On Strongly Solving Chinese Checkers

Nathan R. Sturtevant(✉)

University of Alberta, Edmonton, AB, Canada
nathanst@ualberta.ca

Abstract. Chinese Checkers is a game for 2–6 players that has been used as a testbed for game AI in the past. The game is easily scalable to different size boards, different numbers of players, and different numbers of pieces for each player. In this paper we provide an overview of what is required to strongly solve versions of the game, including a complete set of rules needed to solve the game. We provide results on smaller boards with result showing that these games are all a first-player win.

Keywords: Solve · Game · Chinese Checkers

1 Introduction

This paper studies the problem of strongly solving variants of Chinese Checkers, and provides the foundation needed to strongly solve them. Chinese Checkers is closely related to the game of Halma, which was invented around 1883–1884, except that Chinese Checkers is played on a star-shaped board while Halma is played on a square board [4]. It is typically played by 2–6 players, with the goal of moving your pieces across the board into a goal area before your opponent. There are two reasons why we are interested in studying this game and strongly solving variants of the game.

First, many traditional two-player perfect information games, such as Connect-Four [2], Awari [9], Checkers [13], and Hex [6] have been strongly solved or weakly solved, so solving Chinese Checkers follows in this line of work. One common board size for Chinese Checkers has a 1.73×10^{24} states, and thus might be weakly solvable, given a good proof strategy. However, in our work we have found it difficult to construct small proofs for the game. Thus, by strongly solving small versions of the game, we can study the nature of the game and learn how to build compact proofs for the game.

Second, recent work on AlphaZero [14] has suggested a common approach for learning to play deterministic two-player games with perfect information. As Chinese Checkers falls into this category, we expect that the AlphaZero approach is able to learn to play the game. But, given that we can strongly solve some board sizes, this offers the opportunity to precisely measure and evaluate how learning takes place in the game. Thus, we suggest that Chinese Checkers will be a good testbed for evaluating learning in a two-player deterministic perfect information game.

T. Cazenave et al. (Eds.): ACG 2019, LNCS 12516, pp. 155–166, 2020.
https://doi.org/10.1007/978-3-030-65883-0_13

Given this reasoning, we have built a number of solvers for the game of Chinese Checkers with different board sizes and different numbers of pieces on the board. Through analysis of these solvers we can now provide a deeper analysis of Chinese Checkers than has previously been found in the literature, including comprehensive rules for wins, losses, draws, and illegal positions in the game. This paper provides an overview of these insights, as well as the results of strongly solving games with up to the 6×6 board with 6 pieces per player which has 2,313,100,389,600 positions. All games that we have solved have been a first-player win.

2 Background and Related Work

Victor Allis defined three different types of solved games. These are defined as [1]:

- **Ultra-weakly solved.** For the initial position(s), the game-theoretical value has been determined.
- **Weakly solved.** For the initial position(s), a strategy has been determined to obtain at least the game-theoretical value of the game, for both players, under reasonable resources.
- **Strongly solved.** For all legal positions, a strategy has been determined to obtain the game-theoretic value of the position, for both players, under reasonable resources.

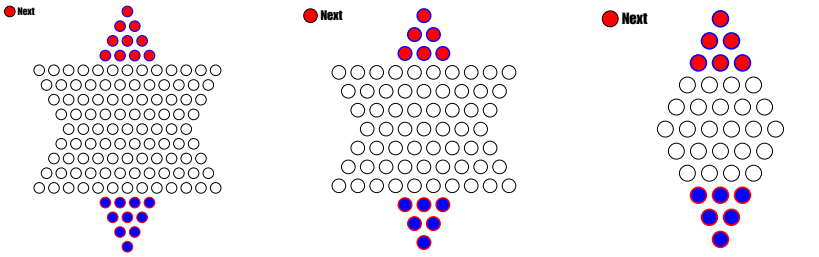
While reasonable resources might change over time, it is suggested that a limit of several minutes of computation per move should be allowed. Thus, storing the result of every position would qualify for a game being strongly solved. But, sometimes there are too many positional combinations to store efficiently, requiring the use of search to dynamically re-compute this data when making queries about a state. In such cases, the a game would still be considered to be strongly solved even if it took a few minutes to perform these re-computations.

