MAPlan: Reductions with Fact-Alternating Mutex Groups and h^m Heuristics

Daniel Fišer and **Anonín Komenda**
Department of Computer Science, Faculty of Electrical Engineering,
Czech Technical University in Prague, Czech Republic
danfis@danfis.cz, antonin.komenda@fel.cvut.cz

Introduction

MAPlan (Fišer, Štolba, and Komenda 2015) is, originally, a multi-agent planner that we adapted for the deterministic optimal track of the International Planning Competition (IPC) 2018. The planner uses state-based heuristic search with a translator from PDDL to STRIPS (Fikes and Nilsson 1971) and then to the finite domain representation (FDR, or SAS^+) (Bäckström and Nebel 1995). The translator tries to reduce the STRIPS problem (i.e., to remove spurious facts and operators) using mutexes found by h^2 and h^3 heuristics (Haslum and Geffner 2000; Haslum 2009) in both progression and regression, and by removing dead-end operators (operators that cannot be part of any plan) detected using fact-alternating mutex groups (fam-groups) (Fišer and Komenda 2018). The inferred fam-groups are also useful in the disambiguation process (Alcázar et al. 2013) that is essential for the h^m heuristics, especially in regression. The fam-groups can also provide more concise encodings of the problems in FDR than the most commonly used translator from Fast Downward (Helmert 2006), although its impact on the performance is very limited (Fišer and Komenda 2018).

For IPC 2018, we prepared two configurations of the MAPlan planner. Both of the configurations use the aforementioned reductions and the $A^\star$ search algorithm. They differ only in the heuristics. The first one (`maplan-1`) uses the LM-Cut heuristic (Helmert and Domshlak 2009). The second configuration (`maplan-2`) uses a simplified abstraction heuristic based on merging a certain subset of the inferred fam-groups, similarly to what is done in the merge and shrink heuristic (Helmert, Haslum, and Hoffmann 2007) and pattern databases (Culberson and Schaeffer 1998; Edelkamp 2001). This heuristic is just a preliminary work testing how big merges can fit into the memory and how many overlapping fam-groups can these merges cover.

In the following sections we briefly introduce all the methods we used.

Fact-Alternating Mutex Groups

A mutex group is a set of facts out of which maximally one can be true in any reachable state. Mutex groups are invariants that are primarily used in the translation from STRIPS to FDR for the creation of FDR variables. The inference of mutex groups is in general PSPACE-Complete. The fact-alternating mutex groups (fam-groups) were first introduced by Fišer and Komenda (2018) as a subclass of mutex groups of which inference is NP-Complete and they described an algorithm based on the integer linear programming that is complete with respect to maximal fam-groups. This algorithm is implemented in the MAPlan planner using CPLEX solver.

Fam-group of a certain form can be used for a detection of operators that can produce only dead-end states, i.e., states from which it is impossible to reach any goal state. Such operators can be safely removed from the planning task, because these operators cannot be part of any plan. This is one of the methods we use for a reduction of the input planning problem.

Another useful application of fam-groups is in the disambiguation of operators' preconditions and the goal specification. Disambiguation is a simple process that extends a set of facts with the fact that is the only possibility given a set of facts out of which exactly one is a part of every state. For example, consider an operator's precondition $\{f_1, f_2\}$ and let us assume that every state must contain one of the facts $\{f_3, f_4, f_5\}$. If we know that there are no reachable states that contain either $\{f_1, f_3\}$ or $\{f_1, f_4\}$, then we can safely extend the precondition with f_5, because it is the only possibility. A certain subset of fam-groups can be easily identified as mutex groups that has not maximally one, but exactly one fact in every reachable state. These fam-groups together with the h^m mutexes, described in the next section, are used for the disambiguation which in turn improves a pruning of operators and facts.

h^2 and h^3 Mutexes

A generalization of the $h^{\max}$ heuristic to a family of h^m heuristics (Haslum and Geffner 2000; Haslum 2009) offers a method for the generation of mutex invariants that can be used for reductions of the planning problems (Alcázar and Torralba 2015). $h^{\max}$ is a widely known and a well understood admissible heuristic for STRIPS planning. The heuristic value is computed on a relaxed reachability graph as a cost of the most costly fact from a conjunction of reachable facts. The heuristic works with single facts, but it can be generalized to consider a conjunction of at most m facts instead.

h^1 would then be equal to $h^{\max}$, h^2 would build the reachability graph with single facts and pairs of facts, etc. Unfortunately, the cost of the computation increases exponentially in m, which is why, usually, only h^1 and h^2 variants are used.

