# Chinese Chess

Chinese Chess

Supervised by Dr Dayou Li

Supervised by Dr Dayou Li

University of Luton
Education that works

Zhihua Lai ID:0405005
BSc Computer Science

## Abstract:

Chess is always an important part in the whole Artificial Intelligence field. The famous Chess program, Deep Blue, which used to beat human world Champion, will remain immortal in history. In the last few years, with the rapid improvement of computer hardware, the AI engines are becoming more and more efficient. However, due to more complex search space and rules, in the filed of Chinese Chess, computer has not been able to beat the human champion. The project explores the mysteries of a chess program and proposes possible solutions to a Chinese Chess Computer Game (CCCG).

## Aim & Objectives:

The aim is to develop a CCCG. Searching for the optimal step is the core of the development of the CCCG. This work has focused on development of a search algorithm and evaluation which offset the shortcomings of those used by others. Because of a larger game board, the techniques used to develop a western chess might not be so effective in Chinese Chess. Even slight improvement on search algorithms or evaluation can drastically increase the overall performance.

✓To learn theory in conjunction to Chinese chess of major problems used to be solved in CCCG by studying computer game
✓To literature search for CCCG products and algorithms used and the problems existing in those algorithms.
✓To develop an algorithm which can offset the problems in the existing algorithms.
✓To test and compare the developed algorithm and other algorithms.
✓To develop a user-friendly CCCG.

## Introduction:

A chess program is composed of "state representation", "move generator", "search engine", and "evaluation". [1, 3, 6] (Figure 1). The project includes literature reviews of the "evaluation" and "search", which are the most important parts of a chess program. Pros and Cons of the algorithms developed before are discussed and possible improvements are proposed.

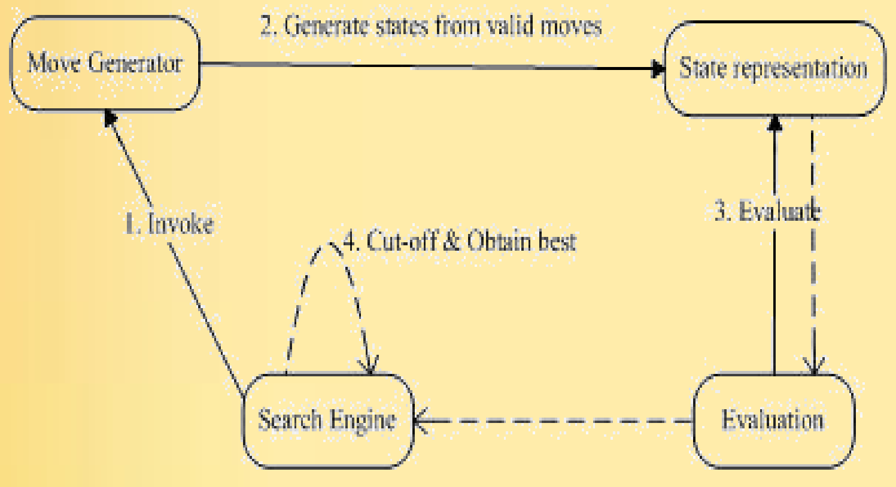
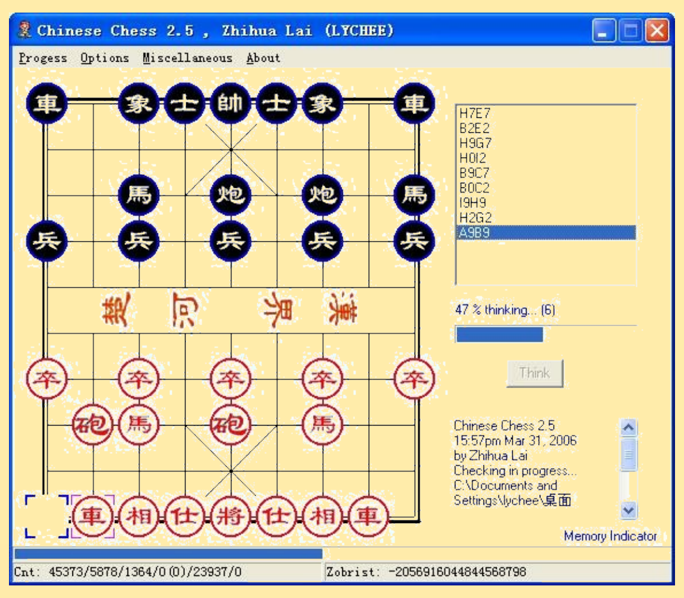


Figure 1 – Components of a chess program



H7E7
B2E2
H9G7
H0I2
B9C7
B0C2
I9H9
H2G2
A9B9

47 % thinking... (6)

Chinese Chess 2.5
15:57pm Mar 31, 2006
by Zhihua Lai
Checking in progress...
C:\Documents and Settings\lychee\桌面

