Checkers: A Preview of What Will Happen in Chess?

Jonathan Schaeffer

Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada T6G 2H1

ABSTRACT

The checkers program *Chinook* has earned the right to challenge for the World Checkers Championship. The experiences gained in developing the program and preparing for the match illustrate several important obstacles to successfully mounting a challenge to the human World Champion. These points appear to be directly applicable to chess, and the checkers experience may well be a preview of what is to come in chess.

1. Introduction

The last five years have seen chess programs achieve tremendous success against strong human chess players, including Grandmasters [5]. Even though the *Deep Thought* program was handily defeated by World Champion Garry Kasparov [5], the impression prevails that it will only be a short time until the two (or their successors) square off for the World Chess Championship. In fact, with recent advances in hardware and software technology, it is hard to be pessimistic when forecasting the date this match will take place, with most experts predicting the human demise before the end of this decade [7].

The checkers program *Chinook* is arguably one of the two or three best checker players in the world, and has earned the right to challenge the World Champion, Dr. Marion Tinsley, for his title [10, 11]. By coming second to Tinsley in the biennial U.S. Open, *Chinook* became the next challenger to the World Champion. In an exhibition match in December, 1990, Tinsley narrowly defeated *Chinook* 8.5 - 7.5, winning game 10 and drawing the rest. A 40-game World Championship match is tentatively scheduled to be played in November, 1991.

The games of chess and checkers programs share many similarities, so it is not surprising that programs to play these games should be similar. They are both played on 8 X 8 boards (although in checkers only half the squares are used) with each side alternating moves (black moves first in checkers). Both games have opening, middlegame and endgame phases with similar characteristics. Championship-level players of both games must extensively study the opening literature, understand the subtleties of positional play, have a tactical acuity, and have a delicate touch in the endgame. Chess and checkers programs use deep alpha-beta searching to amaze the masters with long, startling combinations, while sometimes appearing like duffers when it comes to the subtleties of positional play. Other than redefining the move generator and position evaluation function, most of the work on computer chess can be applied to computer checkers.

Programs are closer to toppling the best humans in checkers than in chess. Given the strong similarity between the games, one would expect that the lessons learned from attempting to defeat Tinsley may be applicable to future man-machine World Championship chess matches. Despite *Chinook*'s success, we remain pessimistic about defeating Tinsley for a variety of reasons, many of which are discussed here.

In this paper, the experience gained in mounting a challenge for the checkers World Championship is extrapolated to the game of chess. There are a number of obstacles that will make it difficult to mount a successful challenge to Garry Kasparov this decade. Some of these problems have yet to be seriously looked at by the computer chess community.

2. All Draws are Not Equal

At the U.S. National Open (Tupelo, 1990), in eight of its 32 games *Chinook* announced that the game was drawn by move 10! In other words, the deep searches had found their way safely into the endgame databases, backing up a draw score to the root of the tree. The adversaries could now agree to a draw and move on to their next game. Or should they?

Consider the following four instances of "drawn" games:

(a) Of the eight games *Chinook* announced as draws in the U.S. National Open, the program won two of them. Although the program believed the position was drawn, assuming the opponent could see as far as the program could, the depth of the analysis was beyond what the humans could see and two eventually made mistakes and lost. The other six games ended in quick draws without *Chinook* doing anything to make the opponent's life difficult.

(b) In the Mississippi State Open, *Chinook* reached an endgame with a large advantage that was, however, insufficient to force a win. Given a choice of moves that preserved the draw, *Chinook* made some questionable selections. Over the span of 5 moves, the opponent's position steadily improved to the point where he refused *Chinook*'s draw offer. The program was in its database so there was no possibility of losing the game. Nevertheless, the game was extended an additional 90 minutes while the opponent tried to press home an insufficient advantage.

(c) In a Tinsley match game, the program declined winning a checker. It had correctly seen that all lines led to draws, but Tinsley was surprised that the program did not at least win the checker, thereby forcing him to come up with the correct (non-obvious) sequence of moves (which he had seen).

