



Evento	Salão UFRGS 2015: SIC - XXVII SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
Ano	2015
Local	Porto Alegre - RS
Título	Obtendo Soluções Ótimas para o Sokoban com Poda Dependente de Domínio
Autor	RENATO REIS LEME
Orientador	LUCIANA SALETE BURIOL

Título: Obtendo Soluções Ótimas para o Sokoban com Poda Dependente de Domínio

Autor: Renato Reis Leme

Orientador: Luciana Salete Buriol

Universidade Federal do Rio Grande do Sul

Algoritmos de busca como A* e IDA* podem ser aprimorados através do uso de técnicas dependentes de domínio como podas da árvore de busca. Entretanto, algumas vezes, a aplicação de semelhante técnica pode afetar a garantia de otimalidade da solução encontrada. Em nosso trabalho, propomos o Domain Dependent Move Pruning (DDMP) uma técnica de poda para o Sokoban. O DDMP analisa cada um dos nodos e seleciona o subconjunto de nodos sucessores a ser explorados que preservam todas as soluções ótimas.

O Sokoban é um puzzle bidimensional de único agente que pode ser descrito como um labirinto em uma grade de quadrados. Cada um desses quadrados pode ser um quadrado livre ou estar ocupado por um quadrado fixo (parede), um quadrado móvel (pedra). Disposto ao longo de cada mapa do Sokoban existe um conjunto de k pedras e k quadrados objetivos. Uma pedra pode ser movimentada individualmente através do agente controlado pelo jogador, desde que o quadrado destino esteja livre. O puzzle termina quando todos os quadrados objetivos estiverem ocupados por pedras. O Sokoban apresenta características que o tornam mais difícil de resolver quando comparado com outros problemas de agente único como Cubo Mágico ou 24-puzzle, como deadlocks, maior fator de ramificação, comprimento de solução e espaço de estados.

O DDMP é uma técnica de domínio específico derivada do PI-corrал, uma técnica conhecida na comunidade do Sokoban. Difere deste por acrescentar uma restrição que garante a não-interferência na otimalidade da solução.

Um DDMP é um conjunto de pedras que cerca uma área não alcançável do mapa a partir da posição atual do jogador. Essas pedras precisam atender a três condições: devem ser acessíveis, isto é, o jogador deve poder acessá-las a partir da sua posição atual; devem poder ser empurradas somente para dentro de uma zona imediatamente não acessível pelo jogador; e os movimentos que estas pedras podem fazer deve ser o mesmo que elas poderiam fazer caso as pedras fora da região que elas cercam fossem removidas. Quando detectamos um caso de DDMP, apenas as pedras que o compõe são empurradas. Assim, são gerados sucessores apenas para os movimentos possíveis a essas pedras. Dessa forma, o fator de ramificação da árvore de solução é de um modo geral reduzido, e casos de deadlock são forçados.

Ao longo do nosso trabalho, primeiramente, procuramos encontrar uma versão que se baseasse na intuição do PI-corrал mas que garantisse a otimalidade. Uma vez encontrada, formalizamos o conceito. Essa formalização foi importante para que pudéssemos desenvolver o algoritmo para detecção de DDMP e provar a sua otimalidade, que foi o segundo passo. Após, executamos sucessivos testes sobre o conjunto padrão de instâncias do Sokoban.

Para os testes, limitamos o número de nodos explorados a 5 milhões, buscando a solução para os 90 estados do conjunto padrão do Sokoban, analisando: o Branching Factor geral, o f-value (máximo valor heurístico encontrado durante busca), o número de deadlocks detectados com e sem DDMP em um conjunto de 1.000 estados aleatórios, e finalmente o impacto no tempo e no número de nodos gerados. Nestes testes, pudemos notar os ganhos com o DDMP, ao solucionar uma instância a mais, reduzir a média do Branching Factor em 0,88 e aumentar a média do f-value em 0,3, e, por fim, ao aumentar a média de deadlocks encontrados para a profundidade de busca 2, 4 e 8.