Given these classifications, significant research has gone into both solving games or and subgames of a game, such as in endgame databases [3,12].

Connect-Four was one of the first non-trivial games to be solved. One of the original proofs used specialized knowledge to build a small proof tree for the game [2], although the full game can now be enumerated efficiently [5], so the game is now strongly solved. Awari [9] was strongly solved using parallel hardware; the value of 889,063,398,406 positions was determined during this proof. Pentago [7] has also been strongly solved on parallel hardware, with 3,009,081,623,421,558 positions in the game. In the Pentago solution positions with more than 18 stones are not recorded on disk, as it requires too much storage and the value of the states can be re-computed easily.

Weakly solved games include Checkers [13], which is solved for the primary opening moves, and 8×8 Hex [6], which is solved for any opening move.

Chinese Checkers has been a common domain used in testing for multi-player games [11,16,18]. A common heuristic used for play is the single-agent distance



(a) 9x9 start state (b) 7x7 start state (c) 6x6 start state

Fig. 1. Examples of different board sizes.

to the goal [16], which can be easily calculated on smaller board sizes. The single-agent distances on the full board size can also be calculated [19] and used for game play [15], but other approaches such as MCTS [10] and TD learning [15] have also been explored for playing Chinese Checkers.

2.1 Terminology

In this paper we use the terms *board*, *board position*, and *state* interchangeably. The term *jumps* can be substituted anytime we use the term *hops*. We refer to pieces on the board, but these can also be referred to as *marbles*, as physical versions of the game often use marbles for pieces. We will also talk about *corners* of the board and goal or start *areas* interchangeably.

3 Rules of Chinese Checkers

If we wish to strongly solve variants of Chinese Checkers, we must be able to determine the value of every state in the game. This requires resolving a number of special cases around how the game is played. In this section we work through and propose resolutions for each of these special cases. This paper focuses primarily on the two-player game.

Chinese Checkers is typically played on a star-shaped board with six corners, as shown in Fig. 1(a) with 10 pieces per player. In the two-player game, player 1 and player 2's pieces start at the top and bottom of the board, respectively, and the goal is for a player to get their pieces into their goal area, which, in this case, is the corner where the opponent's pieces started. Pieces are not typically allowed to move into the other corners of the board, although the rules sometimes allow for pieces to move through these areas as long as they do not remain in one of the other corners at the end of a turn.[1]

[1] Our current two-player implementation does not allow this, but we are considering adding this for 7×7 boards. The rule does not seem to play an important role in optimal play and may be more useful in the n-player version of the game ($n > 2$) when space is more constrained.

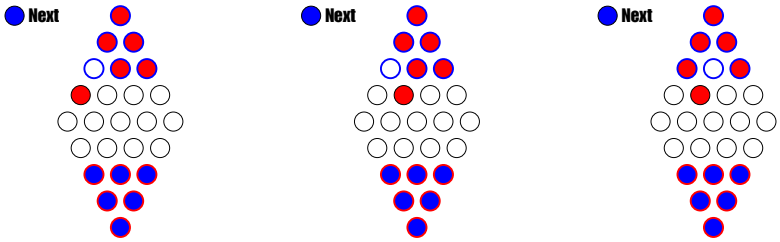


Fig. 2. Adjacent moves in Chinese Checkers.