(d) In *Chinook*'s lone loss to Tinsley, the program announced a draw, only to change its assessment downward a few moves later and eventually lose the game. As early as move 10, Tinsley knew he was winning, yet *Chinook*'s deep searches managed to find inventive ways of postponing realization of the danger (and thereby not offering the stiffest resistance).

These three events call into question the meaning of a draw score. In commenting on *Chinook*'s play against him, Tinsley writes [14]:

> This illustrates the saying that for Chinook one draw is as good as another. The programmers have a challenge to direct him (sic) to select the most

aggressive line of attack.

A draw score backed up to the root of a search tree is not necessarily a *guaranteed* draw. The reliability of the draw assessment depends on the accuracy of leaf node evaluation. For example, if the opponent has a choice of a draw (score = 0) and an inferior position (score = -10), minimax assumes he maximizes his score, resulting in a draw score for this node. This score could be backed up to the root and the program will declare the position drawn. A deeper search (for example, on the next iteration) may search this line an additional ply and discover that the extra depth uncovers a tactical nuance, changing the assessment to a win (score = 100). Now the opponent backs up a winning score (100), possibly changing the root assessment of a draw into a loss.

When a draw score is backed-up to the root, how reliable is it? Alpha-beta gives no clues. Other search algorithms, such as conspiracy numbers [8] do, but with greatly increased search effort. A chess or checkers program faces the dilemma that if two or more moves backup a draw score, it is random which one is selected. Of course, there are "easy" and "hard" draws. The program needs some way of differentiating this.

Points (a), (b) and (c) above illustrate the need to differentiate draw scores. The program must be able to select moves that maximize the program's chances of winning. Point (b) also illustrates that the program can "con" the opponent into exerting himself longer than is required, an exercise that can only end up in tiring the human. Point (d) illustrates that a draw score can hide danger. For example, in this game at the critical position, *Chinook* selected a "drawing" move but an analysis of the game tree shows that path to the draw mirage was narrow and fraught with danger. An alternative move (the correct move), led to a small disadvantage (score = -10) but the program had many acceptable subsequent choices. In fact, an additional 4 ply of search would of been sufficient to see that this line led to a draw. The dilemma? Maximize score or maximize safety.

The move with the highest minimax score may not be the best move to maximize winning chances. Consider a position with the choice of two moves, $m1$ and $m2$:

m1: leads to a persistent advantage but the opponent will have no difficulty finding the right sequence of moves to draw.

m2: leads to a dead draw. However, there are a couple of traps along the way that require the opponent to resist playing the "obvious" move.

Which move would you choose? This is an important issue that has received some attention in the literature ([2, 6, 9], for example) but, unfortunately, has not been addressed in an implementation.

Selecting the move to maximize winning chances is not easy; a static evaluation function is often inadequate. The evaluation of a line of play should not be based solely on the position at the end of the line (as alpha-beta does). The evaluation of a line must also be a function of the *series of moves* that lead to the leaf position. Some considerations would include the depth of search required to see the correct result of the line of play (although this can often be misleading), the number of reasonable choices the opponent has at interior nodes along the line, how "obvious" the moves the opponent must play are, and any traps or pitfalls along the way. Peter Jansen has made some preliminary efforts at resolving these issues as they apply to traps and pitfalls [4].

It is difficult to know what the best solution to the draw differentiation problem is. In *Chinook*, scores can be viewed as 2 parts: integer and decimal (although that is not how they are implemented). The integral value is the usual minimax score, with a draw having a value of zero. Only when the position is scored as a draw are the decimal digits used. When reaching a drawn node (such as by hitting a position in an endgame database), a static evaluation is performed and assigned to the decimal digits. The program tries to maximize its integral score and, in the case of draws, maximize the decimal score. Another way of looking at it is that we have created a large range of values that fall in between the range -1..1, all of which are draw scores. In this way, we can differentiate draw scores by their backed-up static evaluation score, preferring to reach drawn positions that are more favorable to us than otherwise (solving problem (c) above).

