Resolvendo Otimamente o Sokoban com Poda Dependente de Domínio

Renato Leme, André Grahl, Luciana Buriol, & Marcus Ritt

UFRGS Universidade Federal do Rio Grande do Sul

INF Instituto de Informática UFRGS

Apresentação

Sokoban é um jogo singleplayer que, embora simples nas regras, pode ser extremamente desafiador. Se comparado com outros *puzzles* de natureza semelhante (como *Cubo Mágico* ou *24-puzzle*), apresenta características que o tornam mais difícil de resolver, tais como maior fator de ramificação, comprimento de solução e espaço de estados.

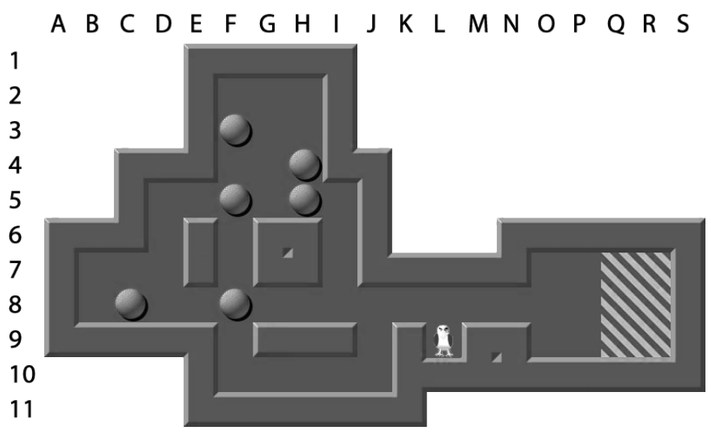

Figura 1. Instância #1 do conjunto padrão de instâncias do Sokoban.

Características

- O agente pode mover uma pedra por vez para um quadrado vazio adjacente
- A solução consiste em um conjunto de ações que levam todas as pedras para as posições finais
- O objetivo do solver é encontrar uma solução com o menor número de movimentos de caixa

Motivação

Um algoritmo de solução gera uma árvore de possibilidades, considerando os movimentos permitidos a cada uma das pedras pertencentes à instância. O nodo a ser explorado é escolhido considerando-se uma aproximação dada por $f(s) = g(s) + h(s)$, em que $g(s)$ é o custo acumulado até o nodo analisado e $h(s)$ uma aproximação heurística.

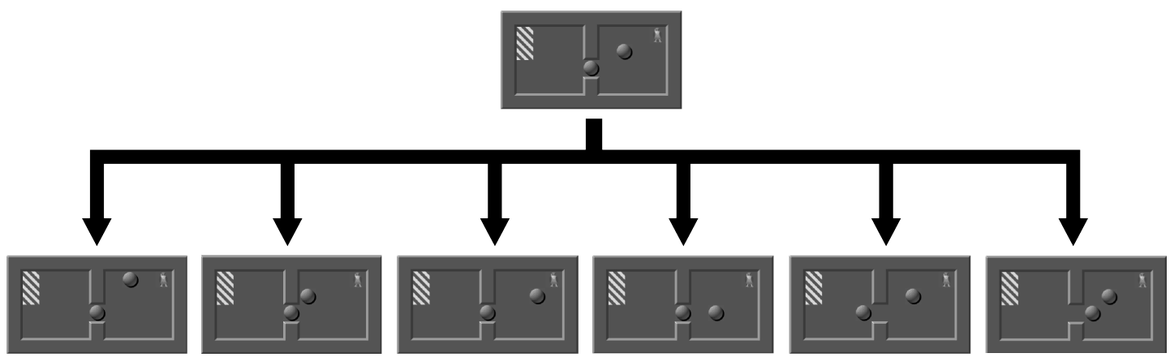

Figura 1. Instância #1 do conjunto padrão de instâncias do Sokoban.

Uma forma de reduzir o número de sucessores

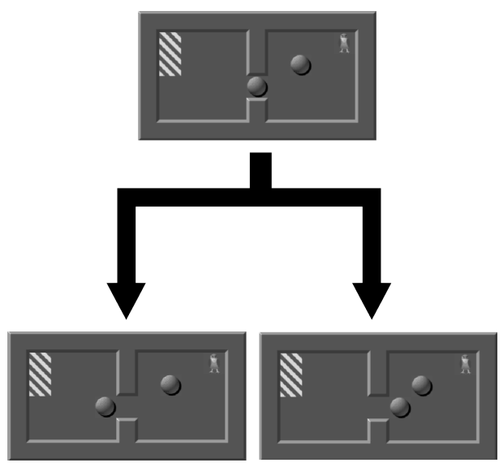

Figura 3. Diminuição na geração de nodos causada pelo uso de PI-Corral.

Passos do PI-Corral

1. Procura por uma zona não alcançável do mapa
2. Verifica as pedras que são fronteiras dessa zona
3. Se todas as pedras atenderem às condições, então explora somente elas
4. Senão, se tiver visto todas as zonas não alcançáveis, então explora normalmente todas as pedras
5. Senão, volta ao passo 1.

PI-Corral

Um PI-Corral é um conjunto $P = s0, s1, s2, ..., s(n1)$, tal que cada uma das n pedras devem, necessariamente, atender às seguintes condições:

1. Pode ser movida, considerando a posição atual do jogador.
2. Pode ser movida somente para dentro da região não-alcançável da qual é fronteira.

O algoritmo abaixo é um pseudo-código para a busca de PI-Corrals no mapa. Se o conjunto PC retornado for diferente de vazio, então apenas as pedras pertencentes a esse conjunto serão exploradas.

input : Uma instancia *inst* do Sokoban
output: Um vetor de PI-Corrals

```
C ← encontraCorrals(inst);
PC ← ∅;
i ← 0;
while C ≠ ∅ do
    F ← encontraFronteira(C[0]);
    remove primeiro elemento de C;
    if verificaCondicoes(F) then
        PC[i] ← F;
        i ← i + 1;
    end
end
return PC;
```

Algorithm 1: Algoritmo para encontrar PI-Corrals

Resultados

Abaixo, breve descrição dos resultados obtidos. Foram testadas as instâncias {*1, 2, 3, 4, 5, 6, 7, 9, 17, 38, 43, 53, 65, 73, 78, 79, 80, 81, 82, 83*} . O símbolo '>' indica que a busca pela solução ultrapassou o limite de 1 milhão de nodos explorados.

#	Com PI-Corral	Sem PI-Corral
1	**149**	160
2	**117489**	161834
3	**420**	20839
4	**1582**	1590
5	>	>
6	>	>
7	>	>
9	>	>
17	**125515**	>
38	183171	93423
43	>	>
53	**1044**	1071
65	>	>
73	>	>
78	**334**	8544
79	>	>
80	**768**	27708
81	>	>
82	>	>
83	706	559

Embora em alguns casos a redução de estados explorados não tenha sido grande, e em alguns outros, utilizando-se PI-Corral, o resultado tenha ficado inferior ao obtido sem o uso do método, em alguns casos o ganho foi considerável, como no caso das instâncias #3, #17 e #80. Assim, vemos no PI-Corral uma boa alternativa de reduzir o número de nodos explorados durante uma busca por solução do Sokoban.