5300) Memory, 120GB HD
Sistema Operacional	Windows 7 Ultimate
Compilador	GCC 4.6.4
IDE	Dev-C/C++ 5.0

Tabela 5.1: Ambiente de implementação e testes

5.3.1 Otimização de entrada de dados

O primeiro programa chamado pelo algoritmo é o *Auxiliar.exe*, que tem a função de ler o grafo com as distâncias entre os nós em metros, na forma de pontos flutuantes e transformar em segundos, a partir do pressuposto de que cada avião tem uma velocidade média constante de 5 m/s.

Este pequeno programa lê um arquivo de texto chamado *input0.txt* e grava o grafo otimizado em *inputX.txt*, que por sua vez serve de entrada para cada vez que um avião tiver sua rota de taxiamento escolhida. Este arquivo conterá o nome de cada nó, os nós que possuem ligação e as distâncias entre estes.

5.3.2 Cálculo de rota para cada avião

Executado para cada voo individualmente, o programa *Colonia.exe* é chamado várias vezes e a cada vez retorna o resultado parcial da escolha de rota para cada avião. O programa principal coleta cada um desses resultados parciais e salva no arquivo de texto *Resultado.txt*. Este programa executa a leitura do grafo, ou seja, da origem, destino e tempo do voo, em seguida, calcula a melhor rota utilizando o princípio de otimização do sistema formiga.

Inicialização de feromônio e ativação da colônia

As funções utilizadas nesta parte da implementação são:

- *void feromonio* (caminho **inicio) - esta função inicializa os valores de feromônio.
- *void colonia* (caminho **cam, caminoper **per) - esta função chama as funções auxiliares de leitura de parâmetros (criação do grafo, leitura da lista de vôos), cria e executa a ação de cada formiga, atualiza valores de feromônios no grafo, armazena os melhores resultados e chama a função auxiliar *void resultado* para escrever a saída.

- *int* formiga (int distanciaP, int atual, int destino, int tempoI, caminho **inicio, caminhoper **percorrido) - esta função verifica os caminhos possíveis a partir do nó atual, calcula as probabilidades de escolha de cada nó, realiza a escolha, e move a formiga para o próximo nó.
- *int* valorFeromonio (int a, int b, int c, caminho **inicio) - chamada dentro da função *int formiga*, retorna o valor do feromônio entre 2 nós *a* e *b*, localizados no grafo *c*.

Percorrimento e atualização de feromônio

As funções utilizadas nesta parte da implementação são:

- *int* valorFeromonio (int a, int b, int c, caminho **inicio) - chamada dentro da função *int formiga*, retorna o valor do feromônio entre 2 nós *a* e *b*, localizados no grafo *c*.
- *void* lerVoos (voo **voos) - chamada dentro da função *int colonia*, executa a leitura do voo a ser escalado.
- *void* atualizaFeromonio (caminho **inicio, caminhoper **inicio2) - chamada na função *int colonia*, é executada sempre que uma formiga termina de percorrer um grafo e encontra um resultado. Ela ajusta o feromônio nos trechos percorridos usando o principio atualização *offline* de feromônio.
- *void* decaimentoFeromonio (caminho **inicio) - essa função é executada sempre que uma formiga se move entre dois pontos, e executa o decaimento de feromônio local no trecho.

5.3.3 Detecção e resolução de conflitos

Uma vez que exista uma solução calculada para cada um dos aviões, é necessário garantir os requisitos mínimos de segurança entre as aeronaves, onde nenhuma aeronave deve se aproximar muito da outra, respeitando uma distância mínima, que na implementação determinamos ser de 30 segundos (o que correspondente a aproximadamente 150 metros). O segundo requisito de segurança é que duas aeronaves não colidam frontalmente.

O programa *Conflitos.exe* analisa um conjunto solução para um grupo de aeronaves e busca por problemas de aproximação, em que duas aeronaves se situam a uma distância inferior à distância mínima. Este programa verifica cada trecho direcionado a cada avião, compara com os aviões anteriores e, caso exista um conflito, é resolvido de forma que o avião escalado anteriormente tenha a preferência, e o segundo avião espere pelo tempo necessário para se obter a distancia mínima permitida.

O programa *ColisaoFrontal.exe* tem a função de encontrar possíveis colisões frontais e trata-las. Semelhante à política adotada em *Conflitos.exe*, o primeiro avião tem a preferência e o avião que foi alocado posteriormente tem um aumento no tempo nos seus trechos, começando a partir daquele que se detectou o conflito e voltando em direção ao ponto inicial, até que não existam mais choques.