Cnt: 45373/5878/1364/0 (0)/23937/0
Zobrist: -2056916044844568798

Figure 2 – The Lychee 2.5

## Literature review of "Evaluation":

Evaluation module is the most important and crucial part of any chess programs. It is done by estimating the situation on a board. It determines how intelligent your program is. Normally, it is done by determining the value that measures the advantage or disadvantage [1]. The evaluation in chess often consists of "Material-value Evaluation", "Mobility-value Evaluation", "Relation between pieces" and etc. [2]

## Literature review of "Search":

So far, the theory of chess playing of perfect information is systematic. MiniMax [4] is the fundamental algorithm of alpha-beta search in the idea of DFS. Negamax is the improved frame of Alpha-beta [8]. The other algorithms that have been developed are Fail-soft alpha-beta, Aspiration Search, Principal Variation, Transposition Table, Iterative Deepening, History Heuristic [7], Killer Heuristic, MTD(f), and Null Move Pruning [5, 10].

## "The Lychee": (Figure 2)

The outcome is a piece of software - "The Lychee", which has been designed in Delphi 7.0 (Object Oriented). The latest version is 2.5, which is available from the website. The AI of "The Lychee" now is above average. Besides, it has a user-friendly GUI and a group of configurable AI options. Moreover, it has a built-in auto-update engine. CBS format is developed and supported to load & save states. Openbook is also one of the important features. On the whole, "The Lychee" can be divided into 4 basic parts: Lychee state representation, Lychee Move generator, Lychee Search and Lychee Evaluation respectively.

Like as pointed out in Figure 1, the components of "The Lychee" has similar relationships between them. After a user has made a move, which is that the user has dragged and dropped a piece (Figure 2), the updated state is passed to "Lychee Search Engine", which invokes move generator to recursively generate a move tree. Search Engine then uses evaluation function to get values of each terminal nodes. Cut-offs occur and a best move is found.

"The Lychee" has been able to do more than that: Openbook (Figure 4) at outmost layer absorbs human's experience and has improved intelligence. Conditional Reflex and Must Respond Cut Off are especially added and they serve to accelerate search by incurring more cut-offs.

## Lychee State Repesentation:

*ChessElement=(NOCHESS=0,*
*B_KING,B_TANK,B_HORSE,B_CANNON,B_GUARD,B_ELEPHANT,B_SOLDIER,*
*R_KING,R_TANK,R_HORSE,R_CANNON,R_GUARD,R_ELEPHANT,R_SOLDIER);*
*statetype=array [1..10,1..9] of ChessElement;*

## Lychee Move Generator:

A good move generator should generate all valid moves fast. In order to save space, dynamic array is used. The size of a dynamic array can be changed during run time, which is useful in implementing a link.

## Lychee Evaluation:

It consists of aspects of "Material Value"," Mobility Value"," Relation-between-piece Value" and "Bonus Value".

## Lychee Search:

Lychee search has been designed in layers. (Figure 3), which gives the advantage to tackle different issues in different layers. The outmost layer is responsible for handling openbook and Iterative Deepening. Then it goes to next layer (Middle), which is on Conditional Reflex and Aspiration Search. MTD Layer is optional as it can be configurable not to do MTD algorithm. And the innermost layer is the core layer of Lychee Search, because it handles the real search in recursive form. This layer uses Fail-soft Alphabeta, HH, PVS, Must Respond-cut-off, TT, Null move pruning and etc.

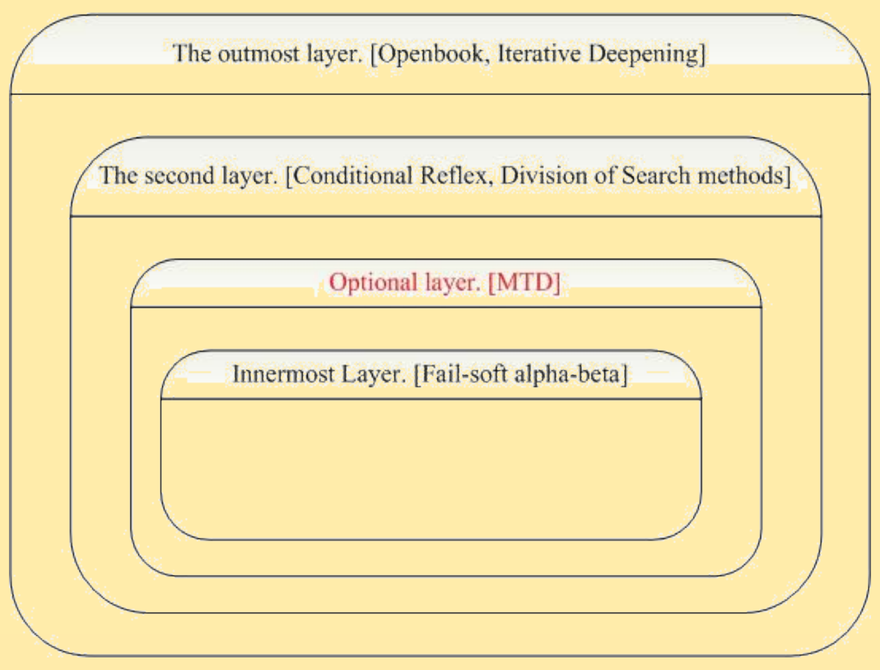

The outmost layer. [Openbook, Iterative Deepening]
The second layer. [Conditional Reflex, Division of Search methods]
Optional layer. [MTD]
Innermost Layer. [Fail-soft alpha-beta]

