Results

The tests done in this report were between two players that were similar in all other aspects other than the tactic in question. This is to show the advantage that the tactic gives to the Player in comparison to smart player without the tactics.

Tactic One.

If the DomBoard is equal to InitBoard and the hand contains a (5,4) domino then play the domino. If the hand does not contain the (5,4) domino then play the highest scoring domino.

Below is a comparison between two players, one of which implements the tactic:

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 100 5

(52,48)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 5

(500,500)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 789

(514,486)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 78

(516,484)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 1

(493,507)

Tactic Two.

If the remaining score to finish the game is less than 9, try and find a domino that can score that score. The reason for nine is because it would be rare for a domino to score more than nine.

Below is a comparison between two players, one of which implements the tactic:

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 100 5

(61,39)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 5

(570,430)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 789

(567,433)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 78

(581,419)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 1

(544,456)

Tactic Three.

If the player is losing to the opponent, then play a domino to make the opponent knock.

Below is a comparison between two players, one of which implements the tactic:

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 100 5

(52,48)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 5

(502,498)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 789

(505,495)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 78

(518,482)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 1

(491,509)

Tactic four

If the Player has similar dominoes in the hand then it can play the highest scoring domino, if it doesn’t then play the second best highest scoring domino if there are similar dominoes, if there aren’t then the third best highest scoring domino is played.

Below is a comparison between two players, one of which implements the tactic:

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 100 5

(52,48)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 5

(502,498)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 789

(505,495)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 78

(518,482)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 1

(490,510)

The tests below show the performance of the smart player against the random player and the highest scoring domino player. This is to show that the tactics lead to an increase in performance in comparison to the highest scoring domino player an the random scoring domino.

Highest Scoring Domino Player:

\*SmartPlayer> domsMatch smartPlayer1 hsdPlayer 100 5

(61,39)

\*SmartPlayer> domsMatch smartPlayer1 hsdPlayer 1000 5

(570,430)

\*SmartPlayer> domsMatch smartPlayer1 hsdPlayer 1000 789

(575,425)

\*SmartPlayer> domsMatch smartPlayer1 hsdPlayer 1000 78

(578,422)

\*SmartPlayer> domsMatch smartPlayer1 hsdPlayer 1000 1

(550,450)

Random Player:

\*SmartPlayer> domsMatch smartPlayer1 randomPlayer 100 5

(99,1)

\*SmartPlayer> domsMatch smartPlayer1 randomPlayer 1000 5

(987,13)

\*SmartPlayer> domsMatch smartPlayer1 randomPlayer 1000 789

(979,21)

\*SmartPlayer> domsMatch smartPlayer1 randomPlayer 1000 78

(988,12)

\*SmartPlayer> domsMatch smartPlayer1 randomPlayer 1000 1

(980,20)

\*SmartPlayer>

The following tests were done to find out which is the best order to place the different arrangements of tactics against each other to determine which would be the best order to put the tactics in. The smart Player is in a logical order chosen by me. To prove that this is the best order to use I have performed the following tests.

Tactic order for Test Player: 1 3 2 4

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 100 5

(52,48)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 5

(504,496)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 789

(509,491)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 78

(524,476)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 1

(495,505)

Tactic order for test: 1 4 2 3

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 100 5

(61,39)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 5

(570,430)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 789

(567,433)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 78

(581,419)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 1

(545,455)

Tactic Order for Test: 1 4 3 2

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 100 5

(61,39)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 5

(570,430)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 789

(567,433)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 78

(581,419)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 1

(545,455)

Tactic order for test: 1 3 4 2

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 100 5

(61,39)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 5

(570,430)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 789

(567,433)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 78

(581,419)

\*SmartPlayer> domsMatch smartPlayer1 testPlayers 1000 1

(544,456)

The player number of losses that the smart player with the logical organisation of tactics took were less than the wins that the player had on average.

The Methods called in smartPlayer include:

Member: This method returns a Boolean value if a domino is a member of a domino list. It takes a domino and a domino list.

Similar: This method returns similar dominoes to the one supplied as a parameter from a hand. Similarity here means dominoes containing the same pip values as the one supplied in the parameters.

Remove: Which removes a domino from a list of dominoes

SmartPlayer: This is the method that implements the tactics

RemScore: This method checks for the remaining score of current player to the end of the game.

Losing: This method checks if the current player is trailing and returns a Boolean value.

Contain: This method returns a list of dominoes from a hand containing a certain value.

Stitch: if the opponent player is knocking the method returns a list of players from the hand that will make the opponent more likely to knock.

ListScore/pickDom/findDom: All three of these methods are used to find dominos that score a certain value.