

MDSAA

Mestrado em
Data Science and Advanced Analytics

CIFO PROJECT

Sports League Optimization

Tiago Pedro Soares

20240655

Index

1. Formal Definition of the Problem	2
2. Description of Selection and Genetic Operators	2
3. Performance Analysis	4
4. Justification of Decisions.....	5
5. Conclusions	6
Appendix A.....	7
Appendix B.....	7

1. Formal Definition of the Problem

In this project, we attempt to form balanced teams based on a set/dataset of players. The goal of this project is to group players in a way that maximizes the overall performance of teams while ensuring that they are balanced. The space for possible solutions to an optimization problem is very large. Our goal is to find a combination that maximizes the evaluation function (fitness).

There are five teams representing each solution (individual), with a goalkeeper, two defenders, two midfielders, and two attackers. Players are aggregated into Team objects, so the vector has a length of five teams, and each element has a length of seven players. Based upon an analysis of the teams and a comparison of the requirements of each team, an integrated function, `is_valid()`, determines if the individual is valid.

The search space corresponds to the set of all possible valid combinations of players distributed across five teams, without repetitions and with the correct positional composition. Also, each team may not exceed the budget limit of 750 million euros. Solutions that do not comply with these conditions will be automatically considered invalid and penalized.

The fitness function can be predefined using the function toolkit, which allows the user to set the fitness function dynamically based on multiple criteria by forming two bins based on the average. This is useful if there is an equal division between two solutions or a larger deviation between the splits. You must also specify the performance of various metric goals, such as recall and F-score, and decide whether you want to minimize the false positive rate or maximize the true positive rate.

It is ensured that composition and budget constraints are met through internal methods that ensure: Per team, there is one goalkeeper, two defenders, two midfielders, and two forwards; There is only one team for each player; The maximum amount per team is € 750 M. In this model, selection, crossover, and mutation operate effectively, preserving solutions and exploring the space with search and diversity.

2. Description of Selection and Genetic Operators

During the different developmental phases, a set of selection and genetic modification operators were developed and implemented. One of the main goals was to construct balanced teams abiding by all assumed constraints, including position budgets at different hierarchical levels of the organization. Consequently, these operators and their selection strategies modify – often in an adverse manner – the fitness landscapes of the solutions evolving by the genetic algorithm.

2.1. Selection Strategies

Different selection strategies were used during reproduction that differed in genetic diversity and selective pressure. We chose these strategies: In the tournament selection, a few (2-3) individuals are selected, and the best one is selected. Providing selective moderation and preventing too fast convergence. Roulette wheels select participants based on their fitness level. All candidates have a chance, but the best performer is more likely to be chosen. A random individual from the top 30% of the population is selected in a truncation selection to direct evolution to integer-based solutions. Furthermore, rank ordering can counteract stronger variation between individuals that could harm evolution. SUS reduces chance-related variance by placing options more randomly. Evolutionary elitism ensures that high-quality solutions endure.

2.2. Crossovers Operators

There are several strategies developed for crossbreeding using positional rosters. Team Swap involves swapping entire teams between two parents. Therefore, the proposed solution can be greatly altered. A player swap involves exchanging players from different countries occupying the same position, so the positional arrangement is maintained. A uniform team has a 50-50 chance of being able to identify either parent's child. Hence, descendants are produced with different parents. A position-based crossing replaces one position holder with someone else. The Gene Level Crossing balances the composition of teams without repeating players. Players from both parents are combined with stylistic rules, and many different techniques increase fragmentation.

2.3. Mutation Operators

To avoid local convergence, some mutation operators were applied: Random Substitution consists of changing players that have not been put to some position with other players. Team Swap: exchange players that have the same position in different teams. Team Replacement is the process of replacing a valid team with another valid one under specified conditions. Most Expensive Player Replacement identifies the least cost effective one and replaces him with a cheaper one. Cross Position Swap replaces the players from distinct teams but assigns them players who occupy the same position. The younger players are to replace the older players in respective positions once they become eligible as strikers. A team rotation might happen on an individual when the teams are restructured within individual which will influence crossover operations afterwards.

3. Performance Analysis

3.1. Global Comparison of Selection Methods

Comparison of selection techniques within a genetic algorithm is presented in two figures. *Fig. 1 shows the best individual over several generations while Fig. 2 shows average fitness by generation with confidence intervals demonstrating method consistency.* The elitism method, as shown in *Fig. 1*, achieves and maintains a maximum fitness value of 1000 from approximately the 6th generation onwards. Both initial exploration and conservation of best solutions occurs based on the portrayed behavior. Optimal fitness may be attained by the rank method as well, but the curve is much shallower and takes many generations to reach optimal fitness. Truncation, tournaments, and roulette achieve these intervals faster; fitness values are achieved slower yet converge far later. The SUS method depicts the most irregular and slowest curve illustrating increased variability and less stability. Along with this analysis, *Fig. 2* depicts the average trends over thirty generations for each individual methodology. Regardless of best execution, all methods overly converge by the 15th generation with slight advantages led by elitism and truncation. Retention of best individuals is aided by intervention alongside confidence variables.

