More than just competing with people, game-playing machines complement human thinking by offering alternative methods to solving problems

Computers, Games and the Real World

by Matthew L. Ginsberg

The world watched with considerable amazement in May 1997 as IBM's chess computer, Deep Blue, beat Garry Kasparov, the world champion, in a six-game match. With a machine's victory in this most cerebral of games, it seemed that a line had been crossed, that our measurements of ourselves might need tailoring.

The truth of who ultimately won and who lost, of course, is not so black-and-white. Kasparov played poorly, resigning a game that would have led to a draw early in the match and making a completely uncharacteristic error in the last game. And while chess-playing computers have been gaining an edge on their human competitors for some time, in many other games, such as Go and bridge, computer players remain relatively weak. Still, in checkers and Othello, machines have been the world's strongest players for years. Backgammon, like chess, is currently too close to call, whereas machines have a slight but definite edge in Scrabble.

The board and card games that researchers build programs to play provide an environment with specific rules and objective outcomes, a closed system allowing theories to be tested and achievements to be tracked. As an element of artificial intelligence, game-playing software highlights the key differences between the brute-force calculation of machines and the often intuitive, pattern-matching abilities of humans.

Considering Moves

People have been designing machines and programs to play games almost as long as computer code has been written. In 1946 British mathematician Alan M. Turing began designing a chess player on a souped-up code-breaking machine used by Britain during World War II. The computer became the first machine to play a full game of chess, albeit at an extremely slow beginner level. Turing and his colleagues at the University of Manchester later went on to tackle the more basic programming challenges of tic-tac-toe and checkers.

Since then, many people have designed different types of programs and computers with varying degrees of success. The most recent and best game programs [*see box on pages 86 and 87*] are interesting because they generally play their games quite well. But perhaps more interesting is that they achieve this level of performance by using techniques very different from those used by their human counterparts. In games such as chess, a human player will consider some tens (or perhaps hundreds) of positions when selecting a move. A machine, on the other hand, will consider billions, searching through many possible lines of play in the process of selecting an action.

Considering the far greater computational capacity of computers, how is it that humans can win at all? The answer is that although we look at a relative handful of successor positions, we look at the right handful. Emanuel Lasker, world chess champion from 1894 to 1920, was once asked how many moves he considered when analyzing a chess position. "Only one," he replied. "But it's always the best move."

We identify the best positions to consider by a process known as pattern matching. It is how we immediately identify the four-legged platform in a room as a table or the images in a police mug shot as those of the same person despite the different orientations of the front and profile views. An expert chess player compares a given chess position to the tremendous number of such positions that he has seen over the course of his career, using lessons learned from analyzing those other posi-

LOUISE AND WALTER ANNENBERG COLLECTION, PHILADELPHIA MUSEUM OF ART

Portrait of Chess Players, **by Marcel Duchamp**

JONATHAN ELDERFIELD *Gamma Liaison*

MAN AGAINST MACHINE: Although the IBM computer Deep Blue officially beat world champion Garry Kasparov (two wins, one loss, three ties), it is not entirely clear whether the computer would have won had Kasparov played in his usual top form.

tions to identify good moves in the current game.

Pattern matching is a parallel process; if you had *n* computers, you could do it *n* times faster. For example, suppose you are trying to compare a particular chess position to the 100,000 positions that you have seen previously. Each comparison is in some sense independent: given 100,000 computers, you could do the overall comparison 100,000 times more quickly.

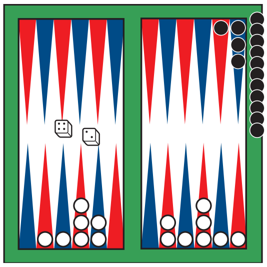
Searching is not an inherently parallel process; problems cannot be split up, because what you want to do next depends on what you just did. For Deep Blue, there is no preexisting list of positions that it evaluates. Rather it can examine a position only after it has constructed its predecessor.

It is important to note that when I say that pattern matching is parallel, whereas brute-force searching is not, I am referring not to the problem being solved but only to the method being used to solve it. I have not said that playing a good game of chess is a parallel problem or that it is not. In fact, the evidence in some sense is that chess is not inherently parallel or serial, because parallel techniques (human pattern matching) and serial ones (Deep Blue's brute-force searching) can be applied equally well to the game.

Function Follows Form

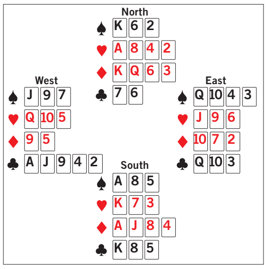
There is a good reason that Kasparov and other human players use a parallel technique to play chess: the human brain appears to be a parallel machine, consisting of about 100 billion neurons, each capable of operating 1,000 times a second.

