Artificial Intelligence 134 (2002) 277–311

Artificial Intelligence

www.elsevier.com/locate/artint

Games solved: Now and in the future

H. Jaap van den Herik [a], Jos W.H.M. Uiterwijk [a,*], Jack van Rijswijck [b]

[a] *Institute for Knowledge and Agent Technology (IKAT), Department of Computer Science, Universiteit Maastricht, Maastricht, The Netherlands*

[b] *Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada*

Abstract

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.

Keywords: Solving games; Search methods; Game characteristics; Brute-force methods; Knowledge-based methods; Initiative

1. Introduction

The domain of strategic games is an intriguing one. Already for half a century building strong game-playing programs has been an important goal for Artificial Intelligence researchers [57,58,60,74]. The principal aim is to witness the "intelligence" of computers. A second aim has been to establish the *game-theoretic value* of a game, i.e., the outcome when all participants play optimally. This value indicates whether a game is won or lost (or

* Corresponding author.

E-mail addresses: herik@cs.unimaas.nl (H.J. van den Herik), uiterwijk@cs.unimaas.nl (J.W.H.M. Uiterwijk), javhar@cs.ualberta.ca (J. van Rijswijck).

0004-3702/01/$ – see front matter
PII: S0004-3702(01)00152-7

drawn if that possibility exists) from the perspective of the player who has the first move. In this article we give an overview of results accomplished thus far.

1.1. Conventions

We have restricted ourselves to two-person zero-sum games with perfect information [83]. In this class of games we deal with connection games, mancala games, endgame problems of chess-like games, and a subclass of other games, such as Domineering and Othello. Thus all one-player games such as the 15-puzzle and Rubik's Cube, however interesting, are omitted, as are the games involving racing (Halma), chance (Backgammon), uncertainty (Bridge), imperfect information (Stratego), or negotiation (Monopoly). Moreover, we only take into account games where the computer plays a role in obtaining the solution. This excludes games which can be solved solely by mathematics, such as Nim; many of such mathematical games, either solved or not, are described elsewhere [12,13]. Finally, no trivial games are taken into consideration, although some of them might be described concisely as "little-brother" versions of more complex games, such as Tic-Tac-Toe being a simplified version of Go-Moku.

When discussing the solution of games, at least three different definitions of a solution can be used. We use the terminology proposed by Allis [7]. Here *ultra-weakly solved* means that the game-theoretic value of the initial position has been determined, *weakly solved* means that for the initial position a strategy has been determined to achieve the game-theoretic value against any opposition, and *strongly solved* is being used for a game for which such a strategy has been determined for all legal positions.

In this article we adopt the following conventions. All names of games are capitalized, e.g., Awari, Chess, and Go-Moku, as are the names of the players: Black and White. Lowercase letters are used when referring to a group of games, such as mancala games. We remark that different games have different first players. For instance, in Chess the first player is White, and in Go it is Black. Hence we refrain from identifying either White or Black to be the first player. If we use White or Black, it is explicitly stated whether it is the first or second player. Finally, we assume that the readers are acquainted with the rules of the games. Many of them have been played in one of the five Computer Olympiads held so far (1989 [69], 1990 [70], 1991 [56], 1992 [17], and 2000 [59]).

1.2. Classification by complexity

In 1990 a prediction for the year 2000 was made concerning the expected playing strength of computer programs for the games played at the first two Computer Olympiads [4]. This prediction is reproduced in Table 1.

Of the predictions then made, in particular concerning games solved, we confirm below that three of the four games listed have been solved. Connect-Four is not included in this count, since it was already solved at the time of the prediction. Qubic is included although it was then solved too, but the authors were not aware of its solution at that time. Only Awari escaped solution so far, though it seems probable that its solution is a matter of a few years at most [48,71]. We thus can legitimately say that the predictions then were rather accurate.

Table 1
Predicted program strengths for the Computer Olympiad games in the year 2000

Solved or cracked	Over champion	World champion	Grand master	Amateur
Connect-Four	Checkers (8 × 8)	Chess	Go (9 × 9)	Go (19 × 19)
Qubic	Renju	Draughts (10 × 10)	Chinese chess	
Nine Men's Morris	Othello		Bridge	
Go-Moku	Scrabble			
Awari	Backgammon			

↑
log log
state-space
complexity

Category 3 if solvable at all, then by knowledge-based methods	**Category 4** unsolvable by any method
Category 1 solvable by any method	**Category 2** if solvable at all, then by brute-force methods

log log game-tree complexity →

Fig. 1. A double dichotomy of the game space.

In addition to the predictions, the same paper also introduced a characterisation of games with respect to their position in the game space. It was defined as a two-dimensional space with the state-space (there called search-space) complexity as one dimension and the decision complexity as the second dimension. The notion of decision complexity was rather vague; later it was replaced by the notion of game-tree complexity [7], which can be seen as a good indication of a game's decision complexity. A dichotomy applied to each dimension roughly yields four categories according to the games' state-space complexity being low or high and the games' game-tree complexity being low or high. This categorisation is given in Fig. 1. To dampen the exponential nature of both complexities, the measurements along the axes are given in log log format.

It was easy to predict that the games of category 1 were most suitable to being solved. Further it was postulated that the games in categories 2 and 3 were possibly solvable, though with different types of methods, i.e., with brute-force methods for category 2 and knowledge-based methods for category 3. Finally, category 4 contained games assumed to be unsolvable in practice.

In this article we investigate to what extent the game characteristics are most appropriate for solving games. In particular we examine whether a low state-space complexity (category 2) or a low game-tree complexity (category 3) of a game is most promising for solving the game in question. Clearly, solving category-2 games is dependent on the increase in computer power, and solving category-3 games on the development of new AI techniques.

Below we explore the nature of the methods used in solving games. In particular we check whether the category-2 games were indeed mainly solved by brute-force methods and the category-3 games by knowledge-based methods. Thereafter, we consider the role of the *initiative* by a player as a characteristic in a game [117]. Can it be seen as a determining factor in the efforts needed to solve a game?

The most important AI techniques that enable us to solve a game are listed and discussed. It is argued that research in the field of solving games has resulted in the development of many new techniques applicable elsewhere in the field of Artificial Intelligence. Nevertheless numerous worthwhile research questions of a general nature remain, especially in the exceptional cases in which knowledge can be justifiably said to be perfect. Below we pose three questions taken from [32], and return to them at the end of this article.

(1) Can perfect knowledge obtained from solved games be translated into rules and strategies which human beings can assimilate?
(2) Are such rules generic, or do they constitute a multitude of ad hoc recipes?
(3) Can methods be transferred between games? More specifically, are there generic methods for all category-n games ($n = 1, 2, 3, 4$), or is each game in a specific category a law unto itself?

1.3. Classification by game type

An important aspect of any two-player game is *convergence*. We consider a game to be convergent when the size of the state space decreases as the game progresses. If the size of the state space increases, the game is said to be *divergent*. Informally, convergent games start with many pieces on the board and pieces are gradually removed during the course of the game, while divergent games start with an empty or almost empty board and pieces are added during the game.

The course of the article is as follows. In Sections 2 and 3 we discuss results for convergent and divergent games, respectively. Within each section, the games are presented roughly in order of difficulty for ultra-weakly solving them. The game characteristics investigated as determining factors for solving games are reviewed in Section 4. In Section 5 we go into some detail concerning methods developed during game-solving research. Finally, Section 6 deals with the three research questions above, contains our conclusions and makes some predictions on future developments in the domain of games solved.

2. Convergent games

In this section we deal with convergent games. Unlike divergent games, convergent games by their definition allow for the construction of *endgame databases*. These databases are a powerful tool for solving the games by retrograde analysis, starting at the goal states and working backward. Section 2.1 covers the game of Nine Men's Morris, which is unusual in that it first diverges, but then converges. Section 2.2 discusses the mancala family of games, with results for Awari, Kalah, and Dakon. The final two games in this section, Checkers in Section 2.3 and Chess in Section 2.4, have so far only seen solutions for particular endgames.

2.1. Nine Men's Morris

Nine Men's Morris is a game that is at least 3000 years old [11]. It is played on a horizontal board with a grid as depicted in Fig. 2. Initially the board is empty. Two players, White and Black, each have 9 stones in hand. White starts. The game consists of three phases. In the first phase (opening) players alternately place a stone on a vacant point of the board. This phase necessarily takes 18 turns. A *mill* is formed by three stones of the same colour in a row. When a player during any phase of the game obtains a mill, an opponent's stone not in a mill may be removed. If all opponent stones are part of a mill, any stone may be removed. When two mills are formed simultaneously, only one opponent stone may be removed. After 18 turns the second phase (middle game) starts, in which a player may move a stone to an adjacent vacant point on the board. When a player only has three stones left, the third phase (endgame) starts. The player in question then may jump with a stone to any vacant square. A player to move with fewer than three stones left or with no more legal moves loses the game. If a position is repeated, the game ends in a draw.

Gasser solved the game in 1995 [44,45]. This was accomplished by combining two techniques: building databases and performing a database-guided search. First, he built all 28 w-b middle-game and endgame databases, with w and b being the number of white and black stones on the board. The numbers w and b necessarily have values between 3 and 9, both inclusive. When unreachable positions and symmetrical positions are taken into account, the total number of states is 7,673,759,269. The databases were independently checked. Second, an 18-ply search from the start position using the databases was performed. Exploiting the 9-9, 9-8, and 8-8 databases it was established that the game-theoretic value of the game is a draw.

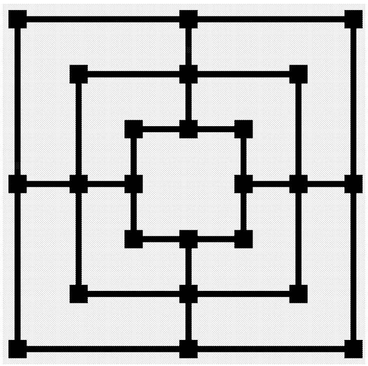

Fig. 2. The Nine-Men's-Morris board.

2.2. *Mancala games*

There are literally hundreds of mancala games [92]. They all use a playing board with a number of holes, commonly called *pits*. The pits are distributed over two or more rows; in addition there are zero, one, or two large holes, called *stores*. The number of *stones* used is variable. Traditionally, easily obtainable objects are used for the stones, such as pebbles, shells, or seeds. Usually the game starts with an equal number of stones in each pit, and with no stones in the stores, if any. The mancala games have in common that a turn consists of (1) taking all stones out of a pit and (2) distributing (*sowing*) them one after another over the subsequent pits. However, there is a wide variety of differences as is evident from the number of pits, the number of stones per pit, the sowing direction, the inclusion of the stores, the inclusion of the starting pit, and the continuation after a sowing has finished (mostly depending on owner and number of stones in the final pit).

Given the large number of rule variations no exhaustive account can be given on solved mancala games in this article. We restrict ourselves to those games which are played by a large community, and which are not known as trivial. Of these Awari is the most well-known game, played worldwide. It is not yet solved, and there is a large interest for this game among computer scientists. Therefore we briefly introduce computer results for Awari, before turning to two other types of mancala games for which solutions have been found: Kalah and Dakon.

2.2.1. *Awari*

The Awari board contains two rows of six pits. The players, traditionally called *North* and *South*, each control one row. The game starts with four stones in each pit. An extra pit for each player is used to collect captured stones, but these are not used in sowing stones. A player to move chooses one of their non-empty pits and starts sowing. Whenever the sowing reaches the start pit again, it is ignored. If the last stone lands in an enemy pit which then contains 2 or 3 stones, the stones are captured. After the capture, the previous pit is examined, and when it too contains 2 or 3 stones, they are captured also, and so on. Capturing only occurs in enemy pits. The game terminates when a player has no moves left or when a player has collected 25 or more stones. The rules force each player to leave the opponent a move whenever possible.

