

## Introdução

Na resolução de problemas logísticos e operacionais, buscar a máxima eficiência e o menor custo é essencial. Em cenários dinâmicos, onde o ambiente sofre alterações constantes, abordagens tradicionais e algoritmos estáticos de pré-processamento tornam-se ineficazes, pois falham em recalculando rotas de custo mínimo em tempo real.

Para superar essa limitação, a estrutura de dados em **grafos** surge como a modelagem matemática ideal, traduzindo espaços físicos complexos em uma malha navegável de vértices e arestas. É neste contexto de abstração que **algoritmos de caminhos mínimos**, como **Dijkstra** e **A\*** (*A-Estrela*), destacam-se, determinando a rota mais vantajosa com a eficiência computacional necessária para viabilizar tomadas de decisão ágeis.

Neste pôster, exploramos a resolução de trajetórias em ambientes mutáveis utilizando o algoritmo heurístico A\* e sua variante otimizada, o algoritmo *A-estrela com expansão de Triângulos (REA\*)* (ZHANG et al., 2016). Para tangibilizar a teoria matemática, demonstramos sua aplicação prática na automação da **Indústria 4.0**, otimizando as rotas de robôs autônomos que precisam recalculando caminhos e desviar de obstáculos dinamicamente.

## Metodologia

O algoritmo de Dijkstra garante a descoberta da rota ótima com uma complexidade assintótica de  $O(|E| + |V| \log |V|)$  quando operado com uma Fila de Prioridade Mínima (*Min-Priority Queue*). Contudo, sua expansão omnidirecional explora vértices em um padrão radial "cego". Em malhas densas de 8 conexões, isso gera uma explosão de estados avaliados, criando gargalos inaceitáveis para cálculos de rotas em tempo real.

Para mitigar o alto custo de processamento, implementamos o algoritmo A\*. Ele mantém a garantia de otimização do caminho mínimo, mas direciona a busca de forma inteligente através da função de avaliação de custo total estimado:

$$f(n) = g(n) + h(n)$$

onde  $g(n)$  é o custo real acumulado do início até o nó atual  $n$ , e  $h(n)$  é a heurística (o custo estimado até o destino).

Como a movimentação ocorre em 8 eixos, métricas clássicas como a Distância de Manhattan perdem sua eficácia ou admissibilidade. A métrica matematicamente ideal para este cenário é a Distância Octil. Definindo  $\Delta x = |x_n - x_{destino}|$  e  $\Delta y = |y_n - y_{destino}|$ , a função heurística é dada por:

$$h(n) = D \cdot \max(\Delta x, \Delta y) + (D_2 - D) \cdot \min(\Delta x, \Delta y)$$

onde  $D$  é o custo do movimento ortogonal e  $D_2$  é o custo do movimento diagonal, tipicamente  $\sqrt{2}$ . Essa formulação direciona a exploração geometricamente para o alvo, cortando caminhos diagonais precisos e minimizando o uso de memória. Veja em Figura 1.

Para superar o gargalo de memória gerado pela expansão célula a célula em áreas abertas, aplicamos o conceito do algoritmo REA\*. Em vez de avaliar pontos individuais, esta abordagem explora o mapa em unidades de retângulos livres de obstáculos. Todos os pontos desnecessários dentro desses retângulos são podados, e apenas as fronteiras externas são utilizadas como nós de busca. Por operar de forma totalmente online e não exigir nenhum pré-processamento offline, o REA\* mantém a flexibilidade necessária para armazéns dinâmicos, gerando uma fila de prioridade (*open list*) consideravelmente menor em comparação ao A\* padrão.