Games Computers Play



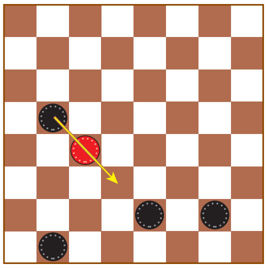
BACKGAMMON

- 2 players
- 15 pieces each
- Goal: Move all pieces off the board
- Rules:
  - Dice roll determines number of moves
  - Players move in opposite directions
  - Piece cannot land on a point occupied by 2 or more of opponent's pieces
  - Single piece can be "hit" if landed on by opponent; hit piece must start anew
- Program: TD-Gammon*
- Web site: www.research.ibm.com/massdist/tdl.html
- Advantage: Too close to call



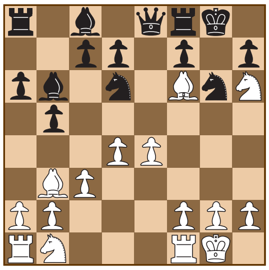
BRIDGE

- 4 players in 2 teams
- 13 cards dealt to each player
- Goal: Make 2 "game contracts," or a "rubber"
- Rules:
  - The bid: Each player predicts how many times his or her card will be the highest (a trick)
  - The play: Put down 1 card at a time and compare it with others; this occurs 13 times
  - The scoring: Points scored if bid is made or exceeded; otherwise points go to the opposing team
- Program: GIB*
- Web site: www.gibware.com
- Advantage: Human



CHECKERS

- 2 players
- 12 pieces each
- Goal: Avoid being the player who can no longer move (usually when a player has no pieces left)
- Rules:
  - Move forward on dark diagonal, 1 square at a time
  - Opponent's piece captured when jumped to empty square diagonally behind opponent's piece
  - Creation of a "king," a piece that can move backward and forward, occurs when piece is moved to opponent's last row
- Program: Chinook
- Web site: www.cs.ualberta.ca/~chinook
- Advantage: Machine



CHESS

- 2 players
- 16 pieces each (1 king, 1 queen, 2 rooks, 2 bishops, 2 knights, 8 pawns)
- Goal: Capture opponent's king (checkmate)
- Rules:
  - Pieces are captured when landed on by opponent's piece
  - Type of piece dictates movement options
- Program: Deep Blue
- Web site: www.chess.ibm.com/meet/html/d.3.html
- Advantage: Too close to call

*Indicates commercial software that runs on personal computers

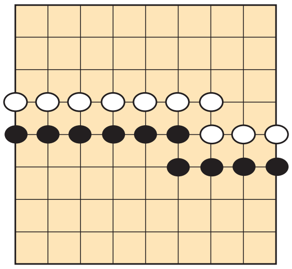
About 30 billion neurons are laid out in six layers of the cortex, the gray matter ("thinking" neurons) making up the outer folds of the human brain. An additional 70 billion constitute the white matter ("connecting" neurons). This massively parallel configuration is good at recognizing patterns but is challenged by serial calculations, such as searching.

In contrast, Deep Blue contains only 480 chess-specific processors, each of which is capable of examining about two million chess positions a second. This setup enables it to search many positions in very little time. Although this search-intensive approach provides a certain advantage, it has limitations. Even with the most powerful computer imaginable—say, one performing perhaps 10^{17} operations per second (one operation in the amount of time it takes light to traverse the space of a hydrogen atom)—you still would not be able to make a dent in solving a game such as Go, for which there are some 10^{170} possible positions.

When humans play a game that depends on a brute-force search through an enormous number of positions and strategies, such as Othello, we cannot use methods that depend on the possibility of executing a billion instructions a second; the neurons in our brain fire at a millionth of that rate. Because our brains operate so slowly, serial methods to solve problems are generally ineffective for us.

Of course, 1,000 operations per second may be relatively slow, but humans can still use the serial method in a pinch. We use serial methods when we multiply large numbers, for instance. We also commonly use them to solve puzzles, such as Rubik's cube, and brainteasers, such as the old problem of figuring out how to get three missionaries and three cannibals across a river using a single rowboat that can hold two people, subject to the condition that the cannibals cannot outnumber the missionaries on either bank because they will eat them. (In the more modern version of the problem, the missionaries cannot outnumber the cannibals because they will convert them.) Although we are capable of applying reasoning to solve brainteasers, the reasoning itself seems very unnatural because we typically give up on parallel methods and instead search through the possible combinations. For artificial intelligence, this method is referred to as "puzzle mode" reasoning. Machines are good at it, because their hardware is designed for it, but humans are not.

