Solving Motion Planning Problems

André Grahl Pereira, Marcus Ritt, and Luciana S. Buriol

Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
{agpereira,mrpritt,buriol}@inf.ufrgs.br

This work considers a family of motion planning problems with movable blocks. Such problem is defined by a maze grid occupied by immovable blocks (*walls*) and free squares. There are k movable blocks (*stones*) and k fixed goal squares. The *man* is a movable block that can traverse free squares and move stones between them. The problem goal is to move the stones from their initial positions to the goal squares with the minimum number of stone moves.

Moving blocks is a simplified model of general robot motion planning which is a fundamental problem in robotics [3]. Sokoban is a challenging single-agent search problem and the best known problem in this class. In Sokoban the man can push one stone at a time. It is one of the remaining puzzles which humans solve better than computers. Sokoban is harder to solve than other well-known single-agent search problems, e.g. Rubik's cube or the 24-puzzle, due to its larger branching factor, greater solution length and larger search space size [5]. The presence of deadlocks and real world problem characteristics also contribute to the difficulty of solving it. Pukoban is another well known motion planning problem in which the man can push or pull one stone at a time. It has similar characteristics as Sokoban, but in general, leads to a greater branching factor. Moreover, the absence of deadlocks can open new research possibilities.

Due to difficulty of solving Sokoban most solvers use non-admissible techniques, returning a solution that is not necessarily the optimal. Up to our results, using only admissible techniques, Rolling Stone solver [5] presented the best results for Sokoban. It uses multiple domain-independent and -dependent enhancements, and solves six instances from the standard set of 90 instances, limited to 20 million expanded nodes.

While no complexity results are known for Pukoban, Sokoban was shown to be PSPACE-complete [1]. Variants of Sokoban in which the man can push k stones or an unbounded number of stones at a time, or the moved stone slide until hitting an immovable block, have been shown to be PSPACE-complete or NP-hard. When only pull moves are allowed the problem was shown to be NP-Hard [6]. When pull and push moves were allowed, the problem was proved to be NP-hard [3] for the case in which the man can move at least five stones at a time.

Pattern databases (PDBs) [2] were able to improve the state of the art of several single-agent search problems. In order to effectively apply PDBs for a problem the goal state must be explicitly defined. However, in movable blocks the goal state is defined implicitly, since we do not know without solving the problem the destination goal square of each stone. We propose a method that solves this problem decomposing an instance problem into two subproblems and introducing an explicit defined intermediate goal state for the harder subproblem. In this way, PDBs can be applied to the harder subproblem, while a domain specific lower bound is computed for the other subproblem. For movable blocks the tie breaking rule used for the search has significant impact on the algorithm effectiveness. We use domain-dependent knowledge to propose a tie breaking rule that uses the fact that the goal squares cannot be filled in an arbitrary order. We define a fill order which respects the restrictions of the instance that depends on the placement of the goal squares in the maze. Then, we use this fill order to tie break nodes during the search. Applying both techniques to Sokoban we are able to solve with optimality guaranties more than four times more instances than the best previous results, provided by Rolling Stone solver, reducing by two orders of magnitude the total computational effort.

From the problem complexity side, the nondeterministic constraint logic model [4] provides a set of tools to construct PSPACE-hardness proofs for motion planning problems. Using this model we prove that several variants of movable blocks with pull moves are PSPACE-complete strengthen previous results. We also use this model to investigate the hardness of Pukoban and some of its variants with push and pull moves. So far, we are able to prove that all variants in which the man can move at least three stones at a time are PSPACE-complete, again strengthen previous results.

As future work, we plan to improve the proposed method further. Moreover, we plan to apply similar techniques to Pukoban, using domain specific techniques for exploring the absence of deadlocks. In addition, we plan to explore the idea of decomposing instances for applying PDBs in other problems with implicit defined goal states. Finally, we will continue investigating the hardness results for movable blocks with push and pull moves, as well as investigating a possible PSPACE-hardness prove for Pukoban.

References

1. Culberson, J.C.: Sokoban is PSPACE-Complete. In: Fun With Algorithms. pp. 65–76 (1999)
2. Culberson, J.C., Schaeffer, J.: Searching with pattern databases. In: Canadian Conference on Artificial Intelligence. pp. 402–416 (1996)
3. Dor, D., Zwick, U.: Sokoban and other motion planning problems. Computational Geometry 13(4), 215–228 (1999)
4. Hearn, R., Demaine, E.: PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. Theoretical Computer Science 343(1-2), 72–96 (2005)
5. Junghanns, A., Schaeffer, J.: Sokoban: Enhancing general single-agent search methods using domain knowledge. Artificial Intelligence 129(1–2), 219–251 (2001)
6. Ritt, M.: Motion planning with pull moves. The Computing Research Repository pp. 1–9 (2010)