



An AI for Shogi

Raphael R. Eguchi
CS221

1. Introduction

Chaturanga, the world's first chess variant, is thought to have originated in India some time during the seventh century. While chaturanga eventually evolved into western chess, in the east the game evolved into Xiang Qi in China, and into Shogi in Japan. Shogi differs from western chess in several ways. First, board size: Shogi is played on a 9x9 board, while chess is played on an 8x8 board. Second, piece promotion is not restricted to pawns: unlike chess, most pieces in Shogi can be promoted upon reaching the third rank, and the movement of the promoted piece is pre-determined. Third, in Shogi, a captured piece can be redeployed anywhere on the board, given that (1) the piece is able to make at least one legal move, and (2) that two un-promoted pawns do not fall in the same rank. Computationally, these characteristics make Shogi the most complex form of chess that is commonly played, with an average game length of 115 moves, and a branching factor of 80, yielding a game tree complexity of $\sim 10^{226}$. In comparison, chess has a complexity of $\sim 10^{120}$.



2. Development Strategy

2.1 Model

- A Two-player zero-sum game
- Players = {0, 1} (0 plays first)
- State S: position of all pieces, player-held pieces, player turn
- Actions(S): the set of legal chess moves that the can be made
- IsEnd(S): whether S is checkmate, stalemate or draw
- Utility(S): $+\infty$ if A wins, 0 if draw, $-\infty$ if B wins

2.2 Current Algorithm

- Fixed-depth Minimax with Alpha-Beta Pruning.
- Evaluation Function from domain knowledge.

2.3 Evaluation Function

Currently very simple. Comprised of 3 sub-evaluation functions:

1. Piece Value

$$PieceEval(s, agent) = \sum_{p \in agentPieces} PieceValue(p) - \sum_{p \in opponentPieces} PieceValue(p)$$

2. King Safety

$$KingSafetyEval(s, agent) = KingState(s, agent) - KingState(s, opponent)$$

$$KingState(s, agent) = w_1 EscapeSquares(s, agent) + w_2 Attackers(s, agent) + w_3 Defenders(s, agent)$$

3. Square Control

$$SquareCtrlEval(s, agent) = controlledSquares(s, agent) - controlledSquares(s, opponent)$$

Total Evaluation:

$$Evaluation(s, agent) = c_1 PieceEval(s, agent) + c_2 KingSafetyEval(s, agent) + c_3 SquareCtrlEval(s, agent)$$

3. Agent Set

3.1 My Agents

- RandomAgent — Randomly choses from all legal moves.
- GreedyReflexAgent — Immediately maximizes the evaluation.
- AlphaBetaAgent — Fixed-depth minimax with alpha-beta pruning.
- EpsGreedy Agent — Greedy, but chooses a random move with probability ϵ .

3.2 Oracles

- Apery-SDT5 — Highest rated AI, with a rating of 4135.
- Gikou2 — 4th strongest AI, with a rating of 3798. Variable Elo at different depths.

4. Current Results

The table below summarizes the results of preliminary testing. The Win-Draw-Loss counts are written from the perspective of the subject. Parentheses next to subjects' names indicate depth. PE indicates the pieceValue evaluation function. "full" indicates the full evaluation function. The runs were done on Sherlock CPUs.

Subject	Opponent	Win Rate	# Games	Win	Draw	Loss
Random	Random	0.5219	205	107	0	98
AlphaBeta(3,PE)	Random	1.0000	200	200	0	0
Random	Greedy	0.0000	200	0	2	198
AlphaBeta(3,PE)	Greedy	1.0000	200	200	0	0
AlphaBeta(3,PE)	AlphaBeta(2,PE)	1.0000	200	200	0	0
AlphaBeta(3,PE)	Gikou(1)	0.0000	200	0	0	200
AlphaBeta(3,PE)	Gikou(Unlim.)	0.0000	200	0	0	200
AlphaBeta(3,PE)	Apery	0.0000	200	0	0	200
AlphaBeta(2,full)	AlphaBeta(3,PE)	0.9911	227	225	2	0
AlphaBeta(3,full)	AlphaBeta(3,PE)	1.0000	100	100	0	0
AlphaBeta(3,full)	Gikou(1)	0.0000	74	0	0	74

Overall, the results agree with the expected strength ordering:

Random < Greedy < AlphaBeta(2, PE) <

AlphaBeta(3, PE) < AlphaBeta(3,full) < Gikou(1) < Gikou(Unlim.), Apery

	Western Chess	Shogi	Go
Board Size	8x8	9x9	19x19
Promotion	Pawn only, Backrow	Most Pieces, 3rd rank	—
Redeployment	No	Yes	—
Avg. game Length	70	115	150
Stalemate	Draw	Loss of the mated.	—
Move Repetition	Draw	Draw	—
Branching Factor	35	92	250
Complexity	10^{123}	10^{226}	10^{360}

↑ Table 1: Comparison of Chess and Shogi

Piece	Value	Piece (Promoted)	Value
Pawn (歩)		1 Pawn+ (と)	6
Lance (香)		3 Lance+ (成香)	6
Knight (桂)		4 Knight+ (成桂)	6
Silver General (銀)		5 Silver+ (成銀)	6
Gold General (金)			
Bishop (角)		8 Bishop+/Horse (馬)	10
Rook (飛)		10 Rook+/Dragon (龍)	12
King (王/玉)		10000	

↑ Table 2: Piece Values used in Evaluation

9	8	7	6	5	4	3	2	1	9	8	7	6	5	4	3	2	1
v香	v桂	v銀	v金	v玉	v金	v銀	v桂	v香	v香	v桂	.	.	v玉	v金	.	v桂	v香
.	v飛	v角	.	.	v飛	v金	v銀	.	v銀	.	v角	.
v歩	v歩	v歩	v歩	v歩	v歩	.	v歩	v歩	v歩	v歩	v歩	.	v歩	.	v歩	v歩	v歩
.	v歩	v歩	.	角
.
.	.	歩	歩
歩	歩	.	歩	歩	歩	歩	歩	歩	歩	歩	.	歩	歩	歩	歩	歩	歩
.	角	飛	銀	金	玉	銀	飛	.
香	桂	銀	金	玉	金	銀	桂	香	香	桂	.	.	.	金	.	桂	香

l	n	s	g	k	g	s	n	l	l	n	.	.	k	g	.	n	l
.	r	b	.	.	r	g	s	.	s	.	b	.
p	p	p	p	p	p	p	p	p	p	p	p	.	p	.	p	p	p
.	p	p	.	B
.
.	P	P
P	P	.	P	P	P	P	P	P	P	P	.	P	P	P	P	P	P
.	B	R	.	.	.	S	G	K	S	R	.	.
L	N	S	G	K	G	S	N	L	L	N	.	.	.	G	.	N	L

↑ Figure 1: Recovery of the 7g7f opening (left), and recovery of the Mino Formation during midgame (right).

5. Future Directions

- Improve Performance using Cython Numba.
 - Allow greater depth.
- TD-Learning to learn weights of the current evaluation.
 - Use an epsilon-greedy policy to explore, self-learn.
 - Try to retain interpretable features.
- A neural network based evaluation function.
 - Learn weights using TD-Learning.