First Compounds Analysis

1st Lorenzo Serafini
University Of Padua
Padua, Italy
lorenzo.serafini.1@studenti.unipd.it

2nd Michele Sprocatti
University Of Padua
Padua, Italy
michele.sprocatti@studenti.unipd.it
https://orcid.org/0009-0005-7886-441X

3rd Riccardo Zuech
University Of Padua
Padua, Italy
riccardo.zuech@studenti.unipd.it

***Abstract*—This project aims at finding if the two players can have advantages by starting or by being the second player in the game of First Compounds. To test if this is true or not, we defined some well-known strategies, and we simulated the game for different assignments of initial cards for the different pairs of strategies that the two players can play. Then we have done some analysis on the results to see if there are some advantages, and in case this is true, understand also for which player or for which pair of strategies.**

I. Introduction

In this work, we want to apply game theoretic tools to the analysis of a real-life tabletop game, called Primi Composti.

Our objective is first of all to understand whether this specific game is unfair, in the sense that it favours one player over another independently of their ”skill” level. Additionally, we also want to study the efficacy of approximated strategies for players, taking inspiration from fundamental concepts like security strategies but adapting them to a real-world scenario, where computational power is limited.

However, before delving into the specifics of those topics, we want to first present the game rules and then introduce our approach to its modelling.

A. Game's rules

In this work we focus on the 2 players scenario, where the rules are as follows.

There are 24 different cards whose value vary between 2 and 25, for a total of 9 primes and 15 composites in play. The 24 cards get distributed at random to the players, building 2 hands of 12 cards; the player that received the 2 gets to play first.

The game is turn-based: at each turn a player has to play a card from its hand, receiving a certain amount of points that gets added to its score, and then pass the turn to its opponent, until they both run out of cards in the hand.

Each of the players has two piles of cards in front of himself, one for cards of prime values and one for composites; when a card is played, it gets placed on the top of its player's primes or composites pile depending on its value.

To conclude, Table I-A shows how the points are assigned to a certain play. At the end of the game, whoever obtained the highest score wins.

It is easy to see that, under this formulation, ties can happen. However, we consider a tie as an invalid result that needs to be repeated, in order to simplify the analysis.

Column 1	Column 2	Column 3
C	1	Whoever plays a composite card, whose value is not the result of any operation among the visible cards on the table, gets 1 point.
P	2	Whoever plays a prime card, whose value is not the result of any operation among the visible cards on the table, gets 2 points.
C = C # C	3	Whoever plays a composite card, whose value can be seen the result of an operation between two composite cards visible on the table, gets 3 points.
C = P # C	4	Whoever plays a composite card, whose value can be seen the result of an operation between a composite card and a prime card, both visible on the table, gets 4 points.
P = C # C	4	Whoever plays a prime card, whose value can be seen the result of an operation between two composite cards visible on the table, gets 4 points.
P = P # C	5	Whoever plays a prime card, whose value can be seen the result of an operation between a composite card and a prime card, both visible on the table, gets 5 points
P = P # P	6	Whoever plays a composite card, whose value can be seen the result of an operation between two prime cards visible on the table, gets 6 points.

TABLE I
RULES FOR ASSIGNING POINTS TO A PLAYER AFTER IT PLAYS ONE OF ITS CARDS. SYMBOL # INDICATES AN ARITHMETIC OPERATION, WHILE P STANDS FOR PRIME AND C FOR COMPOSITE.

B. Modelling approach

It is easy too see that this game can essentially be modelled as a finite multi-stage Stackelberg game, where at each stage the hands of the players and the cards visible on the table depend on the outcomes of the previous stages.

In particular, it is interesting to notice how this game is of complete imperfect information, since the content of the other player's hand is deterministic given one's hand, that is a Nature's choice. However it is worth observing how, after Nature's choice, the game becomes of complete perfect information. This is interesting because, after observing the Nature's choice, each player can theoretically ”solve” the game by always playing a best response through backward induction on the resulting decision tree.