And unlike the human brain, which is stuck where it is, computers can improve their game-solving ability with faster hardware and more efficient programs. In most game-playing programs, a computer is programmed to analyze only those moves close to the current position, to avoid a massively deep search. Its move options can be assigned values using a procedure known as minimaxing. An additional technique, called alpha-beta pruning, allows the computer to compute these

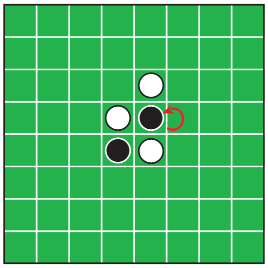


GO

- 2 players
- Black-and-white stones
- Grid size of board can vary: typical game is on 19-by-19 grid points
- Goal: Conquer a larger part of the board (conquered part encompasses stones placed on board plus stones that could be added safely—that is, within the player's walls)
- Rules:

    Both sides alternate in placing stones on the board

    Stones surrounded by an opponent's stones are captured and removed from the board
- Program: Handtalk*
- Web site: www.webwind.com/go
- Advantage: Human, by a huge margin

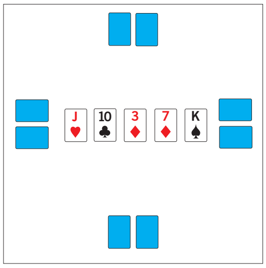


OTHELLO

- 2 players
- Black-and-white disks
- Goal: Have most disks on the board at the end of the game
- Rules:

    Players alternate placing disks on unoccupied board spaces

    If opponent's disks are trapped between other player's disks, opponent's disks are flipped to the other player's color
- Program: Logistello
- Web site: www.neci.nj.nec.com/homepages/mic/log.html
- Advantage: Machine



POKER (Texas Hold 'Em)

- 3 to 20 players
- 2 cards dealt to each player; 5 cards placed in center of table
- Goal: Obtain the best hand and win the "pot"
- Rules:

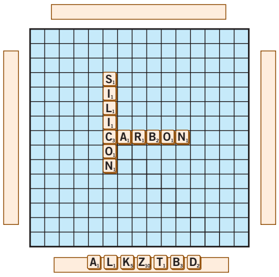
    5 center (community) cards start facedown

    First round of betting ensues; 3 community cards are turned over

    Subsequent rounds of betting ensue; 4th and 5th community cards turned over

    Players select best 5 from the community cards and their hands to obtain identical kinds of cards (pairs, 3- and 4-of-a-kind), flushes (all same suit), straights (sequential) or their combinations

    Final round of betting ensues
- Program: LOKI
- Web site: www.cs.ualberta.ca/~games/poker
- Advantage: Human, by a huge margin



SCRABBLE

- 2 to 4 players
- 100 tiled letters
- Goal: Accumulate most points by creating high-scoring words
- Rules:

