Computer Chinese Chess

Tsan-sheng Hsu

徐讚昇

tshsu@iis.sinica.edu.tw

http://www.iis.sinica.edu.tw/~tshsu
Abstract

- An introduction to problems and opportunities in Computer Chinese Chess.
  - Open game
  - Middle game
  - End game

- How to generate endgame databases efficiently?
  - Exhaustive enumeration.
  - Memory addressing space.
  - Speed.

- How to use endgame databases during searching?
Introduction

- **Western chess programs.**
  - One of the important areas since the dawn of computing research.
  - Pioneer paper by C.E. Shannon (1950).
  - Beat the human champion at 1997.
  - Many techniques can be used in computer Chinese chess programs.

- **Computer Chinese chess programs.**
  - About 7-dan.
  - Computing research history: more than 30 years late.
    - Started at about 1981.
Chess Related Researches

- **Chess related research:**
  - Open game.
    - Many pseudo theories.
    - Heuristics.
  - Middle game searching.
    - Traditional game tree searching.
  - Endgame.
    - Databases.
    - More heuristics.
Books about Chinese Chess

- First written book: South Sung (about 1127–1279 AD)
Properties of Chinese Chess

- Several unique characteristics about Chinese chess.
  - The usage of Cannon.
  - Categories of defending and attacking pieces.
  - The positions of Pawns.
  - Complex Chinese chess rules.
  - Palace and the protection of kings.
  - Material combinations:
    - Although Knight is roughly equal to Cannon, \( \text{Rook + Knight + Cannon} \) is better than \( \text{Rook + 2 Cannons} \).
    - Knowledge inferencing among material combinations [Chen et al. 2007].
Some research opportunities.

- Open game theories.
  - Learning form a vast amount of prior human knowledge [Chen et al. 2006].

- Much larger searching space:
  - Western chess: $10^{123}$
  - Chinese chess: $10^{150}$
  - Deeper searching depth and longer game.

- Game tree searching.
  - The usage of materials.
  - Knowledge inferencing among material combinations [Chen et al. 2007].

- Endgame: contains lots of pieces.
- Rules.
Endgame Databases

- **Chinese chess endgame database:**
  - Indexed by a sublist of pieces $S$, including both Kings.

<table>
<thead>
<tr>
<th>K</th>
<th>G</th>
<th>M</th>
<th>R</th>
<th>N</th>
<th>C</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>King</td>
<td>Guard</td>
<td>Minister</td>
<td>Rook</td>
<td>Knight</td>
<td>Cannon</td>
<td>Pawn</td>
</tr>
<tr>
<td>🍀</td>
<td>♟️</td>
<td>☽</td>
<td>♜️</td>
<td>♔️</td>
<td>☻</td>
<td>☑️</td>
</tr>
</tbody>
</table>

- $KCPGGMMKGGMM$ (炮兵仕仕相相 vs. 士士象象):
  - the database consisting of RED Cannon and Pawn, and Guards and Ministers from both sides.

- A **position** in a database $S$: A legal arrangement of pieces in $S$ on the board and an indication of who the next player is.

- **Perfect information of a position:**
  - What is the best possible outcome, i.e. win/loss/draw, that the player can achieve starting from this position?
  - What is a strategy to achieve the best possible outcome?

- Given $S$, to be able to give the perfect information of all legal positions formed by placing pieces in $S$ on the board.

- **Partial information of a position:**
  - win/loss/draw; DTC; DTZ; DTR.
Usage of Endgame Databases

- Improve the “skill” of Chinese chess computer programs.
  - KNPKGGMM (俠 兵 vs. 士 士 象 象)

- Educational:
  - Teach people to master endgames.

- Recreational.
An Endgame Book
TCG: Computer Chinese Chess, 20100112, Tsan-sheng Hsu
TCG: Computer Chinese Chess, 20100112, Tsan-sheng Hsu
Definitions

- **State graph for an endgame** $H$:
  - **Vertex**: each legal placement of pieces in $H$ and the indication of who the current player (Red/Black) is.
    - Each vertex is called a **position**.
    - May want to remove symmetry positions.
  - **Edge**: directed, from a position $x$ to a position $y$ if $x$ can reach $y$ in one ply.
  - **Characteristics**:
    - Bipartite.
    - Huge number of vertices and edges for non-trivial endgames.
    - Example: KCPGGMMKGGMM has $1.5 \times 10^{10}$ positions and about $3.2 \times 10^{11}$ edges.
Overview of Algorithms

- **Forward searching:** doesn’t work for non-trivial endgames.
  - AND-OR game tree search.
  - Need to search to the terminal positions to reach a conclusion.
  - Runs in exponential time not to mention the amount of main memory.
  - Heuristics: A*, transposition table, move ordering, iterative deepening ...

```
  OR search
  ...
  AND search
  ...
  ...
  ...
```
Retrograde Analysis (1/2)

- First systematic studies by Ken Thompson 1986 for Western chess.

