Silicon Champions of the Game

Computers have conquered tic-tac-toe, checkers, and chess. What's next?

By IVARS PETERSON

The final game of the match lasted barely more than an hour. A rattled Garry Kasparov conceded defeat after falling into a trap that had been set by the IBM chess computer Deep Blue.

Deep Blue's triumph last May marked the first match victory by a chess-playing computer over a reigning world champion (SN: 5/17/97, p. 300). This week, the team of researchers who developed Deep Blue, led by Chung-Jen Tan of the IBM Thomas J. Watson Research Center in Yorktown Heights, N.Y., received the prestigious Fredkin Prize for Computer Chess. Established in 1980 by computer scientist Edward Fredkin, now at Carnegie Mellon University in Pittsburgh, the $100,000 award honors the first computer program to defeat a world champion in a regulation match.

The victory also represented the culmination of nearly 50 years of scientific and engineering effort. The field of computer chess got its start in 1950 with the ideas of applied mathematician Claude E. Shannon, then at Bell Telephone Laboratories, who proposed the basic search and evaluation strategies that still underlie the way computers generate chess moves.

Since that time, one chess-playing computer after another has held center stage, each eventually falling to a faster, more powerful successor: KAISSA, MAC HACK, CHESS 4.6, Belle (SN: 10/8/83, p. 236), CRAY BLITZ (SN: 10/29/83, p. 276), Hitech (SN: 10/26/85, p. 260), and Deep Thought (SN: 10/28/89, p. 276), the immediate predecessor of Deep Blue.

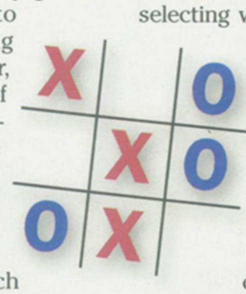
"The beauty of computer chess was that ideas could be tested in competition," says computer scientist Monty Newborn of McGill University in Montreal. "The good ideas went from one generation to the next, and the bad ideas fizzled out. That's science at its best."

Chess isn't the only game being played by computers at or near the championship level. At this week's Fourteenth National Conference on Artificial Intelligence in Providence, R.I., the Hall of Champions event brought together some of the world's top computer programs playing backgammon, bridge, checkers, chess, Go, Othello, and Scrabble.

"We're at a unique point in time," says Matthew L. Ginsberg of the University of Oregon in Eugene, who organized the event. "Ten years ago, no computers were close to the championship level in any of these games. Now, they even have the edge over human players in several of them. We can have the best computers competing against the best people."

Indeed, anyone can try his or her hand at playing top programs in many games just by going to the World Wide Web. Researchers and game developers monitor play and use the data to improve their programs.

Even in the earliest days of computers, researchers couldn't resist programming them to play games. It was an entertaining way to show off one's programming prowess, to test the computer, and to evaluate the efficacy of various techniques for organizing information in massive databases or searching among a wide range of possibilities to determine the best choice.



Chess was often the chosen battleground, though much simpler games such as tic-tac-toe served as handy programming exercises. Indeed, it's not difficult to write a short computer program that plays tic-tac-toe flawlessly, in effect demonstrating that no matter what the first move, the worst you can do is tie.

In recent years, researchers have solved a number of games similar to, but more challenging than, tic-tac-toe. In connect-4, two players take turns dropping white or black balls into seven tubes, each of which holds a maximum of six balls. The first person to create a line of four balls in a row, column, or diagonal wins. In this game, by playing correctly, the player going first can always win.

Go-Moku (or five-in-a-row), which is played on a 19-by-19 square grid, is also a guaranteed win for the savvy player moving first. The same applies to Qubic, a three-dimensional version of tic-tac-toe played on a 4-by-4-by-4 lattice. In nine-men's morris, an alignment-and-capture game popular in Europe, neither player can be assured of a triumph.

In such solved games, where a good player can recognize all the alternatives for any situation, a computer can be programmed to make the best possible moves at all times, and a win or a draw is guaranteed. Games such as chess, checkers, and Go are, in principle, solvable, and a computer could be programmed to play a perfect game. However, the number of possible moves is so enormous that no existing computer can figure out the entire game from beginning to end.