    Each player draws 7 letters

    Each letter has a value

    Squares on the board have values

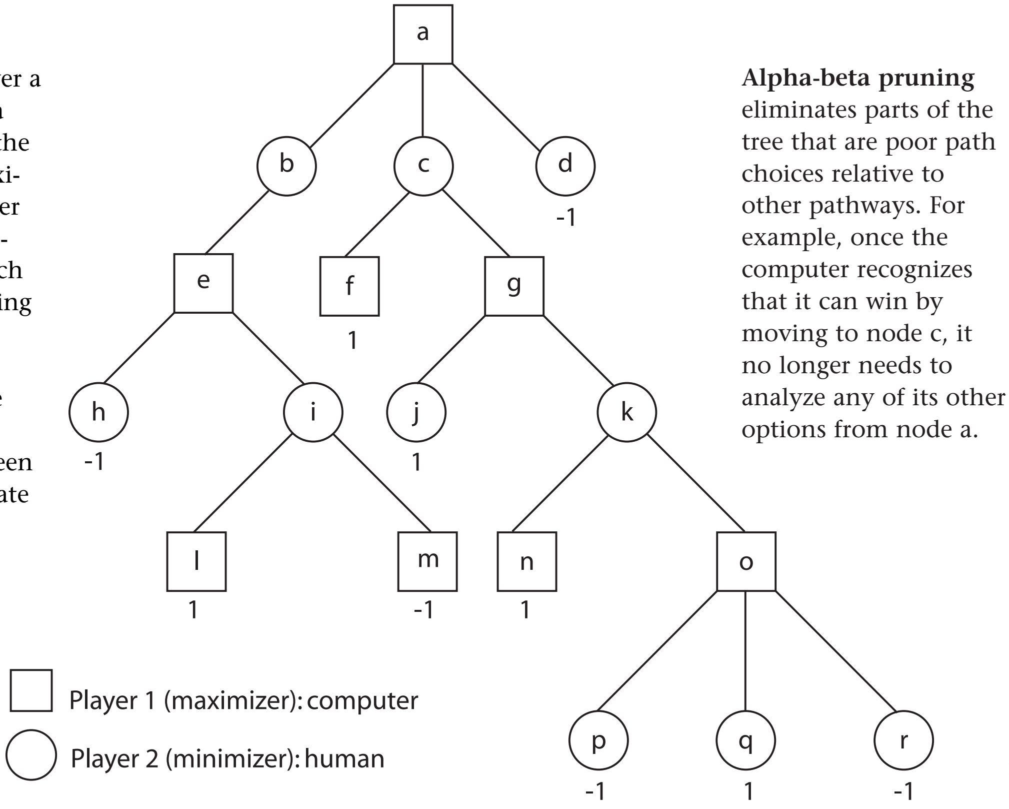
    Words created must join an array
- Program: Maven* (used in Scrabble CD-ROM)
- Web site: www.hasbroscrabble.com/cd/cd.html
- Advantage: Machine, by a slight margin

*Indicates commercial software that runs on personal computers

JOHN MENGEL *Ponzi & Weill*

How Game-Playing Computers Think

Minimaxing assigns one player a positive value and the other a negative value. (In this case, the computer is player 1, the maximizer, and the human is player 2, the minimizer.) Once established, the values assigned each node can be judged as benefiting the maximizer (the "positive" player) or the minimizer (the "negative" player). Each node represents the positions of all the pieces after a move has been made—more generally, the state of the game at the moment.


a
b
c
d
-1
e
f
g
1
h
-1
i
j
1
k
l
1
m
-1
n
1
o
p
-1
q
1
r
-1
□ Player 1 (maximizer): computer
○ Player 2 (minimizer): human

Alpha-beta pruning eliminates parts of the tree that are poor path choices relative to other pathways. For example, once the computer recognizes that it can win by moving to node c, it no longer needs to analyze any of its other options from node a.

MEG CRANE

values without examining every imaginable possibility. This strategy enables the computer to perform a more effective search given its fixed computational resources [*see illustration above*].

There have been attempts to produce chess computers that play the game in a more humanlike way. But the performance of these systems has been modest at best. Nobel laureate Herbert A. Simon of Carnegie Mellon University believes that the basic differences in architecture may prevent efficient humanlike reasoning in computers: "It could be that because of the radical differences between electronic devices and brains, programs designed to be efficient [intelligent programs] would be totally different in architecture and process from systems designed to simulate human thinking." Thus, the hardware may make what we consider efficient reasoning not so efficient when run on a silicon-based system.

Prowess Follows Process

In summary, some problems are best solved using pattern matching, whereas others are best solved using serial search. Go seems to be a fundamentally parallel challenge; Othello seems to be serial. Other games, such as chess, seem equally amenable to both methods.

This division also exists for games of imperfect information, where the players do not know what cards or other assets are held by their opponents. (A game of perfect information provides all players with complete data about the state of the game; backgammon, chess, checkers and Othello are examples.) Imperfect-information games have historically eluded competent computer play. In these games, such as poker and bridge, humans often rely on experience and intuition to play well. Computers are at a disadvantage; there is just too much information to process. But recently massive computational resources have brought games of imperfect information closer to being solved. Daphne Koller and Avi Pfeffer of Stanford University suggest that it will be possible to "solve poker." They base this conclusion on successful algorithms they designed to play perfectly using an eight-card deck. The question now is if enough computational power will ever be available to apply the same approach to a 52-card deck.

In bridge, too, the power of exhaustive methods is beginning to be felt. Earlier programs modeled human thinking, but a modern approach works as follows. The opponents are repeatedly given specific random hands, and the result is then analyzed assuming that the locations of all the cards are known. This approach identifies the best play or plays in one particular situation, and the best play overall is the one that is best in as many of the random situations as possible. Once again, this style of analysis is possible primarily because of increasing hardware power. Although the best humans still outplay the best programs, the gap is narrowing. As in chess, programs that attempt to model human thinking are no match for their search-based competitors.

