Scorpion 2023

Jendrik Seipp
Linköping University
Linköping, Sweden
jendrik.seipp@liu.se

This planner abstract describes "Scorpion 2023", the planner configuration we submitted to the sequential optimization track of the International Planning Competition 2023. Scorpion 2023 is implemented within the Scorpion planning system, which is an extension of Fast Downward (Helmert 2006). Like the original Scorpion configuration, which participated in IPC 2018, Scorpion 2023 uses A^* (Hart, Nilsson, and Raphael 1968) with an admissible heuristic (Pearl 1984) to find optimal plans. The overall heuristic is based on component abstraction heuristics that are combined by saturated cost partitioning (Seipp, Keller, and Helmert 2020).[1]

In this abstract we only list the components of Scorpion and the settings we used for them. For a detailed description of the underlying algorithms we refer to Seipp, Keller, and Helmert (2020).

Abstraction Heuristics

Depending on whether or not a given task contains conditional effects, we use a different set of abstraction heuristics.

Tasks Without Conditional Effects

For tasks without conditional effects we use the combination of the following heuristics:

- Cartesian abstraction heuristics (CART):
  We consider Cartesian abstractions of the landmark and goal task decompositions (Seipp and Helmert 2018). We limit the total number of non-looping transitions in all abstractions underlying the Cartesian heuristics by one million.
- pattern databases selected by saturated cost partitioning SYS-SCP algorithm (SYS-SCP):
  We iteratively generate larger *interesting* patterns and let saturated cost partitioning choose the ones whose projection contains non-zero goal distances under the remaining cost function.

Tasks With Conditional Effects

For tasks with conditional effects we only use the SYS-SCP patterns as described above.

[1] We chose the name "Scorpion" since it contains the letters s(aturated) c(ost) p(artitioning) in this order.

Saturated Cost Partitioning

We combine the information contained in the component heuristics with saturated cost partitioning (Seipp and Helmert 2018). Given an ordered collection of heuristics, saturated cost partitioning iteratively assigns each heuristic h only the costs that h needs for justifying its estimates and saves the remaining costs for subsequent heuristics. Distributing the operator costs among the component heuristics in this way makes the sum of the individual heuristic values admissible.

The quality of the resulting saturated cost partitioning heuristic strongly depends on the order in which the component heuristics are considered (Seipp, Keller, and Helmert 2017). Additionally, we can obtain much stronger heuristics by maximizing over multiple saturated cost partitioning heuristics computed for different orders instead of using a single saturated cost partitioning heuristic (Seipp, Keller, and Helmert 2017). We therefore iteratively sample a state (using the sampling algorithm by Haslum et al. 2007), use a greedy algorithm for finding an initial order for the state (more concretely, we use the static greedy ordering algorithm with the $q_{\frac{h}{stolen}}$ scoring function) and afterwards optimize the order with simple hill climbing in the space of orders for at most two seconds (Seipp 2018). If the the saturated cost partitioning heuristic computed for the resulting optimized greedy order yields a higher estimate for one of a set of 1000 sample states than all previously added orders, we add the order to our set of orders. We limit the time for finding orders in this way to 100 seconds.

Operator Pruning Techniques

We employ two operator pruning techniques:

- atom-centric strong stubborn sets (Röger et al. 2020):
  We switch off pruning in case the fraction of pruned successor states is less than 20% of the total successor states after 1000 expansions.
- h^2 mutexes (Alcázar and Torralba 2015):
  This operator pruning method can remove irrelevant operators. We invoke it after translating a given input task to SAS^+ and before starting the search component of Fast Downward.

References

Alcázar, V., and Torralba, Á. 2015. A reminder about the importance of computing and exploiting invariants in planning. In Brafman, R.; Domshlak, C.; Haslum, P.; and Zilberstein, S., eds., *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS 2015)*, 2–6. AAAI Press.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2):100–107.

Haslum, P.; Botea, A.; Helmert, M.; Bonet, B.; and Koenig, S. 2007. Domain-independent construction of pattern database heuristics for cost-optimal planning. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI 2007)*, 1007–1012. AAAI Press.

Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

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

Röger, G.; Helmert, M.; Seipp, J.; and Sievers, S. 2020. An atom-centric perspective on stubborn sets. In Harabor, D., and Vallati, M., eds., *Proceedings of the 13th Annual Symposium on Combinatorial Search (SoCS 2020)*, 57–65. AAAI Press.

Seipp, J., and Helmert, M. 2018. Counterexample-guided Cartesian abstraction refinement for classical planning. *Journal of Artificial Intelligence Research* 62:535–577.

Seipp, J.; Keller, T.; and Helmert, M. 2017. Narrowing the gap between saturated and optimal cost partitioning for classical planning. In Singh, S., and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*, 3651–3657. AAAI Press.

Seipp, J.; Keller, T.; and Helmert, M. 2020. Saturated cost partitioning for optimal classical planning. *Journal of Artificial Intelligence Research* 67:129–167.

Seipp, J. 2018. *Counterexample-guided Cartesian Abstraction Refinement and Saturated Cost Partitioning for Optimal Classical Planning*. Ph.D. Dissertation, University of Basel.