A decade ago the first serious attempts were made to construct endgame databases for Awari. At that time a 13-stone database was constructed, leading to the first strong Awari program LITHIDION [3]. Since then the number of stones for which databases are

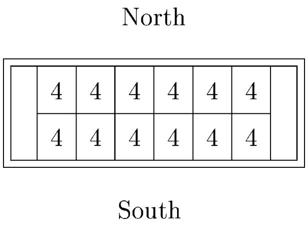

Fig. 3. Initial position of Awari.

constructed rapidly increased and databases up to 35 stones were built [48,71]. At present even databases for up to 38 stones are calculated [72] and the 39-stone database is under construction at the University of Alberta. The 36-stone database shows that the little brother of Awari with three stones per pit in the initial position is a draw. In the 5th Computer Olympiad in 2000, MARVIN and SOFTWARI played with 35-stone databases. The match was very balanced, although MARVIN won the last game [73]. There is a tendency to believe that the game is drawn. Keeping in mind that the standard start position has 48 stones it is clear that fast forward searchers encounter the databases relatively quickly in many lines, but in some lines the captures may be delayed for a large number of moves. Finding the weak solution of Awari depends on how long these captures can be delayed, but is expected within a few years at most [48,71].

2.2.2. Kalah

Kalah is a game quite similar to Awari. It traditionally has six pits per row and starts with 4 stones per pit, as in Awari. In addition, it has two stores, one for each player. Again there are different versions, e.g., with a different number of pits and/or with a different number of stones per pit. A version is denoted by Kalah(m,n), m being the number of pits per side and n the number of stones per pit. The rules clearly differ from Awari. First, when sowing the own store is included, but the opponent's store is ignored. Second, if the last stone lands in the player's own store, the player moves again. Third, if the last stone lands in an own empty pit, the last stone plus the stones in the opposite hole, if any, are captured and put in the player's store, which ends the move. In any other case, the move is over without any side effects. As in Awari, if a player cannot move, lacking stones at the player's side of the board, the game is over and all remaining stones are captured by the opponent. The player with the most stones wins the game. These differences in the game rules between Awari and Kalah result in smaller complexity measures for Kalah than for Awari for the same configurations. Recently all game-theoretic values for Kalah up to six pits per player and up to five stones per pit have been determined, using a combination of endgame databases and search [62]. It means that these games are weakly solved. The results are listed in Table 2, where 'W' indicates a first-player win, 'D' a draw, and 'L' a first-player loss. There we see that the standard game, Kalah(6,4), is a first-player win.

Table 2
Game values for Kalah with m pits per side and n stones per pit

$m\backslash n$	1	2	3	4	5	6
1	D	L	W	L	W	D
2	W	L	L	L	W	W
3	D	W	W	W	W	L
4	W	W	W	W	W	D
5	D	D	W	W	W	W
6	W	W	W	W	W	

We may conclude that in Kalah having the initiative is a large advantage [117], since 23 out of 35 games are first-player wins (66%). It may be that one needs sufficient space in order to bring the initiative to fruition. From Table 2 we see that for Kalah(m,n) with $m \geqslant 3$ and $n \geqslant 3$ as much as 13 out of 15 games (87%) are first-player wins. Nevertheless, the exceptions Kalah(3,6) and Kalah(4,6) deserve our attention.

2.2.3. Dakon

Like Kalah, the game of Dakon is played on a board with two rows of pits and two stores. The number of stones in each hole normally equals the number of pits per player. This form is indicated by Dakon-n. The rules differ from Kalah in the following rule. If the last stone of a sowing lands in a non-empty pit, all stones of this pit are taken out and sowing continues starting with the next pit. When a player runs out of moves, this denotes the end of a *round*, but not necessarily the end of the *game*. The opponent collects all remaining stones and puts them in the own store. The player having most stones wins the round. Next, both players take out their stones and redistribute them over the holes as follows. From right to left the holes are filled with as many stones as there are pits per side. As soon as a pit cannot be filled, this pit and all remaining pits to the left are left empty and eliminated from future play. The remaining stones are put in the store. The player who has won the last round starts the next round. The game ends when a player has not sufficient stones to fill one hole at the start of a round. The player who won the most rounds wins the game.

The game of Dakon has a special property that might be present in other mancala games too: winning openings. A winning opening for Dakon-n is a sequence of moves that is played in the first turn of the starting player and that captures at least $2n^2 - n$ stones. It is winning because the opponent will have fewer than n stones left and cannot fill a single hole in the next round, which means that the game is over in a single move. Of course, all moves of a winning opening but the last one must end in the player's store. Recently it was reported that a winning opening for Dakon-8 was found "by hand" by players of the Maldives [33]. The winning opening is: 1, 8, 8, 6, 4, 6, 2, 3, 7, 4, 5, 1, 8, 8, 3, 8, 4, 8, 7, 8, 5, 2, 7, 8, 6, 5, 6, 3, 7, 4, 5, 2, 5, 8, 8, 6, 8, 3, 8, 5, 8, 7, 4, 8, 7, 8, 7, 8, 8, 6, 8, 7, 8, 4, 8, 6, 8, 3, 8, 6, 8, 5, 8, 6, 1, 8, 7, 8, 5, 8, 4, 6, 7, 8, 8, 5, 6, 8, 3, 8, 1, 8, 7, 8, 2, 8, 6, 8, 5, 8, 6, 8, 3. The numbers indicate the holes that have to be selected at every move. Holes are numbered from 1 to 8 from left to right. This sequence consists of 93 moves and in total 168 *laps* (a lap is a complete tour around the board). After the complete sequence there are only three stones left in the opponent's holes.

With the help of a computer program, it was then investigated whether winning openings exist in other Dakon-n instances. Except for Dakon-3, for which the existence of a winning opening can easily be proved to be impossible, many winning openings for Dakon versions up to Dakon-18 were found [33]. Thus these games were weakly solved. In fact it was proved that Dakon-n for $n \neq 3,\ n \leqslant 18$ are one-person games, where the player has to find one of many solutions. In this aspect Dakon resembles the game of Tchuka Ruma, which is a rare one-player mancala game [26].

2.3. Checkers

Unlike in Chess, large endgame databases have been critical for success in Checkers. Schaeffer and co-workers made them instrumental when developing the strong program CHINOOK [95,96], which officially became the first man-machine World Champion in any game in 1994. CHINOOK has reached a level of play significantly stronger than all humans. Since retiring from tournament play in 1996, the CHINOOK project has only one goal left: weakly solving the game of checkers for all 144 valid three-move opening sequences played in tournaments. The solution is likely to be achieved by combining three approaches.

- *Endgame databases.* The program has perfect information about all positions involving eight or fewer pieces on the board, a total of 443,748,401,247 positions, compressed into 6 GB for real-time decompression.
- *Middle-game databases.* Whenever the program is able to determine the game-theoretic value of a position during a game, the position is added to the middle-game database. In practice, positions with as many as 22 pieces on the board have been solved, i.e., CHINOOK has announced the result of the game as soon as move 5.
- *Verification of opening analysis.* Starting with published lines of play for each opening, or lines as played by the program, deep searches are used to try and solve positions as close to the start of the game as possible. These solved positions are added to the middle-game database. If the lines are stable and do not get refuted, the solution eventually reaches the opening position.

With current technology it will likely be possible to solve the game of Checkers weakly within the next decade, and create a program that is close to strongly solving Checkers in practice [95].

Endgame databases have been used to correct many errors in the Checkers literature, for endgame positions as well as middle-game positions. The strength of CHINOOK's endgame databases is illustrated by the position of Fig. 4, previously known as the 100 *year position*. During the 1800s, extensive analysis was published considering this position. The eventual conclusion after about a century of debate was that the position is a win for White. In 1997,

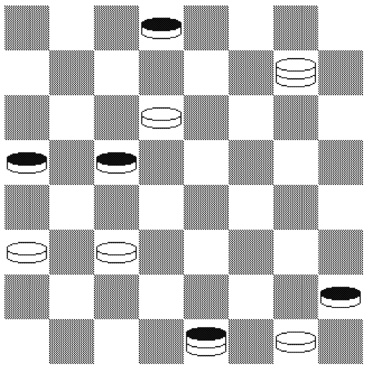

Fig. 4. The 197 year position: White to move.

Grandmaster Lafferty asked whether CHINOOK could resolve the problem. The program took less than a second to prove that the result is actually a draw. The position has since been renamed the “197 year position”.

2.4. Chess endgames

Endgames play a most prominent role in Chess. Using the technique of retrograde analysis ([53,106], see also Section 5), Thompson built all important 5-piece endgames [109]. He made them accessible to the public by placing the results on CDs. Other researchers in this area experimented with other techniques, e.g., using parallel construction algorithms [104] or other indexing schemes [82]. In the late nineties, several 6-piece endgames were constructed [105,110] and it seems only a matter of time before the most important ones will also become widely available. However, it is unlikely that important endgames with more than six (non-blocked) pieces will be built in the near future, due to the extortionately growing size of the databases involved.

The existence of endgame databases has had a large impact on the chess community, both human and machine. They are to be considered as *strong* solutions of particular endgames. When the first results became available, showing that some endgames by optimal play needed more than 50 moves to conversion, the official rules were changed. For some endgames the 50-move rule was transformed into an n-move rule with $n > 50$. Later on, when endgames were discovered which required several hundreds of moves to conversion under optimal play, it was realized that it was unfair to the defender of a position to impose similar n-move rules with such large n. It was then decided to return to the 50-move rule, even knowing that this means that many endgames, which are in some sense game-theoretically won, cannot be converted into a win if the opponent defends adequately [32,61]. As an extreme example, see Fig. 5, showing the current longest distance-to-mate of 262 moves.

The position is the max-to-mate from the KRNKNN database, independently constructed by Thompson [110] and Stiller [104]. It is a bizarre experience to go through the moves. Even grandmasters hardly understand what is going on. The example clearly

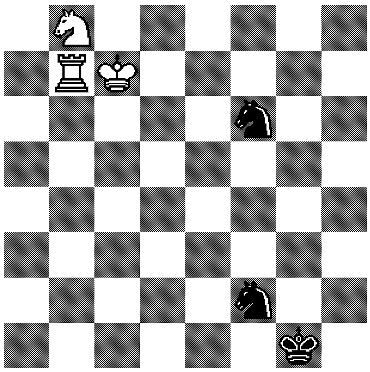

Fig. 5. The longest (262) chess distance-to-mate position known at present.

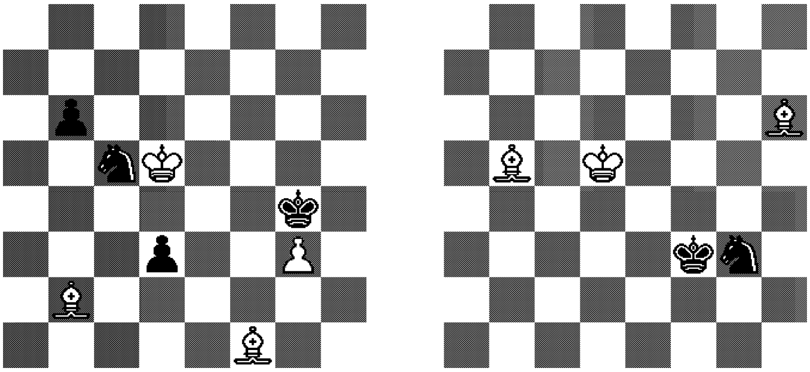

Fig. 6. Timman–Speelman, after 60. … b6 (left) and 66. Bxb5 (right).

shows that the most difficult part of the increased chess knowledge is not generating, but *understanding*, preferably by discovering general rules and patterns assimilatable for human beings. One of the pioneers in this respect is Chess grandmaster Nunn, who devoted much time to explore the mysteries of this fascinating area [84].

A second impact of endgame databases on the chess world was the correction of false statements in theory books and a better understanding of the principles involved. As an example of the impact of newly-discovered endgame knowledge in tournament practice we refer to a game between the grandmasters Timman and Speelman, in Linares, 1992. Timman knew that his game, adjourned after 60 moves (see Fig. 6, left), would quickly result in the KBBKN position of Fig. 6, right (61. Be5 Kf3 62. Bf4 d2 63. Bxd2 Ne4 64. Bh6 Nxg3 65. Bd3 b5 66. Bxb5). Kling and Horwitz [65] presented the first thorough study on the KBBKN endgame 150 years ago. Their analysis showed that Black can hold out indefinitely against White's attacks if a "fortress" can be built in what is nowadays called a Kling–Horwitz position. Thompson's endgame database for KBBKN [108,109] refuted this analysis, showing that the strong side, apart from some degenerate positions, can always win, even when the opponent reaches a Kling–Horwitz position. Knowing this and using all information available, Timman was able to convert the adjourned position into a win [16].

