A Database of Databases

Jonathan Schaeffer

Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada T6G 2H1
jonathan@cs.ualberta.ca

Dedication

This paper is to honour Bob Herschberg for his 14 years of dedication to the ICCA Journal. Over this period of time, Bob Herschberg and Jaap van den Herik turned the ICCA Newsletter into the ICCA Journal, a quality publication that our research community is justifiably proud of. Both are endgame database advocates, and their enthusiasm has inspired many people (including this author) to use this tool to uncover new secrets.

1. Introduction

Endgame databases were pioneered in the game of chess. Chess endgame positions with a few pieces on the board (typically less than six) have been exhaustively enumerated and each position solved (for example, [11, 12, 21]). Using retrograde analysis [22], the game-theoretic result (win, loss or draw) for each position can be determined. In effect, these databases are a repository of perfect information [14].

Endgame databases are useful in two ways. First, the database positions can be regarded as having heuristic values with no error. Each such position encountered in the search improves the accuracy of the final search result. Second, once encountering a database position, no further search is required beyond that point. Database positions can eliminate large portions of the search tree, allowing the program to search deeper and thereby further improve the accuracy of the result.

Although endgame databases are best known for their impact on the game of chess (for example, changing the rules for declaring a draw [6, 7, 10, 13, 15]), they have also been successfully applied to a number of two-player games and one-player puzzles. This article surveys some of the domains where endgame databases have been used.

2. Checkers

Endgame databases are an integral part of the *Chinook* checkers (8x8 draughts) program [19, 20]. Over a five-year period, the *Chinook* team computed all positions with eight or fewer pieces on the board. The resulting database has 443,748,401,247 (4.4×10^{11}) positions and is compressed into roughly six gigabytes of data [17]. The data has been organized to allow for real-time decompression during the search. Because of the forced capture rule in checkers, searches at the start of the game (with 24 pieces on the board) can lead to endgame database positions being reached (with 8 or fewer pieces).

The result is a significant increase in the accuracy of the search. For example, as early as move five in a tournament game, *Chinook* has been able to declare a game drawn. This suggests that it may be possible to solve the game. Unfortunately, there are roughly 5×10^{20} possible checkers positions, meaning that potentially a lot more positions will have to be computed before the game can be solved.

The endgame databases have overturned a number of results in the checkers literature. For example, the well-known 100 Years Problem was through to be a forced win. Chinook's databases quickly show this position to be a draw (the problem is now to be called the 197 Years Problem).

The *Chinook* team have also computed the 5-piece databases for 10x10 checkers (international checkers) and Italian checkers.

3. Nine Men's Morris

Nine Men's Morris is popular in the German speaking part of Europe. The game is played on a square board which has 24 empty squares (or points), with connecting lines between some of the squares. The game has two phases. In the opening phase, each player alternates putting one of their nine stones on an empty square. After all the stones have been placed, then the middlegame commences. Here players alternate making moves by sliding a stone on the board. If a player gets three men in a row, then they can remove an opponent's stone. A side having less than three stones on the board loses.

Ralph Gasser used endgame databases to solve the game: with perfect play it is a draw [8-]. He computed 7,673,759,269 endgame positions: all the reachable positions with nine white and nine black (or fewer) stones on the board. This database essentially solves the middlegame. To solve the game, an 18-ply search is needed to work out all the possible ways that players can reach the middlegame from the opening.

4. Awari

Awari is an African game played on a board with 12 pits in a circle with six consecutive pits for each player. Initially, each pit contains four stones (hence, 48 stones to start). Players alternately empty one of their pits by distributing the contents in a counter clockwise fashion, one per pit. If a player's move ends in an opponent's pit, and that pit contains two or three stones, then these stones are captured and removed from the playing area. The goal is to capture all the opponent's stones.

Endgame databases have been built for Awari. All positions with 23 or fewer stones on the board have been solved, a total of 834,451,800 positions [2]. Although this database is beneficial for an Awari-playing program [1], there is still a long way to go before the game can be solved (estimated 10^{12} positions).

5. 15-Puzzle

All the examples above have been for two-player games. Endgame databases have also been used to improve the search efficiency of single-agent search. For the 4x4 sliding tile puzzle (the 15-Puzzle), all positions within 25 moves of the goal state have been computed [3]. Note that two-player games usually build databases based on the number of pieces on the board. Since no pieces are removed in the sliding tile puzzles, the database is computed as a function of the number of moves to the solution. The database contains 36,142,146 positions and is compressed into 106 megabytes.