In the early days of computer chess, some researchers attempted to mimic the way humans play the game, building in pattern recognition, invoking various rules of thumb, and developing criteria for selecting which moves to consider while discarding the rest. However, the programmers found it extremely difficult to furnish the computer with enough knowledge to avoid making major mistakes.

The alternative that proved much more powerful was the brute-force search—simply checking out all the moves. The program looks ahead a fixed number of moves, evaluates the strength of each move, and selects the best one. Adding knowledge about the game and refined algorithms has made searches more responsive to actual game situations and turned this strategy into a remarkably effective mode of operation.

At the same time, the steadily increasing speed of computers has allowed chess programs to search more and more moves into the future. Experiments have clearly demonstrated that the faster the computer, the better a program plays, simply because it can perform a more extensive search. "That's counter to what a lot of people argued a number of years ago," Newborn says.

"The message from chess is profound and widely applicable," says Carnegie

Mellon's Hans Berliner. "Brute force is a practical way of doing things."

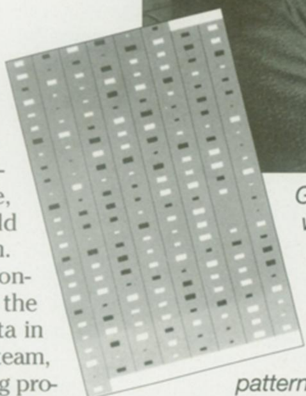
The success of computers like Deep Blue also highlights the fact that the way computers play a game differs fundamentally from the way people play it. From a human perspective, computers sometimes make weird moves; yet more often than not, the best programs somehow manage to succeed in the end.

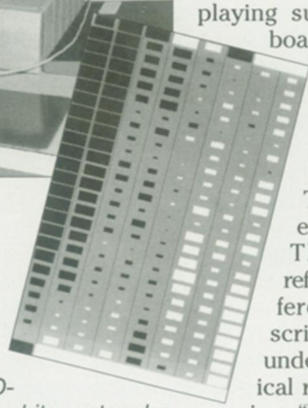
That difference in style can be very valuable. "We're good at pattern matching, and we're good at applying rules," Ginsberg says. "Machines are good at searching."

"This means that the capabilities of computers are complementary to ours," he continues. "Together, we can solve problems that neither of us can solve individually."

Moreover, "we need to face the fact that things that once could be done only through human intelligence can now be done in other ways as well," says former U.S. chess champion Patrick G. Wolff of Cambridge, Mass. "The intriguing question is, how many things are there like that?"

IBM Research





Gerald Tesauro of IBM Research with an early version of his backgammon-playing program TD-Gammon. The computer is playing black and has established a strong position on the board (top). The patterns (left and right) illustrate how TD-Gammon develops its playing skill. The black and white rectangles represent the numerical strengths of the connections between artificial neurons in the neural network programmed to play backgammon. The size of the rectangles indicates the magnitude of excitation (white) or inhibition (black). Initially, the neural network has a random distribution of connection strengths (left). After about 200,000 games, the network's connection strengths exhibit a strong pattern, reflecting the computer program's recognition of features important to winning the game (right).

Even before Deep Blue defeated Kasparov, a program named Chinook had become, in effect, the world checkers champion.

Created by Jonathan Schaeffer of the University of Alberta in Edmonton and his team, the checkers-playing program incorporates the types of search strategies originally developed for chess. It also includes enormous databases covering every possible position that can be reached once there are fewer than a certain number of pieces on the board (SN: 7/20/91, p. 40).

With such databases at its disposal and with the game down to a manageable number of pieces, Chinook can look up all possible outcomes and select an appropriate sequence of moves to ensure a win, maintain a draw, or delay a loss. From then on, it plays flawlessly.

In 1994, Chinook played world checkers champion Marion Tinsley, a retired mathematician from Tallahassee, Fla., and a formidable opponent. Since 1975, he had lost only a handful of the thousands of games he had played in tournaments and exhibition matches. Two of those losses had occurred in 1992, when Tinsley successfully defended his world title against Chinook in a man-versus-machine match of 40 games (SN: 10/3/92, p. 217).

In the 1994 rematch, the first six games between Tinsley and Chinook ended in draws. Then, Tinsley had to resign for health reasons. He was diagnosed as having cancer, and he died a year later.

