



Evento	Salão UFRGS 2014: SIC - XXVI SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
Ano	2014
Local	Porto Alegre
Título	Obtendo Soluções Ótimas para o Sokoban com Poda Dependente de Domínio
Autor	RENATO REIS LEME
Orientador	LUCIANA SALETE BURIOL

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, exploramos o uso de uma técnica de domínio específico para Sokoban chamada de PI-Corral, procurando defini-lo formalmente e elaborando um algoritmo capaz de detectá-lo, e vimos, através de resultados empíricos, que o PI-Corral reduz significativamente o número de nodos explorados e não afeta a otimalidade da solução.

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 quadrado destino esteja desocupado. O puzzle termina quando todos os quadrados objetivos estiverem ocupados por pedras. Sokoban apresenta características que o tornam mais difícil de resolver quando comparado com outros problemas de agente único como Cubo Mágico ou 24-*puzzle*. Sokoban apresenta maior fator de ramificação, comprimento de solução e espaço de estados.

O PI-Corral é uma técnica de domínio específico criada por Matthias Meger e aprimorada posteriormente através de discussões entre entusiastas do Sokoban. Mostrou-se através de resultados empíricos que de fato o PI-Corral proporciona uma redução significativa no número de nodos gerados durante a busca por soluções.

Um PI-Corral 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 duas condições: devem ser acessíveis, isto é, o jogador deve poder acessá-las a partir da sua posição atual, e elas devem poder ser empurradas somente para dentro de uma zona imediatamente não acessível pelo jogador.

Assumindo que PI-Corral garante a otimalidade da solução, quando o nosso algoritmo detecta um caso de PI-Corral, apenas as pedras que o compõe são empurradas. Assim, são gerados sucessores apenas para os movimentos possíveis a essas pedras. E, embora isso possa levar a tornar o algoritmo não completo, isto é, fazer com que ele não retorne uma solução, o fator de ramificação da árvore de solução é de um modo geral reduzido quando acontece de se detectar um caso de PI-Corral durante a busca pela solução. Além disso, movimentando apenas as pedras que pertencem a um PI-Corral, aumenta-se a chance de se detectar *deadlocks*, o que auxilia no processo de busca por solução.

Ao longo do nosso trabalho, primeiramente, procuramos formalizar o conceito de PI-Corral e todos os conceitos que ele pressupõe. Essa formalização foi importante para que pudéssemos desenvolver o algoritmo para detecção de PI-Corral, que foi o segundo passo. Após, executamos sucessivos testes sobre o conjunto padrão de instâncias do Sokoban para que pudéssemos ter a ideia da relevância no desempenho da busca pela solução do puzzle.

Para os testes, limitamos o número de nodos explorados a um milhão, buscando a solução em um conjunto de 20 estados do conjunto padrão do Sokoban e somando apenas o número de nodos das soluções que foram encontradas tanto sem PI-Corral quanto com PI-Corral. Desse modo, obtemos, sem o uso de PI-Corral, um total de 294.889 nodos explorado. Já com PI-Corral, este número caiu para 248.928, obtendo, assim, uma redução total de cerca de 15,5% do número de nodos explorados. Além disso, com PI-Corral e levando em consideração a limitação de um milhão de nodos, nosso algoritmo foi capaz de encontrar a solução para duas instâncias (#3 e #17) que sem o PI-Corral não foi possível.