Forthcoming Papers

Chips Challenging Champions: Games, Computers, and Artificial Intelligence (Special Issue edited by Jonathan Schaeffer and Jaap van den Herik)

J. Schaeffer and H.J. van den Herik, Games, computers, and Artificial Intelligence

R. Korf, A. Felner, Disjoint pattern database heuristics

We describe a new technique for designing more accurate admissible heuristic evaluation functions, based on pattern databases [J. Culberson, J. Schaeffer, Comput. Intelligence 14 (3) (1998) 318–334]. While many heuristics, such as Manhattan distance, compute the cost of solving individual subgoals independently, pattern databases consider the cost of solving multiple subgoals simultaneously. Existing work on pattern databases allows combining values from different pattern databases by taking their maximum. If the subgoals can be divided into disjoint subsets so that each operator only affects subgoals in one subset, then we can add the pattern-database values for each subset, resulting in a more accurate admissible heuristic function. We used this technique to improve performance on the Fifteen Puzzle by a factor of over 2000, and to find optimal solutions to 50 random instances of the Twenty-Four Puzzle. © 2001 Published by Elsevier Science B.V.

M.L. Littman, G.A. Keim and N. Shazeer, A probabilistic approach to solving crossword puzzles

We attacked the problem of solving crossword puzzles by computer: given a set of clues and a crossword grid, try to maximize the number of words correctly filled in. After an analysis of a large collection of puzzles, we decided to use an open architecture in which independent programs specialize in solving specific types of clues, drawing on ideas from information retrieval, database search, and machine learning. Each expert module generates a (possibly empty) candidate list for each clue, and the lists are merged together and placed into the grid by a centralized solver. We used a probabilistic representation as a common interchange language between subsystems and to drive the search for an optimal solution. PROVERB, the complete system, averages 95.3% words correct and 98.1% letters correct in under 15 minutes per puzzle on a sample of 370 puzzles taken from the New York Times and several other puzzle sources. This corresponds to missing roughly 3 words or 4 letters on a daily 15 × 15 puzzle, making PROVERB a better-than-average cruciverbalist (crossword solver). © 2001 Published by Elsevier Science B.V.

M. Campbell, A.J. Hoane Jr. and F.-H. Hsu, Deep Blue

Deep Blue is the chess machine that defeated then-reigning World Chess Champion Garry Kasparov in a six-game match in 1997. There were a number of factors that contributed to this success, including:

0004-3702/2001 Published by Elsevier Science B.V.
Pii: S0004-3702(01)00167-9
• a single-chip chess search engine,
• a massively parallel system with multiple levels of parallelism,
• a strong emphasis on search extensions,
• a complex evaluation function, and
• effective use of a Grandmaster game database.

This paper describes the Deep Blue system, and gives some of the rationale that went into the design decisions behind Deep Blue. © 2001 Published by Elsevier Science B.V.

M. Buro, Improving heuristic mini-max search by supervised learning

This article surveys three techniques for enhancing heuristic game-tree search pioneered in the author’s Othello program LOGISTELLO, which dominated the computer Othello scene for several years and won against the human World-champion 6–0 in 1997. First, a generalized linear evaluation model (GLEM) is described that combines conjunctions of Boolean features linearly. This approach allows an automatic, data driven exploration of the feature space. Combined with efficient least squares weight fitting, GLEM greatly eases the programmer’s task of finding significant features and assigning weights to them. Second, the selective search heuristic PROBCUT and its enhancements are discussed. Based on evaluation correlations PROBCUT can prune probably irrelevant sub-trees with a prescribed confidence. Tournament results indicate a considerable playing strength improvement compared to full-width $\alpha$–$\beta$ search. Third, an opening book framework is presented that enables programs to improve upon previous play and to explore new opening lines by constructing and searching a game-tree based on evaluations of played variations. These general methods represent the state-of-the-art in computer Othello programming and begin to attract researchers in related fields. © 2001 Published by Elsevier Science B.V.

V. Anshelevich, A hierarchical approach to computer Hex

Hex is a beautiful game with simple rules and a strategic complexity comparable to that of Chess and Go. The massive game-tree search techniques developed mostly for Chess and successfully used for Checkers and a number of other games, become less useful for games with large branching factors like Hex and Go. In this paper, we describe deduction rules, which are used to calculate values of complex Hex positions recursively starting from the simplest ones. We explain how this approach is implemented in HEXY—the strongest Hex-playing computer program, the Gold medallist of the 5th Computer Olympiad in London, August 2000. © 2001 Published by Elsevier Science B.V.

H. Iida, M. Sakuta and J. Rollason, Computer shogi

This paper describes the current state of the art in computer shogi. Shogi (Japanese chess) promises to be a good vehicle for future research into game-playing programs that are based on tree-searching paradigms. This paper shows where chess and shogi are similar, and details the important areas that make shogi programming of particular interest. A crucial difference is the game-tree complexity, which is significantly higher in shogi than in chess. Three important differences are the “drop” rule, the diverging character of the game, and the slow build-up of forces. They make it difficult to have effective opening and endgame procedures.

