

Modelo para o artigo do Módulo 5

André Luís Lessa Junior, Arthur Alberto Cardoso Reis, Cristiane Andrade Coutinho, Giovan

Janeiro de 2023

Abstract

Como parte das atividades do módulo 5, cada grupo deverá redigir um texto descrevendo os resultados do projeto no formato de um artigo científico. Este arquivo no formato markdown contém a estrutura básica deste artigo. Cada grupo deverá editar este arquivo com a descrição do projeto que desenvolveu.

Introdução

O meio militar envolve operações que visam o planejamento estratégico. Esse conceito foi publicado em junho de 2011 pela Asplan, órgão de assistência direta do Ministério da Defesa brasileiro e utilizado para definir uma direção ou caminho almejado pela corporação, a fim de que gastos de recursos sejam minimizados e a eficiência do trabalho para alcançar o objetivo seja maximizada. De acordo com o INPI, Instituto Nacional de Pesquisa Industrial, para que esse planejamento estratégico consiga êxito no futuro, diversas empresas ou autarquias necessitam de tecnologias que potencializem essa ideia de estratégia, além de desempenhar um papel na persecução dos objetivos organizacionais e garantir o alinhamento da empresa com esses mesmos objetivos. Dentro do planejamento estratégico existem diversas missões que necessitam das mais variadas logísticas e estratégias, entre elas existem missões aeroespaciais de reconhecimento, resgate e ataque em percursos próximos ao solo. Assim sendo, uma das ferramentas aeroespaciais militares utilizadas nesse contexto é o radar *Terrain-Following*, permitindo que o piloto, ao fazer o reconhecimento de uma região (muitas vezes desconhecida), voe em baixa altitude de modo constante, ou seja, sem variação da elevação da aeronave. A importância desse sistema se deve ao impedimento de uma possível detecção de um radar inimigo, além de facilitar o reconhecimento de um alvo ou até mesmo compreender a estrutura do relevo local.

Entretanto, a tecnologia *Terrain-following* necessita de segurança e ponderações para a sua utilização durante as missões aéreas. Conforme o portal de comunicações da AEL, o sistema se baseia na fusão de dados provenientes do mapeamento radar e de um banco de dados de terreno e obstáculos. Assim sendo, é de suma importância que ocorra um reconhecimento prévio da região e a escolha de um melhor caminho possível. Todavia, a tecnologia *Terrain-following* não

supre essas necessidades de reconhecimento, gerando uma desvantagem inerente ao sistema. Portanto, o problema de pesquisa está relacionado ao desenvolvimento de um modelo matemático computacional que, através do conhecimento prévio da região (dados armazenados nesse banco), construa possíveis rotas que levem o avião de um ponto ao outro e especifique o caminho preferível para o seu deslocamento. Um artigo correlato foi publicado por estudantes de Ciência da Computação da Unesc (Universidade Estadual de Criciúma) e utilizava os conhecimentos de algoritmo de Dijkstra, grafos e estrutura de dados como embasamento e possuía o intuito de estudar as rotas realizadas pelo transporte público de Criciúma e encontrar as mais viáveis para a solução do problema. Esse trabalho contém similiaridades com o descrito anteriormente, contudo com ênfase na área de defesa, visto que a criação do algoritmo propicia que empresas e as forças armadas realizem o *Terrain-following*, em conjunto com a tecnologia planejadora de voos, sem a preocupação de não percorrerem um trajeto viável. O resultado esperado é que o *software* consiga destacar uma ou mais trajetórias viáveis através de dados de *input* das cidades de saída e chegada feitos pelo usuário.

Para que tal solução fosse viável, os principais materiais e métodos que foram utilizados para a construção do algoritmo de alta performance, capaz de identificar o melhor trajeto possível, são a linguagem de programação Java, necessária para a construção desses modelos computacionais, o banco de dados NoSql, que contém as informações de relevo da região disponibilizadas pela empresa AEL. Desse modo, a linguagem java com o *framework Spring Boot Web* como arquitetura MVC (*Model, View, Control*) para a realização do CRUD (*Create, Read, Update, Delete*), necessário para manipular a estrutura do banco de dados orientado à grafos através de uma *API RESTful*. Além disso, esses dados estão na extensão de arquivo DTED e por isso se faz necessário o uso da biblioteca *Geospatial Data Abstraction Library* (GDAL), disponível na linguagem Java, capaz de extrair essas informações geográficas. De tal maneira, convertemos os dados do arquivo DTED com essa biblioteca e convertemos em objetos Java. Por outro lado, para o NoSql, utilizamos outro banco de dados denominado Neo4j, orientado a grafos (manipulação dos dados através da linguagem Cypher), responsável pela modelagem matemática e a construção do raciocínio algorítmico dos fluxos. Por fim, para a visualização das rotas e a identificação do melhor caminho possível, utilizamos a biblioteca do Javascript chamada D3js. Desse modo, para que tais ferramentas fossem possíveis, todas essas tecnologias foram hospedadas através da aplicação Docker que isola as mesmas em contêineres únicos, justamente para garantir a padronização do ambiente de desenvolvimento.