O programa principal chama cada um dos programas auxiliares 3 vezes, intercaladamente, de modo a eliminar todos os conflitos. Em casos de grande fluxo de aeronaves, em que mais choques poderão ocorrer, esta quantidade de execuções poderá ser maior, porém para uso no aeroporto de Brasília foi suficiente.

Capítulo 6

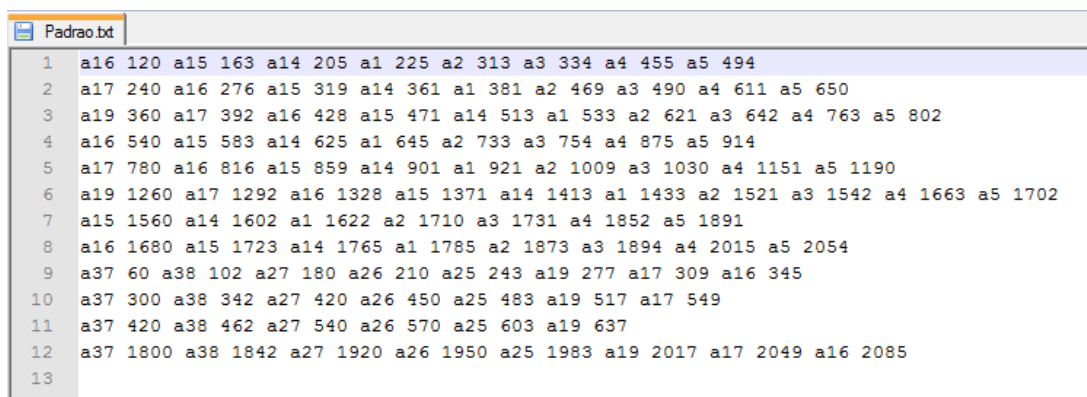
Simulações e resultados

Os testes e simulações realizados no sistema foram feitos alterando-se parâmetros para fins de comparação de resultados. Estas alterações consistem em mudanças de cenário (variação no fluxo de aeronaves) e de parâmetros utilizados no algoritmo formiga. Este capítulo apresenta os resultados obtidos com a aplicação do sistema formiga para sequenciamento de táxi de aeronaves.

6.1 Planejamento de estudo

Através dos dados referentes ao aeroporto de Brasília, contendo cada um dos nós com suas respectivas coordenadas geográficas, foi possível calcular as distâncias entre estes. Além disso, com os algoritmos desenvolvidos foi possível otimizar o mapa o aeroporto, de forma que a grande quantidade de nós não causasse uma distorção de resultados.

As rotas apresentadas como solução do taxiamento para 12 aeronaves é mostrada na figura 6.1, seguindo o padrão de [nome do nó] [tempo em segundos]. Pela análise das rotas e dos tempos, percebe-se que não houve choque entre aeronaves.



```
1 a16 120 a15 163 a14 205 a1 225 a2 313 a3 334 a4 455 a5 494
2 a17 240 a16 276 a15 319 a14 361 a1 381 a2 469 a3 490 a4 611 a5 650
3 a19 360 a17 392 a16 428 a15 471 a14 513 a1 533 a2 621 a3 642 a4 763 a5 802
4 a16 540 a15 583 a14 625 a1 645 a2 733 a3 754 a4 875 a5 914
5 a17 780 a16 816 a15 859 a14 901 a1 921 a2 1009 a3 1030 a4 1151 a5 1190
6 a19 1260 a17 1292 a16 1328 a15 1371 a14 1413 a1 1433 a2 1521 a3 1542 a4 1663 a5 1702
7 a15 1560 a14 1602 a1 1622 a2 1710 a3 1731 a4 1852 a5 1891
8 a16 1680 a15 1723 a14 1765 a1 1785 a2 1873 a3 1894 a4 2015 a5 2054
9 a37 60 a38 102 a27 180 a26 210 a25 243 a19 277 a17 309 a16 345
10 a37 300 a38 342 a27 420 a26 450 a25 483 a19 517 a17 549
11 a37 420 a38 462 a27 540 a26 570 a25 603 a19 637
12 a37 1800 a38 1842 a27 1920 a26 1950 a25 1983 a19 2017 a17 2049 a16 2085
13
```

Figura 6.1: *Output* da execução do código para fluxo médio de aeronaves

Para a primeira aeronave, por exemplo, temos que o nó de origem é o *a16*, onde aeronave se encontra no tempo $t = 120s$. Em seguida ela parte para o nó *a15*, chegando lá em $t = 163s$ e assim por diante, até atingir seu destino, o nó *a5*, em $t = 494s$, gastando, portanto, um tempo de 374 segundos ou 6,2 minutos para completar o trajeto.