After a short summary of the rules of shogi and an outline of the main areas of current work in computer shogi, we provide an overview of the history of computer shogi, in which computer-shogi activities both in human tournaments and in exhibition events are given. We conjecture that by the year 2010 a computer will be comparable in strength to the best human players. The most important techniques used in computer shogi are described. We focus on issues such as opening play, selective search, quiescence search, solving tactical exchanges without tree searching, position
evaluation and endgame play. At the end the key challenges in computer shogi are enumerated, and finally, concluding remarks are given. © 2001 Published by Elsevier Science B.V.

M. Müller, Computer Go
Computer Go is one of the biggest challenges faced by game programmers. This survey describes the typical components of a Go program, and discusses knowledge representation, search methods and techniques for solving specific subproblems in this domain. Along with a summary of the development of computer Go in recent years, areas for future research are pointed out. © 2001 Published by Elsevier Science B.V.

G. Tesouro, Programming backgammon using self-teaching neural nets
TD-Gammon is a neural network that is able to teach itself to play backgammon solely by playing against itself and learning from the results. Starting from random initial play, TD-Gammon’s self-teaching methodology results in a surprisingly strong program: without lookahead, its positional judgement rivals that of human experts, and when combined with shallow lookahead, it reaches a level of play that surpasses even the best human players. The success of TD-Gammon has also been replicated by several other programmers; at least two other neural net programs also appear to be capable of superhuman play.

Previous papers on TD-Gammon have focused on developing a scientific understanding of its reinforcement learning methodology. This paper views machine learning as a tool in a programmer’s toolkit, and considers how it can be combined with other programming techniques to achieve and surpass world-class backgammon play. Particular emphasis is placed on programming shallow-depth search algorithms, and on TD-Gammon’s doubling algorithm, which is described in print here for the first time. © 2001 Published by Elsevier Science B.V.

D. Billings, A. Davidson, J. Schaeffer and D. Szafron, The challenge of poker
Poker is an interesting test-bed for artificial intelligence research. It is a game of imperfect information, where multiple competing agents must deal with probabilistic knowledge, risk assessment, and possible deception, not unlike decisions made in the real world. Opponent modeling is another difficult problem in decision-making applications, and it is essential to achieving high performance in poker.

This paper describes the design considerations and architecture of the poker program Poki. In addition to methods for hand evaluation and betting strategy, Poki uses learning techniques to construct statistical models of each opponent, and dynamically adapts to exploit observed patterns and tendencies. The result is a program capable of playing reasonably strong poker, but there remains considerable research to be done to play at a world-class level. © 2001 Published by Elsevier Science B.V.

B. Sheppard, World-championship-caliber Scrabble
Computer Scrabble programs have achieved a level of performance that exceeds that of the strongest human players. Maven was the first program to demonstrate this against human opposition. Scrabble is a game of imperfect information with a large branching factor. The techniques successfully applied in two-player games such as chess do not work here. Maven combines a selective move generator, simulations of likely game scenarios, and the B* algorithm to produce a world-championship-caliber Scrabble-playing program. © 2001 Published by Elsevier Science B.V.
H.J. van den Herik, J.W.H.M. Uiterwijk and J. van Rijswijck, Games solved: Now and in the future

In this article we present an overview on the state of the art in games solved in the domain of two-person zero-sum games with perfect information. The results are summarized and some predictions for the near future are given. The aim of the article is to determine which game characteristics are predominant when the solution of a game is the main target. First, it is concluded that decision complexity is more important than state-space complexity as a determining factor. Second, we conclude that there is a trade-off between knowledge-based methods and brute-force methods. It is shown that knowledge-based methods are more appropriate for solving games with a low decision complexity, while brute-force methods are more appropriate for solving games with a low state-space complexity. Third, we found that there is a clear correlation between the first-player’s initiative and the necessary effort to solve a game. In particular, threat-space-based search methods are sometimes able to exploit the initiative to prove a win. Finally, the most important results of the research involved, the development of new intelligent search methods, are described. © 2001 Published by Elsevier Science B.V.

K. Markert and U. Hahn, Understanding metonymies in discourse

X. Nguyen, S. Kambhampati and R.S. Nigenda, Planning graph as the basis for deriving heuristics for plan synthesis by state space and CSP search

T. Sandholm, Algorithm for optimal winner determination in combinatorial auctions

J.F. Hory, Skepticism and floating conclusions

F. Wotawa, On the relationship between model-based debugging and program slicing

J. Amilhastre, H. Fargier and P. Marquis, Consistency restoration and explanations in dynamic CSPs—Application to configuration

B. Guijarro-Berdiñas, A. Alonso-Betanzos and O. Fontenla-Romero, Intelligent analysis and pattern recognition in cardiotocographic signals using a tightly coupled hybrid system

K. Engesser and D.M. Gabbay, Quantum logic, Hilbert space, revision theory

O. Bernard and J.-L. Gouzé, Global qualitative description of a class of nonlinear dynamical systems