The 25-move endgame database has been used with an iterative deepening A* (IDA*) searcher. Surprisingly the benefits of the database are small, yielding a 1.9-fold reduction in the search tree size on a standard set of 100 problems [16]. In IDA*, the heuristic evaluation function allows many branches to be cut off quickly, before they can reach the databases.

So far we have discussed computing databases off-line; the computation is saved to disk and reused as often as necessary. Recently work has been done on creating endgame databases on the fly. With *perimeter search*, the program dynamically builds a perimeter (endgame database) around the goal node as needed [18]

Endgame databases operate on the entire state of the game. This idea has been applied to partial states of the game. Pattern databases operate on portions of the board. In the 15-Puzzle, a subset of the board (7 tiles plus the blank) were enumerated and the minimum distance required to correctly place the tiles was computed. This was done for two different subsets of tiles, resulting in two databases each containing 518,918,400 positions [4, 5]. Using these databases in a search resulted in an improved lower bound estimator, leading to an average 1,000-fold reduction in the search effort.

6. Conclusion

Endgame databases have become an important enhancement to single-agent and two-player search. With pattern databases, there is the potential to generalize the idea, making it useful in a wider set of applications.

References

1. L. V. Allis, M. Meulen and H. J. Herik, Databases in Awari, in *Heuristic Programming in Artificial Intelligence 2, The Second Computer Olympiad*, D. N. L. Levy and D. F. Beal (ed.), Ellis Horwood, 1991, 232-243.
2. H. Bal and L. V. Allis, Parallel Retrograde Analysis on a Distributed System, *Supercomputing ’95*, 1995.
3. J. Culberson and J. Schaeffer, Efficiently Searching the 15-Puzzle, Tech. Rep. 94-08, Department of Computing Science, University of Alberta, 1994.
4. J. Culberson and J. Schaeffer, Searching with Pattern Databases, *Advances in Artificial Intelligence*, Berlin, 1996, 402-416. 11th biennial Canadian Society for Computational Studies of Intelligence (CSCSI) Conference.
5. J. Culberson and J. Schaeffer, Pattern Databases, *Computational Intelligence*, 1997. To appear.
6. S. T. Dekker, H. J. Herik and I. S. Herschberg, Perfect Knowledge Revisited, *Artificial Intelligence 43*, 1 (1990), 111-123..
7. L. Falk, H. J. Herik and I. S. Herschberg, So Help me the Database, *ICCA Journal 12*, 3 (1984), 165-166.
8. R. Gasser, Harnessing Computational Resources for Efficient Exhaustive Search, Ph.D. thesis, Department of Computer Science, Swiss Federal Institute of Technology (ETH) Zurich, 1995.

9. R. Gasser, Solving Nine Men's Morris, *Computational Intelligence 12*, 1 (1996), 24-41.

10. H. J. Herik and I. S. Herschberg, Elementary Theory Improved, A Conjecture Refuted, *ICCA Journal 8*, 3 (1985), 141-149.

11. H. J. Herik and I. S. Herschberg, A Data Base on Data Bases, *ICCA Journal 9*, 1 (1986), 29-34.

12. H. J. Herik, I. S. Herschberg and N. Nakad, A Six-Men-Endgame Database: KRP(a2)KbBP(a3), *ICCA Journal 10*, 4 (1987), 163-180.

13. H. J. Herik and I. S. Herschberg, The 50-move Rule Revisited, *ICCA Journal 12*, 3 (1989), 192.

14. I. S. Herschberg and H. J. Herik, Calling All Mates, *ICCA Journal 8*, 2 (1985), 45-46.

15. I. S. Herschberg and H. J. Herik, A Gauge of Endgames, *ICCA Journal 8,* 4 (1985), 225-229.

16. R. Korf, Iterative Deepening: An Optimal Admissible Tree Search, *Artificial Intelligence 27*, (1985), 97-109.

17. R. Lake, J. Schaeffer and P. Lu, Solving Large Retrograde Analysis Problems Using a Network of Workstations, *Advances in Computer Chess 7*, Maastricht, Netherlands, 1994, 135-162.

18. G. Manzini, BIDA*: An Improved Perimeter Search Algorithm, *Artificial Intelligence 75*, (1995), 347-360.

19. J. Schaeffer, J. Culberson, N. Treloar, B. Knight, P. Lu and D. Szafron, A World Championship Caliber Checkers Program, *Artificial Intelligence 53*, 2-3 (1992), 273-290.

20. J. Schaeffer, *One Jump Ahead*, Springer-Verlag, 1997.

21. L. Stiller, Parallel Analysis of Certain Endgames, *ICCA Journal 12*, 2 (1989), 55-64.

22. K. Thompson, Retrograde Analysis of Certain Endgames, *ICCA Journal 9*, 3 (1986), 131-139.