Os horários dos vôos que serviram de base para os testes foram obtidos no site do aeroporto de Brasília [7], e referem-se ao plano de vôos do dia 31 de julho de 2013, em horários variados.

Para os casos de decolagem, foi escolhido um nó de origem localizado em um portão de embarque e um nó de destino localizado em uma das *runways*. Para os casos de aterisagem, foram escolhidos dois nós no sentido inverso, da mesma maneira.

Na tabela 6.1 são apresentados os valores padrão de parâmetros que foram usados nas simulações com fluxos variados.

Valores padrões de parâmetros	
α	1
β	1
Q	100
τ_0	10
ϕ	0.01
ρ	0.01
τ inicial	50
Iterações	50.000

Tabela 6.1: Valores para testes

Observamos que foram necessárias pelos menos 50.000 iterações para que o algoritmo convergisse para um resultado, executando em aproximadamente 4 segundos para cada voo. Para um valor de 100.000 iterações por formiga, obtivemos um desempenho pior, com o programa demorando cerca de 11 segundos por avião.

6.1.1 Caso 1: Fluxo baixo

Para simulações com baixo fluxo de aeronaves, foram escolhidos voos entre 06h00 e 06h45. Os resultados são mostrados na figura 6.2

	A	B	C	D	E	F	G	H	I	J
1	Voo	Data	Previsao	Confirma	Destino	TIPO	Nó origem	Nó destino	Tempo inicial(s)	Tempo Final(s)
2	4749	31/jul	06:00	05:53	São Paulo	Partida	a16	a5	0	347
3	1514	31/jul	06:17	06:17	Goiânia	Partida	a17	a37	1020	1269
4	1481	31/jul	06:20	06:20	Rio de Janeiro	Partida	a19	a5	1200	1642
5	6217	31/jul	06:45	06:45	Rio de Janeiro	Partida	a15	a37	2700	3028
6	3543	31/jul	06:30	06:44	Manaus	Chegada	a5	a16	1800	2147
7	2081	31/jul	06:32	06:11	Porto Velho	Chegada	a37	a19	1920	2137
8	Tempo Total:								8640	10570

Figura 6.2: Tempos de taxiamento em horário de baixo fluxo

Pela análise dos resultados acima, percebemos que o tempo total de taxiamento foi $T_{final} - T_{inicial} = 1930s$, ou 32,16 minutos para os 6 voos analisados, uma média de 5,36 minutos por voo.

6.1.2 Caso 2: Fluxo médio

Para simulações com fluxo médio de aeronaves, foram escolhidos voos com previsão de chegada ou partida entre 12h00 e 12h30. Os resultados são mostrados na figura 6.3

	A	B	C	D	E	F	G	H	I	J
1	Voo	Data	Previsão	Confirma	Destino	TIPO	Nó origem	Nó destino	Tempo inicial(s)	Tempo Final(s)
2	6300	31/jul	12:02	12:08	Recife	Partida	a16	a5	120	494
3	6222	31/jul	12:04	12:04	Aracaju	Partida	a17	a5	240	650
4	3578	31/jul	12:06	12:06	Guarulhos	Partida	a19	a5	360	802
5	6125	31/jul	12:09	12:09	Juazeiro do Norte	Partida	a16	a5	540	914
6	3845	31/jul	12:13	13:48	Belo Horizonte	Partida	a17	a5	780	1190
7	1760	31/jul	12:21	12:15	Cuiabá	Partida	a19	a5	1260	1702
8	6322	31/jul	12:26	12:26	Petrolina	Partida	a15	a5	1560	1891
9	6306	31/jul	12:28	13:09	Maceió	Partida	a16	a5	1680	2054
10	3718	31/jul	12:01	11:57	São Paulo	Chegada	a37	a16	60	345
11	3597	31/jul	12:05	12:05	Manaus	Chegada	a37	a17	300	549
12	3024	31/jul	12:07	12:19	Rio de Janeiro	Chegada	a37	a19	420	637
13	1684	31/jul	12:30	12:48	Guarulhos	Chegada	a37	a16	1800	2085
14	Tempo total:								9120	13313

Figura 6.3: Tempos de taxiamento em horário de médio fluxo

Pela análise dos resultados acima, percebemos que o tempo total de taxiamento para as 12 aeronaves foi $T_f - T_i = 4193s$, ou 69,88 minutos, uma média de 5,82 minutos por voo.

