ComplementaryPDB Planner

Santiago Franco[1], Stefan Edelkamp[2], Ionut Moraru[3]

[1] Department of Computer Science, Royal Holloway University of London, Egham, UK
[2] Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, CZ
[3] School of Computer Science, University of Lincoln, Lincoln, UK
[1]Santiago.francoaixela@rhul.ac.uk, [2]stefan.edelkamp@fel.cvut.cz, [3]imoraru@lincoln.ac.uk

Abstract

ComplementaryPDB is a planner that uses heuristic search via Symbolic Pattern Databases (PDBs) and uses a greedy pattern selection algorithm – Partial Gamer – combined with pattern collections from bin-packing pattern selection algorithms. For more information on this method, we direct the reader to the main paper describing this method [Moraru et al. 2019].

Introduction

The automated generation of search heuristics is one of the holy grails in AI, and goes back to early work of Gaschnik [Gaschnig 1979a], Pearl [Pearl 1985], and Prieditis [Preditis 1993b]. In most cases, lower bound heuristics are problem relaxations: each plan in the original state space maps to a shorter one in some corresponding abstract one. In the worst case, searching the abstract state spaces at every given search nodes exceeds the time of blindly searching the concrete search space [Valtorta 1984a].

With pattern database heuristic (PDBs), all efforts in searching the abstract state space are spent prior to the plan search, so that these computations amortize through multiple lookups. The ComplementaryPDB planner bases its search on the PDB heuristic, combining the work from [Franco et al. 2017a] and [Moraru et al. 2019]. It is an evolution of the Complementary planners submitted at the 9th International Planner Competition.

In this planner abstract, we will briefly describe Pattern Databases. For more information on this method, we direct the reader to the main paper describing this method [Moraru et al. 2019]. The results, evaluation and discussion sections will be added after the full results of the competition will be made public.

Pattern Databases

Initial results of Culberson and Schaeffer [Culberson and Schaeffer 1998a] in sliding-tile puzzles, where the concept of a pattern is a selection of tiles, quickly carried over to a number of combinatorial search domains, and helped to optimally solve random instances of the Rubik's cube, with non-pattern labels being removed [Korf 1997]. When shifting from breadth-first to shortest-path search, the exploration of the abstract state-space can be extended to include action costs.

The combination of several databases into one, however, is tricky [Haslum et al. 2007a]. While the maximum of two PDBs always yields a lower bound, the sum usually does not. Korf and Felner [Korf and Felner 2002] showed that with a certain selection of disjoint (or additive) patterns, the values in different PDBs can be added while preserving admissibility. Holte et al. [Holte et al. 2004a] indicated that several smaller PDBs may outperform one large PDB. The notion of a pattern has been generalized to production systems in vector notation [Holte and Hernádvölgyi 1999], while the automated pattern selection process for the construction of PDBs goes back to the work of Edelkamp [Edelkamp 2006].

Many planning problems can be translated into state spaces of finite domain variables [Helmert 2004], where a selection of variables (pattern) influences both states and operators. For disjoint patterns, an operator must distribute its original cost, if present in several abstractions [Katz and Domshlak 2008; Yang et al. 2008].

During the PDB construction process, the memory demands of the abstract state space sizes may exceed the available resources. To handle large memory requirements, symbolic PDBs succinctly represent state sets as binary decision diagrams [Edelkamp 2002a]. However, there are an exponential number of patterns, not counting alternative abstraction and cost partitioning methods. Hence, the automated construction of informative PDB heuristics remains a combinatorial challenge. Hill-climbing strategies have been proposed [Haslum et al. 2007a], as well as more general optimization schemes such as genetic algorithms [Edelkamp 2006; Franco et al. 2017a]. The biggest area of research in this area remains the quality evaluation of a PDB (in terms of the heuristic values for the concrete state space) which can only be estimated. Usually, this involves generating the PDBs and evaluating them [Edelkamp 2014; Korf 1997].

Results

Discussion

Acknowledgments

We would like to thank all contributors to these the Fast-Downward planner, as ComplementaryPDB is built on top of it.

References

Culberson, J. C.; and Schaeffer, J. 1998a. Pattern Databases. *Computational Intelligence*, 14(4): 318–334.

Culberson, J. C.; and Schaeffer, J. 1998b. Pattern Databases. *Computational Intelligence*, 14(4): 318–334.

Derval, G.; Regin, J.-C.; and Schaus, P. 2017. Improved Filtering for the Bin-Packing with Cardinality Constraint. In *CP*.

Dósa, G. 2007. The tight bound of first fit decreasing bin-packing algorithm is FFD (I) 11/9OPT (I)+ 6/9. In *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, 1–11. Springer.

Edelkamp, S. 2001. Planning with pattern databases. In *ECP*.

Edelkamp, S. 2002a. Symbolic Pattern Databases in Heuristic Search Planning. In *AIPS*, 274–283.

Edelkamp, S. 2002b. Symbolic Pattern Databases in Heuristic Search Planning. In *AIPS*, 274–293.

Edelkamp, S. 2006. Automated creation of pattern database search heuristics. In *International Workshop on Model Checking and Artificial Intelligence*, 35–50. Springer.

Edelkamp, S. 2007. Automated Creation of Pattern Database Search Heuristics. In *MOCHART*, 36–51.