- Algorithm:
  - List all positions.
  - Find all positions that are initially “stable”, i.e., solved.
  - Propagate the values of stable positions backward to the positions that can reach the stable positions in one ply.
    ▶ Watch out the and-or rules.
  - Repeat this process until no more changes is found.
Critical issues: time and space trade off.
- Information stored in each vertex can be compressed.
- Store only vertices, generate the edges on demand.
- Try not to propagate the same information.
Stable Positions

- Another critical issue: how to find stable positions?
  - Checkmate, stalemate, King facing King.
  - It maybe the case the best move is to capture an opponent’s piece and then win.
    - so called “distance-to-capture” (DTC);
    - the traditional metric is “distance-to-mate” (DTM).

- Need to access values of positions in other endgames. For example,
  - KCPKGGM needs to access
    - KCKGGMM
    - KPKGGMM
    - KCPKGMM, KCPKGGM

- A lattice structure for endgame accesses.
- Need to access lots of huge databases at the same time.

[Hsu & Liu, 2002] uses a simple graph partitioning scheme to solve this problem with good practical results.
An Example of the Lattice Structure

TCG: Computer Chinese Chess, 20100112, Tsan-sheng Hsu
Yet another critical issue: cycles in the state graph.

- Can never be stable.
- In terms of graph theory,
  - a stable position is a pendant in the current state graph;
  - a propagated position is removed from the state graph;
  - no vertex in a cycle can be a pendant.
For most games, a cyclic sequence of moves means draw.
- Positions in cycles are stable.
- Only need to propagate positions in cycles once.

For Chinese chess, a cyclic sequence of moves can mean win/loss/draw.
- Special cases: only one side has attacking pieces.
  - Threaten the opponent and fall into a repeated sequence is illegal.
  - You can threaten the opponent only if you have attacking pieces.
  - The stronger side does not need to threaten an opponent without attacking pieces.
  - All positions in cycles are draws.
- General cases: very complicated.
Western chess: general approach.
- Complete 3- to 5-piece, pawn-less 6-piece endgames are built.
- Selected 6-piece endgames, e.g., KQQKQP.
  - Roughly $7.75 \times 10^9$ positions per endgame.
  - Perfect information.
  - $1.5 - 3 \times 10^{12}$ bytes for all 3- to 6-piece endgames.

Awari: machine and game dependent approach.
- Solved in the year 2002.
- $2.04 \times 10^{11}$ positions in an endgame.
  - Using parallel machines.
  - Win/loss/draw.

Checkers: game dependent approach.
- $1.7 \times 10^{11}$ positions in an endgame.
  - Currently the largest endgame database of any games using a sequential machine.
  - Win/loss/draw.

Many other games.
Earlier work by Prof. S. C. Hsu (許舜欽) and his students, and some other researchers in Taiwan.

- KRKGGMM (仕士象象 vs. 兵仕仕相) [Fang 1997; master thesis]
  - About $4 \times 10^6$ positions; Perfect information.

- Memory-efficient implementation: general approach.

- KCPGMKGGMM (炮兵仕仕相 vs. 兵士象象) [Wu & Beal 2001]
  - About $2 \times 10^9$ positions; Perfect information.

- KCPGGMMKGGGM (炮兵仕仕仕仕相 vs. 兵士士象) [Wu, Liu & Hsu 2004]
  - About $8.8 \times 10^9$ positions; $2.6 \times 10^{-5}$ seconds per position; Perfect information.
  - The largest single endgame database and the largest collection reported.

- Verification [Hsu & Liu 2002]

- Special rules: more likely to be affected when endgames get larger.
Chinese Chess Special Rules (1/3)

- A player cannot avoid the losing of the game or important pieces by forcing the opponent to do repeated counter-moves.
  - Checking the opponent’s king repetitively with no hope of checkmate.
    - Asia rule example #2.
  - Chasing an unprotected opponent’s piece repetitively with no hope of capturing it.
    - Asia rule example #19.
  - Threatening (to checkmate) repetitively with no hope of realizing the threat.
    - Asia rule example #31.

- Sometimes it is difficult to check whether a piece is truly or falsely protected.
  - Asia rule example #39.
  - Asia rule example #105.