6.1.3 Caso 3: Fluxo alto

Para simulações com fluxo médio de aeronaves, foram escolhidos voos com previsão de chegada ou partida entre 09h30 e 10h00. Os resultados são mostrados na figura 6.4

Pela análise dos resultados, percebemos que o tempo total de taxiamento para as 19 aeronaves escaladas foi $T_f - T_i = 6757s$, ou 112,61 minutos, uma média de 5,92 minutos por voo.

6.2 Modificação de parâmetros

Esta sessão apresenta os resultados obtidos ao realizar-se a modificação dos diversos parâmetros presentes no algoritmo. Os valores nos gráficos estão mostrados em termos da variação do tempo. Utilizamos para os testes um fluxo médio de aeronaves, com os valores padrões apresentados na tabela 6.1.

1. Fator inspiração: é a constante α . Este parâmetro controla a importância relativa do feromônio em uma rota i . O resultado de sua variação é mostrado no gráfico 6.5.

A partir da variação de α , podemos perceber que a carga de feromônio em cada caminho tem um grande peso no fator de convergência, pois para valores inferiores ao padrão percebemos grande diferença nos resultados encontrados, chegando a mais de 500 segundos quando $\alpha = 0$, o que representa um impacto bastante significativo

	A	B	C	D	E	F	G	H	I	J
1	Voo	Data	Previsao	Confirma	Destino	TIPO	Nó origem	Nó destino	Tempo inicial(s)	Tempo Final(s)
2	1864	31/jul	09:36	09:36	Salvador	Partida	a16	a5	2160	2534
3	3457	31/jul	09:38	09:38	Guarulhos	Partida	a17	a5	2280	2690
4	3025	31/jul	09:40	09:40	Rio de Janeiro	Partida	a19	a5	2400	2862
5	3057	31/jul	09:42	09:42	Porto Alegre	Partida	a16	a5	2520	2894
6	5019	31/jul	09:44	09:41	Porto Alegre	Partida	a17	a5	2640	3050
7	248	31/jul	09:46	09:37	Miami	Partida	a19	a5	2760	3202
8	1912	31/jul	09:48	09:51	São Luis	Partida	a15	a5	2880	3211
9	1407	31/jul	09:50	09:50	São Paulo	Partida	a16	a5	3000	3374
10	3540	31/jul	09:52	09:50	Boa Vista	Partida	a17	a5	3120	3530
11	1764	31/jul	09:54	09:50	Teresina	Partida	a19	a5	3240	3682
12	1654	31/jul	09:56	09:56	Manaus	Partida	a16	a5	3360	3734
13	3840	31/jul	09:58	09:58	Fortaleza	Partida	a17	a5	3480	3890
14	1485	31/jul	10:00	09:54	Rio de Janeiro	Partida	a19	a5	3600	4042
15	3716	31/jul	09:30	09:20	São Paulo	Chegada	a37	a16	1800	2085
16	1082	31/jul	09:35	09:27	Belo Horizonte	Chegada	a37	a17	2100	2349
17	3518	31/jul	09:41	09:41	Vitória	Chegada	a37	a19	2460	2677
18	4430	31/jul	09:43	09:38	Belo Horizonte	Chegada	a37	a16	2580	2865
19	3783	31/jul	09:45	09:32	João Pessoa	Chegada	a37	a17	2700	2949
20	1817	31/jul	09:59	12:59	Natal	Chegada	a37	a19	3540	3757
21	Tempo total:								52620	59377