2.5. Summary of results

Nine Men's Morris, the mancala games, and endgames in Checkers and Chess all belong to category 2. Endgame databases combined with forward search have been successful in attacking these games. Nine Men's Morris has been solved, as well as the smaller versions of Kalah and Dakon. Increasing computing power and storage capacity are expected to be instrumental in solving Awari within a few years, and Checkers within a decade. The full game of Chess belongs to category 4.

Not mentioned in this section is the game of Xiang Qi, or Chinese Chess. Efforts in this domain have traditionally not concentrated on solving endgames. The first results have been published only recently, with the construction of 151 endgame databases [37,120].

3. Divergent games

The games in this section are immune to the retrograde-analysis methods that were prevalent in Section 2. Instead, search and knowledge-based methods must be used. This section contains results for connection games and polyomino games, as well as the games of Othello, Shogi, and Go. Strictly according to the definitions, Shogi is neither convergent nor divergent. Solutions can only be obtained for Tsume-Shogi problems. For computers, Go is the most challenging game of all; solutions can only be computed close to the end of the game or on very small boards.

3.1. Connection games

In this section we deal with connection games. Most games are two dimensional, but the board can be placed horizontally or vertically. In Section 3.1.1 we discuss Connect-Four, which has a vertical board and thus uses the law of gravity. Section 3.1.2 describes the three-dimensional game Qubic in which the role of gravity is neglected. In Section 3.1.3 the results of the games Go-Moku and Renju are mentioned. Section 3.1.4 provides the results of many k-in-a-row games, so-called mnk-games. Finally, in Section 3.1.5 the game of Hex is discussed. This is an example of a game that is ultra-weakly solved.

3.1.1. Connect-Four

Connect-Four is a four-in-a-row connection game played on a vertically-placed board. The standard game has a board of 6 rows and 7 columns. The game has been solved by Allen, using a brute-force approach [1], and simultaneously by Allis, using a knowledge-based approach [2,114]. It is a first-player win. Allen's program uses brute-force depth-first search with alpha-beta pruning, a transposition table, and killer-move heuristics. Moreover, he used expert knowledge to "tweak" the program. The game was weakly solved on a Sun-4/200-series workstation in about 300 hours. In contrast, Allis used nine strategic rules that identified potential threats of the opponent. He weakly solved the game, also on a Sun-4 in about 350 hours. The conditions of each rule were exactly formulated, as were the allowable combinations of the nine rules. These rules were implemented in the program VICTOR, which in addition exploited a combination of straightforward depth-first search and conspiracy-number search [77,93].

Relying on its knowledge rules, VICTOR easily solved smaller Connect-Four games. The game-theoretic value for $m \times n$ boards (m rows, n columns) is a trivial draw if $n < 4$. The 4×4, 6×4, 4×5, 6×5, 4×6, and 6×6 boards yielded a game-theoretic draw too. Only boards with an even number of rows were investigated, since the strategic rules used by VICTOR are based on the concept of control of zugzwang, [1] for which an even number of rows is a prerequisite. For all these boards no search is needed, i.e., the program declares the draw in the start position without looking forward. The 4×7 board is the first one in which VICTOR needed a little search, demonstrating again a game-theoretic draw. The standard 6×7 board appeared to be the smallest board with an even number of

[1] Zugzwang (mostly used in Chess): an obligation to move in one's turn even when this must be disadvantageous (German, from Zug 'move' and Zwang 'compulsion').

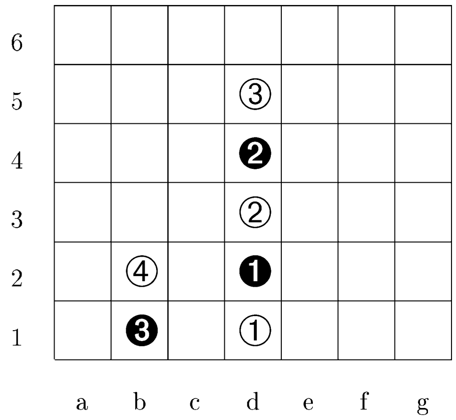

Fig. 7. Optimal opening-move sequence in standard Connect-Four.

rows on which the first player (considered to be White) wins. However, the winning moves are rather hard to find, as illustrated by the following continuation: **1. d1! d2 2. d3! d4 3. d5! b1 4. b2** (see Fig. 7), in which an exclamation mark signals that a move is unique in obtaining the win. The first three white moves all are forced, any other choice allows Black at least a theoretical draw. The fourth white move leads to a win; other white alternatives were not examined.

3.1.2. Qubic

Qubic is a three-dimensional version of Tic-Tac-Toe, where the goal is to achieve a straight chain of four own stones in a $4 \times 4 \times 4$ cube. No gravity conditions apply. In 1980 Patashnik weakly solved the game by combining the usual depth-first search with his own expert knowledge for ordering the moves [86]. However, for some reason this solution escaped the attention of computer-games researchers. In the early 1990s Allis and Schoo wrote the program QBIG to solve Qubic. Initially they were not aware of Patashnik's results. When they were notified in the last phase of the project, they decided to continue, since their main goal was not only the solution of Qubic, but also the testing of two new methods: threat-space search, developed for Go-Moku and Renju, and proof-number search, already tested in the domain of Connect-Four and Awari. Thus, Allis and Schoo weakly solved Qubic again [6], confirming Patashnik's result that Qubic is a first-player win.

3.1.3. Go-Moku and Renju

Go-Moku and Renju are played on the same board of 15×15 points of intersections. Two players alternately place a stone of their own colour on an empty intersection. Black starts the game and must place a black stone on the centre intersection (h8). The goal is to obtain a straight chain (horizontally, vertically or diagonally) of five stones of the player's colour. If the board is completely filled, and no one has five-in-a-row, the game is drawn. The rules of Go-Moku and Renju differ [118]. To complicate matters, even for the game of Go-Moku there are different versions. For instance, in *free-style* Go-Moku an *overline* of six-in-a-row wins, whereas in *standard* Go-Moku it does not win. We remark that both versions have the property that the same rules hold for both players, i.e., symmetric ruling. In Renju additional restrictions are imposed on the first player, making the game asymmetrical: when making an overline or a so-called *double three* or *double four* Black loses the game immediately, whereas White is allowed to make an overline, double three or double four.

Go-Moku

For both free-style and standard Go-Moku, Allis established that the game-theoretic value is a first-player win [8]. To prove this he used a combination of threat-space search, proof-number search and database construction (cf. Section 5). Thus, he weakly solved Go-Moku. At that time (1995) he did not see any possibility to apply these methods successfully on the game of Renju. Another five years had to pass before this happened.

Renju

Since 1936—when the Japanese Federation of Renju was founded—it was felt that in Go-Moku the first player has an overwhelming advantage. Consequently, most professional players favoured the game of Renju, in which several restrictions were imposed on the first player. Still it was generally believed that Renju also is a first-player win. This conjecture has been verified by Wágner and Virág [118], who weakly solved Renju in 2000 by a search process that used transposition tables, threat-sequence search [8], and expert knowledge for no-threat moves. The program took advantage of an iterative-deepening search based on threat sequences up to 17 plies. The best solution of a winning line (not necessarily the shortest solution) found by Wágner and Virág reads: **1. h8 2. h7 3. i7 4. g9 5. j6 6. i8 7. i6 8. g6 9. j9 10. k5 11. k6 12. l6 13. j7 14. j8 15. k8 16. l9 17. l7 18. m6 19. j4 20. i4 21. k7 22. m7 23. i5 24. h4 25. g7 26. h6 27. h5 28. g4 29. f5 30. g5 31. j10 32. f4 33. e4 34. e3 35. d2 36. g3 37. g2 38. n8 39. o9 40. i9 41. i10 42. h11 43. k10 44. k9 45. h10 46. g10 47. l10** (see Fig. 8).

3.1.4. k-in-a-row games

Next to Go-Moku and Renju there are many games with the aim of obtaining a straight chain of stones of one colour on a flat rectangular board. Such games can be characterized as k-in-a-row games, or more precisely, as mnk-Games, with m and n denoting the number of rows and columns of the board, and k giving the length of the straight chain to be obtained. Thus, an mnk-game is an abbreviation for a k-in-a-row game on an $m \times n$ flat board [115]. The standard 5-in-a-row Go-Moku game can thus be characterized as the 15,15,5-Game, whereas the children's game Tic-Tac-Toe is the 3,3,3-Game. The rules of the mnk-Games are equivalent to the rules of the free-style Go-Moku game.

The game-theoretic values of many mnk-Games have been published recently [117]. These results were based on using knowledge-based rules. The results are collected in Table 3.

3.1.5. Hex

Hex is played on a rhombic hexagonal board of any size; the most commonly used board sizes are 10×10 and 11×11. The players, Black and White, move alternately by placing

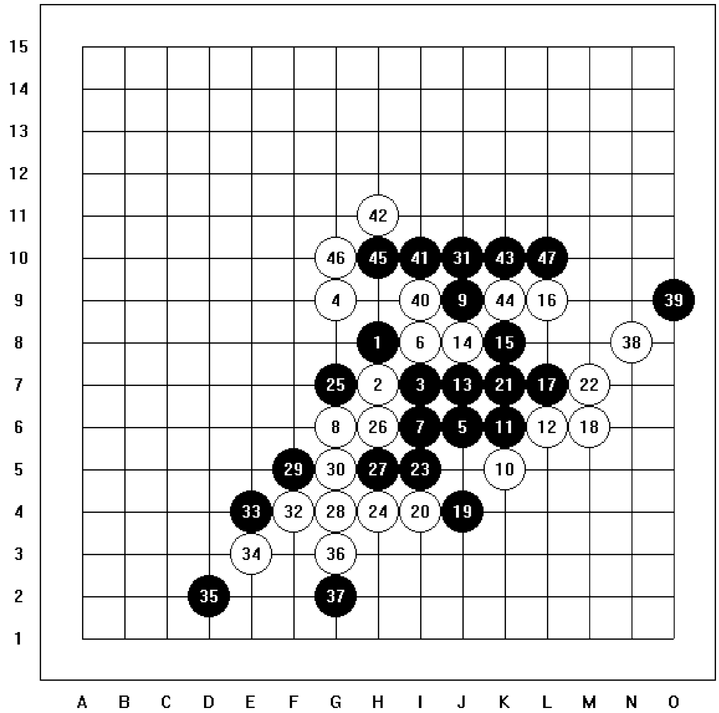

Fig. 8. A first-player win in Renju.

Table 3
Game values of mnk-games

mnk-games ($k = 1, 2$)	W
333-game (Tic-Tac-Toe)	D
$mn3$-games ($m \geqslant 4, n \geqslant 3$)	W
$m44$-games ($m \leqslant 8$)	D
$mn4$-games ($m \leqslant 5, n \leqslant 5$)	D
$mn4$-games ($m \geqslant 6, n \geqslant 5$)	W
$mn5$-games ($m \leqslant 6, n \leqslant 6$)	D
15,15,5-game (Go-Moku)	W
mnk-games ($k \geqslant 8$)	D

a stone in any unoccupied cell, trying to connect their two opposite borders in Fig. 9 with a chain of stones of their own colour. As a consequence of the board topology, draws are impossible [42].