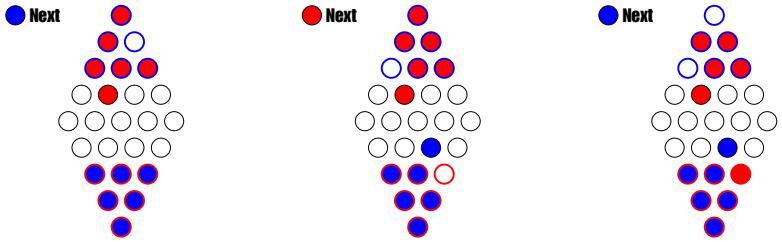
Given that a player cannot move into corners of the board besides their start and goal corners, the primary game play takes place on the center diamond of the board, which can be represented as a $m \times m$ grid. The game in Fig. 1(a) has a 9×9 gameplay area with 10 pieces per player. Another common board size used in research [16] is shown in Fig. 1(b), which is a 7×7 gameplay area with 6 pieces per player. It is possible to draw the board as a star when the size of the gameplay area is odd. If we use an even-sized gameplay area, then it is not possible to draw uniform-sized corners, and thus it is only to play these sized boards with two players, as shown in Fig. 1(c). In the remainder of the paper we will only draw the main gameplay area, and we will give each player anywhere from one to six pieces.

3.1 Movement Rules

In the game of Chinese Checkers there are two types of moves that can be performed, moves to an adjacent location, or hops over adjacent pieces to non-adjacent locations. Figure 2 shows moves to adjacent locations from the starting position of the game. There are six possible adjacent moves at the beginning of the game; three of the moves are shown in the figure. The remaining moves result in positions that are symmetric to the ones shown, something that will be discussed in Sect. 3.5.

The second type of move hops over an adjacent piece to land on an empty location on the other side. A hop from the starting position is illustrated in Fig. 3(a). If multiple hops are available for a single piece, they can be chained together into longer hops that cross the entire board. Figure 3(b) sets up an alternate board position where additional hops can be chained, allowing a piece to move across the board, giving the position in Fig. 3. A piece can take any number of legal hops in a turn, and can stop when further hops are still possible. Additionally, pieces are not removed from the board when they are hopped over by another pieces.

One notable feature of Chinese Checkers is that the game is not acyclic – it is possible to return to the same position and not make progress. This makes Chinese Checkers more complex to analyze than games like Pentago, which adds a new piece to the board every turn.



(a) Single hop

(b) Board position that allows multiple hops.

(c) Board position after three hops are taken.

Fig. 3. Hopping moves in Chinese Checkers.

3.2 Winning Conditions

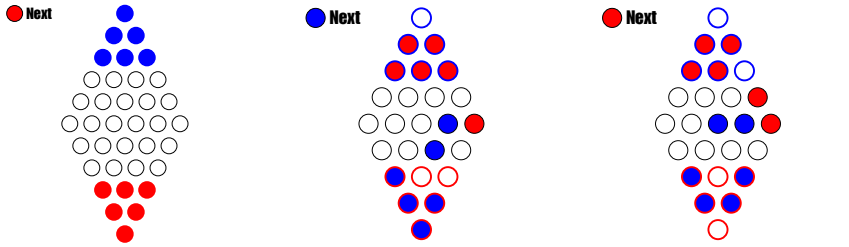
In the most general case a player wins a game of Chinese Checkers if they get all of their pieces into their goal area on the opposite side of the board. In normal play this is achievable, but for exhaustive analysis we need a more precise definition of the winning conditions. In particular, it has been observed that a single player can leave one piece in their start area, preventing the opponent from ever getting their pieces into the goal. This rule is typically not handled in rule books for the game, although web sites discussing the game often make suggestions for handling this. One suggested handling is as follows:[2]

> "If one or more of a player's marbles are imprisoned in his/her original starting point so that the marbles cannot be moved, such player forfeits the game. In the event of multiple players, the imprisoned marbles are removed and the game continues as before."

However, this rule is inadequate. It is possible for a player to place a piece on the outer row of their goal area so that it cannot be imprisoned, but it will still block the other play from winning. Other suggestions that we have seen for handling this have also been inadequate. Thus, we propose the combination of two rules for wins and illegal states to force an end to the game in these types of situations. The first has been traditional used in our implementation of the game, while the second was adapted based on our ongoing work.

Definition 1. *A state in Chinese Checkers is* won *for player n if player n's goal area is filled with pieces, and at least one of the pieces belongs to player n.*

[2] http://www.abstractstrategy.com/chinese-checkers-g.html#c-checkers-rules.



