From: AAAI-00 Proceedings. Copyright © 2000, AAAI (www.aaai.org). All rights reserved.

Automatic Generation of Memory Based Search Heuristics

István T. Hernádvölgyi
University of Ottawa
School of Information Technology & Engineering
Ottawa, Ontario, K1N 6N5, Canada
Email: istvan@site.uottawa.ca

Our goal is to automatically generate heuristics to guide state space search. The heuristic values are distances computed in an abstract space which is automatically derived from the original space. The search space is described in a production system. Simple syntactic transformations of this description give rise to another search space. The distances of abstract states from the abstract goal state are stored in a look-up table and provide admissible and monotonic heuristics for search algorithms such as IDA*. The size of the abstract space is the size of the look-up table and different transformations on the description of the space give rise to abstract spaces of different size. We are interested in the relationship between the memory required to store the heuristic and the speed of search. We are also interested in ranking abstractions which generate abstract spaces of the same cardinality with respect to their predicted performance without actually performing searches in the original space. We also plan to use our technique to search for macro operators to find suboptimal paths very quickly. A macro operator is a sequence of operators which immediately reaches a subgoal state applied to a state without performing search.

Culberson and Schaeffer (Culberson & Schaeffer 1996) developed a technique (*pattern database*) to represent heuristic look-up tables and effectively used it on the 15-Puzzle. Korf used pattern databases to find optimal paths for random instances of the Rubik's Cube for the first time. In his paper he conjectured that the size of the pattern database and the speed of search can be linearly traded for each other. We verified his conjecture in a large scale experiment and reported it in (Holte & Hernádvölgyi 1999). Korf and Reid in (Korf & Reid 1998) gave a more formal derivation of the expected number of states generated by the search algorithm based on the distribution of heuristic values. We used their ideas to select the best heuristics in a large pool of heuristics with equal memory requirements. We devised a simple vector notation for representing state spaces and a method for automatically creating abstractions based on this notation. Our technique based on mapping labels (*domain abstraction*) is guaranteed to create abstract spaces where the distances provide admissible and monotonic heuristic values. Some abstractions are non-surjective; there are states in the abstract space which have no pre-image in the original space. These states take up space in the pattern database and they often shorten the optimal path in the abstract space resulting in small heuristic values. Non-surjective abstractions arise frequently in our experience.

We have identified some structural properties which may be used to avoid non-surjective abstractions, but to date we have no automatic way of avoiding them in general. We are working on efficiently computable methods which can quickly determine if the state has a pre-image. These involve invoking suboptimal but very fast search techniques – such as refinement and macro search – to test membership. Traditionally macro operators were found by blind search techniques (Korf 1985) and by macro composition due to the lack of heuristics for searching for the macros. Domain abstraction can provide heuristics to search for macro operators and we plan to use it to find shorter macro operators and to build macro tables for very large spaces. Domain abstraction proved very successful in many problem spaces but it has no use in binary domains. We are investigating other techniques which also give rise to abstract spaces by reducing the dimension of the state representation rather than map labels. We are also working on extending our current production system to encode more complex spaces. Our preliminary data suggests that combining more than one smaller abstractions with total size m results in faster searches than a single one with size m. We plan to study how to select those abstractions from a pool of abstractions whose combined performance is the best.

Copyright © 2000, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

References

Culberson, J. C., and Schaeffer, J. 1996. Searching with pattern databases. *Advances in Artificial Intelligence (Lecture Notes in Artificial Intelligence 1081)* 402–416.

Holte, R. C., and Hernádvölgyi, I. T. 1999. A space-time tradeoff for memory-based heuristics. *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)* 704–709.

Korf, R. E., and Reid, M. 1998. Complexity analysis of admissible heuristic search. *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)* 305–310.

Korf, R. E. 1985. Macro operators: A weak method for learning. *Artificial Intelligence* 26:35–77.