No entanto, é de suma importância a compreensão do funcionamento do algoritmo e que tipo de modelo foi desenvolvido para construção lógica do projeto planejador de voos. Assim sendo, utilizamos um algoritmo que é conhecido por muitos estudiosos na área da computação. Ele foi elaborado Edsger *Dijkstra*, cientista da computação holandês, no final da década de 50 e até hoje é aplicado para propor rotas mais otimizadas. O algoritmo de *Dijkstra* começa em um vértice de origem e visita todos os outros vértices do grafo para encontrar o caminho mais curto

de origem para todos os outros vértices. Ele mantém uma lista de distâncias conhecidas a partir da origem para cada vértice do grafo, que são atualizadas à medida que o algoritmo avança. O algoritmo também mantém uma lista de vértices que ainda precisam ser visitados, e a cada iteração, escolhe o vértice com a menor distância conhecida para visitar em seguida. O algoritmo de *Dijkstra* funciona apenas para arestas ponderadas não negativas e é implementado usando uma fila de prioridade para manter a lista de vértices a serem visitados, o que torna o algoritmo eficiente em termos de tempo. Ele é amplamente utilizado em aplicações práticas, como sistemas de roteamento em redes de computadores, sistemas de navegação em mapas e jogos de estratégia. Todavia, o algoritmo de *Dijkstra* possui alguns problemas que no ponto de vista do projeto foram negativos pelo seu uso em específico. O funcionamento da aplicação são as requisições que o usuário faz no *front-end* e essas mesmas requisições são passadas para o *SPRINGBOOT* pelo protocolo *Rest*. Se é a primeira vez que a aplicação está sendo rodada, o usuário terá que popular no banco de dados com as informações do arquivo *DTED*. Para popular no banco de dados é utilizada a biblioteca *GDAL* e a aplicação manda os dados manipulados para o *Neo4j* que retorna um objeto *Json* para a aplicação no *SpringBoot* que, por sua vez, envia para o *front-end*. Existiam muitos dados no arquivo e a representação das regiões por nós desse grafo eram muito pesadas e exigiam uma alta capacidade de processamento, assim sendo a preferência por um outro algoritmo que possuía a mesma lógica do *Dijkstra* e resolvesse os problemas de processamento era considerável, portanto, o algoritmo escolhido foi o *A** (A-star) com algumas determinações heurísticas.

De maneira mais precisa, nos baseamos no *Hierarchical A* que é uma variante do algoritmo *A* que usa uma estrutura de grafo hierárquico para melhorar o desempenho e a eficiência do *A**. Esse algoritmo possui a capacidade de suprir as necessidades do algoritmo de *Dijkstra*. Dessa maneira, as principais vantagens do *A** sobre o *Dijkstra* são eficiência, visto que pode ser mais rápido do que o algoritmo de *Dijkstra* em grafos grandes. Isso ocorre porque o *H-A** usa uma estrutura de grafo hierárquico para reduzir o número de nós visitados. Isso pode levar a uma economia significativa de tempo em comparação com o algoritmo de *Dijkstra*, especialmente em grafos grandes e complexos. Também exige menos memória, pois o *H-A** usa uma estrutura de grafo hierárquico, ele pode ser mais eficiente em termos de uso de memória do que o algoritmo de *Dijkstra*. Isso ocorre porque o *H-A* armazena informações sobre subgrafos em vez de armazenar informações sobre cada nó individualmente. A terceira vantagem é a adaptabilidade do algoritmo, já que o *H-A* é mais adaptável do que o algoritmo de *Dijkstra*. Isso ocorre porque o *H-A* pode ser facilmente ajustado para lidar com diferentes tipos de heurísticas, enquanto o algoritmo de *Dijkstra* usa apenas uma medida de distância (peso da aresta). Por último existe a precisão do modelo, que geralmente é mais preciso do que o algoritmo de *Dijkstra*, pois usa uma heurística que leva em consideração informações adicionais sobre o grafo. A heurística usada pelo *H-A* pode ser personalizada para levar em conta fatores como a topologia do grafo, a localização do destino e a natureza das arestas.