- Not a problem for Western chess.
  - Cycles mean draw.
Asia Rule Example #2

- Checking the opponent’s king repetitively with no hope of checkmate.
  - $R4=5, K5=6, R5=4, K6=5, ...$
  - *Red Rook checks Black King.*
Asia Rule Example #19

- Chasing an unprotected opponent’s piece repetitively with no hope of capturing it.
  - $C2-1, R4-2, C2+2, R4+2, ...$
  - *Red Cannon at the 2nd column chases Black Rook.*
Threatening (to checkmate) repetitively with no hope of realizing the threat.

- $R2=1, C9=8, R1=2, C8=9, \ldots$
- Black Cannon at the 9th column threatens to checkmate.

TCG: Computer Chinese Chess, 20100112, Tsan-sheng Hsu
Sometimes it is difficult to check whether a piece is *truly* or *falsely* protected: the definition of a protector is complicated.

- $R^8 + 2, G^6 + 5, R^8 - 3, G^5 - 6,\ldots$
- *Red Knight at the 2nd column is not protected.*
- *Black Rook at the 6th column cannot threaten.*
Sometimes it is difficult to check whether a piece is truly or falsely protected: you can block a protector.

- $P7=6,M1+3,P6=7,M3−1,...$
- The protector of Black Knight at the 7th column is blocked.
Two main categories:
  - Supported by Asian Chinese Chess Association.
  - Simple and effective.
  - Is not really “fair” in certain complex cases.
  - Taiwan version (2007) is based on Asian version.
- Mainland version (1999)
  - Supported by the PRC Chinese Chess Association.
  - A national standard.
  - Try to be as complete and ”fair” as possible.

Problems in computer implementation:
- “Rules” are vague.
- Often illustrated with examples.
Rules: Taiwan Version

Rules: Asian Version

Rules: Mainland Version

Rules: Problems About the Mainland Version

Chinese Chess Special Rules (3/3)

- **Current treatment of special rules:**
  - Avoid them at all: do not play repeated positions.
    - *May lose advantage.*
    - *Must allow loops in endgame construction.*
  - **Special cases:**
    - *Only one side has attacking pieces: all are implemented.*
    - *One side has only a pawn and some defending pieces: can be affected by special rules.*
  - **Partial treatment:**
    - *Implement only the rules related to “checking.”*
    - *Implement some “chasing” rules.*
    - *Verify whether special rules can affect an endgame.*

- **We need a throughout understanding of special rules to build larger endgame databases.**
Special Rules: Results

- Partial treatment may build imperfect databases.
  - [Fang, Hsu & Hsu 2000].

- Upto 17.3% for the checking rule in KRKNMM ( 黑 vs. 马 象 象 ) [Fang, Hsu & Hsu 2002].

- Jih-tung Pai [Private communication 2003] implemented a variation of [Fang, Hsu & Hsu 2002].

- Look for necessary conditions when databases can be stained by special rules.
  - Selected 50+ databases are verified [Fang 2004].
Special Rules: Work in Progress

- May affect the correctness of evaluation functions.
  - Xie Xie vs. Contemplation in the first WCCCC (Year 2004).
    - Less than 3% of the games played.
  - About 5% of the games played in the 10th Computer Olympiad (October 2005) need to utilize special rules.

- Usage of logic and graph theory in an algorithmic context to describe the Asian version.
  - To explain all examples.
  - To abstract hidden experts’ knowledge.
  - To obtain fast computer implementations.

- Still a long way to go for the Mainland version.
Red: Contemplation.

N3+4,R7–6,N4–3,R6–7,...

- *Red Knight at 3rd column is protected.*
- *The game ended in a draw.*
Usage of Endgame Knowledge

- Databases of endgames are too large to be loaded into the main memory due to searching.

- Human experts:
  - Studies the degree of “advantageous” by considering only positions of pawns and material combinations.
  - Lots of endgame books exist.

- How to verify whether these knowledge are consistent?
  - Piece additive law: If endgame $W$ is advantageous to the Red, then
    - adding a red piece to $W$ will never make it worse.
    - deleting a red piece to $W$ will never make it better.

- Inferencing the degree of “advantageous” of an unknown endgame $W$ by values of endgames that we have already known.
  - [Chen et. al. 2008].

- Checking whether a set of endgame knowledge is consistent according to the piece additive law.
  - [Chen et. al. 2009].
Concluding Remarks

- Many open problems.
- Research opportunities:
  - Algorithm and complexity.
  - Algorithmic engineering.
  - External memory algorithms.
  - System implementation.
  - Parallel computing.
  - A.I.
    - Knowledge extracting.
    - Data mining.
    - ...
  - Discrete Math., e.g., Graph theory.
- Commercial opportunities.
- Fun.
References and further readings


