

Relatório de Análise

Felipe A. Giovanella, Gustavo H. Campestrini, Julia Welter, Nicolas A. Ceruti

Etapa 1 – Agente Reativo Simples

O agente reativo simples foi implementado com base em regras locais: ele se movimenta em uma direção até encontrar um obstáculo ou parede. Quando ocorre a colisão, a direção atual é registrada e o agente escolhe uma nova direção aleatória para continuar a exploração. Esse comportamento demonstra como agentes que não armazenam memória dependem apenas da percepção imediata do ambiente.

O ambiente utilizado foi um grid de tamanho $n \times n$, contendo obstáculos em posições aleatórias. Os resultados mostraram que o agente consegue explorar partes do ambiente, mas frequentemente revisita células já percorridas e pode acabar se movendo de forma redundante. Apesar disso, ele cumpre o objetivo inicial de detectar as bordas do espaço e adaptar seu movimento.

De maneira geral, a principal limitação observada foi a falta de memória, que leva a movimentos repetitivos e baixa eficiência. Como melhoria futura, seria possível incluir um registro das células já visitadas, tornando o agente capaz de planejar melhor seus deslocamentos.

Etapa 2 – Agente Baseado em Modelo

O agente baseado em modelo foi projetado para superar a limitação da falta de memória, implementando um algoritmo de exploração sistemática de Busca em Profundidade (DFS). Ele utiliza uma pilha para memorizar o caminho percorrido, avançando para vizinhos não visitados sempre que possível. Ao atingir um beco sem saída, o agente realiza um backtracking inteligente, desempilhando suas últimas posições para retornar ao ponto de decisão anterior e explorar rotas alternativas.

Nos experimentos, essa abordagem garantiu 100% de completude da exploração de todas as células acessíveis do grid. O comportamento do agente foi extremamente eficiente, eliminando movimentos aleatórios e reduzindo os passos redundantes ao mínimo necessário para o backtracking. A animação do caminho demonstrou claramente a natureza metódica da exploração, preenchendo todo o espaço de forma lógica.

A análise evidencia que a adição de um modelo de mundo mais sofisticado (o mapa de visitados junto à pilha de caminho) transforma o agente em um explorador altamente competente. Ele representa uma ponte clara para agentes de planejamento, embora suas decisões ainda sejam tomadas passo a passo.

Etapa 3 – Agente Baseado em Objetivos

O agente baseado em objetivos foi desenvolvido com a capacidade de planejar seu movimento até um destino específico. Para isso, foi utilizado o algoritmo A*, que calcula o caminho mais curto entre um ponto inicial e um ponto final, considerando apenas a distância de Manhattan como heurística. O agente segue o caminho encontrado de forma animada até alcançar o objetivo.

Nos testes realizados, o agente apresentou alto índice de sucesso, conseguindo encontrar o destino em diferentes cenários, desde que existisse um caminho acessível. O comprimento do caminho obtido foi consistente com a solução ótima, confirmando a eficácia do uso da heurística para o planejamento. Quando não havia rota possível devido a obstáculos, o algoritmo retornava falha de forma apropriada.

Esse agente mostra um salto em complexidade, pois não apenas reage ao ambiente, mas também planeja antecipadamente. Entretanto, ele não leva em conta custos diferenciados das células, tratando todo movimento como equivalente. Isso abre espaço para a evolução para um agente baseado em utilidade.

Etapa 4 – Agente Baseado em Utilidade

O agente baseado em utilidade amplia a lógica do agente baseado em objetivos, incorporando a noção de custo associado a cada célula do ambiente. Dessa forma, ele não busca apenas o caminho mais curto em termos de distância, mas sim a rota de menor custo total. Para explorar essa capacidade, o agente foi implementado em duas variações com desafios distintos.

Variação 1: Ambiente Completamente Observável

Na primeira variação, o agente possuía conhecimento prévio de todo o mapa de terrenos. Para encontrar a rota de menor custo total, foi implementado o **Algoritmo de Dijkstra**. Diferente do A* da etapa anterior, que usa uma heurística de distância, o Dijkstra prioriza unicamente o custo real acumulado desde o ponto de partida. Nos testes, o agente demonstrou sua capacidade de ignorar caminhos mais curtos que passavam por terrenos caros, optando por rotas mais longas, porém mais "baratas", provando a eficácia do algoritmo para problemas de otimização de custo.

Variação 2: Ambiente Parcialmente Observável

A segunda e mais avançada variação colocou o agente em um ambiente onde ele só podia "enxergar" uma área limitada ao seu redor. Isso o forçou a combinar exploração com planejamento, seguindo um ciclo contínuo de "perceber-planejar-agir". A cada passo, o agente escaneava a vizinhança, atualizava seu mapa interno e recalculava a melhor rota até o destino com base no

conhecimento parcial que possuía. Foi notável sua capacidade de se adaptar, recalculando o caminho ao descobrir "armadilhas" de alto custo que não eram visíveis no início. Este comportamento é o mais complexo entre os quatro agentes, simulando a tomada de decisão em cenários realistas onde a informação é incompleta.