Motivação

A gama de utilidades de voos de baixa altitude é bastante grande, abrangendo desde a aplicação de fertilizantes e defensivos agrícolas em grandes extensões de plantio até operações militares de busca, resgate, reconhecimento de terreno, dentre outras missões. Por sua vez, a operação desse tipo de voo apresenta riscos inerentes de colisão com o solo ou CFIT do inglês Controlled Flight Into Terrain (DA COSTA, 2019). Atualmente, a empresa parceira do projeto, AEL Sistemas, provê sistemas de Terrain Following, todavia o planejamento de uma rota possível é feita à mão. A elaboração dessa rota pode ser caracterizada como o problema do Caminho Mínimo da teoria de grafos, tendo como objetivo encontrar a rota com menor peso entre dois vértices (inicial e final) em um dado grafo G com pesos. Para suprir a lacuna de planejamento de rotas de voos de baixa altitude não automatizado pela AEL e, conseqüentemente, diminuir os riscos de colisão com o solo, a presente pesquisa trouxe um sistema que automatiza todas essas funções, desde carregar um mapa suprido pelo banco de dados, gerar um grafo a partir de suas coordenadas geográficas, adicionar pesos às arestas e, por fim, apontar o melhor caminho mínimo para a missão. # Descrição do problema

Trabalhos relacionados

Descrição da estratégia adotada para resolver o problema

Análise da complexidade da solução proposta

Neste artigo, cada grupo precisará fazer a análise de complexidade da solução proposta, utilizando as notações $O(\cdot)$, $\Omega(\cdot)$ e $\Theta(\cdot)$.

A seguir temos a citação de alguns trechos de DASGUPTA et. al. (2011) para mostrar como estas notações são em L^AT_EX.

Sejam $f(n)$ e $g(n)$ duas funções de inteiros positivos em reais positivos. Dizemos que $f = O(g)$ (que significa que “ f não cresce mais rápido do que g ”) se existe uma constante $c > 0$ tal que $f(n) \leq c \cdot g(n)$.

Ainda em outro trecho de DASGUPTA et. al. (2011), temos:

Assim como $O(\cdot)$ é análogo a \leq , podemos definir análogos de \geq e $=$ como se segue:

$f = \Omega(g)$ significa $g = O(f)$

Análise da corretude da solução proposta

Resultados obtidos

Conclusão

Referências Bibliográficas

AEL, Portal de comunicações. **Terrain Following**. Disponível em: <https://www.ael.com.br/terrain-following.html>. Acesso em: 20 fev. 2023.

ASPLAN. **Sistema de Planejamento Estratégico e Defesa**. Importância do Planejamento Estratégico, cap.3. Disponível em: <https://portal.tcu.gov.br/lumis/portal/file/fileDownload.jsp?fileId=8A8182A24F0A728E014F0AC5DCB653E4>. Acesso em: 20 fev. 2023

DA COSTA, Rodolfo Lobo. **Causa e prevenção de acidentes resultantes da colisão controlada com o solo – CFIT**. Universidade do Sul de Santa Catarina – UNISUL, Palhoça, SC. 2019.

GERSTING, Judith L. **Fundamentos matemáticos para Ciência da Computação**. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora, 2017.

GRAMANI, Cristina M; **Abordagem de Caminho Mínimo para problemas de otimização**, XXXVI - SBPO, São João Del-Rei, MG. 23 a 26 de novembro de 2004.

L., A. et al. **Utilização do Algoritmo de Dijkstra para Cálculo de Rotas no Trabalho Público do Município de Criciúma/SC**. Disponível em: <https://periodicos.unesc.net/ojs/index.php/sulcomp/article/download/1815/1717/5477>. Acesso em: 20 fev. 2023.

LINTZMAYER, CN; OLIVEIRA MOTA, G. **Análise de Algoritmos e Estruturas de Dados**. Estrutura de dados, cap.3. Disponível em: https://www.ime.usp.br/~mota/livros/livro_AAED.pdf. Acesso em: 20 fev. 2023.

LUIZ, O. **PETI**. Introdução, cap.2. Disponível em: https://www.gov.br/inpi/pt-br/aceso-a-informacao/tecnologia-da-informacao/arquivos/documentos/peti_20162019.pdf. Acesso em: 20 fev. 2023.

PRESTES, E. **Introdução à Teoria dos Grafos**. Fundamentação Teórica, cap.1. Disponível em: <https://www.inf.ufrgs.br/~prestes/Courses/Graph%20Theory/Livro/LivroGrafos.pdf>. Acesso em: 20 fev. 2023.