This report presents improvements to baseline algorithm to play isolation game. Baseline algorithm is minimax search with alpha-beta pruning and iterative deepening. Baseline heuristic function is defined as follows:

```
def score(self, state):
    own_loc = state.locs[self.player_id]
    opp_loc = state.locs[1 - self.player_id]
    own_liberties = state.liberties(own_loc)
    opp_liberties = state.liberties(opp_loc)
    return len(own_liberties) - len(opp_liberties)
```

As improvement option 1 was selected – a custom heuristic development. Two different heuristics were developed and tested. Test results are described in the following sections.

Heuristic 1 – Block the opponent

This heuristic is offensive heuristic that would prioritize moves that would block opponent moves. It might be beneficial because it encourages direct limitation of opponent moves. It is done by adding number of both player actions that end on the same field.

The heuristic is defined as follows:

```
def score(self, state):
    own_loc = state.locs[self.player_id]
    opp_loc = state.locs[1 - self.player_id]
    own_liberties = state.liberties(own_loc)
    opp_liberties = state.liberties(opp_loc)
    count = 0
    for own_liberty in own_liberties:
        for opp_liberty in opp_liberties:
            if own_liberty == opp_liberty:
                count = count + 1
    return len(own_liberties) - 2 * len(opp_liberties) + count
```

Tests results are presented in table below.

Matches won	Time limit	Number of rounds	Fair matches
54%	150ms	50	True
48%	300ms	50	True

Following game consisting of 50 rounds 10 rounds game was run to log maximal search depth for both heuristics. They are presented in table below. Number in bold means that the match was won.

Depth of baseline algorithm	**Depth of heuristic 1 algorithm**
69	34

32	32
113	113
10	**10**
113	**79**
113	113
15	**13**
50	**51**
113	**113**
113	**113**
10	**9**
113	113
113	**113**
9	11
58	113
113	**113**
97	**74**
113	**113**
32	25
113	**113**

As it can be seen from tables above block opponent algorithm seemed to do slightly better for 150ms timeout than baseline one. For longer periods (300ms) results were slightly worse. The depth of the search was very similar for both algorithms (time needed for calculation of score function was very close). It seems that accuracy of new heuristic is not better than base one.

Heuristic 2 – From offensive to defensive

This heuristic changes weights related to number of own and opponent moves. At the beginning of the game strategy is offensive, but with each move is getting more defensive. It might be beneficial because at the beginning of the game it tries to limit moves of the opponent and towards the end it prefers moves that give bigger flexibility, when number of choices is getting more limited. It is done by adding weight that depends on number of actions applied on board and board size.

The heuristic is defined as follows:

```
def score(self, state):
    own_loc = state.locs[self.player_id]
    opp_loc = state.locs[1 - self.player_id]
    own_liberties = state.liberties(own_loc)
    opp_liberties = state.liberties(opp_loc)
    return (2 * len(own_liberties) * state.ply_count / _SIZE) - len(opp_liberties)
```

Tests results are presented in table below.

Matches won	Time limit	Number of rounds	Fair matches
56.5%	150ms	50	True
53.5%	300ms	50	True

Following game consisting of 50 rounds 10 rounds game was run to log maximal search depth for both heuristics. They are presented in table below. Number in bold means that the match was won.

Depth of baseline algorithm	**Depth of heuristic 2 algorithm**
113	**113**
113	**113**
8	10
11	10
113	113
113	113
113	**113**
113	113
113	113
13	**10**
113	**113**
59	**13**
113	**113**
113	**113**
49	113
113	113
113	**113**
35	49
113	**113**
113	113

As it can be seen from tables above offensive to deffensive algorithm seemed to do slightly better for 150ms and 300ms timeout than baseline one. The depth of the search was very similar for both algorithms (time needed for calculation of score function was very close). It seems that accuracy of new heuristic is slightly better than base one.