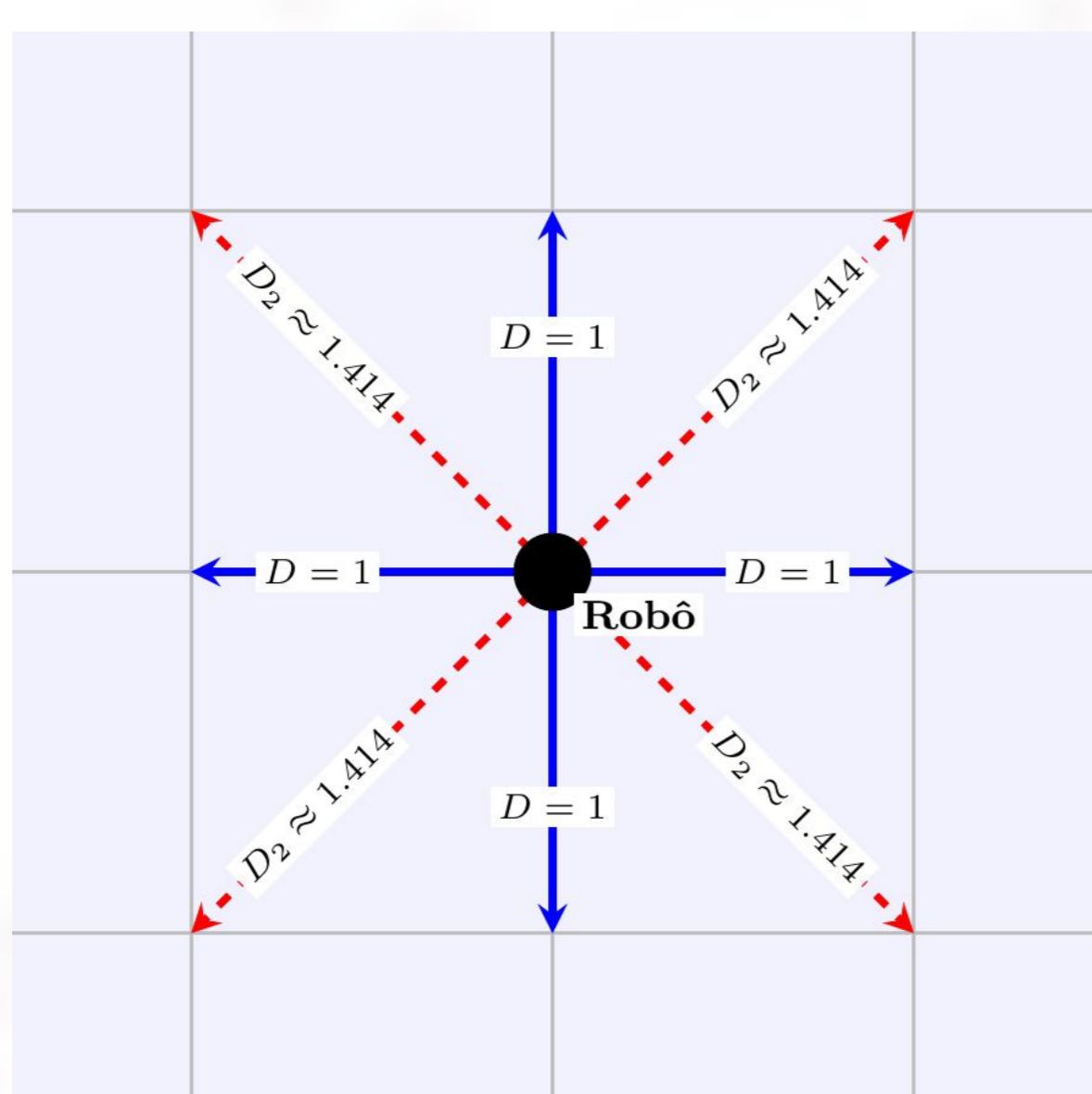


Figura 1. Representação geométrica da Distância Octil. (Fonte: Autores)

## Resultados

Em redes topológicas, o A\* supera o Dijkstra ao introduzir a heurística na busca (Figura 2). Enquanto o Dijkstra faz uma expansão radial cega, alocando todos os vértices vizinhos na memória, o A\* usa uma estimativa de custo para focar a exploração no eixo mais promissor. Essa direcionalidade reduz drasticamente os nós avaliados, tornando o A\* o padrão-ouro para rotas mínimas. Contudo, sua avaliação discreta (nó a nó) impõe severos gargalos de memória em malhas densas de ambientes logísticos.