(a) Unreachable state (b) Illegal state for red (c) Illegal state for blue

Fig. 4. Illegal states in Chinese Checkers. (Color figure online)

Under this definition if a player leaves some pieces in their goal area, the other player can just fill in around these pieces in order to win the game. For very small versions of the game this can have unintended consequences, as shown in Fig. 3(c). After only 3 ply this state qualifies as a win for the red player, because the red player has a piece in their goal area, and the remainder of the home is filled with blue pieces. Such shallow goal states are not found on larger boards, and the small boards are not used widely, so the shallow goal states seem acceptable. Note that even Chess has possible goal states just 4 ply into the game.

This definition has been adequate for our past experiments with Chinese Checkers playing agents, but it is not adequate for strongly solving the game when there are six pieces on the board, as there are still some conditions it cannot catch. However, instead of modifying our definition of a winning state, we declare certain board positions to be illegal, which we discuss in the next section.

3.3 Illegal States

In Chinese Checkers there are states that are unreachable via normal play. One example of an illegal position is shown in Fig. 4(a). In this state both players have their pieces in the goal area. But, because a player should win as soon as their pieces are moved into the goal area, it is impossible for both players to simultaneously have their pieces in their respective goals. Thus, this state is not reachable by normal play and should be excluded from any analysis that attempts to strongly solve the game. We do this by declaring the state to be illegal. As long as all illegal states are marked or checked appropriately, they can then be ignored in any proof procedure.

Given the definition of a winning state in the previous section, there are a large number of illegal states in the game – all combinations of states where both goals are filled with pieces. Thus, we form a general rule for illegal states, the first of two rules that are required for strongly solving the game.

Definition 2. *(Part 1) A state in Chinese Checkers is* illegal *for player n if the winning conditions are met for player n and it is player n's turn to move.*

This definition implies that it is illegal for a player to take a suicidal action that causes the other player to win. For instance, a player might move back into their own start area to fill it up when the opponent already has a piece in this area, thus creating a win for the other player even when it was their turn to move.

The other type of state that we declare to be illegal are states where a player is blocking the other player from filling their goal area. This type of position is shown in Fig. 4(b). In this state the red player has blocked the outer two rows of their start area. In such a position it is impossible for the blue player to move a single piece into their goal area, even though one of the locations in the goal area is open. This allows the red player to prevent the blue player from winning, because a game is only won when the goal area is filled with pieces, and in this case the red player can always move its other piece instead of unblocking the goal.

While we could declare a state to be illegal if two consecutive rows are blocked with empty locations behind them, our experiments showed that this rule is still inadequate, as shown in Fig. 4(c). This state shows an arrangement of the blue pieces at the bottom of the board that is sufficient to prevent the red player from filling the goal without blocking two consecutive rows. We use this position to formulate the second condition for a state to be illegal.

Definition 3. *(Part 2) A state in Chinese Checkers is* illegal *if there are one or more unoccupied locations in player n's goal area that are unreachable by player n due to another players pieces.*

These rules are only necessary on game variants with more pieces. When there are six pieces, it suffices to check the two outer edges of the goal area to see if they are occupied, and if the tip of the goal is unoccupied. On larger boards, when a player has 10 pieces, the conditions are more complicated because there are more ways to block locations within the goal area.

Under this rule the states in both Fig. 4(b) and (c) are illegal, meaning it is illegal to take an action that leads to this state. This solves one particular problem; the question remains whether it creates alternate problems in gameplay. In looking at the shortest single-player sequence of actions needed to cross the board, such positions are never reached. This is because players are focused on getting their pieces out of the start area as quickly as possible, not on fortifying the start area.

For human play against suboptimal opponents, these positions could appear as part of normal play, and we do not dismiss the possibility that we could be missing some important cases. But, in human play it is also possible to use reasonable judgement to determine if a player is intending to block the goal, and thus such a rule would only need to be applied selectively. But, since we do not have that luxury in optimal computer vs computer play, we choose to err on the side of simpler rules.