However, the backward induction approach, i.e. a strategy choice that would lead to a SPE, is infeasible to implement in practice, given that the resulting decision tree would have $2^24 - 1$ nodes and a total of $24!$ possible different sequences of plays (number of leaves). This major practical limitation hinders a rigorous analysis of the game, limiting us to only consider sub-optimal strategies due to the computational infeasibility of computing strategies that would yield subgame-perfect Nash equilibria.

Additionally, another major problem in analysing this game arises when considering the randomness of the hands: Nature has a total of ${}_{24}C_{12}$ possible choices, meaning that the number of games to consider would be well beyond practical feasibility. However this problem is easily solvable, since Nature's choice involves randomness, by using an high enough number of trials we can obtain results that are good in approximation.

Another interesting aspect of this game is that it can be modelled also as a zero-sum game: it is straightforward to prove that if we do not only add the points of a play to the score of the relative player, but also subtract them from the score of its opponent, the final result, i.e. winning or losing the tabletop game, remains the same. This aspect is particularly interesting when we consider the possible strategies for our players, something that we will delve into in Section II.

II. Strategies

We decided to test different strategies against each other to see if one of them outperformed the others. In particular, we implemented the following strategies:

- random strategy: the player plays a random card from their hand;
- max strategy: the player plays the card with the maximum number written on it;
- min strategy: the player plays the card with the minimum number written on it;
- prime_first strategy: the player plays at first only prime numbers cards, if any, chosen at random, otherwise he plays composite numbers card;
- composite_first strategy: the player plays at first only composite numbers cards, if any, chosen at random, otherwise he plays prime numbers card;
- max_val strategy: the player plays the card that gives him the highest score; TODO
- security strategy: the player plays the card that maximized his minimum score; TODO
- minimax strategy: the player plays the card that minimizes his maximum score.

III. Simulation

To understand if the first player has some moving advantage, we had to run several simulations by comparing the different strategies explained in section II in the two cases that we want to compare:

- General case: the starting player is decided by the random cards that they got.

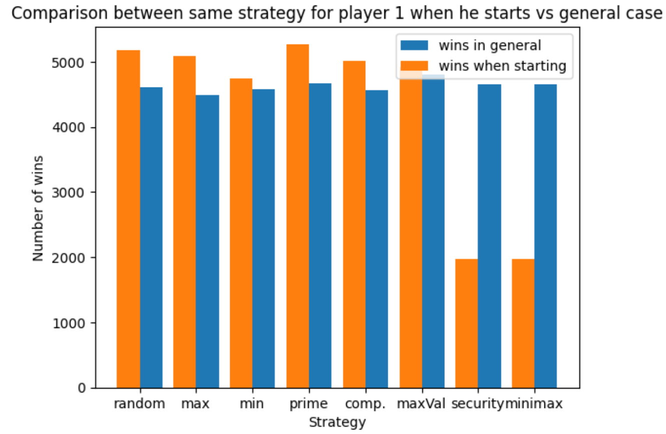

Fig. 1. Wins of player 1 in the two cases when the two players play the same strategy.

- Player 1 always starting: we forced the fact that the player 1 has the card corresponding to number two so he starts always the game.

In order to achieve this, firstly we generate the different possible configurations of cards by assigning each card uniformly at random between the two players. Then for each assignment, we run the game comparing every possible pair of strategies from the set explained in II.

A. Results

To have enough data to make statistical analyses but also to have limited computation time, we decided to do 10000 samples of possible assignments; then we simulate the different strategies.

By looking at figure 1, that plots the win of player 1 when the two players play the same strategy, we can see that there is some small advantage for some strategies (random, max, min, prime, comp, max_val) versus a great worsening for two particular strategies (security, minimax).

This is also reflected in the opposite plot, so the one for player 2 reported in figure 2.

After these, we have computed the probability that player 1 wins the game for every possible pair of strategies. The probability is computed as follows:

$$prob = \frac{number - of - wins}{number - of - total - matches} \qquad (1)$$

To visualize the results in these two cases, we decided to use the heat-maps; the plots are reported in figure 3 and in figure 4.