"Tinsley was without a doubt the best checker player of all time—an absolutely incredible talent," Schaeffer says. Having beaten the top remaining checker players, Chinook qualifies as the current champion.

Whether Chinook could ever have defeated Tinsley remains a nagging question, and Schaeffer has considered the possibility of calculating the game from beginning to end and building a perfect checker player to settle the issue.

"I certainly believe we're capable of solving the game," Schaeffer says. "The technology is here. It's just a matter of committing the time and resources."

Chinook is also a research experiment. For instance, Schaeffer and his team have built a database of 444 billion positions—every position with eight or fewer pieces on the board. "This is a vast repository of information. To a checker player, it's a gold mine," he says. Whatever data-mining techniques are developed to sift through the information and identify what's important would benefit many fields.

Meanwhile, Chinook continues to play in tournaments and exhibitions. The only major change in the program since 1994 has been the removal of restrictions that gave it an extremely cautious style specifically designed to counter the near-perfect play of Tinsley.

Instead of achieving draw after draw after boring draw, Chinook has started to play games that are truly exciting, Schaeffer says. "The program's winning percentage has gone up and up, and its losing percentage has remained the same—zero.

"That was a relatively minor change in the [computer program], but it had a dramatic impact on the play," he adds.

The world's top backgammon programs differ markedly from those that play checkers and chess. Instead of relying on brute-force searches, the software incorporates a model brain—an artificial neural network—that allows the program to learn the game from scratch.

In backgammon, two players race their pieces around a track on a rectangular board. Each player uses two dice to determine how far to move one or two pieces at a time with the objective of winning the race by conveying all of one's 15 pieces around the playing surface and off the board.

The neural network approach to playing backgammon was pioneered by IBM's Gerald Tesauro, who created a program called TD-Gammon. "TD" refers to "temporal difference," which describes the program's underlying mathematical recipe for self-learning. "We turn the neural net loose on this task, and it just learns by playing lots and lots of games against itself," Tesauro says. "It learns very well—though some things are learned better than others."

The original concern was that such an approach would lead to a program that lacks flexibility and is unable to cope with unexpected situations presented by players using unconventional tactics. "It actually does very well against all kinds of different strategies," Tesauro says. The random rolls of the dice during the learning phase seem to force the neural network to explore all sorts of situations and develop remarkably robust strategies.

"Unfortunately, there are strategies and situations that never occur when you play just against yourself," says Brian Sheppard, a software developer in Concord, Mass., who is working on a new expert backgammon player. "You have to be told about them. An expert [human] player can make these situations arise with some regularity."

Backgammon nevertheless remains the one major success for automated learning in the domain of games. The neural network approach has generally not worked as well for deterministic games such as chess, checkers, Othello, and Go, which have no element of chance.

Other leading backgammon programs, such as JellyFish, have followed TD-Gammon's lead, also incorporating neural network learning and sometimes adding search techniques. Several of these programs rank among the top 20 backgammon players in the world.

"Games are good proving grounds for testing learning algorithms," Tesauro remarks. "There's lots of complexity, but the task is clear-cut and the rules extremely clean."

In card games such as contract bridge and poker, players deal not only with chance but also with incomplete information about what cards the other players hold. It's just this sort of uncertainty that makes these games so alluring to their practitioners—and so difficult for programmers.

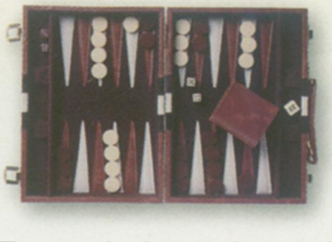
Bridge is a card game for four players who form two partnerships. The deck of cards is dealt evenly to the four players, so each gets 13 cards. Players start by bidding for the right to play the hand, and whichever side makes the highest bid then tries to take the number of tricks indicated by its bid.

The two key elements of the game are bidding and card play. The sticking point is that no single player knows precisely how the cards are distributed around the table.

Of the commercial bridge-playing programs now available, none ranks highly as a contender at the tournament level, though several are useful for teaching novices to play. At the research level, Ginsberg, who is a strong bridge player himself, has developed a program called GIB, for Goren in a Box (named after Charles H. Goren, a prominent bridge expert and instructor). "It's the first expert-level computer bridge player," Ginsberg asserts.