3.4 Draws

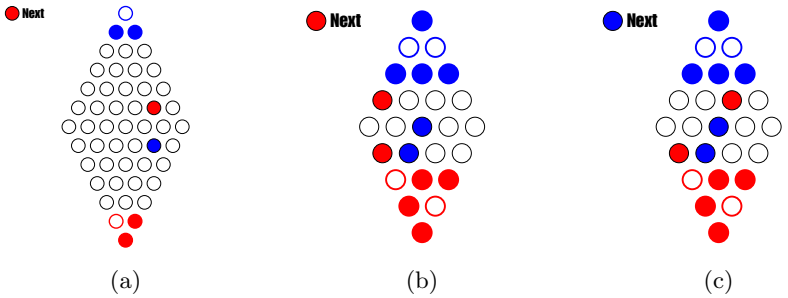


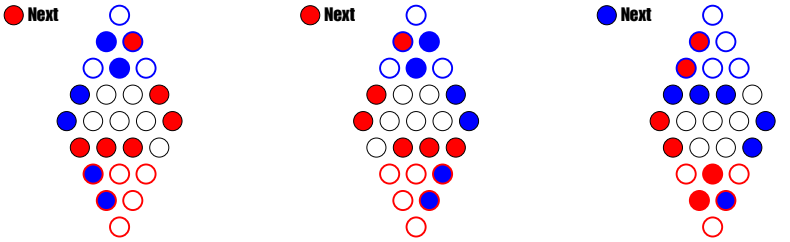
Fig. 5. Drawn states in Chinese Checkers.

In some games, such as Hex, all states can be determined to be a win or loss. The possible value of states in Chinese Checkers has not, to our knowledge, been previously determined. That is, we have unable to find any rules for the game that discuss drawn positions. If we prove all possible states that are wins, losses and illegal, yet have states that are still unable to be proven, these remaining states are drawn. Drawn states begin to appear in Chinese Checkers once there are three pieces on the board.

We illustrate one such state in Fig. 5(a). On this board each player has two pieces in the goal area, and one piece left in the middle of the board. Ignoring the blue piece, the red piece in the middle can reach the goal area in six moves. Similarly, the blue piece in the middle can also reach its goal area in six moves. However, if the red player takes the first move, it will allow the blue player to perform a jump and reach the goal in five moves, thus winning the game. Thus, this similar to a *zugzwang* position, where the player to move is at a disadvantage, except that there are other pieces that can be moved in order to delay the disadvantageous move. Instead of making progress, both player will alternate moving one of their two pieces that have already reached the goal.

Figures 5(b) and 5(c) illustrate a more complicated position that is also drawn. In 5(b) the red player moves to the state shown in 5(c) to block the blue player from performing a double hop into their goal area. As a result, the blue player will move its piece in the middle row one step to the left in order to enable a different double hop. After this the red player will undo its previous move, returning to block the blue piece. This then causes the blue player to move back to the position in 5(b). In this set of positions blue has an advantage, but red can continually block that advantage.

These figures illustrate that drawn positions can occur in Chinese Checkers. As existing rules do not account for repeated positions, we suggest the following rule.



(a) Original state (b) Left-right symmetry (c) P1-P2 symmetry

Fig. 6. Symmetric states in Chinese Checkers.

Definition 4. *A game of Chinese Checkers is* drawn *if any board state is repeated during play.*

Under this rule, if a drawn position is reached players can either repeat the position to draw the game, or a player can move into a position that is a theoretical loss, hoping that the opponent will make a mistake later in game play. While other games require positions to be repeated multiple times to cause a draw, it seems that one repetition should be sufficient.

3.5 Chinese Checkers Symmetry

Chinese Checkers has two types of symmetry. The first is right-left symmetry where flipping the board results in a position that must have an equivalent value to the original position. This type of symmetry is shown in Fig. 6(a) and 6(b). This symmetry was exploited in previous work when solving the single-agent version of Chinese Checkers [19], but also applies to the two-player version of the game. While the symmetry can be used to reduce the size of the stored solution to the game, the symmetric states will still be reached during the process of solving the game. Note also that there are some positions which are identical when flipped, and thus the savings from this symmetry is close to, but does not reach a factor of 2. On the 6×6 board with 6 pieces our implementation reduces the number of stored states by a factor of 1.998.