Edelkamp, S. 2014. Planning with pattern databases. In *Sixth European Conference on Planning*.

Franco, S.; Torralba, A.; Lelis, L. H.; and Barley, M. 2017a. On creating complementary pattern databases. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 4302–4309. AAAI Press.

Franco, S.; Torralba, Á.; Lelis, L. H. S.; and Barley, M. 2017b. On Creating Complementary Pattern Databases. In *IJCAI*, 4302–4309.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractibility, A Guide to the Theory of NP-Completeness*. Freeman & Company.

Gaschnig, J. 1979a. A Problem Similarity Approach to Devising Heuristics: First Results. In *IJCAI*, 434–441.

Gaschnig, J. 1979b. A Problem Similarity Approach to Devising Heuristics: First Results. In *IJCAI*, 434–441.

Hart, N.; Nilsson, J.; and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on System Science and Cybernetics*, 4(2): 100–107.

Haslum, P.; Bonet, B.; and Geffner, H. 2005. New admissible heuristics for domain-independent planning. In *AAAI*, volume 5, 9–13.

Haslum, P.; Botea, A.; Helmert, M.; Bonet, B.; and Koenig, S. 2007a. Domain-Independent Construction of Pattern Database Heuristics for Cost-Optimal Planning. In *AAAI*, 1007–1012.

Haslum, P.; Botea, A.; Helmert, M.; Bonet, B.; and Koenig, S. 2007b. Domain-Independent Construction of Pattern Database Heuristics for Cost-Optimal Planning. In *AAAI*, 1007–1012.

Helmert, M. 2004. A Planning Heuristic Based on Causal Graph Analysis. In *ICAPS*, 161–170.

Helmert, M. 2006. The Fast Downward Planning System. *J. Artif. Intell. Res.*, 26: 191–246.

Holland, J. 1975. *Adaption in Natural and Artificial Systems*. Ph.D. thesis, University of Michigan.

Holte, R.; Newton, J.; Felner, A.; Meshulam, R.; and Furcy, D. 2004a. Multiple Pattern Databases. In *ICAPS*, 122–131.

Holte, R.; Newton, J.; Felner, A.; Meshulam, R.; and Furcy, D. 2004b. Multiple Pattern Databases. In *ICAPS*, 122–131.

Holte, R. C.; Grajkowski, J.; and Tanner, B. 2005. Hierarchical Heuristic Search Revisited. In *SARA*, 121–133.

Holte, R. C.; and Hernádvölgyi, I. T. 1999. A space-time tradeoff for memory-based heuristics. In *AAAI/IAAI*, 704–709. Citeseer.

Holte, R. C.; and Hernádvölgyi, I. T. 2001. A Space-Time Tradeoff for Memory-Based Heuristics.

I. Moraru, M. M., S. Edelkamp. 2018. Automated Pattern Selection using MiniZinc. In *CP-Workshop on Constraints AI Planning*.

Karpas, E.; Katz, M.; and Markovitch, S. 2011. When Optimal Is Just Not Good Enough: Learning Fast Informative Action Cost Partitionings. In *ICAPS*.

Katz, M.; and Domshlak, C. 2008. Optimal Additive Composition of Abstraction-based Admissible Heuristics. In *ICAPS*, 174–181.

Korf, R. E. 1997. Finding Optimal Solutions to Rubik's Cube Using Pattern Databases. In *AAAI*, 700–705.

Korf, R. E. 2002. A new algorithm for optimal bin packing. In *AAAI*, 731–736.

Korf, R. E. 2003. An improved algorithm for optimal bin packing. In *IJCAI*, 1252–1258.

Korf, R. E.; and Felner, A. 2002. *Chips Challenging Champions: Games, Computers and Artificial Intelligence*, chapter Disjoint Pattern Database Heuristics, 13–26. Elsevier.

Lelis, L. H. S.; Franco, S.; Abisrror, M.; Barley, M.; Zilles, S.; and Holte, R. C. 2016. Heuristic Subset Selection in Classical Planning. In *IJCAI*, 3185–3191.

Moraru, I.; Edelkamp, S.; Franco, S.; and Martinez, M. 2019. Simplifying automated pattern selection for planning with symbolic pattern databases. In *KI 2019: Advances in Artificial Intelligence: 42nd German Conference on AI, Kassel, Germany, September 23–26, 2019, Proceedings 42*, 249–263. Springer.

Pearl, J. 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley.

Pearl, J. 1985. *Heuristics*. Addison-Wesley.

Pommerening, F.; Helmert, M.; and Bonet, B. 2017. Higher-Dimensional Potential Heuristics for Optimal Classical Planning.

Preditis, A. 1993a. Machine Discovery of Admissible Heurisitics. *Machine Learning*, 12: 117–142.

Preditis, A. 1993b. Machine Discovery of Admissible Heuristics. *Machine Learning*, 12: 117–142.

Valtorta, M. 1984a. A result on the computational complexity of heuristic estimates for the A* algorithm. *Information Sciences*, 34: 48–59.

Valtorta, M. 1984b. A result on the computational complexity of heuristic estimates for the A* algorithm. *Information Sciences*, 34: 48–59.

Yang, F.; Culberson, J.; Holte, R.; Zahavi, U.; and Felner, A. 2008. A general theory of additive state space abstractions. *Journal of Artificial Intelligence Research*, 32: 631–662.