To overcome the limitation imposed by incomplete information about card distribution, Ginsberg has programmed GIB to simulate play by dealing out a large number of potential hands for the other players, none of them containing the cards it holds. GIB then selects the playing strategy that works best on average.

"GIB can analyze a bridge hand in



about a second and a half," Ginsberg says. "In a way, the simulations stand in for judgment. I've shown that you can effectively bring raw computational power to bear in the game."

The program is already a member of the American Contract Bridge League. In July, it played in its first serious tournament, and despite the glitches that inevitably bedevil a freshly minted computer program still under development, it made a respectable showing and earned master points.

In the realm of games, Go presents a particularly tough challenge to software developers. Usually played on a 19-by-19 grid, the game is deceptively simple. Two players alternate in placing black and white stones on the grid's intersection points, each with the goal of capturing more territory and taking more prisoners than the other.

Of the computer programs participating in the Hall of Champions, the one that plays Go is farthest from the championship level. This program, Handtalk, developed by Zhixing Chen of ZhongShan University in Guangzhou, China, is perhaps the strongest computer Go player of recent years. Though details about how the program operates are sketchy, it appears to mix some pattern matching with a limited search strategy. At this stage, it lags far behind the performance of chess programs.

So the game isn't over yet.

Go remains an unsolved puzzle; computer bridge is still missing a few hands; backgammon programs lack the killer instinct of a champion; and there are moves still to be made even in chess.

"Deep Blue will continue to improve its play," Newborn predicts. "But there's a long way to go before computers play perfect chess."

Chess experts who helped the IBM team identify weaknesses in strategy proposed refinements that contributed significantly to Deep Blue's remarkable level of play against Kasparov in May. "Its performance was truly marvelous," Berliner says. "It played as if it had some goals. Almost certainly, that was done with some mechanism other than depth of search."

Researchers are keenly interested in seeing Deep Blue play more games against Kasparov and other opponents in order to evaluate its performance in greater detail. Kasparov also learns his lessons, and if he plays Deep Blue again, there are sure to be new surprises.

"We've seen tremendous progress, and there have been a lot of scientific surprises along the way," Newborn contends. "The whole field of [artificial intelligence] has a lot to learn from what's happened in computer chess." □

Word play

In Scrabble, players create words from letters selected at random from a stockpile of 100 tiles. The tiles are laid down on a board 15 squares high by 15 squares wide to form an interlocking, crossword arrangement.

At first glance, getting a computer to play Scrabble requires little more than supplying it with a huge dictionary from which it can choose words that incorporate the available letters. Actually, the dictionary isn't nearly as important as knowing the relative value of the different letters in terms of play, says computer programmer Brian Sheppard of Concord, Mass.

"You need a clever move-generation algorithm to do all the searching within the time limit," Sheppard says. "However, high-scoring moves tend to use letters that are valuable, and there's a trade-off between using those letters and saving them for future turns."

The most important aspect of Scrabble is judging which tiles to keep and which to play, he says. For example, a balance of vowels and consonants is good, while having a Q, even with a U, is bad. Sheppard's success in developing a strong Scrabble player hinged on establishing exact values for tiles in different situations. His program, Maven, assigns 25 points to a blank, 7.75 points to S, and negative values to those dreaded letters like Q. The average tile has a value of zero.

In its first major tournament involving some of the top human players in the United States, Maven outscored its opponents by an average of 70 points out of several hundred, winning eight games and losing two. It now ranks as one of the best players in the world.

"Programs play better than their authors, as a rule. Scrabble is a great example of that," Sheppard says. "It's also an example of programs teaching their authors how to play. I had never played any Scrabble at all before I started Maven, and I'm now an expert player."

One thing Sheppard learned was to disregard the pattern developing on the board. "Tile positions hardly matter at all," he says. One of the few exceptions comes up when tiles can be placed on certain squares to triple word scores at a crucial moment in the game.

Maven is available as a commercial product (under the Scrabble brand name), and Sheppard is now working on an expert backgammon player.

—*I.P.*