Figura 6.4: Tempos de taxiamento em horário de alto fluxo

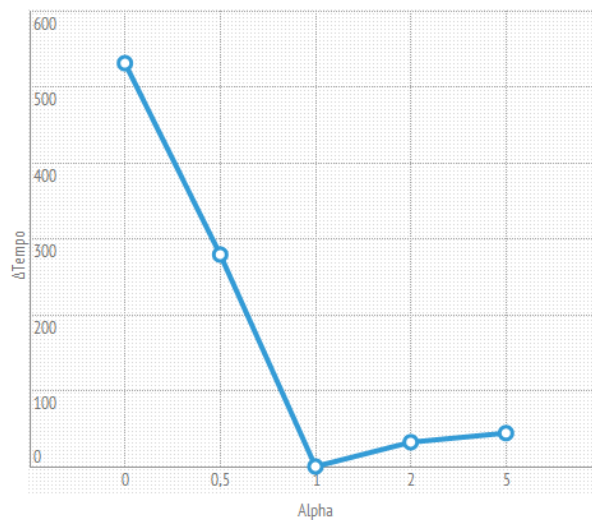


Figura 6.5: Alterações no tempo de táxi provocadas por α

no desempenho. De modo similar, para valores de α maiores que 1 temos variações no desempenho, que apesar de serem menores que para valores inferiores a 1, ainda sim são significativas, mostrando que o melhor valor para α deve estar próximo de 1.

2. Fator de visibilidade: é a constante β . Este parâmetro influencia na importância do comprimento da rota, impactando na probabilidade da escolha de uma formiga.

O resultado de sua variação sobre o tempo total de taxiamento é mostrado no gráfico 6.6.

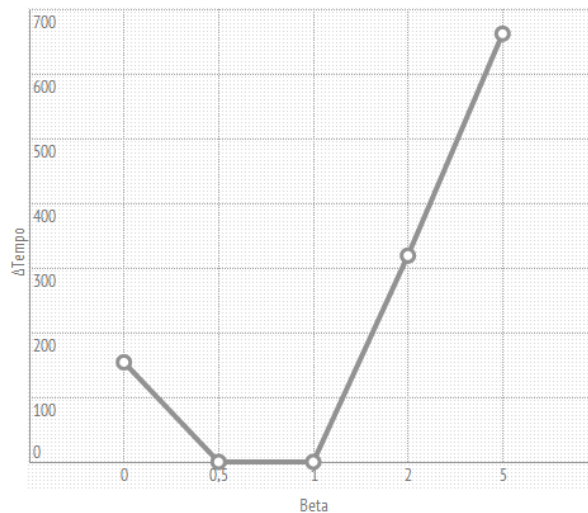


Figura 6.6: Alterações no tempo de táxi provocadas por β

Analisando o gráfico da variação de desempenho em relação a constante β , podemos observar que obtivemos bons resultados tanto para $\beta = 1$, como $\beta = 0,5$. Para $\beta = 0$, temos uma distorção significativa, e com valores de $\beta \geq 2$, o grafo apresenta resultados muito piores à medida que β aumenta.

Podemos justificar essas variações pelo fato do aeroporto apresentar distâncias pequenas entre os pontos presentes em seus terminais e em curvas, enquanto os pontos presentes em linhas retas possuem distâncias maiores. Ao aumentar o fator de visibilidade (β), é causada uma distorção nos resultados pela tendência da aeronave a seguir para menor o caminho, que nem sempre é o mais interessante.

3. Fator de convergência: É o parâmetro Q . O fator de convergência é uma constante que, juntamente com a informação do comprimento da rota, determina a quantidade de feromônio que será depositada.

A influência que esta constante exerce no tempo de taxiamento é mostrada no gráfico da figura 6.7

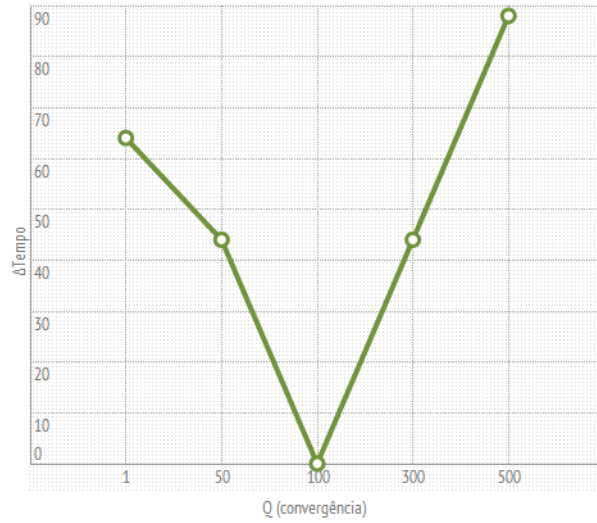


Figura 6.7: Alterações no tempo de táxi provocadas por Q

Podemos observar que os melhores resultados foram obtidos com $Q = 100$. No algoritmo da colônia de formigas em [13] esse parâmetro é sempre igual a 1, mas devido às características do grafo, onde o peso da visibilidade pode ser determinante, decidimos testar outros valores para esse parâmetro. Apesar de não existirem divergências tão grandes como as ocorridas com as variações dos outros parâmetros, podemos perceber que a variação de Q também afetou o desempenho final obtido.

