and fifth losses, but won the match 20.5-18.5. A rematch at the 1994 world championship ended prematurely when Tinsley had to withdraw for health reasons. CHINOOK became the official world champion. Schaeffer kept on building on his database of endgames, and in 2007 "solved" checkers (Schaeffer et al., 2007; Schaeffer, 2008). This had been predicted by Richard Bellman (1965). In the paper that introduced the dynamic programming approach to retrograde analysis, he wrote, "In checkers, the number of possible moves in any given situation is so small that we can confidently expect a complete digital computer solution to the problem of optimal play in this game." Bellman did not, however, fully appreciate the size of the checkers game tree. There are about 500 quadrillion positions. After 18 years of computation on a cluster of 50 or more machines, Jonathan Schaeffer’s team completed an endgame table for all checkers positions with 10 or fewer pieces! over 39 trillion entries. From there, they were able to do forward alpha—beta search to derive a policy that proves dial checkers is in fact a draw with best play by both sides. Note that this is an application of bidirectional search (Section 3.4.6). Building an endgame table for all of checkers would be impractical: it would require a billion gigabytes of storage. Searching without any table would also be impractical: the search tree has about $8^{47}$ positions, and would take thousands of years to search with today’s technology. Only a combination of clever search, endgame data, and a drop in the price of processors and memory could solve checkers. Thus, checkers joins Qubic (Patashnik, 1980), Connect Four (Allis, 1988), and Nine-Men’s Morris (Gasser, 1998) as games that have been solved by computer analysis.

Backgammon, a game of chance, was analyzed mathematically by Gerolamo Cardano (1653), but only taken up for computer play in the late 1970s, first with the BKG program (Berliner, 1980b); it used a complex, manually constructed evaluation function and searched only to depth 1. It was the first program to defeat a human world champion at a major classic game (Berliner, 1980a). Berliner readily acknowledged that BKG was very lucky with the dice. Gerry Tesauro’s (1995) TD-GAMMON played consistently at world champion level. The BGBLITZ program was the winner of the 2008 Computer Olympiad.

Go is a deterministic game, but the large branching factor makes it challenging. The key issues and early literature in computer Go are summarized by Boozy and Cazenave (2001) and Muller (2002). Up to 1997 there were no competent Go programs. Now the best programs play most of their moves at the master level; the only problem is that over the course of a game they usually make at least one serious blunder that allows a strong opponent to win. Whereas alpha—beta search reigns in most games, many recent Go programs have adopted Monte Carlo methods based on the UCT (upper confidence bounds on trees) scheme (Kocsis and Szepesvari, 2006). The strongest Go program as of 2009 is Golly and Silver’s MoGo (Wang and Golly, 2007; Gelly and Silver, 2008). In August 2008, MoGo scored a surprising win against top professional Myungwan Kim, albeit with MoGo receiving a handicap of nine stones (about the equivalent of a queen handicap in chess). Kim estimated MoGo’s strength at 2-3 dan, the low end of advanced amateur. For this match, MoGo was run on an 800-processor 15 teraflop supercomputer (1000 times Deep Blue). A few weeks later, MoGo, with only a five-stone handicap, won against a 6-dan professional. In the 9 x 9 form of Go, MoGo is at approximately the 1-dan professional level. Rapid advances are likely as experimentation continues with new forms of Monte Carlo search. The Computer Go
Newsletter, published by the Computer Go Association, describes current developments.

Bridge: Smith et al. (1998) report on how their planning-based program won the 1998 computer bridge championship, and (Ginsberg, 2001) describes how his GIB program, based on Monte Carlo simulation, won the following computer championship and did surprisingly well against human players and standard book problem sets. From 2001-2007, the computer bridge championship was won five times by JACK and twice by WBRIDGE5. Neither has had academic articles explaining their structure, but both are rumored to use the Monte Carlo technique, which was first proposed for bridge by Levy (1989).

Scrabble: A good description of a top program, MAVEN, is given by its creator, Brian Sheppard (2002). Generating the highest-scoring move is described by Gordon (1994), and modeling opponents is covered by Richards and Amir (2007).

Soccer (Kitano et al., 1997b; Visser et al., 2008) and billiards (Lam and Greenspan, 2008; Archibald et al., 2009) and other stochastic games with a continuous space of actions are beginning to attract attention in AI, both in simulation and with physical robot players.

Computer game competitions occur annually, and papers appear in a variety of venues. The rather misleadingly named conference proceedings Heuristic Programming in Artificial Intelligence report on the Computer Olympiads, which include a wide variety of games. The General Game Competition (Love et al., 2006) tests programs that must learn to play an unknown game given only a logical description of the rules of the game. There are also several edited collections of important papers on game-playing research (Levy, 1988a, 1988b; Marsland and Schaeffer, 1990). The International Computer Chess Association (ICCA), founded in 1977, publishes the ICGA Journal (formerly the ICCA Journal). Important papers have been published in the serial anthology Advances in Computer Chess, starting with Clarke (1977). Volume 134 of the journal Artificial Intelligence (2002) contains descriptions of state-of-the-art programs for chess, Othello, Hex, shogi, Go, backgammon, poker, Scrabble, and other games. Since 1998, a biennial Computers and Games conference has been held.

EXERCISES

5.1 Suppose you have an oracle, O_M(s), that correctly predicts the opponent’s move in any state. Using this, formulate the definition of a game as a (single-agent) search problem. Describe an algorithm for finding the optimal move.

5.2 Consider the problem of solving two 8-puzzles.
   a. Give a complete problem formulation in the style of Chapter 3.
   b. How large is the reachable state space? Give an exact numerical expression.
   c. Suppose we make the problem adversarial as follows: the two players take turns moving; a coin is flipped to determine the puzzle on which to make a move in that turn; and the winner is the first to solve one puzzle. Which algorithm can be used to choose a move in this setting?
   d. Give an informal proof that someone will eventually win if both play perfectly.