Chinese Checkers also has symmetry between player 1 and player 2. That is, if we flip the board top-to-bottom and swap the colors and the player to move, we also end up in a symmetric position. The state in Fig. 6(a) is symmetric to the state in Fig. 6(c).

4 Strongly Solving the Chinese Checkers

We have built a number of different solvers that strongly solve the game of Chinese Checkers in different ways, including in-memory solvers and external-memory solvers. The various solvers have been used to test the efficiency of solving techniques and to verify the correctness of each of the new implementations.

Many of the structural choices have been based on our earlier work studying the influence of solving parameters [17]. One focus of the work has been strongly solving the game efficiently using moderate hardware, as opposed to using massive parallelization to solve the game more quickly, yet less efficiently (from a CPU/power usage perspective).

Three important choices in the solving process are (1) the ranking function which orders the states in the state space (2) the order in which states are proven, and (3) immediate propagation of wins to parent states once a win is proven. On 7×7 Chinese Checkers with 3 pieces per player and well-optimized choices for these parameters, we can determine the value of 99% of the states in the state space in one pass through the data, and only 8 passes through the data are needed to complete the proof. Without parent propagation or an optimized proof ordering, the proof takes up to 34 passes through the data with only 0.5% of the states proven in the first iteration. Both proofs find the same result, but the first implementation performs the computation more efficiently.

5 Results

Results from our solves are found in Table 1. We have solved a variety of board sizes of Chinese Checkers with up to the 6×6 board with 6 pieces per player, which has 2 trillion states. The solvers use two bits per state, so given the two types of symmetry, this solution requires storing 500 billion states and 135 GB of storage, which exceeds the main memory available in the computer used for building the solution. As a result, external memory (disk) was used for this solution. A full description of this solver is outside of the scope of this paper, but our solver borrows ideas from external memory BFS implementations such as search with structured duplicate detection [20] and TBBFS [8].

Table 1. Proven results and win/loss/draw/illegal counts from various size games. All results except for the 6×6 board with 6 pieces have been solved multiple times with different solvers to validate the results.

Board Size	# Pieces	Positions	Wins	Draws	Illegal	Start
7×7	1	4,704	2,304	0	96	P1 Win
7×7	2	2,542,512	1,265,851	0	10,810	P1 Win
7×7	3	559,352,640	279,297,470	180,860	576,840	P1 Win
7×7	4	63,136,929,240	31,532,340,944	51,686,042	20,561,310	P1 Win
4×4	6	3,363,360	1,205,441	547,058	405,420	P1 Win
5×5	6	9,610,154,400	4,749,618,788	47,056,118	63,860,706	P1 Win
6×6	6	2,313,100,389,600	1,153,000,938,173	5,199,820,604	1,898,692,650	P1 Win
7×7	6	170,503,381,976,928	?	?	?	?

Because the game is symmetric between the two players, the number of wins and losses is the same. Thus, in Table 1 we only report the number of won states in the game. These counts are for the full game, ignoring symmetry. Note that

all results, except for the largest game, were computed multiple times and by multiple solvers, validating the counts. We are in the process of building a more efficient external-memory solver which will validate these results with significantly better performance. The exact count of wins should not be considered correct until this verification has been completed.

Our goal is to strongly solve the 7×7 board with 6 pieces per player, as this is the largest game that can likely be strongly solved on hardware that we have readily accessible.

6 Conclusions and Future Work

This paper has outlined the steps required to begin strongly solving games of Chinese Checkers, including rules for ending the game and for drawn states. The games solved thus far are all first-player wins, something we do not expect to change on larger board sizes. Besides providing new information about a game that has not previously been solved, the solved information provides a new testbed for studying games.