There is no standard convention as to which colour has the first move. Hex in its *unrestricted form* is a first-player win, which is easily shown by the “strategy-stealing”

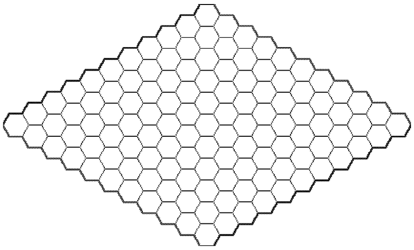

Fig. 9. The 11 × 11 Hex board.

argument, stated first by John Nash in 1949 [13]: if there were a winning strategy for the second player, the first player can still win by making an arbitrary first move and using the second-player strategy from then on. Since an arbitrary extra move can never be a disadvantage in Hex, the resulting contradiction shows that there cannot be a winning strategy for the second player. In practice, the advantage of moving first is offset by adding the *one-move-equalization* rule, which states that one player plays an opening move and the other player then has to decide which colour to play for the remainder of the game. The resulting game, *competitive Hex*, is a second-player win. Hex is thus solved ultra-weakly.

Weak and strong solutions are only known for small board sizes. For unrestricted Hex, a winning strategy on the 7 × 7 board was fully described in the early 1990s in an unpublished manuscript by Yang [34]. For competitive Hex, Enderton solved all 6 × 6 openings by computer in 1994 [35]. The first *strong* solution for this game was constructed by Van Rijswijck's program QUEENBEE in 1999 [90,91]. Experienced human players can probably play perfectly in any 5 × 5 position, but likely not in every 6 × 6 position. One example is the opening move most frequently played by experts. The general consensus among top players was that this move would win the unrestricted game on a 6 × 6 board, but in fact it loses.

Hex exhibits considerable mathematical structure, which is exploited by the top programs. Currently the strongest program is HEXY, which uses a method called *virtual-connection search*. This is an automatic theorem-proving method that deduces properties of subsets of the board [9,10]. Unfortunately this method is known to be unable to solve some positions and thus cannot be used to solve the game strongly without relying on additional game-tree search. QUEENBEE uses a different theorem-proving method called *pattern search* (see Section 5.2), which is a game-tree search enhancement that in principle can prove the value of any position.

Hex has been proved to be PSPACE-complete [36,89]. The state-space and decision complexities are comparable to those of Go on equally-sized boards. Both the branching factor of the game tree and the typical length of the games are of $O(n^2)$ on an $n \times n$ board. Given the search effort involved in solving 6 × 6 competitive Hex and the proposed enhancements to the pattern-search algorithm, solving 7 × 7 competitive Hex may be within reach in the next four years. Solving the 8 × 8 game is entirely intractable without fundamental breakthroughs.

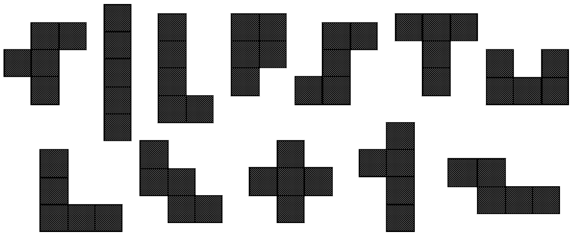

Fig. 10. The twelve distinct pentominoes.

3.2. Polyomino games

A *polyomino* is a two-dimensional piece that forms a connected subset of a square grid. In polyomino games, these pieces are placed on the square grid according to specified rules. The first player unable to place a piece loses the game. In this section we cover two such games: Pentominoes and Domineering. Pentominoes is played on a fixed board size, whereas Domineering can be played on any rectangular board size.

3.2.1. Pentominoes

Pentominoes are shapes made up of five squares of equal size joined edge to edge. There are 12 topologically-distinct pentominoes, see Fig. 10.

The invention of Pentominoes is ascribed to Golomb around 1953, who published a book *Polyominoes* in 1965 [47]. Pentominoes can be used as one-player puzzles to form prescribed geometric shapes. However, besides its intrigue as a puzzle, the placement of pentominoes on an 8×8 board also makes it an exciting competitive game of skill. Played by two or three players, the object of the game is to be the last player to place a pentomino piece on the board. Players take turns choosing a piece and placing it on the board. The pieces must not overlap or extend beyond the boundary of the board, but they do not have to be adjacent. Recently the two-player version of this Pentominoes game has been weakly solved by Orman [85]. Using a straightforward search program, based on opening-move suggestions by the user, she showed that of all 296 distinct first moves at least two assure a first-player win. She also showed that one move, in particular the one restraining the second-ply moves mostly, is a second-player win. Out of the 1181 replies exactly two refute the first-player's opening move. Checking how many opening moves guarantee the first player a win still has to be done.

3.2.2. Domineering

The Domineering game is also known as Crosscram. It was proposed by Andersson around 1973 [31,43]. In Domineering the players alternately place a domino (2×1 tile) on a board, i.e., on a finite subset of Cartesian boards of any size or shape. The game is usually played on $m \times n$ boards (m rows and n columns). The two players are denoted by Vertical and Horizontal. In standard Domineering the first player is Vertical, who is only allowed to place dominoes vertically on the board. Horizontal may play only horizontally. Dominoes are not allowed to overlap. As soon as a player is unable to move the player loses. As a consequence, draws are not possible in Domineering. Although Domineering can be played

Table 4
Game-theoretic values of Domineering games on $m \times n$ boards

$m\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	**L**	**L**	**L**	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***
2	**W**	**W**	**W**	**L**	**W**	**W**	**W**	***L***	**W**	**W**	**W**	***L***	**L**	**W**	**W**	***L***	***L***	**W**	**W**	***L***	***L***	**L**	**W**	***L***	***L***	***L***	**W**	***L***	***L***	***L***
3	**W**	**W**	**W**	**L**	**L**	**L**	**L**	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***
4	***W***	**W**	**W**	**W**	***W***	**W**	**W**	**L**	***W***	**L**	**W**	**L**	**L**	**L**	**L**	***L***	**L**	***L***		*L*		*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*
5	***W***	**L**	**W**	***L***	**L**	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	***L***	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*
6	***W***	**W**	**W**	**W**	***W***	**W**	**W**	**L**	***W***	**W**	**W**	**L**	*W*			*L*				*L*				*L*				*L*		
7	***W***	**W**	**W**	**L**	***W***	**L**	**W**	***L***	**L**	***L***	**L**	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*
8	***W***	***W***	***W***	**W**	***W***	**W**	***W***	**W**	***W***		*W*		*W*			*L*														
9	***W***	**L**	***W***	***L***	***W***	***L***	**W**	***L***	**W**	*L*		*L*		*L*		*L*		*L*		*L*		*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*	*L*
10	***W***	**W**	***W***	**W**	***W***	**W**	***W***	**W**	*W*		*W*		*W*							*L*										
11	***W***	**W**	***W***	**L**	***W***	**W**	**W**	*L*	*W*			*L*				*L*				*L*		*L*		*L*		*L*		*L*		*L*
12	***W***	***W***	***W***	**W**	***W***	**W**	*W*		*W*		*W*		*W*											*L*						
13	***W***	**L**	***W***	***L***	***W***	*L*	*W*	*L*		*L*		*L*		*L*		*L*		*L*		*L*		*L*		*L*		*L*		*L*		*L*
14	***W***	**W**	***W***	**W**	***W***	*W*	*W*		*W*		*W*		*W*															*L*		
15	***W***	**W**	***W***	**W**	***W***		*W*		*W*		*W*																			*L*
16	***W***	***W***	***W***	***W***	***W***	*W*	*W*		*W*		*W*		*W*																	
17	***W***	***W***	***W***	**W**	*W*		*W*		*W*		*W*		*W*																	
18	***W***	**W**	***W***	***W***	*W*	*W*	*W*		*W*		*W*		*W*																	
19	***W***	**W**	***W***	*W*	*W*	*W*	*W*		*W*		*W*		*W*																	
20	***W***	***W***	***W***	*W*	*W*	*W*	*W*		*W*		*W*		*W*																	
21	***W***	***W***	***W***	*W*	*W*		*W*		*W*		*W*		*W*																	
22	***W***	***W***	***W***	*W*	*W*	*W*	*W*		*W*	*W*	*W*		*W*																	
23	***W***	**W**	***W***	*W*	*W*	*W*	*W*		*W*		*W*		*W*																	
24	***W***	**W**	***W***	*W*	*W*	*W*	*W*	*W*	*W*		*W*		*W*																	
25	***W***	***W***	***W***	*W*	*W*		*W*		*W*		*W*		*W*																	
26	***W***	***W***	***W***	*W*	*W*	*W*	*W*		*W*	*W*	*W*		*W*																	
27	***W***	**W**	***W***	*W*	*W*	*W*	*W*		*W*		*W*		*W*																	
28	***W***	**W**	***W***	*W*	*W*	*W*	*W*		*W*		*W*		*W*																	
29	***W***	***W***	***W***	*W*	*W*		*W*		*W*		*W*		*W*																	
30	***W***	**W**	***W***	*W*	*W*	*W*	*W*		*W*		*W*		*W*	*W*																

on any board and with Vertical as well as Horizontal to move first, the original game was played on an 8×8 board with Vertical to start, and this instance has generally been adopted as standard Domineering.

Many weakly-solved instances of Domineering have been published [21,117]. Meanwhile, some forty new results have been obtained by computer analyses. Table 4 provides an overview of all currently-available results. We use **bold**, *italics*, and its combination ***bold-italics***. In **bold** face (i.e., **bold** and ***bold-italics***) all results of computer analyses are given.

The game-theoretic values of many boards can be obtained from the values of smaller boards [68]. A simple example is as follows. On the 2×8 board, Horizontal can decide never to cross the vertical line between the fourth and fifth column, thus effectively playing in either of two 2×4 subboards. Since the 2×4 board is a win for Horizontal, irrespective of who plays first or second, it is clear that the 2×8 board, and indeed all $2 \times 4n$ boards, are a win for Horizontal too. With this and other strategies the game-theoretic values of many boards can be derived from the game-theoretic values of smaller boards. These results have been indicated in Table 4 in *italics* (i.e., *italics* and ***bold-italics***).

The following two conjectures about Domineering games have previously been published [117].

Conjecture 1. Domineering on $m \times (2n+1)$ boards, $m > 2n+1$, are first-player wins. Equivalently, Domineering on $(2m+1) \times n$ boards, $n > 2m+1$, are second-player wins.

Conjecture 2. Domineering on $m \times 4n$ boards, $m < 4n$, are second-player wins. Equivalently, Domineering on $4m \times n$ boards, $n < 4m$, are first-player wins.

All new results are still in agreement with both conjectures. Moreover, the same article contained a discussion on the outcome of the 9×9 game. The row for $m = 9$ suggested a first-player win, but both the diagonal and the column for $n = 9$ suggested a second-player win. The 9×9 game turned out to be a first-player win.

3.3. Othello

The game of Othello is played on an $n \times n$ board, with n being even. A move consists of adding a piece to the board and changing the colour of some of the pieces already present on the board. The game ends when no more moves can be played, at which point the winner is the player with the most pieces of their colour on the board.

The strongest Othello program today is Buro's program LOGISTELLO [24]. It plays stronger than the human World Champion. In 1997 it beat the reigning World Champion Murakami convincingly by 6–0 [23]. So far the standard 8×8 Othello game has not been solved, but the 6×6 game (see Fig. 11) was weakly solved by the former British Othello Champion Feinstein.

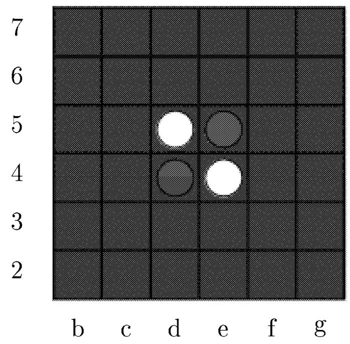

Fig. 11. The initial position for 6×6 Othello.

It took Feinstein two weeks of computer time to show that the second player has a forced win by 20–16 discs [39]. The principal variation is as follows: **1. d3 2. c5 3. d6 4. e3 5. f5 6. f4 7. e2 8. d2 9. c2 10. e6 11. e7 12. g5 13. c4 14. c3 15. g4 16. g3 17. f3 18. c7 19. b5 20. d7 21. b7 22. b3 23. c6 24. b6 25. f7 26. f6 27. b4 28. b2 29. g6 30. g7 31. pass 32. f2 33. g2**.

