

PROJECT REPORT

Problem Statement:

Part 1: MINIMAX

The first program plays a move in the opening phase of the game using MiniMax algorithm. The second program plays in the midgame/endgame phase using MiniMax algorithm.

Part 2: ALPHA BETA

The first program plays a move in the opening phase of the game using Alpha Beta pruning algorithm. The second program plays in the midgame/endgame phase using Alpha Beta pruning algorithm.

PART 3:PLAY A GAME FOR BLACK

Same programs as in Part I, but the computed move should be Black's move instead of White's move.

PART 4:STATIC ESTIMATION

Improved static estimation function. New function is better than new handout.

Methodology:

1. MiniMax algorithm

1. It first check what is the value of depth. If it is zero means it is a leaf then perform normal estimation as specified on handout and update the value of MINIMAX estimate and number of positions evaluated.

2. If flag is 1 and Opening/Game for white then call function `GenerateMovesOpening/GenerateMovesMidgameEndgame` which has list of possible positions of white. Else call function `GenerateMovesOpeningBlack/GenerateMovesMidgameEndgameBlack` which returns list of possible positions of black. If flag is 1 and Opening/Game for black perform the opposite as specified.

3. Then check the flag value which specify whether to perform MAX or MIN at that level. According set the value of data which is minimax estimate then recursively call MINIMAX function by reducing the depth by 1 and changing flag 1 to flag 0. Flag 1 means Max while flag 0 means Min calculation.

2. Alpha Beta Pruning Algorithm

1. It first check what is the value of depth. If it is zero means it is a leaf then perform normal estimation as specified on handout and update the value of ALPHA BETA estimate and number of positions evaluated.

2. If flag is 1 and Opening/Game for white then call function GenerateMovesOpening/GenerateMovesMidgameEndgame which has list of possible positions of white. Else call function GenerateMovesOpeningBlack/GenerateMovesMidgameEndgameBlack which returns list of possible positions of black. If flag is 1 and Opening/Game for black perform the opposite as specified.

3. While call the AlphaBeta pruning function it should pass value of Alpha and Beta. Alpha has value Integer.Min while beta has value Integer.Max. Then recursively call the Alpha Beta pruning function by reducing depth and updating values of alpha and beta. Return the final board state with number of evaluation and estimate value.

3. Static Estimation Improved

1. In normal static estimation for Opening the game for Black or White we just count the number of white and black and then subtract it.

2. If static estimation is for Midgame/Endgame phase for Black and white then perform calculation as specified in the handout.

3. In improved estimation step 1 and step 2 is same for the corresponding programs. Changes made are to make mills. As forming a mill will be better to reduce the moves in the game and can reduce the opposite players moves also. So for example we need to perform MiniMaxOpening for White. I will perform step 1. For each possible board check which all positions have X as the value. Then select that position and check whether if we replace it by W will it form mill. If it forms mill then increment the counter. Update this counter value in data calculation and update data. This makes changes in the final move. As we are checking in static estimation phase many moves are reduced and game gets over early.

Development Platform:

Operating System: Windows 8

Hardware Specifications: Intel CORE i5 processor, 8GB RAM

Development IDE: Eclipse

Comparison Results:

Case 1:

Positions evaluated by MiniMaxGame and MiniMaxGameImproved is different

Input File: BWBWBWBWBWWXXXWBWBXXWXB

Output File :

MiniMaxGame:

BoardPosition: BWBWBWBWBWWXXWBXBXXWXB

Positions Evaluated by static estimation: 43

MINIMAX estimate: 965

MiniMaxGameImproved:

BoardPosition: BWBWBWBXBWBWWXXWBWBXXWXB

Positions Evaluated by static estimation: 43

MINIMAX estimate: 2978

Case2:

Positions evaluated by MiniMaxOpening and MiniMaxOpeningImproved is different.

Input File: BWBXXBWBWWWBXBWBXBXWWXB

Output File:

MiniMaxOpening:

BoardPosition: BWBWXBWBWWWBXBWBXBXWWXB

Positions Evaluated by static estimation: 640

MINIMAX estimate: 0

MiniMaxOpeningImproved:

BoardPosition: BWBXXBWBWWWBWBWBXBXWWXB

Positions Evaluated by static estimation: 640

MINIMAX estimate: 1

Case 3:

Estimation time taken by ABGame is less compared to MiniMaxGame.

Estimation time for ABOpening is less compared to MiniMaxOpening, MiniMaxOpeningImproved .

Input File:

BWBWBWBWBWWXXWBWBXXWXB

Output File:

ABGame:

BoardPosition: BWBWBWBWBWWXXWBXBXXWXB

Positions Evaluated by static estimation: 28

Alpha Beta estimate: 965

MiniMaxGame:

BoardPosition: BWBWBWBWBWWXXWBXBXXWXB

Positions Evaluated by static estimation: 43

MINIMAX estimate: 965

ABOpening:

BoardPosition: XWBWBWBWBWWWBXWBWBXXXXB

Positions Evaluated by static estimation: 35

Alpha Beta estimate: -1

MiniMaxOpening:

BoardPosition: XWBWBWBWBWWWBXWBWBXXXXB

Positions Evaluated by static estimation: 213

MINIMAX estimate: -1

MiniMaxOpeningImproved:

BoardPosition: XWBWBWBWBWWWBXWBWBXXXXB

Positions Evaluated by static estimation: 213

MINIMAX estimate: -1

Case 4:

Output for all the conditions

Input File:

BWBXXBWWWWBWBWBXBXWWXB

Output File:

ABGame:

BoardPosition: BWBXXWBXWWWBXWBXBXWWXB

Positions Evaluated by static estimation: 280

Alpha Beta estimate: -6

ABOpening:

BoardPosition: BWBWXXBWWWWBWBWBXBXWWXB

Positions Evaluated by static estimation: 74

Alpha Beta estimate: 0

MiniMaxGame:

BoardPosition: BWBXXWBXWWWBWBWBXBXWWXB

Positions Evaluated by static estimation: 681

MINIMAX estimate: -6

MiniMaxGameImproved:

BoardPosition: BWBWXXBWWXWBWBWBXBXWWXB

Positions Evaluated by static estimation: 681

MINIMAX estimate: 1985

MiniMaxOpening:

BoardPosition: BWBWXXBWWWWBWBWBXBXWWXB

Positions Evaluated by static estimation: 640

MINIMAX estimate: 0

MiniMaxOpeningBlack:

BoardPosition: BXBBXXBWWWWBWBWBXBXWWXB

Positions Evaluated by static estimation: 1206

MINIMAX estimate: 3

MiniMaxOpeningImpRoved:

BoardPosition: BWBXXXBWWWWBWBWBXBXWWXB

Positions Evaluated by static estimation: 640

MINIMAX estimate: 1

Conclusion:

- 1.Alpha Beta pruning requires less positions for evaluation compared to MiniMax .
- 2.After making changes in MiniMax estimation function the result is better than suggested on handout.

References:

Classnotes