From the plot in figure 3 we can see that in the general case the probability of winning for player 1 is very high when these combinations arise:

1) max_val vs random
2) security vs random
3) minimax vs random
4) max_val vs min

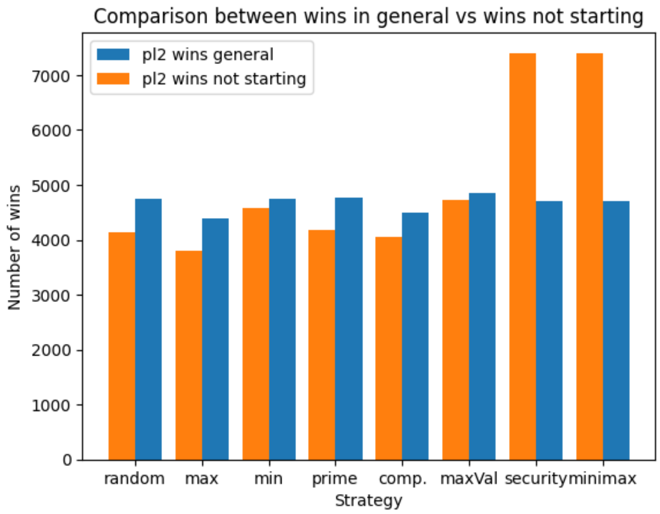

Fig. 2. Wins of player 2 in the two cases when the two players play the same strategy.

5) security vs min
6) minimax vs min
7) security vs prime first
8) minimax vs prime first

The first 3 were somehow obvious because the second player is choosing the card randomly without a strategy, so a good strategy for player 1 will very likely beat player 2. The other five results instead are more interesting. Since the probability of winning for player 2 is $1-probability-winning-player-1$ we can see the dark part at the top right of the heat-map.

By looking at figure 4 we can see what we have pointed out before using figure 1 and 2 that is the fact that we have a significant drop in the chance of winning for the strategies in the bottom right part of the heat-map, but we have an increasing probability for all the other combinations.

To visualize in a better way this last observation we plot also the difference between the two sets of probabilities and we obtained the heat-map in 5, using this plot we can see that we have a 5 % increment in the top left but a 25 % decrease in the bottom right. So given these values we can say that maybe the player that does not start has an advantage because in certain pairs of strategies the probability of the starting player increases by a small percentage but it decreases a lot for other strategies so the not-starting player has in general a greater chance to win.

For all possible simulations that we have run, we also collected the different scores of each match. These results are used in order to plot a box plot for each pair of strategies in the two cases that we have analyzed. These can be seen in figure 6 and figure 7. From the comparison between these plots, we can see again the bottom left part where we have a worsening of the scores of player 1 from general to the case where it starts confirming again what we have said before. Also, we can see that, in general, we have very small boxes but very large whiskers; this means that we have a large variance of

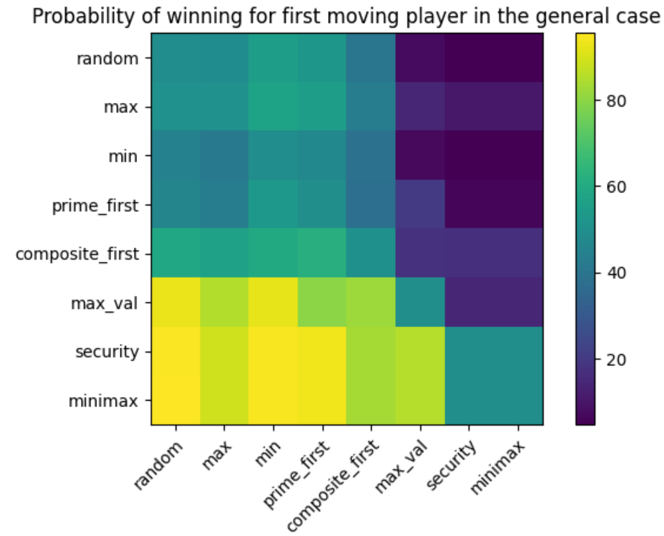

Fig. 3. Probability of winning for player 1 in general case (strategy of player 1 on the rows).