3.4. Shogi

Shogi (Japanese Chess) differs from Western Chess and Chinese Chess in its nature. The latter two are convergent games and Shogi is not. This implies that the initial position does not converge to "smaller games", which can be solved in its own. However, in the last phase of a Shogi game most Shogi programs apply a technique that searches for a mate. This technique has been cultivated so as to solve effectively Tsume-Shogi problems.

A Tsume-Shogi problem is a Shogi composition with only one solution. The answer of a Tsume-Shogi problem is usually presented as a solution sequence of moves, i.e., a sequence of moves from the given position to a mate position. In Tsume-Shogi Black is the attacking side which moves first. White is assumed to play moves that delay the mate as long as possible. The solution sequence must satisfy a number of additional conditions, not given here, stating among others that a composition should have a single well-defined solution [75].

The last decade several strong Tsume-Shogi solving programs have been developed. Of these, SEO is one of the strongest [98]. It is based on a depth-first recursive iterative-deepening version of proof-number search. It was able to weakly solve almost all problems from a test set with some three hundred, mostly notably difficult problems. In addition, it weakly solved a well-known problem, called "Microcosmos" (see Fig. 12). This is a Tsume-Shogi problem with a remarkable solution length of 1525 steps, the longest known at present. It is a world record regarding the length of the solution sequence of Tsume-Shogi problems solved.

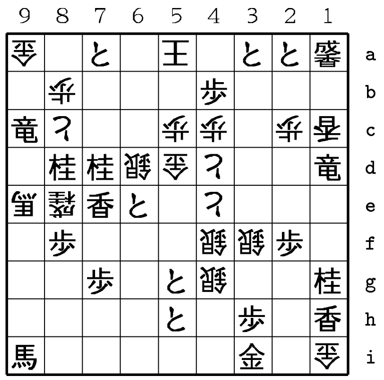

Fig. 12. Microcosmos (1525 steps).

3.5. Go

Although Go played on a 19×19 board is much too complex to solve with the current means, Go endgame positions are often decomposable into separate local fights and as such they may be subject to exhaustive analysis. Local fights occur in parts of the board that are separated by walls of safe stones. Combinatorial game theory provides methods and tools to analyse these Go endgames. The specific theory, called *thermography*, follows methods analysed by Conway [12,31]. It provides a *value* indicating who is ahead and by how much in a given position, and an *incentive* indicating how strong the next move is. Using this theory a program can solve ko-free Go positions of moderate size [79].

A new method for analysing local Go positions with loops (i.e., kos) was developed by Berlekamp [14]. The method is called *generalized thermography*, providing *mast* values and *temperatures* with similar meaning as values and incentives for non-loopy games. An implementation of this generalized thermography was presented by Müller [80]. It is expected that this theory for completely-separated endgame fights in Go can be used as good heuristic evaluations for more open Go positions.

The earliest computer analyses for Go on smaller boards were reported in 1972 by Thorp and Walden [112]. The largest boards investigated by them were 1×5, 2×4, and 3×4. The results on small boards are quite sensitive to the details of the rules; since the publication mentioned above the Japanese rules have been changed, thereby affecting the optimal playing lines published. Under the Chinese rules, the results were not affected.

The largest square board for which a computer proof has been published is 4×4, by Sei [97]. It is a first-player win by two stones. No larger boards have been solved yet. A solution of the 5×5 board is exceedingly subtle, and depends on the possibility of suicide as well as on whether superko is situational or positional. Table 5 lists the results of weakly-solved games on small boards under the Chinese rules. Results confirmed by computer analysis [40] have been indicated in **bold** face; all results were originally obtained by human analysis [103]. A question mark indicates that the result is presumed, but not yet certain.

From professional-level play, the first-player advantage on 19×19 boards is believed to be approximately 7. From the analyses we conjecture that in Go the importance of being the first player grows with the board size up to a certain size and then diminishes.

3.6. Summary of results

Many of the *mnk*-Games, as well as all of the other connection games described in this section, are first-player wins. Connect-Four and Qubic belong to category 1; Go-Moku, Renju, and the *mnk*-Games to category 3; and Hex to category 4. The games of category 1 are solved by both approaches, i.e., the knowledge-based approach and the brute-force approach. The knowledge-based methods used are: game-dependent knowledge rules, threat-space search and proof-number search. The category-3 games are solved by a combination of expert knowledge, threat-space search, threat-sequence search, proof-number search (and database construction). Finally, small versions of Hex are solved by a mixture of methods, whereas solving the standard game of Hex, which is in category 4, is beyond reach.

Table 5
First-player scores for Go on $m \times n$ boards

$m\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**0**	**0**	**3**	**4**	**0**	**1**	**2**	**3**	**0**	**1**	**2**	1?	2
2	**0**	**1**	**0**	**8**	**10**	**12**	14	16	18	4			
3	**3**	**0**	**9**	**4**	15	18	5?	24?					
4	**4**	**8**	**4**	**2**	20	1							
5	**0**	**10**	15	20	25	0	9						
6	**1**	**12**	18	1	0	4							
7	**2**	14	5?		9								
8	**3**	16	24?										
9	**0**	18											
10	**1**	4											
11	**2**												
12	1?												
13	2												

On the mnk-Games we provide two tentative conclusions.

(1) If k is sufficiently small (i.e., smaller than 8): the larger the board the higher the probability that the first player will win.

(2) If k increases from 3 to 8 the probability of winning decreases; for $k \geqslant 8$ the game is always drawn.

The polyomino games, Othello, and Go on small boards have in common that their state-space complexities are considerably smaller than their game-tree complexities. Hence, these games belong to category 2. It is in agreement with the fact that the solution of these games (many Domineering instances, 6×6 Othello, many small board versions of Go, and the game of Pentominoes) all are solved by brute-force methods. Instances on much larger boards will typically belong to category 4. If such a game is *not* unsolvable by any method, the solution will require substantially more computer power, probably in combination with sophisticated knowledge-based methods.

The Tsume-Shogi problem-solving methods are mainly knowledge-based methods as is the theory of thermographs. The Tsume-Shogi problems were not classified previously; they belong to category 3. Go belongs to category 4, but the Go endgames solved belong to category 3.

The advantage of being a first player is clearly predominant in Domineering, as long as the board shape does not favour the second player excessively, e.g., in very horizontally-stretched boards. The 6×6 Othello game is a rare genuine second-player win. So far it is rather uncertain what the outcome of the standard 8×8 Othello will be. Is it a second-player win too, or is having the initiative here an advantage sufficient to win the game? In Go second-player wins are impossible, due to the fact that passing is allowed. The "strategy-stealing" argument proves this. Therefore only game-theoretic first-player wins and draws are possible. The results so far suggest that theoretic draws mainly occur at small boards, where the first player has not sufficient space to build living chains with eyes. Moreover, at larger boards the current results suggest that the first-player's initiative diminishes. Additional outcomes are needed to conclude whether this statement is justified or only a result of human's inability to play optimally in a complex game.

4. Game characteristics determining solvability

In the preceding sections we investigated the solutions of many two-person zero-sum games with perfect information. The games were divided into four categories, which were defined by type of state-space complexity and game-tree complexity. In this section we focus on the complexities mentioned (Section 4.1), on the methods used (Section 4.2), and on the role of the initiative (Section 4.3).

4.1. State-space complexity versus game-tree complexity

For all games discussed previously Table 6 lists the state-space complexities and the game-tree complexities. The first column contains the identification numbers (Id.).

The state-space complexity is defined as the number of legal game positions reachable from the initial position of the game. The game-tree complexity is defined as the number of leaf nodes in the solution search tree of the initial position of the game. The solution search tree of a node J is the full-width search tree with a depth equal to the solution depth of J. Here, the solution depth of a node J is the minimal depth (in ply) of a full-width search sufficient to determine the game-theoretic value of J [7]. Thus, the solution search tree should not be confused with the solution tree [22]. The last column provides the references, if applicable. For Dakon-6 we used the complexities of the Kalah instance played on the same board, i.e., of Kalah(6,6) [62]. For 8×8 Domineering we used an average game length of 30 and an average branching factor of 8 as obtained from preliminary experiments by Breuker [20], yielding a game-tree complexity of $O(10^{27})$. An educated guess for the state-space complexity derived from the game-tree complexity and based on the number of possible transpositions then gives $O(10^{15})$. Finally, a similar estimate for Pentominoes based on an average game length of 10 and an average branching factor of 75 [85] yields a game-tree complexity of $O(10^{18})$ and a state-space complexity of $O(10^{12})$.

An impression of the game space filled with the games listed in Table 6 is given in Fig. 13. The numbers refer to the identification numbers of Table 6.

The main conclusion is that a low state-space complexity is more important than a low game-tree complexity as a determining factor in solving games. The state-space complexity provides a bound on the complexity of games solvable by complete enumeration. In 1994, the boundary of solvability by complete enumeration was set at 10^{11} [7]. The application of Moore's law will bring it to 3×10^{12}, a number that fits with the current state of the art reasonably well. The game-tree complexities form a real challenge for intelligent search methods. Knowledge of the game is very important to apply effectively a knowledge-based method (see Section 4.3).

Table 6
State-space complexities and game-tree complexities of various games

Id.	Game	State-space compl.	Game-tree compl.	Reference
1	Awari	10^{12}	10^{32}	[3,7]
2	Checkers	10^{21}	10^{31}	[7,94]
3	Chess	10^{46}	10^{123}	[7,29]
4	Chinese Chess	10^{48}	10^{150}	[7,113]
5	Connect-Four	10^{14}	10^{21}	[2,7]
6	Dakon-6	10^{15}	10^{33}	[62]
7	Domineering (8×8)	10^{15}	10^{27}	[20]
8	Draughts	10^{30}	10^{54}	[7]
9	Go (19×19)	10^{172}	10^{360}	[7]
10	Go-Moku (15×15)	10^{105}	10^{70}	[7]
11	Hex (11×11)	10^{57}	10^{98}	[90]
12	Kalah(6,4)	10^{13}	10^{18}	[62]
13	Nine Men's Morris	10^{10}	10^{50}	[7,44]
14	Othello	10^{28}	10^{58}	[7]
15	Pentominoes	10^{12}	10^{18}	[85]
16	Qubic	10^{30}	10^{34}	[7]
17	Renju (15×15)	10^{105}	10^{70}	[7]
18	Shogi	10^{71}	10^{226}	[76]

4.2. Brute-force versus knowledge-based methods

In Section 5 we discuss the methods developed to solve games, i.e., the brute-force methods and the knowledge-based methods. Below we provide a comparison of the two methods in relation to the games solved. The comparison is given in three statements, which make a clear connection with state-space complexity and game-tree complexity.

- Games with a relatively low state-space complexity have mainly been solved with brute-force methods. Examples are Nine Men's Morris, Kalah, and Domineering.
- Games with a relatively low game-tree complexity have mainly been solved with knowledge-based methods. Examples are Go-Moku, Renju, and k-in-a-row games.
- Games with both a relatively low state-space complexity and a low game-tree complexity have been solved by both methods. Notable examples are Connect-Four and Qubic.

