



Sokoban

desenvolvimento de um agente autónomo

João Tiago Rainho 92984
Diogo Amaral 93228
José Vasco Sousa 93049

Projeto
Introdução à inteligência artificial
11 dezembro 2020

Descrição do algoritmo

O algoritmo usado neste projeto, utiliza parte da lógica associada ao algoritmo facultado nas aulas práticas. Ao mesmo, foram adicionados mecanismos de otimização relativo ao jogo do sokoban, explicados no próximo slide.

Técnica de pesquisa

Para a pesquisa do caminho, isto é, para a obtenção da solução é usado o algoritmo A^* , com uma heurística baseada numa distância ótima pré-calculada no início do algoritmo e ajustado quando existem caixas no caminho. Para que o algoritmo seja mais eficiente são calculado os deadlocks, tanto estáticos como dinâmicos (“area deadlocks”, “freeze deadlocks” e “corrals deadlocks”), também é feita a verificação de *macros* tal como túneis e por fim, também são eliminados estados repetidos. Com o objetivo de limitar a expansão da árvore é usado um limite com o valor de 800.

Resultados obtidos

Usando a técnica explicada anteriormente, correndo o programa no servidor do professor chegámos aos seguintes resultados:

O agente conseguiu chegar ao nível 146, com 4593 pushes e 29101 steps atingindo assim uma pontuação de 6811599. A maior parte dos níveis, o nosso agente consegue encontrar a solução em menos de 1 segundos. Dos níveis feitos, o que tem pior tempo é o nível 131 com 210.987 segundos e o segundo pior é o nível 141 com 161.21 segundos.

Conclusão

Para melhorar o algoritmo desenvolvido poderia-se implementar uma versão do algoritmo de pesquisa BFS e usar para indexar toda a informação necessária (posição de caixas, posição do agente, etc) dicionários em vez de listas.

Nós ainda chegámos a implementar esta técnica porém tinha prós e contras. Fazia o menor número de passos possível mas demorava mais tempo em certos níveis (por exemplo: 256.627 segundos no nível 131). Com esta solução, muito provavelmente, no server do professor não ia passar tantos níveis então decidimos optar pela versão descrita anteriormente.

É de notar que os corral deadlocks estão comentados na entrega final pois não estão otimizados ao ponto de serem eficientes numa visão geral do algoritmo.