In all these cases, Simon's speculation on the innate importance of architecture has proved correct. The strongest machine players do indeed bear little resemblance to their human counterparts. The only game that seems to be an exception is backgammon. In this case, Gerald J. Tesauro of IBM created a program called TD-Gammon that appears, at least on the surface, to have a humanlike architecture. Specifically, it relies on an artificial neural network, which is software designed to mimic the function and structure of neurons.

Games People (and Computers) Play

The nature of the game and the type of solution processing most conducive to success often dictate whether human or machine plays better. Machines excel at checkers, a game that involves deep-search methods to evaluate possible combinations. Chinook, currently the best checkers program, knows immediately which player (if either) will win once there are eight or fewer checkers on the board. It determines the outcome by accessing its enormous database of endgames. This task is not a pattern-matching problem, because there are no reliable patterns that characterize all the positions; instead Chinook simply stores a huge table of the positions and values and looks in that table when a result is needed.

Machines also have a tremendous advantage in Othello because it is a difficult game for humans to get a feel for; it is almost impossible to tell at a glance who is winning at any particular point. The markers flip back and forth so frequently that having a preponderance of markers with your color in the middle of the game does not necessarily indicate that you are winning. In other words, Othello is a game that is difficult to play using pattern matching but is perfectly suited to a machine's brute-force search methods. Therefore, machines excel.

Go, however, is the computer's Achilles' heel. It is a pattern-matching game in which human experts can generally recognize large configurations of stones that are dead (surrounded and fated to eventual capture), alive (permanently safe from capture) or whose fate is currently undetermined. Playing Go using brute-force search, however, is extremely difficult, because at any given point there will be some 250 legal moves. (In comparison, there are only approximately 30 legal moves in chess and seven in Othello.) Building an endgame database such as Chinook's is completely out of the question because the number of possible ending positions in a Go game is beyond a computer's computational power.

And other games? The best in backgammon is a close call between human and computer and is very likely to remain so: because of the relative simplicity of the game, both people and machines play backgammon nearly perfectly. Scrabble is also close but not very likely to remain so, as machines become better at the strategic parts of the game. (Both humans and machines have no difficulty playing the highest scoring move at any point, but high-level Scrabble is also a matter of keeping a good balance of tiles in your rack and minimizing your opponent's opportunities.) Humans won a two-game match in 1997 but lost an 11-game match in 1998. Bridge is likely to become close in about five years, as hardware becomes faster and algorithms improve. Humans narrowly won a two-hour match this past July, and at the 1998 World Bridge Championships, held in August in Lille, France, the bridge program I wrote placed 12th in a field of 34 of the top human bridge players.

Playing Nicely Together

As artificial intelligence has developed, success has come most often through the application of good serial algorithms. These successes are not limited to game playing: serial algorithms are the most effective known solutions for selecting the order in which to assemble the component parts of a Boeing 747. These algorithms lead to production schedules some 10 to 20 percent shorter than the best schedules produced by humans. Pattern matching and other parallel techniques are not terribly well suited to scheduling complex tasks, because a schedule that looks good may in fact not be.

At least for the foreseeable future, humans will be better than machines at solving parallel problems, and machines will be better at solving serial ones. There should be nothing particularly surprising or threatening here; we have designed machines to achieve other results that we ourselves are not capable of achieving, such as airplanes to fly us or forklifts to raise objects our muscles cannot move. And we do not feel compelled to pit humans against forklifts in Olympic weight lifting.

The lesson is that intelligent machines are not our competitors but our collaborators. The complementary skills of humans and machines enable problems to be solved that neither could have figured out alone. And that is exceptionally good news for us all, carbon and silicon alike.

About the Author

ROBERT KAISER

MATTHEW L. GINSBERG is a senior research associate at the University of Oregon and founder of the university's Computational Intelligence Research Laboratory (CIRL). He received his doctorate in mathematics from the University of Oxford in 1980, where he also captained the bridge team. Although he currently plays little competitive bridge, he has remained in the game via his alter ego, an expert-level bridge-playing program called GIB (Ginsberg's Intelligent Bridge player). "On a personal level, GIB has been a blast," Ginsberg says. "I can interact with top players, but no one is angry when the computer beats them." Pending software adjustments, Ginsberg predicts that GIB will become the world's top bridge player in five years. When not conducting research, supervising graduate work at CIRL or tinkering with GIB, Ginsberg likes to relax with a few loops and rolls in his kit-built stunt plane. Lately he has cut down on that thrill to once a week, choosing instead to spend time with his wife, his infant daughter and his four-year-old son, who has just learned to play chess.