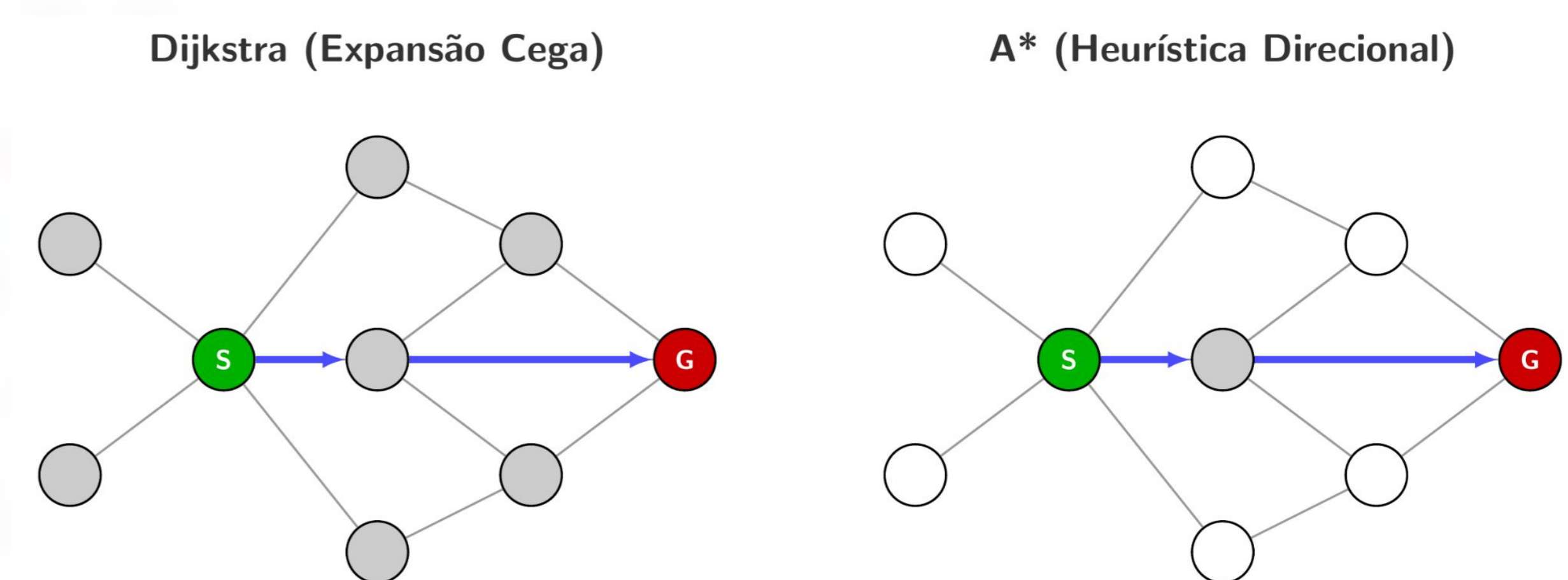


Figura 2. Comportamento de busca entre Dijkstra e A\* (Fonte: Autores)

Para contornar o esgotamento de memória do A\* em Grid Maps industriais, o REA\* aplica a poda geométrica, acelerando a busca em mais de uma ordem de grandeza (Figura 3). Em vez da expansão célula a célula, que gera trajetórias em zigue-zague, o algoritmo aglutina áreas vazias em grandes retângulos e usa as extremidades dos obstáculos como "portas". Essa conversão da busca discreta para a vetorial permite traçar rotas por linhas diretas entre os pontos de inflexão, reduzindo drasticamente o processamento e gerando caminhos físicos muito mais eficientes.

