

Relatório de Análise

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

Etapas 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.

Etapas 2 – Agente Baseado em Modelo

O agente baseado em modelo foi projetado para superar a limitação da falta de memória. Ele mantém um registro das posições já visitadas e, a cada passo, prioriza a exploração de vizinhos não visitados. Apenas quando não há novas células disponíveis, ele escolhe aleatoriamente entre vizinhos já conhecidos, o que gera passos redundantes.

Nos experimentos realizados no grid, o agente obteve maior cobertura do ambiente quando comparado ao reativo simples. O percentual de completude aumentou significativamente, e a contagem de passos redundantes permitiu avaliar sua eficiência. O agente também apresentou um comportamento mais sistemático, evitando revisitar continuamente as mesmas células.

A análise evidencia que a adição de memória melhora consideravelmente a exploração. Porém, ainda existe margem de melhoria: o agente não tem planejamento global, apenas registra localmente as visitas. Uma evolução natural seria integrar algoritmos de busca mais avançados, como BFS ou DFS, para garantir uma cobertura completa do ambiente.

Etapas 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, considerando terrenos ou obstáculos mais custosos.

Nos experimentos, o agente foi capaz de identificar caminhos alternativos que, embora mais longos em número de passos, apresentaram menor custo acumulado. Isso mostrou um comportamento mais “inteligente”, pois em ambientes reais muitas vezes não é a rota mais curta que é a melhor, mas a que oferece maior eficiência em termos de recursos.

A análise indica que esse agente é o mais avançado dos quatro, pois equilibra planejamento com avaliação de utilidade. Contudo, sua complexidade computacional também aumenta, já que a busca deve considerar pesos diferentes em cada movimento. Futuramente, poderiam ser explorados algoritmos de otimização mais sofisticados para melhorar ainda mais o desempenho em ambientes maiores.