In this framework, we are unsure how to modify draw scores to reflect the difficulty of the line required to reach that position. Although we have many ideas how to incorporate depth of the line, obviousness of the moves and probability of human error into the scores, it is very difficult to assess the utility of any solution. There are no books with collections of test positions where, in a drawn position, one attempts to choose the move that gives the opponent the most chances to go wrong.

3. Databases

Endgame databases are important in both chess and checkers because of the perfect knowledge they provide to the program. In this area, more so than any other aspect of the games, computer technology is vastly superior to human abilities.

Chinook currently has access to all the 6-piece and some of the 7-piece databases, a total of 15 billion positions. As well, by inference, the program has solved all the *n* versus 1 and *n* versus 2 endgames (e.g. 12 checkers against 2). Very soon, we expect to complete the 7-seven piece databases (a further 20 billion positions) and start tackling the 8-piece databases (400 billion positions).

In checkers, since capture moves are forced, it is easy for a long forcing line to terminate in a database. For example, a 3 minute search of the checkers starting position already reaches positions in the database. Hence, the databases must be online and accessible throughout a game.

As the number and size of the databases grow, so will the need to access them inexpensively. Our estimate is that an exact representation (win, loss or draw) of the 8-piece databases might be compressible to 15-20 gigabytes. Better compression ratios are possible, but may render the database unusable in a real-time application. Given a position, the program must compute where in the database the value will be found, access from disk the relevant portion of the file, and decompress it to extract the value. Since the database is being accessed during the search, the time taken to decompress and access values must be comparable to the time it takes to generate a move list or evaluate a position. This immediately rules out many of the standard compression algorithms. Unfortunately, even with optimal compression the databases still require too much storage, and each position reached will require a disk access, significantly slowing down the searcher. Although work has been done on compressing endgame databases (in chess [1] and Nine Men's Morris [3], for example), the size of the databases considered is not as big as we require (and will eventually be needed by chess), nor do they necessarily have the online and fast access requirements.

In constructing databases this large, it is too expensive to save anything but the value (win, loss or draw) for a position. Additional information, such as the best move to play in a position, would increase the database size roughly four-fold. A consequence is that in some 6-piece endgames, *Chinook* may reach a winning position and not know how to win, even with 25-ply searches.

An alternative approach to storing databases is to trade time for space. Examination of the checkers databases shows definite patterns that might be exploited. A series of pattern recognition tests might, for example, be able to ascertain the value of a position 90% of the time. Only for the exceptions would you need to reach into the database. The problem, of course, is how to extract patterns from the billions of position-value pairs in the database. Humans are very good at playing many endgames using only a few rules. Ultimately, computers must be able to do the same thing, even if it means introducing a small probability of error into an assessment. If so, then a few gigabytes of disk space may be replaced by a few pages of code.

As more chess and checkers databases are computed, the problems of how to store them compactly and access them in real-time will become more acute. In chess, Stiller's 5-piece endgame work done on the Connection Machine [12] is being extended to 6 pieces [13]. Once this work is complete, the total number of endgame positions computed for chess will dwarf our planned 400 billion for checkers.

4. Search and Knowledge

Chinook searches a minimum of 15-ply (plus extensions) deep in the opening, gradually increasing that to over 20 ply in the endgame. Even at these depths, humans can still out-search the program. How deep does a brute-force searcher need to go to be better than the best humans?

Chinook does not lose a game to a tactical combination. All its (few) losses against the world's best humans stem from the same problem: making a move that has fatal long-term consequences. For example, Marion Tinsley, commenting on his only victory over *Chinook*, said of *Chinook*'s 10th move [14]:

> What a shock! This move didn't seem like *Chinook*. ... From this point I could see quickly a clear unmistakable win.