4. Feromônio inicial (τ_0): Determina o valor mínimo de feromônio que cada rota terá. A figura 6.8 mostra a variação no tempo provocada por alterações do valor de τ_0 .

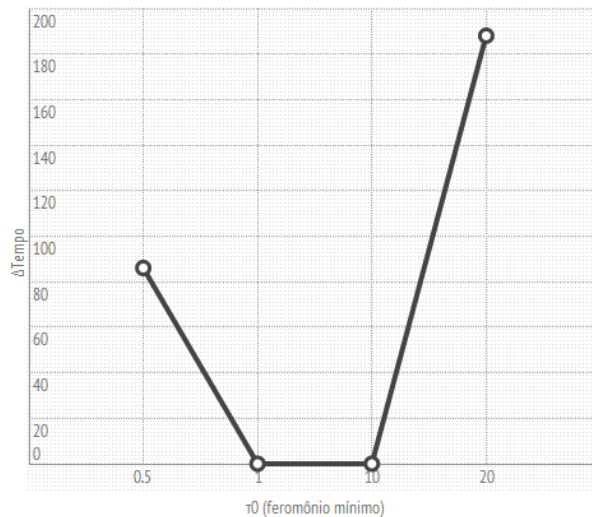


Figura 6.8: Alterações no tempo de táxi provocadas por τ_0

5. Decaimento: É o parâmetro $\phi \in (0, 1]$, que representa o coeficiente de decaimento do feromônio. Este parâmetro influencia na atualização local de feromônio, fazendo com que o feromônio de cada caminho imediatamente percorrido por uma formiga

seja diminuído, com o objetivo de aumentar a variação de escolha de rotas por formigas subsequentes.

O impacto de sua variação no tempo de táxi é mostrado no gráfico 6.9

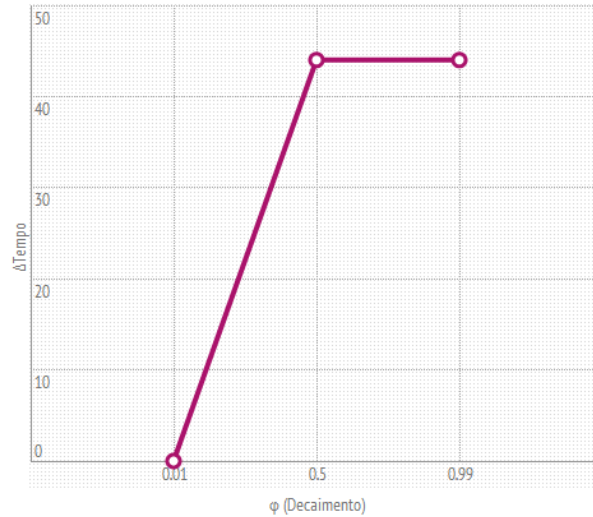


Figura 6.9: Alterações no tempo de táxi provocadas por ϕ

Podemos observar que o melhor valor para essa constante foi próximo de 0,01 e que se aumentarmos, a partir de certo ponto, obteremos uma queda constante no desempenho.

6. Volatilidade: É a constante ρ , utilizada na atualização *offline* de feromônio. Se, ao final de uma iteração, a rota percorrida fizer parte do melhor trajeto, ρ influencia no aumento do feromônio da rota.

O impacto de sua variação é mostrado no gráfico 6.10

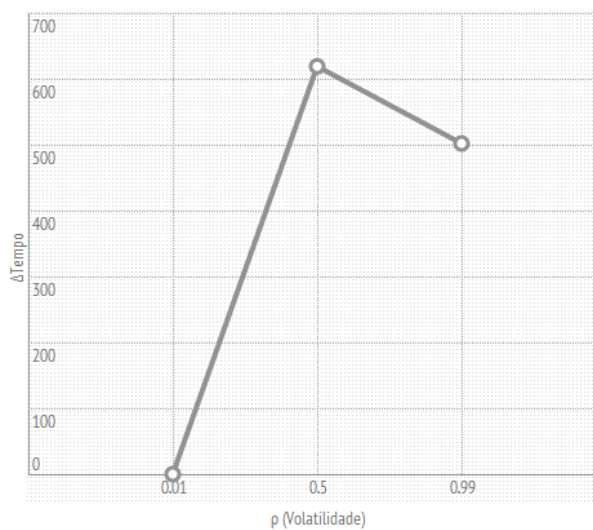


Figura 6.10: Alterações no tempo de táxi provocadas por ρ

Como podemos perceber, variações na volatilidade causam grandes distorções no desempenho. Mais uma vez o melhor valor foi obtido próximo de 0,01.