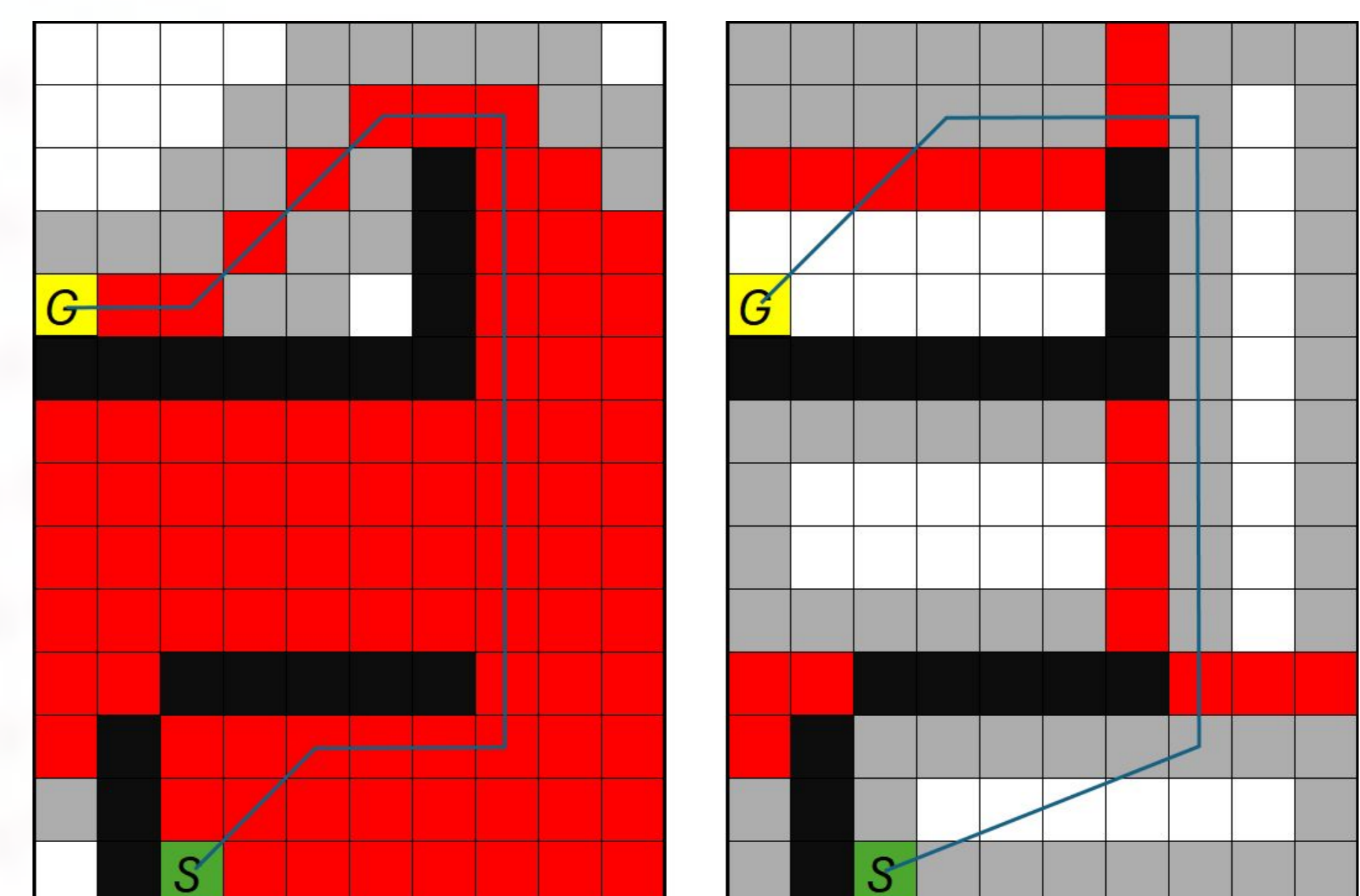


Figura 3. Comparação Visual entre A\* e REA\* (Fonte: Adaptado de ZHANG et al. 2016)

## Conclusões

A escolha do algoritmo ideal depende do cenário operacional. O Dijkstra oferece uma base robusta para encontrar caminhos mínimos em ambientes desconhecidos, mas com alto custo computacional. O A\* evoluiu essa abordagem ao introduzir heurísticas, direcionando a busca de forma eficiente e reduzindo os nós avaliados. Contudo, para o nosso contexto logístico de robótica industrial, o REA\* prova-se a solução mais robusta. Ao expandir áreas retangulares inteiras ao invés de processar o espaço nó a nó, o REA\* supera os gargalos do A\* em malhas densas. Conclui-se que o REA\* garante trajetórias altamente otimizadas, traduzindo essa eficiência algorítmica em redução direta de custos operacionais e máxima agilidade.

## Bibliografia

- CORMEN, T.H. et al. Introduction to Algorithms. 3. ed. [S.l.]: MIT Press, 2009.  
 RUSSEL, S.; NORVIG, P. Inteligência Artificial: Uma Abordagem Moderna. 3 ed. [S.l.]: Elsevier, 2013.  
 ZHANG, Jingwei; LI, Xiang.; BI, Danyan. Rectangle expansion A\* pathfinding for grid maps. Chinese Journal of Aeronautics, v. 29, n. 5, p. 1385-1396, 2016.