Figure 3 – Lychee Search Layers

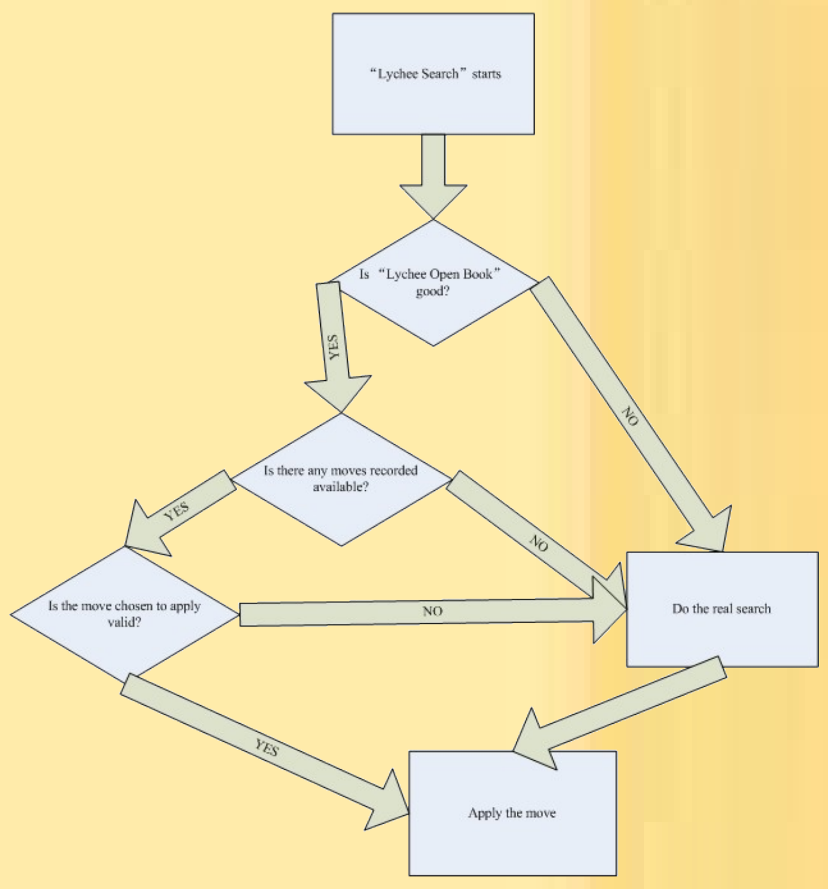

"Lychee Search" starts
Is "Lychee Open Book" good?
YES
NO
Is there any moves recorded available?
YES
NO
Is the move chosen to apply valid?
NO
YES
Do the real search
Apply the move

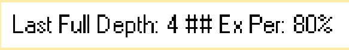
Figure 4 – Lychee Openbook

## Testing methods:

✓Counters *eg.*

Cnt: 22873/9170/8710/0(0)/10109/4

✓Extra piece of information *eg.*



Last Full Depth: 4 ## Ex Per: 80%

## Conclusion:

The literature review includes "evaluation" and "search", which help to design "Lychee Evaluation" and "Lychee Search". The "Lychee Evaluation" includes the evaluation aspects of the "material value", "mobility value", "relation value", and "bonus value". The "Lychee Search" is combined of Iterative Deepening, Conditional Reflex, MTD, Must Respond Cut-off, and etc algorithms. The combined search module aims to prune out as many unnecessary branches as possible. "The Lychee" is programmed using Delphi 7.0 and it can now beat above average.

## Further work:

✓Pattern "Tank Cannon" & Pattern "Horse Cannon"
✓prune out repetitive moves in advance [3]
✓"Please allow me more time"
✓make use of the results obtained in the unfinished new depth.
✓quiescence search [3]

## References:

[1] "Computer Chinese Chess", Shi-Jim Yen, Taiwan, March, 2004
[2] http://www.ics.uci.edu/~eppstein/180a/970415.html
[3] PC Chess playing Xiaocun, Wang
[4] Page 255-311, Introduction to Artificial Intelligence, Drew McDermott
[5] The handbook of AI Avron Barr & Edward A Feigenbaum
[6] "Strategy and board game programming", ICS 180, Winter 1999
[7] "The history heuristic and alpha-beta enhancements in Practice" Jonathan Schaeffer, November 1989
[8] "A New Paradigm for Minimax Search" (a.k.a. EUR-CS-95-03)
[9] "Game Theory" Theodore L. Turocy Texas A&M University Bernhard von Stengel London School, October 8, 2001
[10] "Verified Null-Move Pruning" Omid David Tabibi, Nathan S. Netanyahu, Ramat-Gan, Israel