Analisando-se os resultados obtidos através da variação de cada um dos parâmetros, mantendo-se os demais constantes, percebe-se que o sistema apresenta maior sensibilidade aos parâmetros que influenciam na escolha de uma rota (α e β) e ao parâmetro ρ , que influencia na quantidade de feromônio que será incrementado nas rotas. Estes fatores indicam que a forma como é calculada uma probabilidade de escolha da rota é altamente impactante no desempenho do código, o que condiz com os preceitos da otimização da colônia de formigas, e que não deve ser depositada uma quantidade muito grande de feromônio nas rotas promissoras, pois isto diminui as chances de formigas subsequentes diversificarem as soluções encontradas.

Capítulo 7

Conclusão

O objetivo deste trabalho foi implementar uma aplicação de solução para o problema do sequenciamento de táxi aéreo utilizando-se o algoritmo de Otimização da Colônia de Formigas (ACO).

Fizemos um extenso estudo a respeito do algoritmo formiga, analisando suas possíveis aplicações para maior compreensão de seu funcionamento. Uma das dificuldades encontradas para consecução deste trabalho foi a obtenção de dados do aeroporto de Brasília e, principalmente, manipulação desses dados para que pudessem ser utilizados no sistema. Foi necessário o desenvolvimento de uma ferramenta que nos permitisse criar nosso próprio arquivo de entrada, contendo o mínimo possível de nós formando a estrutura do aeroporto.

Inicialmente, objetivou-se utilizar a vertente do ACO chamada Sistema Formiga, com a intenção de implementar o algoritmo de forma que o cálculo das melhores rotas de taxiamento fosse feito de forma global. Para tanto, a quantidade de grafos que as formigas percorrem se multiplicariam de acordo com a quantidade de aeronaves a serem escalonadas, sendo que cada grafo representaria um avião.

Contudo, o código implementado não estava convergindo para o caminho mais otimizado, além de não encontrar resultado e ser abortado na maioria das execuções. Aumentando a quantidade de grafos, a quantidade de iterações mínima para o algoritmo convergir cresce exponencialmente, tornando inviável o processamento em termos de recursos computacionais.

Ao alterar para o Sistema da Colônia de Formigas, com a utilização de apenas um grafo, o algoritmo passou a convergir e foram encontrados resultados satisfatórios. O algoritmo tem crescimento linear em relação à quantidade de aviões alocados e depende de variáveis como quantidade de iterações e parâmetros. O desempenho do código foi considerado satisfatório, apresentando um tempo de 4 a 11 segundos para encontrar resultado para cada voo. Dessa forma, conclui-se que o objetivo do projeto foi alcançado com sucesso.

Para trabalhos futuros, temos as seguintes considerações:

- Para otimizar a utilização de recursos em solo de forma ainda mais eficiente, sugerimos que outros elementos aeroportuários sejam consideradas, como os hangares.
- Sugere-se implementar mecanismos de construção de solução global, de forma que os conflitos entre aeronaves sejam tratados durante a construção das soluções.

- A utilização de computação paralela e *multi-threads* pode aumentar de forma significativa a eficiência do algoritmo, além de permitir a utilização de dados em tempo real e adaptação a mudanças de cenário, como por exemplo atraso ou cancelamento de voos.

Esta ferramenta será integrada ao sistema SiGePA e não deverá substituir a experiência do controlador de tráfego aéreo e sim ser utilizada em conjunto com esta, permitindo que o controlador realize eventuais adaptações, caso julgue necessário.