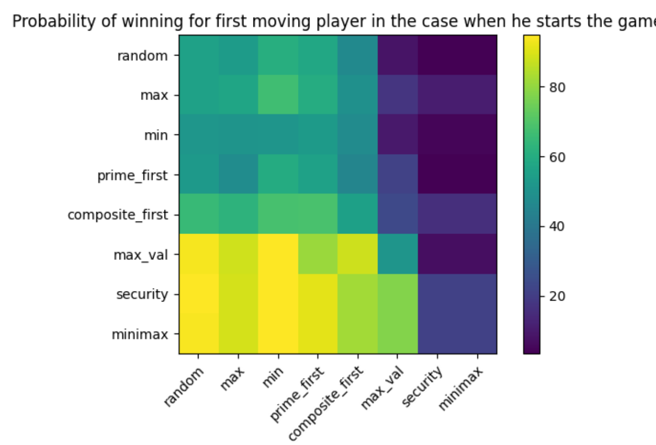

Fig. 4. Probability of winning for player 1 in the case where he always starts (strategy of player 1 on the rows).

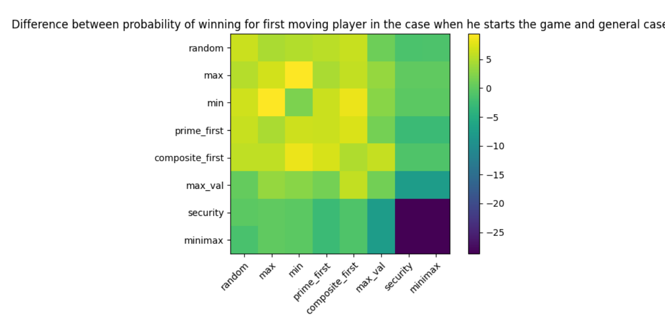

Fig. 5. Difference between probability of winning for player 1 when it starts and the general case.

Fig. 6. Plot that reports the box plots of the different score for each pair of strategies in the general case.

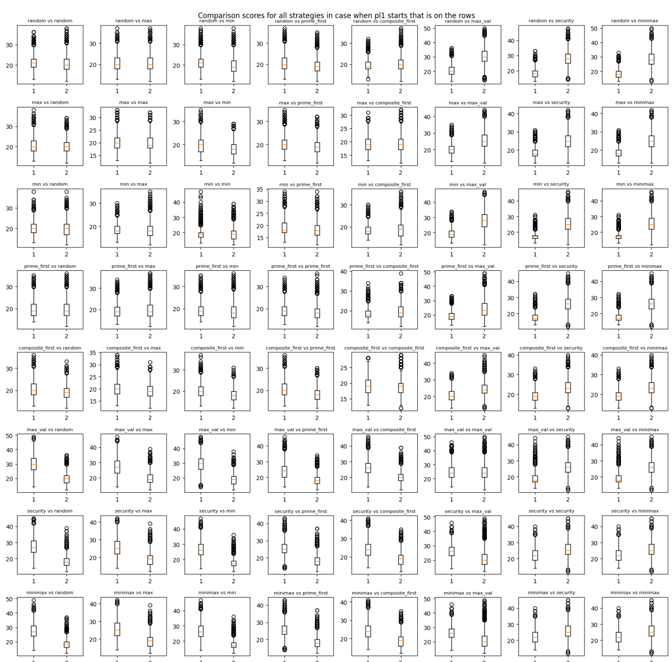

Fig. 7. Plot that reports the box plots of the different score for each pair of strategies when player 1 starts.

the scores but also that in the 2nd and 3rd quantiles we have data that is very concentrated. We can also see that there are a lot of outliers pretty much everywhere. In particular, we can see that when player 1 starts, the number of outliers increases a lot for both players.

At the end of the code, there is also a small part that uses an optimization method to find the binomial distribution that best fits the data in the two cases. The distributions found are very similar to the ones computed before using the equation 1.

IV. Conclusions

Acknowledgment

References