4.3. The advantage of the initiative

In 1981 Singmaster proved a rather conclusive and elegant theorem why first-player wins should abound over second-player wins [101,102]. The positions in a game with two

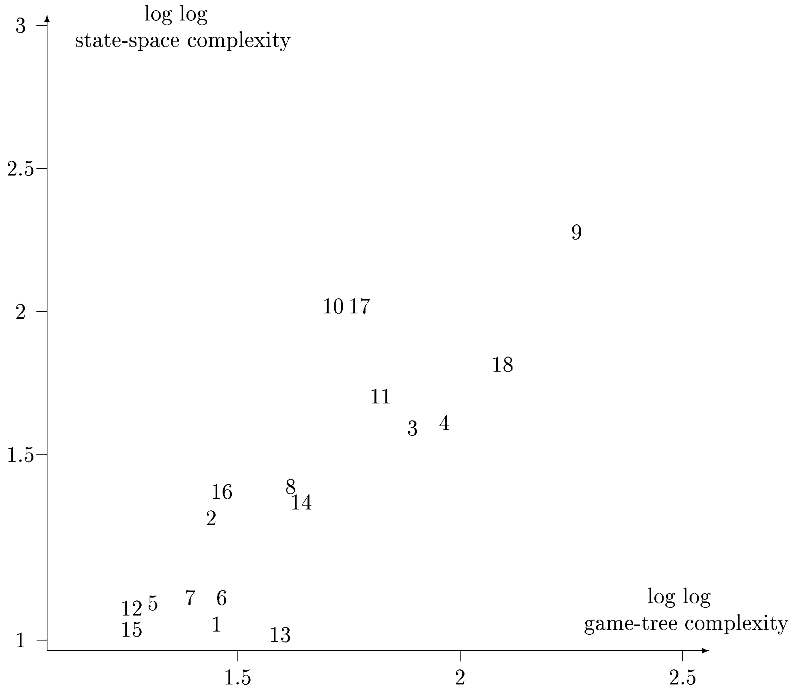

Fig. 13. Approximate positions of games in the game space.

outcomes (won, lost) are split up into P-positions (from which the previous player can force a win) and N-positions (from which the next player can force a win). For the first player to have a forced win, just one of the moves needs to lead to a P-position. For the second player to have a forced win, all of the moves must lead to N-positions. For games with three outcomes, draws can easily be included in this line of reasoning, stating that first-player wins should abound over draws and second-player wins. Two remarks are in order. On the one hand we have seen that in relatively many games on small boards the second player is able to draw or even to win. Therefore we assume that Singmaster's theorem has limited value when the board size is small. On the other hand, there exist unattainable goals, i.e., goals that cannot even be reached on an infinite board. Examples are mnk-Games with $k \geqslant 8$.

To investigate the practical research question as to whether being the first player is such a great advantage, the first two authors previously defined the concept of *initiative* [117], as "the right to move first". From an investigation of solved games they then concluded that the concept of initiative seems to be a predominant notion under the requirement that the first player has sufficient space to fulfil the goals. Using the additional results of the present article we feel strengthened in the belief that there is a relation between the knowledge-based methods, the initiative and solving a game. The idea is as follows. Assume G is a complex game with three outcomes: won, drawn, and lost. We consider G a *fair* game if it is a draw and both players have roughly an equal probability on making a mistake.

A potential strategy S for a player who aims to win G is to try and obtain a small initiative, here taken as a small advantage in the sense that the opponent must react adequately on the moves played by the player. To reinforce the initiative the player searches for threats, and even for a sequence of threats. In the sequence of threats the player hopes to find a double threat which ends the game. An evaluation function E may guide this search process. Now the question is how to implement strategy S in relation to E? This can be done by one of the usual search methods, e.g., α-β search with enhancements [87] or a practical alternative to α-β [64]; the program's task then is to achieve an increasing series of evaluation values. Another method is that once there is an initiative (a small advantage) the player relies on selective search and applies threat-space search. When it results in a win, the strategy is called successful. Otherwise it is a nuisance.

When solving a game, threat-space search is adequate in the final phase of the search process, provided that a win can be proven. Here the question arises: when does the final phase start? In the extreme, the question could be formulated as follows: can the right to move first be considered as a small advantage from where the threat-space search can be started? It seems that games in which the first player has a clear advantage over the other player, such as in Go-Moku, are more easily solvable by threat-space search than games in which the advantage is less clear, such as in Renju.

5. Methods developed for solving games

This section gives an overview of methods developed in the course of solving games. We describe them briefly, and refer the reader to relevant references for more details. Section 5.1 concentrates on brute-force methods, while in Section 5.2 the knowledge-based methods are discussed.

5.1. Brute-force methods

Brute-force methods have been influential tools instrumental in solving games. Many solving programs use basic brute-force methods such as α-β and their enhancements in some way or another. Two specific methods, that have their application especially in solving games, are the construction of databases by retrograde analysis and enhanced transposition-table methods. They are shortly discussed below.

Retrograde analysis

Retrograde analysis is a method in which for each position of some specific game or endgame the number of moves towards the best reachable goal is stored. For instance, in Chess, assuming perfect counterplay, the number of moves to be played by the stronger side up to mate or conversion is stored. Checkers databases sometimes only contain an indication of won, drawn, or lost per position. A database is constructed by starting in terminal positions and then working backwards [53]. Once constructed, perfect play is guaranteed: the stronger side chooses a move with the shortest distance-to-mate (or to conversion) and the weaker side opts for moves with the longest distance-to-mate (or to conversion). In passing we remark that perfect play by a computer in a position which is game-theoretically drawn or lost does not guarantee the best performance against fallible opponents, as was demonstrated by Jansen [63].

Using the retrograde-analysis method, the first endgame database was already constructed by Ströhlein in 1970 [106]. However, his work remained unobserved for a long time. In the mid seventies the method of retrograde analysis was independently re-invented by Clarke [30] and Thompson. Only when the latter researcher's results became widely available in the early eighties did the method gain its well-deserved fame [54,109]. Nowadays the use of retrograde analysis is commonplace for the construction of endgame databases. It has deepened the understanding of such endgames considerably, and resulted in notions as *max-to-mate*, *max-to-conversion*, *max-to-zeroing-move*, and *max-to-the-rule* [51].

Enhanced transposition-table methods

The "traditional" transposition tables used in game-playing programs normally exploit the DEEP replacement scheme, i.e., when two different positions compete for the same entry in the table, the old position is overwritten by the newer one provided that the latter is searched at least as deep as the former. Research on this and other replacement schemes showed that the DEEP scheme is not the most efficient one [18]. In particular, two-level replacement schemes are often more appropriate [19]. They turned out to be essential in solving many instances of Domineering [21].

5.2. Knowledge-based methods

Next to brute-force methods it is often beneficial to incorporate knowledge-based methods in game-solving programs. Their main advantage is providing an appropriate move ordering or selection in the search trees. Four methods are presented in this section.

Threat-space search and λ-search

Allis generalised the idea of threat-space search to a method called dependency-based search [7]. However, the latter name was never adopted and the idea of threat-space search or threat-sequence search is widely known by now. Threat-space search investigates whether by a sequence of threats, to which the opponent at any time has only a limited set of replies, a win can be forced. Since the opponent effectively has no real choices, this search algorithm represents the application of single-agent search to two-player games.

A recent successor of threat-space search, called λ-search, has been proposed by Thomsen [111]. This method uses null moves combined with different orders of threat sequences, called λ-trees. Thomsen introduces λ^1-moves, which threaten to end the game or reach a specified subgoal immediately, followed by λ^2-moves threatening a winning λ^1-sequence, and so on. The method behaves as a goal-directed searcher, with a favourable tree size relative to standard α-β trees. It can be combined with any search method to search the λ-trees. As a relevant example Thomsen mentions proof-number search. A combination of null moves and proof numbers seems a promising method for solving Go endgames.

A close analogue to λ-search is Abstract Proof Search, proposed by Cazenave [27]. This method has reportedly been able to solve Philosopher's Football on 9×9 boards.

Extensions of the algorithm can solve the 11×11 game and may be able to solve 13×13 as well [28].

Proof-number search

Proof-number search [5,7] is a best-first search, in which the cost function used in deciding which node to expand next is given by the minimum number of nodes that have to be expanded to prove the goal. As such it is a successor of conspiracy-number search [77, 93]. Proof-number search is appropriate in cases where the goal is a well-defined predicate, such as proving a game to be a first-player win.

Depth-first proof-number search

The Tsume-Shogi-solving program SEO is based on a newly-developed iterative-deepening depth-first version of proof-number search, named PN* [98]. The idea is partly derived from Korf's RBFS algorithm [67], which was formulated in the framework of single-agent search and as such a successor of the widely-used IDA* algorithm [66].

Pattern search

Pattern search, introduced by Van Rijswijck [90], is a game-tree search enhancement based on Ginsberg's partition search [46] and Allis' threat-space search [7]. It applies to games where immovable counters are placed on the board, and was developed for the game of Hex. The method is able to prove wins and losses while searching a tree smaller than the minimal proof tree of the game, by proving the result for several moves at once in lost positions.

Pattern search concentrates on finding *threat patterns*. A threat pattern is a collection Ψ of empty cells in a position P, with the property that the game-theoretic value of P is unaltered when the winning side is restricted to playing moves within Ψ. A threat pattern is not unique for a position, since adding an empty cell to a threat pattern always creates another valid threat pattern. A threat pattern can be thought of as representing the *relevant area* on the board, an area that human players commonly identify when analysing a position. The patterns can be calculated recursively.

6. Conclusions and future research

In this final section we provide our conclusions and give answers on the three questions posed in the beginning (Section 6.1). Suggestions for future research are split in new predictions (Section 6.2), which might challenge researchers to solve three more games before 2010, and in new games (Section 6.3). There LOA and Amazons are introduced as contenders for being solved in 2020. Finally five new games taken from the Mind Sports Olympiad are mentioned as new challenges.

6.1. Conclusions

In our conclusions we would like to emphasize the methods used rather than the games discussed. The use of retrograde analysis in building databases has had the greatest impact on solving (end)games or almost-solving games (Awari and Checkers). The method will certainly pay off in other game domains too. Recently the first results have been published in the domain of Chinese Chess, for which the construction of 151 endgame databases was reported [37,120].

The knowledge-based methods mostly inform us on the structure of the game. Conjectures on this structure then frequently form a starting point for application of a solving method, e.g., threat-space search and proof-number search. This structure together with the full solution can help to solve the game strongly. In Connect-Four the formulation of the knowledge rules—all understandable to human beings—was in itself already sufficient to solve the game. In contrast, human-understandable rules for Go-Moku were in itself not sufficient to solve the game [116].

The success of retrograde analysis resides in exhaustive enumeration. It seems to imply that knowing the value of a terminal position leads to perfect knowledge of all positions at a distance of one to the terminal position, and so on. Certainly the result is a perfectly-playing database, which however leads to our first question: can perfect knowledge obtained from solved games be translated into rules and strategies which human beings can assimilate? The answer is clearly dependent on the capacity of the human brain. The most difficult solved (end)games are still a mystery for human experts, e.g., the 262-move Chess endgame of Fig. 5. However, there have been various attempts to bridge the gap between perfect knowledge stored in bits and perfect strategies stored in brains [55,78,84,88,99]. Some of these approaches succeeded for relatively small domains. In complex domains, such as KBBKN, the best we can hope for is that perfect knowledge is translated into a *correct* strategy. Timman's play in the game Timman–Speelman (see Section 2.4) is an example of such a translation.

The next question then is: are such rules generic, or do they constitute a multitude of ad hoc recipes? The current state of the art of machine-learning programs is that many ad hoc recipes are produced. Moreover, they are hardly intelligible to human experts. In fact, the database itself is a long list of ad hoc recipes. Hence, the research question is how to combine them into tractable clusters of analogue positions and then to formulate a human-understandable rule. As stated above many attempts have been undertaken. So far they did not result in breakthroughs in complex domains. An overview of the learning methods was given by Fürnkranz [41].

The database can also be used as a corrector of human strategies formulated by experts. Bramer did so, and he expanded his correct set of understandable rules for the endgame KRK to a perfectly-playing set of rules [15]. The technique has not seen many followers, presumably since formulating a correct set of rules for a complex domain is already a difficult task. Nevertheless, the idea that humans could learn from the performances of computer programs is quite clear. Next to Timman, who learned the KBBKN strategy from database publications, it has been seen that Backgammon players learned new strategies from the program TD-GAMMON [107] and Scrabble players from the program MAVEN [100].

This brings us to the third question: can methods be transferred between games? Assume that we have perfect knowledge and that we have almost no other game knowledge. In order to find rules we have to formulate a few concepts. Often, elementary concepts are given to the learning programs, that then generate some relations. Here we arrive at the

Table 7
Predicted program strengths for the Computer Olympiad games in the year 2010

Solved or cracked	Over champion	World champion	Grand master	Amateur
Awari	Chess	Go (9 × 9)	Bridge	Go (19 × 19)
Othello	Draughts (10 × 10)	Chinese Chess	Shogi	
Checkers (8 × 8)	Scrabble	Hex		
	Backgammon	Amazons		
	Lines of Action			

domain of data mining, which has a long list of proven methods [38,49]. Indeed, these methods can be transferred between games. However, until now, they have not provided any success. For accurate strategies it is necessary to understand all the details and subtleties hidden in the perfect knowledge. This means that understanding many intricacies of a game is a prerequisite to applying one of the proven methods out of the data-mining areas successfully. But now we have a chicken-and-egg problem, since the adequate knowledge of the intricacies mentioned is also hidden in the perfect knowledge. Our current conclusion is that each (end)game is a law unto itself and will remain so for a long time, if not forever.