Referências

- [1] T. Araújo. <http://mundoestranho.abril.com.br/materia/como-funciona-o-controle-de-trafego-aereo>, 2012. Editora Abril S.A. [Acessado em: 16/04/2013]. 6
- [2] H. Balakrishnan and I. Deonandan. Evaluation of strategies for reducing taxi-out emissions at airports. In *AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, pages 1–2, 2010. 9
- [3] H. J. C. Barbosa. *Ant Colony Optimization - Techniques and Applications*. InTech, February 2013. vii, 25, 26
- [4] A. Bastos. Impacts of flight delays on airline pricing: Evidence from the US market. volume 2, pages 39–60. *Review of Transportation Literature*, 2008. 12
- [5] M. A. G. Carvalho. Teoria dos grafos - Uma Introdução. Notas de aula. Universidade Estadual de Campinas - UNICAMP. Centro Superior de Educação Tecnológica - CESET, 2005. 22
- [6] J. Chen and P. Stewart. Planning aircraft taxiing trajectories via a multi-objective immune optimisation. In *Natural Computation (ICNC)*, volume 4, pages 2235–2240. Seventh International Conference, 2011. 10
- [7] Aeroporto Internacional de Brasília Presidente Juscelino Kubitschek. <http://www.aerportobrasilia.net/horario-voos-aeroporto-brasilia>, 2012. Acessado em: 01/08/2013. 37
- [8] Departamento de Controle de Espaço Aéreo. Regras do ar e serviços de tráfego aéreo. Technical report, Comando da aeronáutica, 2009. 8
- [9] Portal de notícias do senado federal. <http://www12.senado.gov.br/noticias/materias/2012/02/28/debate-evidencia-problemas-na-aviacao-civil-brasileira>. [Acessado em: 16/12/2012]. 8
- [10] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3:159, 1990. 15
- [11] Problema do Caixeiro Viajante. http://http://pt.wikipedia.org/wiki/Problema_do_caixeiro_viajante. [Acessado em: 11/01/2013]. 19
- [12] M. Dorigo. Ant colony optimization. *Scholarpedia*. 2(3):1461, 2007. 2, 14, 17

- [13] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization: Artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, pages 28–30, 2006. 14, 15, 18, 41
- [14] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26:29–41, 1996. vii, 16, 17
- [15] J. Dréo and P. Siarry. Hybrid continuous interacting ant colony aimed at enhanced global optimization. *Algorithmic Operations Research*, 2(1), 2007. 1, 16
- [16] Boeing 747 fun facts. http://www.boeing.com/commercial/747family/pf/pf_facts.html. [Acessado em: 13/01/2013]. 9
- [17] W. Hickey. <http://www.businessinsider.com/map-of-the-busiest-flight-routes-in-the-world-2013-8>, 2013. Business Insider. Acessado em: 13/08/2013. vii, 4, 5
- [18] International Air Transport Association (IATA). <http://www.iata.org>, 2013. [Acessado em: 14/01/2013]. 5, 10
- [19] B. G. Júnior. Análise de algoritmos de roteamento baseados em formigas. Master’s thesis, Escola Politécnica da Universidade de São Paulo, 2006. 22
- [20] C. N. Lintzmayer, M. H. Mulati, and A. F. Silva. Rt-color ant: Um algoritmo heurístico baseado em colônia de formigas artificiais com busca local para colorir grafos. *Departamento de Informática. Universidade Estadual de Maringá (UEM)*, 2011. 22
- [21] C. Liu and Y. Wang. Aircraft taxi path optimization based on ant colony algorithm. In *Computational Intelligence and Design (ISCID)*, volume 2, pages 226–229. Fourth International Symposium, october 2011. 23, 29
- [22] A. Previdelli. <http://exame.abril.com.br/brasil/noticias/os-aeroportos-brasileiros-que-tem-mais-atrasos>, 2013. Revista Exame. Acessado em: 14/08/2013. 1
- [23] L. Polonio Ribeirinho. Simple interactive UI to mark points and paths between them in google maps. <https://github.com/lucaspolonio/gmaps-graph-maker>, 2013. 29
- [24] V. F. Ribeiro. Decisão colaborativa com utilização de teoria dos jogos para o sequenciamento de partidas em aeroportos. Master’s thesis, Universidade de Brasília, 2013. vii, 6, 7, 13
- [25] Sustainable Aviation CO2 Roadmap. <http://www3.imperial.ac.uk/greenaviation/thechallenge>, 2008. vii, 10
- [26] L. P. Rosa. Utilização de algoritmo genético para gerenciamento de sequenciamento de táxi. Monografia, Departamento de Ciência da Computação, Universidade de Brasília, 2012. 1, 27

- [27] J. Sieber. Overview of the NEWAC Programme. In *Greening of Aircraft Propulsion - Progress and Prospects*, pages 2–3, 2010. vii, 11
- [28] Introduction to Particle Swarm Optimization and Ant Colony Optimization. <http://dc234.4shared.com/img/-hMTdSx8/preview.html>. [Acessado em: 11/01/2013]. 21, 22