Tinsley's vision consisted of a three-stage plan that he knew *Chinook* could not prevent. The program resigned 25 moves later. Only on move 27, 34 ply after the mistake on move 10, did *Chinook* realize it was in trouble. Search alone cannot solve this problem for us.

If the depths of search that *Chinook* reaches are not sufficient to defeat Tinsley, will these depths be sufficient to defeat Kasparov? Possibly. Chess has a higher branching factor than checkers (an average of 35 versus 10 in non-capture positions) and thus a ply of search is more complicated. Deeper searches will reduce the chances of the computer making a mistake and increase the chances for human error. However, the practical limits of search (another 1000-fold speed increase in the next few decades?) will still be insufficient to uncover the subtleties of some "simple" positions.

Even today, it is easy to construct a position that a human can solve easily and a good program, such as *Deep Thought*, won't have the slightest idea of what is going on. Knowing this and trying to exploit it in a game sounds like a difficult challenge. *Chinook*'s games have been extensively analyzed and its weaknesses are known to the Grandmasters. They now adopt a wait-and-see attitude. In most games nothing interesting develops and the Grandmaster plays towards a safe draw. Occasionally, *Chinook* maneuvers itself into a position where the human can force the program to make long-term commitments. Sometimes these commitments are mis-assessed by the program and a mistake occurs. *Chinook*'s last three losses (one to Tinsley and two to Don Lafferty, the world's acknowledged second best human player) were each the result of a poor long-term decision. In all three cases, to the human, while the best move might not have been obvious, *Chinook*'s move was obviously a mistake.

Chinook's static evaluation function is adequate to play Grandmaster checkers. However, without additional understanding of long-term considerations, it is inadequate to defeat Tinsley. The conjecture is that the same is probably true for chess. In a 24-game match, Garry Kasparov will do well if he plays patiently, not forcing things, waiting for a mistake to occur. Unfortunately for many strong players, psychological considerations may prevent them from adopting this objectively best strategy.

5. Theory and Practice

As the chess/checkers program becomes stronger, it becomes harder to find quality opponents to exercise the program's skill. Testing the program in the laboratory is insufficient to cover all the cases that arise in man-machine play. Hence we must go out and seek matches against the world's top players. This is not easy to do without appropriate financial incentives for the human players. It's difficult to get enough games played against top-quality opposition.

Chinook is good enough that it would be the favorite to win any checkers tournament that Tinsley was not competing in. As with any front-runner, the potential opponents have studied the program's play in depth looking for weaknesses. After every public game played, it becomes essential to identify any problems with the program's play that exhibited itself and fix it as quickly as possible.

The problem is most acute in the openings, where the humans have the greatest chance of catching the program by surprise. Grandmasters usually have a large repertoire of opening information at their disposal, either memorized or available in their extensive library. We have learned three rules for opening preparation the hard way:

(a) Do not repeat a game. The need to be non-repeatable dictates the preparation of an opening book that includes all the program's games and maintains alternate lines of play. It is important to avoid an opponent's prepared line, which means the book must be continually updated.

(b) It is not sufficient to play the best move all the time; a surprise move may have greater effect. Against human opposition, it is important to keep them guessing, to not be predictable. For example, *Chinook* has occasionally made "poor" moves by human standards which *Chinook* thinks are quite safe. While the human's assessment may be objectively correct, the surprise value of the move may work to their disadvantage.

(c) Do extensive homework and discover opening innovations. We spend an enormous number of computer cycles analyzing games and published opening analysis. From this, we look for moves that are either better than what appears in published play, or moves that appear to be as good as the published move but is not in the literature. All innovations must be approved for use in a tournament game by a Grandmaster we consult with. In a match against Don Lafferty, *Chinook* played two prepared opening innovations and ended up winning both games.

While these observations may seem obvious, what was not obvious to us is how much manual effort they require. In the past six months, over 75% of all time spent on *Chinook* has been on the opening book. This percentage is expected to increase as we begin intensive pre-match preparation for Tinsley.

