

Simulação de Roteamento de Robôs em Armazéns Utilizando o Algoritmo de Dijkstra para Caminhos Mais Curtos

Carla Cavalcante

Resumo—Este projeto simula a otimização do roteamento de robôs em grandes armazéns, semelhantes aos utilizados por empresas como a Amazon. Utilizando o algoritmo de Dijkstra, o sistema determina os caminhos mais curtos entre as seções do armazém, visando melhorar a eficiência no manejo de estoque e reduzir o tempo de trânsito dos robôs. O algoritmo considera o tempo médio necessário para percorrer os trajetos como o peso das arestas em um grafo ponderado. A abordagem proposta ilustra a aplicação do algoritmo de Dijkstra em uma situação real, demonstrando como ele pode otimizar a navegação e a operação de robôs em ambientes de armazenamento extensos.

Keywords—Grafos; Dijkstra; Caminho Curto;

I. INTRODUÇÃO

Em armazéns de grande porte, como os utilizados por grandes empresas de comércio eletrônico, a movimentação eficiente dos robôs é fundamental para garantir que os produtos sejam manuseados rapidamente e sem erros. À medida que os armazéns ficam fisicamente maiores, se torna cada vez mais necessário encontrar rotas eficientes para esses robôs terem uma otimização de tempo e desgaste. Neste trabalho, implementamos o algoritmo de Dijkstra de forma simplificada, utilizando uma analogia com um armazém real.

Conhecido por encontrar o caminho mais curto em grafos, pode ser aplicado para melhorar a navegação dos robôs em um armazém. Ao representar o armazém como um grafo, onde cada seção é um vértice e os caminhos entre elas são arestas com pesos que indicam o tempo de deslocamento, buscamos mostrar como essa técnica pode tornar a operação dos robôs mais eficiente. O objetivo é reduzir o tempo que os robôs levam para se deslocar de uma seção para outra, contribuindo para uma gestão de estoque mais rápida e eficaz. Esta analogia destaca como conceitos de algoritmos podem ser aplicados na prática para resolver problemas reais em logística.

II. CONCEITOS IMPORTANTES

Para o desenvolvimento do trabalho foi utilizados conceitos importantes aprendidos em sala de aula, são eles:

A. Grafos Ponderados

Grafos são estruturas matemáticas usados para modelados relações entre objetos. Eles são compostos por vértices (ou nós) e arestas (ou linhas) que ligam esses vértices. No nosso caso, cada vértice representa uma seção do armazém, e as arestas representam os caminhos entre essas seções, com um peso que indica o tempo que um robô leva para ir de uma seção a outra.

B. Dijkstra

Algoritmo de Dijkstra é um método usado para encontrar o caminho mais curto entre dois pontos em um grafo ponderado. Ele funciona escolhendo o próximo ponto a ser visitado com base no menor caminho conhecido até o momento. Neste projeto, usamos o algoritmo para encontrar a rota mais rápida para os robôs dentro do armazém.

III. METODOLOGIA

Para fins de demonstração, foi desenvolvido um programa em linguagem C, utilizando Sistema Operacional Windows 11 e ambiente de desenvolvimento Visual Studio Code. O projeto foi desenvolvido nas seguintes etapas:

A. Modelagem do Armazém como um Grafo

Primeiro, o armazém foi representado como um grafo ponderado representado como matriz de adjacências. Cada seção do armazém foi tratada como um vértice do grafo, e os caminhos entre essas seções foram modelados como arestas. O peso de cada aresta foi definido como o tempo médio que um robô leva para percorrer o caminho entre as duas seções conectadas.

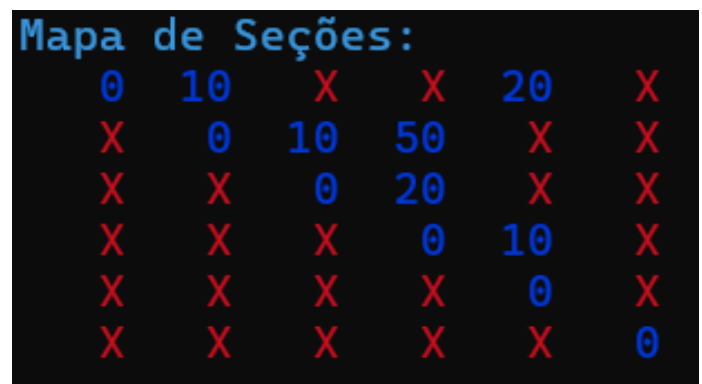


Fig. 1. Demonstração do grafo

B. Implementação do Algoritmo de Dijkstra:

Em seguida, foi implementado o algoritmo de Dijkstra, que é utilizado para encontrar o caminho mais curto em grafos ponderados. O algoritmo foi programado em C, utilizando uma matriz de adjacência para representar o grafo. A função principal do algoritmo calcula a menor distância entre uma seção inicial e todas as outras seções do armazém.

C. Simulação dos Cenários:

Para testar a eficiência do algoritmo, foram simulados diferentes cenários onde o robô precisaria se mover de uma seção a outra. Esses cenários foram definidos em um arquivo de texto, que foi lido pelo programa para carregar as informações do grafo. Seguindo esse modelo:

Quantidade de Vértices: 6			
Origem	Destino	Peso (Tempo)	
0	1	10	
0	4	20	
1	2	10	
1	3	50	
2	3	20	
3	4	10	

TABLE I

DEMONSTRAÇÃO DO ARQUIVO TEXTO CONTENDO COMO PRIMEIRA LINHA A QUANTIDADE DE VÉRTICES, A PRIMEIRA COLUNA O VÉRTICE DE ORIGEM, A SEGUNDA O VÉRTICE DE DESTINO E A TERCEIRA O PESO DAS ARESTAS

IV. RESULTADOS

Ao rodar a simulação, observou-se que o algoritmo foi capaz de identificar os caminhos mais curtos entre as seções do armazém, levando em consideração os pesos atribuídos a cada aresta, que representam o tempo de deslocamento entre as seções. Os tempos obtidos para os trajetos entre os pontos de origem e destino foram consistentemente os menores possíveis, demonstrando a eficiência do algoritmo na otimização do deslocamento dos robôs. Por exemplo, considerando um cenário onde o robô precisaria se mover da seção 0 para a seção 3, o algoritmo identificou que a rota mais curta seria 0 - 1 - 2 - 3, com um tempo total de deslocamento de 40 unidades de tempo. Esse resultado confirma que a aplicação do algoritmo é capaz de minimizar o tempo de operação, reduzindo o tempo que os robôs gastam para se mover entre as seções.

```
Digite a seção de origem: 0
Digite a seção de destino: 3

Calculando o caminho mais curto...
Distância mais curta da seção 0 para seção 3 é 40
Por onde passar: 0 -> 1 -> 2 -> 3

Distâncias atuais:
Distância para o seção 0: 0
Distância para o seção 1: 10
Distância para o seção 2: 20
Distância para o seção 3: 40
Distância para o seção 4: 20
```

Fig. 2. Demonstração do resultado

V. CONCLUSÃO

Este trabalho mostrou que conceitos teóricos de grafos aprendidos em sala de aula podem ser aplicados de forma

eficaz para resolver problemas reais, poder colocar em prática e observar a eficiência do algoritmo de Dijkstra foi uma experiência enriquecedora, servindo para solidificar o conhecimento teórico e compreender na prática os desafios de implementação.