REPORT

In order to select one final strategy, I created a set of best strategies with 50 elements. **A strategy is considered "best" if playing with it a player can win threshold consecutive times in some tournament where threshold I fixed value.**

For example, strategy_1 with threshold = 30 is considered "best" if playing with it I win 30 consecutive games.

Player one 'learns' best strategies while playing against opponents.

Algorithm for generating best strategies set:

1) *Generate random strategy for player _one and name it base_strategy, ind = 0*
   ind indicates the number of games won with base_strategy.
   Number of elements in the set of best strategies = 0
2) *While the number of set elements less than 50 do:*
3) *With base_strategy player _one goes to the tournament against the opponent whose strategy (opponents_strategy) is also randomly generated*
4) *If opponents_strategy is winning strategy than player one takes that strategy and base_strategy = opponents_strategy, ind increases by 1 (ind = ind + 1)*
5) *If ind>threshold than we add base_strategy to the list of best strategies. Number of set elements increases by 1 (number _of_elem = number_of_elem + 1)*

And that is how we generate a set of best strategies when players have the same number of soldiers to allocate to castles. The used threshold is threshold = 300.

When players do not have the same number of soldiers to allocate I used a modification of the previous algorithm:

1) *Generate random strategy for player _one and name it base_strategy, ind = 0, ind indicates the number of games won with base_strategy. Number of elements in the set of best strategies = 0*
2) *While the number of set elements less than 50 do:*
3) *With base_strategy player _one goes to the tournament against the opponent whose strategy (opponents_strategy) is also randomly generated*
4) *If opponents_strategy is winning strategy than player one takes new random strategy, ind increases by 1 (ind = ind + 1)*
5) *If ind>threshold than we add base_strategy to the list of best strategies. Number of set elements increases by 1 (number _of_elem = number_of_elem + 1)*

Note that with the number of soldiers < 100 we are less likely to win so we set threshold = 150 with the number of soldiers > 100 we are more likely to win so we set threshold = 310.

There are three sets with 50 best strategies each. The first set corresponds to the game where a player has 100 soldiers, second where a player has 90 soldiers and third where a player has 110 soldiers to allocate to 10 castles.

Now I have to select ones which will be considered best of the best. This is the reasoning behind this selection:

The player goes through all 50 strategies and enters 10 tournaments with 10000 opponents in each tournament.

Strategies for all opponents are randomly generated but with some additional constraints:

The opponent will randomly allocate 5 soldiers to castles C1, C2, C3. Then he will randomly allocate 40 soldiers to castles C4, C5, C6, C7 and 55 to castles C8, C9, C10.

This strategy comes from the fact that castles C8, C9, C10 have more value to the player than first few castles because they bring more points.

For each strategy I save:

MIN: Minimum number of victories of all 10 tournaments against 10000 opponents

MAX: Maximum number of victories of all 10 tournaments against 10000 opponents

AVG_WIN: Average number of victories 10 tournaments against 10000 opponents

GAP: Gap between the minimum and maximum value, a low gap is preferred

AVG_POINTS: average points won in all tournaments

All data is saved in three tables which can be found in files *data_100_blotto.pdf, data_90_blotto.pdf, data_90_blotto.pdf.*

Selected strategies are ones with maximum AVG_WIN and they are:

For 100 soldiers: [1, 0, 6, 2, 2, 16, 10, 12, 27, 24]

For 90 soldiers: [0, 1, 3, 2, 1, 14, 12, 18, 20, 19]

For 110 soldiers: [0, 3, 4, 1, 7, 4, 16, 17, 20, 38]

Programming language used is Python version 3.6.5.

All tables, problem implementation both in *.ipynb* and *.pdf* format can be found in the dropbox **Blotto** folder by clicking on the link:

https://www.dropbox.com/sh/ug6ru75mwjifdeo/AAAo42x33l_lq05ol1oZ47fca?dl=0

For any suggestions please contact me at donandjela95@gmail.com