6.2. New predictions

As a sequel to the predictions of 1990 we offer a new prediction for 2010 on the strength of computer programs for all games played at the Computer Olympiads. Games that have been solved are eliminated from the list. The new prediction is given in Table 7. In the framework of this article the prediction states that Awari, Othello and Checkers (8 × 8) will be solved in 2010. In Scrabble, computers are believed to be close to perfect play [100].

6.3. New games

Two games not mentioned so far are Lines of Action and Amazons. The prospects of both are rather different. Lines of Action (LOA) is a game for which interest only arose recently. At the fifth Computer Olympiad three strong LOA programs participated: YL and MONA, both from the University of Alberta, and MIA from the Universiteit Maastricht. Though LOA has a complexity similar to Othello [119], research on *solving* LOA only has been marginal and we therefore expect this game not to be solved before 2010. Yet, we assume a weak solution is possible but we expect it in the next decade.

Amazons is a game with a complexity comparable to that of Go. However, for competitive programs it has the advantage that relatively simple evaluation functions work quite reasonable [50]. Moreover, due to its nature, an Amazons game often quickly decomposes into independent subproblems, which are often one-player puzzles. It was shown that solving such an Amazons puzzle is an NP-complete task and consequently that determining the winner of an Amazons endgame is NP-hard [25]. However, approximate solutions and the use of planning techniques may establish that play in the endgame phase by an Amazons program will develop to an extraordinarily strong component, by which programs may rather easily reach expert level. Further progress seems more difficult, mainly due to the excessive branching factor of the game. In the standard game the first player has 2176 possible moves in the opening position and the average branching factor is between 300 and 400 [52]. Therefore it may be expected that Amazons will only be solved on relatively small boards. Since a game starts with eight amazons and every move exactly fills one empty square, we must keep in mind that the initial position on $m \times m$ boards with odd m favours the first player, while the second player has such an advantage if m is even. The advantage of the initiative, if any, may possibly intensify the first-player's advantage on odd boards and counterbalance the second-player's advantage on even boards. At present only some analyses up to the 5×5 board are available [52,81]. The 3×3 board is a trivial first-player win, with only one empty square to be filled in the first move. The 4×4 board, with eight empty squares in the start position, yields a second-player win. The 5×5 board, with seventeen empty squares, is again a first-player win.

Results in the current article show that a merge of combinatorial game theory with methods from computer science is quite successful. It is therefore promising that many additional games with mathematical properties recently have come to the attention of computer scientists. Challenging games include Gipf, Tamsk, Zèrtz, Twixt, and Onyx.

While the inclusion of Awari in the list of games to be solved in this decade seems safe, the other two games included in this class are far from certain. For both, Othello and Checkers, much effort has already been spent to solve them, but there are signs that finding a solution will not be easy. Time will tell whether the current prediction is as accurate as the 1990 prediction.

Acknowledgements

The authors gratefully acknowledge the work performed by all researchers mentioned in this article. Many of them either worked at one of the authors' departments or collaborated closely with them. Without their efforts many games would still have been hidden in the dark and the present overview article would not have been possible.

References

[1] J. Allen, A note on the computer solution of Connect-Four, in: D.N.L. Levy, D.F. Beal (Eds.), Heuristic Programming in Artificial Intelligence: The First Computer Olympiad, Ellis Horwood, Chichester, 1989, pp. 134–135.

[2] L.V. Allis, A knowledge-based approach of Connect Four: The game is over, white to move wins, M.Sc. Thesis, Vrije Universiteit Report No. IR-163, Faculty of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, 1988.

[3] L.V. Allis, M. van der Meulen, H.J. van den Herik, Databases in Awari, in: D.N.L. Levy, D.F. Beal (Eds.), Heuristic Programming in Artificial Intelligence 2: The Second Computer Olympiad, Ellis Horwood, Chichester, 1991, pp. 73–86.

[4] L.V. Allis, H.J. van den Herik, I.S. Herschberg, Which games will survive?, in: D.N.L. Levy, D.F. Beal (Eds.), Heuristic Programming in Artificial Intelligence 2: The Second Computer Olympiad, Ellis Horwood, Chichester, 1991, pp. 232–243.

[5] L.V. Allis, M. van der Meulen, H.J. van den Herik, Proof-number search, Artificial Intelligence 66 (1) (1994) 91–124.

[6] L.V. Allis, P.N.A. Schoo, Qubic solved again, in: H.J. van den Herik, L.V. Allis (Eds.), Heuristic Programming in Artificial Intelligence 3: The Third Computer Olympiad, Ellis Horwood, Chichester, 1992, pp. 192–204.
[7] L.V. Allis, Searching for solutions in games and artificial intelligence, Ph.D. Thesis, University of Limburg, Maastricht, 1994.
[8] L.V. Allis, H.J. van den Herik, M.P.H. Huntjens, Go-Moku solved by new search techniques, Comput. Intelligence: An Internat. J. 12 (1) (1995) 7–24.
[9] V.V. Anshelevich, The game of Hex: An automatic theorem proving approach to game programming, in: Proc. AAAI-2000, Austin, TX, AAAI Press, Menlo Park, CA, 2000, pp. 189–194.
[10] V.V. Anshelevich, A hierarchical approach to computer Hex, Artificial Intelligence 134 (2002) 101–120 (this issue).
[11] R.C. Bell, Board and Table Games from Many Civilisations, Dover Publications, New York, 1979.
[12] E.R. Berlekamp, J.H. Conway, R.K. Guy, Winning Ways for your Mathematical Plays. Vol. 1: Games in General, Academic Press, London, 1982.
[13] E.R. Berlekamp, J.H. Conway, R.K. Guy, Winning Ways for your Mathematical Plays. Vol. 2: Games in Particular, Academic Press, London, 1982.
[14] E.R. Berlekamp, The economist's view of combinatorial games, in: R.J. Nowakowski (Ed.), Games of No Chance, MSRI Publications, Vol. 29, Cambridge University Press, Cambridge, MA, 1996, pp. 365–405.
[15] M.A. Bramer, Machine-aided refinement of correct strategies for the endgame in chess, in: M.R.B. Clarke (Ed.), Advances in Computer Chess, Vol. 3, Pergamon Press, Oxford, 1982, pp. 93–112.
[16] D.M. Breuker, L.V. Allis, H.J. van den Herik, I.S. Herschberg, A database as a second, ICCA J. 15 (1) (1992) 28–39.
[17] D.M. Breuker, J. Gnodde, The AST Fourth Computer Olympiad, ICCA J. 15 (3) (1992) 152–153.
[18] D.M. Breuker, J.W.H.M. Uiterwijk, H.J. van den Herik, Replacement schemes for transposition tables, ICCA J. 17 (4) (1994) 183–193.
[19] D.M. Breuker, J.W.H.M. Uiterwijk, H.J. van den Herik, Replacement schemes and two-level tables, ICCA J. 19 (3) (1996) 175–180.
[20] D.M. Breuker, Memory versus search in games, Ph.D. Thesis, Universiteit Maastricht, Maastricht, 1998.
[21] D.M. Breuker, J.W.H.M. Uiterwijk, H.J. van den Herik, Solving 8×8 Domineering, Theoret. Comput. Sci. 230 (2000) 195–206.
[22] A. de Bruin, W. Pijls, A. Plaat, Solution trees as a basis for game-tree search, ICCA J. 17 (4) (1994) 207–219.
[23] M. Buro, The Othello match of the year: Takeshio Murakami vs. Logistello, ICCA J. 20 (3) (1997) 189–193.
[24] M. Buro, How machines have learned to play Othello, IEEE Intelligent Systems J. 14 (6) (1999) 12–14.
[25] M. Buro, Simple Amazons endgames and their connection to Hamilton circuits in cubic subgrid graphs, in: T.A. Marsland, I. Frank (Eds.), Computers and Games 2000, Lecture Notes in Computer Science, Vol. 2063, Springer, New York, 2001, pp. 263–274.
[26] P.J. Campbell, D.P. Chavey, Tchuka Ruma Solitaire, The UMAP Journal 16 (4) (1995) 343–365.
[27] T. Cazenave, Abstract proof search, in: T.A. Marsland, I. Frank (Eds.), Computers and Games 2000, Lecture Notes in Computer Science, Vol. 2063, Springer, New York, 2001, pp. 40–55.
[28] T. Cazenave, Personal communication, 2001.
[29] S. Chinchalkar, An upper bound for the number of reachable positions, ICCA J. 19 (3) (1996) 181–183.
[30] M.R.B. Clarke, A quantitative study of King and Pawn against King, in: M.R.B. Clarke (Ed.), Advances in Computer Chess, Vol. 1, Edinburgh University Press, Edinburgh, 1977, pp. 108–118.
[31] J.H. Conway, On Numbers and Games, Academic Press, London, 1976.
[32] S.T. Dekker, H.J. van den Herik, I.S. Herschberg, Perfect knowledge revisited, Artificial Intelligence 43 (1) (1990) 111–123.
[33] H.H.L.M. Donkers, A. de Voogt, J.W.H.M. Uiterwijk, Human versus machine problem-solving: Winning openings in Dakon, Board Games Studies 3 (2000) 79–88.
[34] H. Enderton, Personal communication, 1999.
[35] H. Enderton, Answers to infrequently asked questions about the game of Hex, Web page http://www.cs.cmu.edu/People/hde/hex/hexfaq.

[36] S. Even, R.E. Tarjan, A combinatorial problem which is complete in polynomial space, J. ACM 23 (1976) 710–719.

[37] H.-r. Fang, T.-s. Hsu, S.-c. Hsu, Construction of Chinese Chess endgame databases by retrograde analysis, in: T.A. Marsland, I. Frank (Eds.), Computers and Games 2000, Lecture Notes in Computer Science, Vol. 2063, Springer, New York, 2001, pp. 99–118.

[38] U. Fayyad, Advances in Knowledge Discovery and Data Mining, MIT Press, Cambridge, MA, 1996.

[39] J. Feinstein, Amenor wins world 6×6 championships!, British Othello Federation Newsletter (July 1993) 6–9. Also available at URL http://www.maths.nott.ac.uk/othello/Jul93/Amenor.html.

[40] B. Fraser, Personal communication, 2001.

[41] J. Fürnkranz, Machine learning in computer chess: The next generation, ICCA J. 19 (3) (1996) 147–161.

[42] D. Gale, The game of Hex and the Brouwer fixed point theorem, Amer. Math. Monthly (1986) 818–827.

[43] M. Gardner, Mathematical games, Scientific American 230 (2) (1974) 106–108.

[44] R.U. Gasser, Harnessing computational resources for efficient exhaustive search, Ph.D. Thesis, ETH Zürich, Switzerland, 1995.

[45] R.U. Gasser, Solving Nine Men's Morris, in: R.J. Nowakowski (Ed.), Games of No Chance, MSRI Publications, Vol. 29, Cambridge University Press, Cambridge, MA, 1996, pp. 101–113.

[46] M. Ginsberg, Partition search, in: Proc. AAAI-96, Portland, OR, 1996, pp. 228–233.

[47] S.W. Golomb, Polyominoes, Charles Scribner's Sons, New York, 1965; Revised and expanded edition, Princeton University Press, Princeton, NJ, 1994.

[48] R. van der Goot, Awari retrograde analysis, in: T.A. Marsland, I. Frank (Eds.), Computers and Games 2000, Lecture Notes in Computer Science, Vol. 2063, Springer, New York, 2001, pp. 89–98.

[49] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, CA, 2001.