References

1. Allis, L.V.: Searching for Solutions in Games and Artificial Intelligence. Ph.D. thesis, Maastricht University (1994)
2. Allis, V.: A knowledge-based approach of Connect-Four-the game is solved: White wins. Ph.D. thesis, Vrije Universiteit (1988)
3. Björnsson, Y., Schaeffer, J., Sturtevant, N.R.: Partial information endgame databases. In: van den Herik, H.J., Hsu, S.-C., Hsu, T.-S., Donkers, H.H.L.M.J. (eds.) ACG 2005. LNCS, vol. 4250, pp. 11–22. Springer, Heidelberg (2006). https://doi.org/10.1007/11922155_2
4. Carlisle, R.P.: Encyclopedia of Play in Today's Society, vol. 1. Sage, Thousand Oaks (2009)
5. Edelkamp, S., Kissmann, P.: Symbolic classification of general two-player games. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) KI 2008. LNCS (LNAI), vol. 5243, pp. 185–192. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85845-4_23
6. Henderson, P., Arneson, B., Hayward, R.B.: Solving 8x8 hex. In: Twenty-First International Joint Conference on Artificial Intelligence (2009)
7. Irving, G.: https://perfect-pentago.net/details.html, https://perfect-pentago.net/details.html
8. Korf, R.E.: Minimizing disk i/o in two-bit breadth-first search. In: AAAI Conference on Artificial Intelligence, pp. 317–324 (2008)
9. Romein, J.W., Bal, H.E.: Solving awari with parallel retrograde analysis. Computer **36**(10), 26–33 (2003)
10. Roschke, M., Sturtevant, N.R.: UCT enhancements in chinese checkers using an endgame database. In: Cazenave, T., Winands, M.H.M., Iida, H. (eds.) CGW 2013. CCIS, vol. 408, pp. 57–70. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05428-5_5

11. Schadd, M.P., Winands, M.H.: Best reply search for multiplayer games. IEEE Trans. Comput. Intell. AI Games **3**(1), 57–66 (2011)
12. Schaeffer, J., Björnsson, Y., Burch, N., Lake, R., Lu, P., Sutphen, S.: Building the checkers 10-piece endgame databases. In: Van Den Herik, H.J., Iida, H., Heinz, E.A. (eds.) Advances in Computer Games. ITIFIP, vol. 135, pp. 193–210. Springer, Boston, MA (2004). https://doi.org/10.1007/978-0-387-35706-5_13
13. Schaeffer, J., et al.: Checkers is solved. Science **317**(5844), 1518–1522 (2007)
14. Silver, D., et al.: A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science **362**(6419), 1140–1144 (2018)
15. Sturtevant, N.: Challenges and progress on using large lossy endgame databases in chinese checkers. In: IJCAI Workshop on Computer Games (2015)
16. Sturtevant, N.: A comparison of algorithms for multi-player games. In: Schaeffer, J., Müller, M., Björnsson, Y. (eds.) CG 2002. LNCS, vol. 2883, pp. 108–122. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40031-8_8
17. Sturtevant, N.R., Saffidine, A.: A study of forward versus backwards endgame solvers with results in chinese checkers. In: Cazenave, T., Winands, M.H.M., Saffidine, A. (eds.) CGW 2017. CCIS, vol. 818, pp. 121–136. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75931-9_9
18. Sturtevant, N.: Multi-player games: Algorithms and approaches. Ph.D. thesis, UCLA (2003). http://www.cs.du.edu/~sturtevant/papers/multiplayergames thesis.pdf
19. Sturtevant, N., Rutherford, M.: Minimizing writes in parallel external memory search. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 666–673 (2013). http://www.cs.du.edu/~sturtevant/papers/bfs_min_write.pdf
20. Zhou, R., Hansen, E.A.: Parallel structured duplicate detection. In: Twenty-Second AAAI Conference on Artificial Intelligence (AAAI), pp. 1217–1224. AAAI Press, Vancouver, British Columbia, Canada, Jul 2007. http://www.aaai.org/Library/AAAI/2007/aaai07-193.php