The pruning of operators and facts proposed by Alcázar and Torralba (2015) uses h^2 heuristic combined with the disambiguation process as a reachability analysis that can prove that certain operators and facts can be safely removed from the problem. The algorithm runs, in turn, in progression and in regression (on the dual planning problem (Massey 1999; Pettersson 2005; Suda 2013)) removing the unreachable facts and operators in each cycle until a fixpoint is reached. The h^2 heuristic in the algorithm can be easily switched to any heuristic from the h^m family, but with a considerably increased computational time. In the MAPlan planner, we use the pruning with h^2 and fam-groups used for the disambiguation, and we also added the variant with h^3, but we restricted the running time of a single cycle to one minute. That is, if the running time of the pruning using h^3 in any direction exceeds the limit of one minute, the pruning is prematurely terminated and it is not used anymore for that problem. This is a rather weak restriction that still could cause that the whole time limit will be consumed just by computing h^3, but our tests on the domains from IPC 2011 and 2014 showed that this simple rule should be sufficient to disable this type of pruning for large problems.

Merge without Shrink

The heuristic function used in the second configuration of the submitted MAPlan planner (`maplan-2`) is built on the inferred set of maximal fam-groups. It uses a similar approach as is used in pattern databases (PDB) (Culberson and Schaeffer 1998; Edelkamp 2001), but instead of computing the projections on the FDR variables, we use projections on the fam-groups that are usually overlapping in a sense that they contain a common subset of facts.

We start with the complete set of maximal fam-groups and we merge (i.e., compute a synchronized product from the corresponding projected transition systems) as many fam-groups as possible to fit into memory. We use a 7 GB memory block in which we are, usually, able to fit around 20 000 abstract states. From the resulting merge we save the abstract states along with the cost of the cheapest path to a goal state as a heuristic estimate in the same way as is done in PDBs. This process is repeated so that each maximal fam-group is contained in at least one big merge. The selection of the fam-groups used for merging is controlled by a greedy rule that prefers fam-groups that share the most facts. The resulting heuristic estimate is the maximum of the estimates over all merges.

We realize that we could, probably, get much better estimates if we adapted our approach to the framework of either merge and shrink heuristics or PDBs, but this work was not finished in time for IPC. However, we believe that using mutex groups directly instead of FDR variables (which can be considered as derivatives of mutex groups) can lead to better merge and shrink strategies and to improvements in PDB heuristics, which is a focus of our future research.

Acknowledgements

This research was supported by the Czech Science Foundation (grant no. 18-24965Y). Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures".

References

Alcázar, V., and Torralba, Á. 2015. A reminder about the importance of computing and exploiting invariants in planning. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS)*, 2–6.

Alcázar, V.; Borrajo, D.; Fernández, S.; and Fuentetaja, R. 2013. Revisiting regression in planning. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2254–2260.

Bäckström, C., and Nebel, B. 1995. Complexity results for SAS+ planning. *Computational Intelligence* 11:625–656.

Culberson, J. C., and Schaeffer, J. 1998. Pattern databases. *Computational Intelligence* 14(3):318–334.

Edelkamp, S. 2001. Planning with pattern databases. In *Proceedings of the 6th European Conference on Planning*, 13–24.

Fikes, R., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artif. Intell.* 2(3/4):189–208.

Fišer, D., and Komenda, A. 2018. Fact-alternating mutex groups for classical planning. *J. Artif. Intell. Res.* 61:475–521.

Fišer, D.; Štolba, M.; and Komenda, A. 2015. MAPlan. In *Competition of Distributed and Multi-Agent Planners (CoDMAP)*, 8–10.

Haslum, P., and Geffner, H. 2000. Admissible heuristics for optimal planning. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems (AIPS)*, 140–149.

Haslum, P. 2009. $h^m(P) = h^1(P^m)$: Alternative characterisations of the generalisation from $h^{\max}$ to h^m. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, 354–357.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*.

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS)*, 176–183.

Helmert, M. 2006. The Fast Downward planning system. *J. Artif. Intell. Res. (JAIR)* 26:191–246.

Massey, B. 1999. *Directions in planning: Understanding the flow of time in planning*. Ph.D. Dissertation, University of Oregon.

Pettersson, M. P. 2005. Reversed planning graphs for relevance heuristics in ai planning. In *Planning, Scheduling and Constraint Satisfaction: From Theory to Practice*, volume 117, 29–38. IOS Press.

Suda, M. 2013. Duality in STRIPS planning. *CoRR* abs/1304.0897.