[50] T. Hashimoto, Y. Kajihara, N. Sasaki, H. Iida, J. Yoshimura, An evaluation function for Amazons, in: H.J. van den Herik, B. Monien (Eds.), Advances in Computer Games, Vol. 9, Universiteit Maastricht, Maastricht, pp. 191–202.

[51] G.M^{c}C. Haworth, Strategies for constrained optimisation, ICGA J. 23 (1) (2000) 9–20.

[52] P.P.L.M. Hensgens, A knowledge-based approach of the game of amazons, M.Sc. Thesis, Universiteit Maastricht, Maastricht, 2001.

[53] H.J. van den Herik, I.S. Herschberg, The construction of an omniscient endgame data base, ICCA J. 8 (2) (1985) 66–87.

[54] H.J. van den Herik, I.S. Herschberg, A data base on data bases, ICCA J. 9 (1) (1986) 29–34.

[55] H.J. van den Herik, I.S. Herschberg, Omniscience, the rulegiver?, in: B. Pernici, M. Somalvico (Eds.), Proceedings of L'Intelligenza Artificiale Ed Il Gioco Degli Scacchi, III^{o} Convegno Internazionale, 1986, pp. 1–17.

[56] H.J. van den Herik, L.V. Allis (Eds.), Heuristic Programming in Artificial Intelligence 3: The Third Computer Olympiad, Ellis Horwood, Chichester, 1992.

[57] H.J. van den Herik, H. Iida (Eds.), Computers and Games, Lecture Notes in Computer Science, Vol. 1558, Springer, Berlin, 1999.

[58] H.J. van den Herik, H. Iida (Eds.), Games in AI Research, Universiteit Maastricht, Maastricht, 2000.

[59] H.J. van den Herik et al., The Fifth Computer Olympiad, ICGA J. 23 (3) (2000) 164–187.

[60] H.J. van den Herik, B. Monien (Eds.), Advances in Computer Games, Vol. 9, Universiteit Maastricht, Maastricht, 2001.

[61] I.S. Herschberg, H.J. van den Herik, Back to fifty, ICCA J. 16 (1) (1993) 1–2.

[62] G. Irving, H.H.L.M. Donkers, J.W.H.M. Uiterwijk, Solving Kalah, ICGA J. 23 (3) (2000) 139–147.

[63] P.J. Jansen, Using knowledge about the opponent in game-tree search, Ph.D. Thesis, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1992.

[64] A. Junghanns, Are there practical alternatives to alpha-beta?, ICCA J. 21 (1) (1998) 14–32.

[65] J. Kling, D. Horwitz, Chess Studies, or Endings of Games, Skeet, London, 1851.

[66] R.E. Korf, Depth-first iterative-deepening: An optimal admissible tree search, Artificial Intelligence 27 (1) (1985) 97–109.

[67] R.E. Korf, Linear-space best-first search, Artificial Intelligence 62 (1) (1993) 41–78.

[68] M. Lachmann, C. Moore, I. Rapaport, Who wins Domineering on rectangular boards?, MSRI Workshop on Combinatorial Games, Berkeley, CA, 2000.

[69] D.N.L. Levy, D.F. Beal (Eds.), Heuristic Programming in Artificial Intelligence: The First Computer Olympiad, Ellis Horwood, Chichester, 1989.
[70] D.N.L. Levy, D.F. Beal (Eds.), Heuristic Programming in Artificial Intelligence 2: The Second Computer Olympiad, Ellis Horwood, Chichester, 1991.
[71] T.T. Lincke, A. Marzetta, Large endgame databases with limited memory space, ICGA J. 23 (3) (2000) 131–138.
[72] T.T. Lincke, Personal communication, 2001.
[73] T.T. Lincke, R. van der Goot, MARVIN wins Awari tournament, ICGA J. 23 (3) (2000) 173–174.
[74] T.A. Marsland, I. Frank (Eds.), Computers and Games 2000, Lecture Notes in Computer Science, Vol. 2063, Springer, New York, 2001.
[75] H. Matsubara, Shogi (Japanese chess) as the AI research target next to chess, ETL Technical Report ETL-TR-93-23, Electrotechnical Laboratory, Tsukuba, Japan, 1993.
[76] H. Matsubara, H. Iida, R. Grimbergen, Natural developments in game research: From Chess to Shogi to Go, ICCA J. 19 (2) (1996) 103–112.
[77] D.A. McAllester, Conspiracy numbers for min-max search, Artificial Intelligence 35 (3) (1988) 287–310.
[78] D. Michie, I. Bratko, Ideas on knowledge synthesis stemming from the KBBKN endgame, ICCA J. 10 (1) (1987) 3–13.
[79] M. Müller, Computer go as a sum of local games: An application of combinatorial game theory, Ph.D. Thesis, ETH Zürich, Zürich, 1995.
[80] M. Müller, Generalized thermography: A new approach to evaluation in computer Go, in: H.J. van den Herik, H. Iida (Eds.), Games in AI Research, Universiteit Maastricht, Maastricht, 2000, pp. 203–219.
[81] M. Müller, Computer Amazons, Web page http://www.cs.ualberta.ca/~mmueller/amazons/index.html.
[82] E.V. Nalimov, G.M^{c}C. Haworth, E.A. Heinz, Space-efficient indexing of chess endgame tables, ICGA J. 23 (3) (2000) 148–162.
[83] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behavior, 1944. Second Edition, Princeton University Press, Princeton, NJ, 1947.
[84] J. Nunn, Extracting information from endgame databases, in: H.J. van den Herik, I.S. Herschberg, J.W.H.M. Uiterwijk (Eds.), Advances in Computer Chess, Vol. 7, University of Limburg, Maastricht, 1994, pp. 19–34.
[85] H.K. Orman, Pentominoes: A first player win, in: R.J. Nowakowski (Ed.), Games of No Chance, MSRI Publications, Vol. 29, Cambridge University Press, Cambridge, MA, 1996, pp. 339–344.
[86] O. Patashnik, Qubic: $4 \times 4 \times 4$ Tic-Tac-Toe, Mathematical Magazine 53 (1980) 202–216.
[87] A. Plaat, J. Schaeffer, A. de Bruin, W. Pijls, A minimax algorithm better than SSS*, Artificial Intelligence 87 (1–2) (1996) 255–293.
[88] J.R. Quinlan, Discovering rules by induction from large collections of examples, in: D. Michie (Ed.), Expert Systems in the Micro-Electronic Age, Edinburgh University Press, Edinburgh, 1979, pp. 168–201.
[89] S. Reisch, Hex ist PSPACE-vollständig, Acta Informatica 15 (1981) 167–191.
[90] J. van Rijswijck, Computer Hex: Are bees better than fruitflies?, M.Sc. Thesis, University of Alberta, Edmonton, AB, 2000.
[91] J. van Rijswijck, Queenbee's home page, Web page http://www.cs.ualberta.ca/~queenbee.
[92] L. Russ, The Complete Mancala Games Book, Second Edition, Marlowe & Company, New York, 2000.
[93] J. Schaeffer, Conspiracy numbers, in: D.F. Beal (Ed.), Advances in Computer Chess, Vol. 5, Elsevier Science, Amsterdam, 1989, pp. 199–218. Also published in: Artificial Intelligence 43 (1) (1990) 67–84.
[94] J. Schaeffer, J. Culberson, N. Treloar, B. Knight, P. Lu, D. Szafron, Reviving the game of checkers, in: D.N.L. Levy, D.F. Beal (Eds.), Heuristic Programming in Artificial Intelligence 2: The Second Computer Olympiad, Ellis Horwood, Chichester, 1991, pp. 119–136.
[95] J. Schaeffer, R. Lake, Solving the game of checkers, in: R.J. Nowakowski (Ed.), Games of No Chance, MSRI Publications, Vol. 29, Cambridge University Press, Cambridge, MA, 1996, pp. 119–133.
[96] J. Schaeffer, One Jump Ahead: Challenging Human Supremacy in Checkers, Springer, New York, 1997.
[97] S. Sei, T. Kawashima, A solution of Go on 4×4 board by game tree search program, Fujitsu Social Science Laboratory, 2000, Manuscript.
[98] M. Seo, H. Iida, J.W.H.M. Uiterwijk, The PN*-search algorithm: Application to Tsume-Shogi, Artificial Intelligence 129 (1–2) (2001) 253–277.

[99] A.D. Shapiro, Structured Induction in Expert Systems, Turing Institute Press in association with Addison-Wesley Publishing Company, Wokingham, 1987.

[100] B. Sheppard, World-championship-caliber Scrabble, Artificial Intelligence 134 (2002) 241–275 (this issue).

[101] D. Singmaster, Almost all games are first person games, Eureka 41 (1981) 33–37.

[102] D. Singmaster, Almost all partizan games are first person and almost all impartizan games are maximal, J. Combin. Inform. System Sci. 7 (1982) 270–274.

[103] B. Spight, Personal communication, 2001.

[104] L. Stiller, Parallel analysis of certain endgames, ICCA J. 12 (2) (1989) 55–64.

[105] L. Stiller, Exploiting symmetry on parallel architectures, Ph.D. Thesis, The Johns Hopkins University, Baltimore, MD, 1995.

[106] T. Ströhlein, Untersuchungen über kombinatorische Spiele, M.Sc. Thesis, Fakultät für Allgemeine Wissenschaften der Technischen Hochschule München, 1970.

[107] G. Tesauro, TD-Gammon, a self-teaching Backgammon program, achieves master-level play, Neural Comput. 6 (2) (1994) 215–219.

[108] K. Thompson, A.J. Roycroft, A prophecy fulfilled, EG 74 (1983) 217–220.

[109] K. Thompson, Retrograde analysis of certain endgames, ICCA J. 9 (3) (1986) 131–139.

[110] K. Thompson, 6-piece endgames, ICCA J. 19 (4) (1996) 215–226.

[111] T. Thomsen, Lambda-search in game trees—With application to Go, ICGA J. 23 (4) (2000) 203–217.

[112] E.O. Thorp, W.E. Walden, A computer-assisted study of Go on $M \times N$ boards, Information Sciences 4 (1972) 1–33. Reprinted in: D.N.L. Levy (Ed.), Computer Games II, Springer, New York, 1988, pp. 152–181.

[113] K.-M. Tsao, H. Li, S.-C. Hsu, Design and implementation of a Chinese Chess program, in: D.N.L. Levy, D.F. Beal (Eds.), Heuristic Programming in Artificial Intelligence 2: The Second Computer Olympiad, Ellis Horwood, Chichester, 1991, pp. 108–118.

[114] J.W.H.M. Uiterwijk, L.V. Allis, H.J. van den Herik, A knowledge-based approach to Connect-Four. The game is solved!, in: D.N.L. Levy, D.F. Beal (Eds.), Heuristic Programming in Artificial Intelligence: The First Computer Olympiad, Ellis Horwood, Chichester, 1989, pp. 113–133.

[115] J.W.H.M. Uiterwijk, Kennisbehandeling in positionele spelen: Computer-analyse van Four-in-a-Row, in: J. Treur (Ed.), Proc. NAIC'91, Stichting Informatica Congressen, Amsterdam, 1991, pp. 193–207.

[116] J.W.H.M. Uiterwijk, Knowledge and strategies in Go-Moku, in: H.J. van den Herik, L.V. Allis (Eds.), Heuristic Programming in Artificial Intelligence 3: The Third Computer Olympiad, Ellis Horwood, Chichester, 1992, pp. 165–179.

[117] J.W.H.M. Uiterwijk, H.J. van den Herik, The advantage of the initiative, Information Sciences 122 (1) (2000) 43–58.

[118] J. Wágner, I. Virág, Solving Renju, ICGA J. 24 (1) (2001) 30–34.

[119] M.H.M. Winands, J.W.H.M. Uiterwijk, H.J. van den Herik, The quad heuristic in Lines of Action, ICGA J. 24 (1) (2001) 3–14.

[120] R. Wu, D.F. Beal, Computer analysis of some Chinese Chess endgames, in: H.J. van den Herik, B. Monien (Eds.), Advances in Computer Games, Vol. 9, Universiteit Maastricht, Maastricht, 2001, pp. 261–273.