6. What Title is at Stake?

Immediately after *Chinook* won the right to challenge Tinsley for the World Championship, opposition arose to any such contest. In a number of articles appearing in checkers magazines, arguments were put forth why a machine should not play for a human title. The opponents of the match were quite vocal and persuasive. Unfortunately, we were given no opportunity to participate in the discussions. Both the English Draughts Association and American Checker Federation voted against sanctioning the match. A clear case of discrimination!

Eventually an agreement was reached with the American Checker Federation to hold the match as a World "Man-Machine" Championship. Win or lose, Tinsley would retain his title as World (human) Champion. The smoke screen of a fabricated title is just a facade. Tinsley has been the best player in the world for over 40 years and during that period, having played thousands of games, he has lost only five! If *Chinook* defeats Tinsley, there will be no doubt as to who the real champion - human or not - really is.

We are fortunate in that Tinsley wants to play *Chinook* with a title at stake. It would have been easy for the World Champion to hide behind the refusal to sanction the match and avoid playing the computer. It is not hard to imagine that in similar circumstances for another game, such as chess, that the World Champion might refuse to play the computer. We hope that Tinsley's actions set an example that will encourage future encounters between computers and World Champions.

7. Resource Requirements

Perhaps the most frustrating aspect of our work on *Chinook* is the realization that we are incapable of achieving our goals without significant assistance. For the first year of our project, work was conducted using interested students and available equipment. With success came the desire to push beyond what was possible using our limited resources. As our appetite for success grew, so did our resource requirements. Over the past 6 months, many of the projects we would like to undertake remain on hold as we try to acquire access to the necessary resources.

The resources required to mount a successful challenge to the human World Checkers Champion are significant. They accrue in a number of areas:

(a) Compute Time.Checkers may not be as complicated as chess, but still requires access to the equivalent of supercomputers. These resources would be used primarily for building endgame databases and generating a computer opening book. For example, the endgame databases we are now constructing will require years of compute time on a 20 MIPS workstation. As well, this machine would require over a gigabyte of RAM. The only viable solution for solving this problem in a reasonable amount of time is to have access to a fast, parallel machine - in short, a supercomputer. Access to such resources are not possible without the support of a generous sponsor.

(b) Disk Storage.The 8-piece databases will require a minimum of 15-20 gigabytes of storage, and probably more. Most program developers do not have access to that much storage. One can envision going to the World Championship match running *Chinook* on a workstation, with an array of gigabyte disks surrounding the machine.

(c) Literature.Collections of Grandmaster games for checkers are not available in machine readable form, as they are in chess. Consequently, access to a comprehensive library of checkers literature is necessary to assist the opening analysis.

(d) Checkers Consultant.As the strength of the program increases, so must the abilities of a human checkers expert consultant. In our case, we have a volunteer Grandmaster who helps us out. To make faster progress, we really need a full-time Grandmaster, preferably one of the 10 best players in the world. Hiring such a person will cost a lot of money.

(e) Programming.The checkers project to date has involved 7 people with varying degrees of commitment. In a University environment, we must rely on (cheap) student help. Students can be excellent, but the project may suffer from their inexperience and their limited time constraints. Full-time programmer/analyst(s) would expedite things much more quickly.

(f) Salaries.In the University environment, the Faculty members of our Department working on *Chinook* require no financial compensation. This will not necessarily be the case for other, similar projects. A source of funding for the principal researchers on the project might also be required. For example, IBM is paying salaries to the *Deep Thought* principals.

(g) Travel.The need to play matches against strong players and play in strong tournaments means considerable travel expenses.

(h) Prizes.It is necessary to arrange money to sponsor matches against strong Grandmasters, including the World Championship challenge.

All the above points require financial support or access to resources that are usually not readily accessible.

The easiest solution to this problem is to find an interested sponsor. A conservative estimate is that the checkers project could use $100,000 of financial support per year, assuming the sponsor could provide all the computing resources for free. Unfortunately, the costs required for an equivalent chess endeavor may be even greater.

Given the significant costs of this endeavor, there have to be significant benefits in return for the sponsor. Some possible incentives might include:

a) the world-wide publicity that a World Man-Machine Checkers Championship would generate for the sponsor,

b) the opportunity to show off their products (such as hardware),

c) the right to market the program, and

d) access to all the research results produced from the project (although, in our experience, this isn't a big selling point!).

Obviously, how important these items are depends on the sponsor being wooed.

8. Conclusions

Winning the World Checkers Championship is proving to be a harder task than we initially anticipated. The difficulty stems from two sources:

a) the need to solve new, difficult research problems and

b) the limitations imposed by the resources we have access to.

Solving both these problems is critical to success. By presenting some of our obstacles to defeating the World Checkers Champion in this article, it is our hope that the computer chess community can identify the ones that will be relevant to chess, and start work on addressing these problems now.

Acknowledgments

Chinook has been a collective effort of the following people: Joseph Culberson, Brent Knight Paul Lu, Jonathan Schaeffer, Duane Szafron and Norman Treloar. Grandmasters Derek Oldbury and Norman Wexler have provided us with valuable insight into achieving a World-Championship-calibre checkers program.

Don Beal and Joe Culberson provided valuable feedback for improving the contents and clarity of this article.

This work was supported by funding for the Natural Sciences and Engineering Research Council of Canada, grant number OGP-8173.

References

1. Ingo Althofer, Compressing Chess Games with the Help of a Fast Deterministic Chess Program, *Journal of the International Computer Chess Association 13,* 4 (1990), 200-203.

2. Mikhail Donskoy, Fundamental Concepts of Search, *Journal of the International Computer Chess Association 13,* 3 (1990), 133-137.

3. Ralph Gasser, Applying Retrograde Analysis to Nine Men's Morris, in *Heuristic Programming in Artificial Intelligence; The Second Computer Olympiad*, D.N.L. Levy and D.F. Beal (ed.), Ellis Horwood, London, 1991, 161-173.

4. Peter Jansen, Problematic Positions and Speculative Play, in *Chess, Computers, and Cognition*, T. Anthony Marsland and Jonathan Schaeffer (ed.), Springer-Verlag, 1990, 169-181.

5. Danny Kopec, Advances in Man-Machine Play, in *Chess, Computers, and Cognition*, T. Anthony Marsland and Jonathan Schaeffer (ed.), Springer-Verlag, 1990, 9-32.

6. David Levy, Improving the Performance of Endgame Databases, *Journal of the International Computer Chess Association 10,* 4 (1987), 191-192.

7. David Levy and Monty Newborn, *How Computers Play Chess*, W.H. Freeman and Company, New York, 1990.

8. D.A. McAllester, Conspiracy Numbers for Min-Max Search, *Artificial Intelligence 35*, (1987), 287-310.

9. Donald Michie, A Theory of Evaluative Comments in Chess with a Note on Minimaxing, *The Computer Journal 24*, 3 (1981), 278-286.

10. Jonathan Schaeffer, Joseph Culberson, Norman Treloar, Brent Knight, Paul Lu and Duane Szafron, Reviving the Game of Checkers, in *Heuristic Programming in Artificial Intelligence; The Second Computer Olympiad*, D.N.L. Levy and D.F. Beal (ed.), Ellis Horwood, London, 1991, 119-136.

11. Jonathan Schaeffer, Joseph Culberson, Norman Treloar, Brent Knight, Paul Lu and Duane Szafron, Checkers Program to Challenge for World Championship, *SIGART Bulletin 2*, 2 (1991), 3-5.

12. Lewis Stiller, Parallel Analysis of Certain Endgames, *Journal of the International Computer Chess Association 12*, 2 (1989), 55-64.

13. Lewis Stiller, personal communication, electronic mail, 1990.

14. Marion Tinsley, Chinook-Tinsley Exhibition Match, *Checkers*, March 1991, 4-8.