3.2. Analysis of Fig. 3 - 32

The performance of a genetic algorithm is affected by the interplay of selection, crossover, and mutation operators under all six selection schemas. Mutation operators were *random_player*, *replace_team*, *cross_position*, as well as *team_swaps* and *player swaps*. Rank selection guarantees stability and avoidance of stagnation. Due to the high recombination and mutation rate, roulette worked well. It excelled at *uniform_team* and *team_swaps*. Multi-runs with maximum fitness enabled by tournament selection. In combination with strong mutation and crossovers of structure such as *uniform_team*, *team_swap*, and *player_swap*, truncation selection can be extremely robust. These combinations achieved a fitness of 1000 rapidly and consistently, cementing them as global strategies. This one incorporates elitist selection blended with pressure-based selection (truncation or tournament) and high recombination crossovers with intensive mutation, set to 100% over a small number of generations, interleaved by rest phases. We can observe these results with Fig. 3 – 32.

3.3. Analysis of Selection, Crossover, Mutations, and Genetic Rates

Evolutionary operators are strongly correlated with empirical data. In Figure 33, Rank (992.30), Truncation (991.64), and Tournament (990.87) turned out to have the best mean fitness scores, surpassing SUS (990.01), Roulette (989.91), and Elitism (989.43). Most effective methods showed improved robustness and higher convergence stability, demonstrating stronger tighter distribution with less outliers (Figs. 34 and 35). Figure 34 shows that *team_swap_crossover*, *player_swap_crossover*, and *uniform_team_crossover* achieve optimum fitness without fail. *Gene_level_crossover* is less robust, as evidenced by a lower median and higher outcome variability. In Figure 35, *random_player_mutation*, and *replace_team_mutation* results are almost optimal. *Expensive_player_mutation* and *forward_reset_mutation* result in high variation for strategies that require rapid convergence and rigid consistency. Figure 36 shows *uniform_team_crossover* and *rotate_teams_mutation* outperforming *team_swap_crossover* and *replace_team_mutation*, where all methods scored over 1000 with little deviation. Crossovers possess high recombination and structural diversity needed to escape local optima. Both pair plots of fitness against genetic rates (Fig. 37) and mutation rate against fitness area show.

4. Justification of Decisions

This section outlines our extensive work with various combinations of evolution strategies and modifications made to the genetic algorithm. For decision making, fitness averages, system stability, and convergence velocity metrics were employed. Main efforts were devoted to adjusting the balance between selective pressure, population diversity, and the search space exploration. Implementing these strategies will yield optimal outcomes. We can observe/find the information in **TABLE A**.

4.1. Selection Method

The values achieved, such as Truncation 1000.00, Classification 999.50 and Tournament 999.00, represented mean effectiveness. High selective pressure was exhibited while coupled with highly conservative solution removal. Solutions encouraged convergence towards the target. Stable and efficient evolution is demonstrated by the visual representations described earlier in this article.

4.2. Crossover Type

Ranked by averages, Truncation 10000, Classification 999.50, and Tournament 999.00 achieved the best averages. Increased selective pressure removed poorly adapted solutions and accelerated convergence to resolved solutions. In previous sections, the provided visualizations showcased stable and efficient evolution. The order of reliability and speed in evolution was enhanced. Successful crossovers included `players_swap_crossover`, `uniform_team_crossover`, and `team_swap_crossover`. One combination achieved maximum fitness (1000.00) by using `uniform_team_crossover` in the first two combinations in the table. As a result, this operator encourages significant recombination within individuals, preventing local optima stagnation. The traits chosen were genetic diversity and adaptability.

4.3. Mutation Type

The `rotate_team_mutation` was found to occur in three of the five top combinations and was strongly linked to optimal result achievement. Also performing strongly was `replace_team_mutation`. Both examples make structural alterations to dependents that add to diversity and shift away from a local optimum.

4.4. Genetics Rates

The mutation rate exhibits near-synchronization with the attaining of peak fitness scores, reaching 100% achievement in these cases. Additionally, the persistently best combinations showed an 80% crossover rate. In this case, variety is actively preserved while effective recombination of the best population members ensures a high degree of genetic activity.

5. Conclusions

In this genetic algorithm, selection, crossover, mutation and other genetic metrics were used to solve a multifaceted problem. Statistics show that all performance metrics were balanced, visualized, and cross-compared. The described space was effectively evaluated by applying various selection criteria, including truncation, classification and tournament selection, and uniform team mutations. The configuration with truncation selection and uniform team crossover achieved a peak fitness of "1000.00" with 80% crossover and 100% mutation, maintaining a genetic rate. Verification by empirical and comparative analysis is also important. These algorithms demonstrate their optimization and accuracy. The mechanistic approach provides a foundation for combinatorial optimization, demonstrating flexible adaptability across various disciplines.

Appendix A

Selection	Crossover	Mutation	Fitness
<i>truncation</i>	uniform_team_crossover	rotate_teams_mutation	1000.0
<i>rank</i>	uniform_team_crossover	rotate_teams_mutation	999.5
<i>tournament</i>	team_swap_crossover	rotate_teams_mutation	999.0
<i>elitism</i>	team_swap_crossover	replace_team_mutation	999.05
<i>roulette</i>	player_swap_crossover	replace_team_mutation	998.9

Table A - Fitness Evaluation Best Combinations

Appendix B

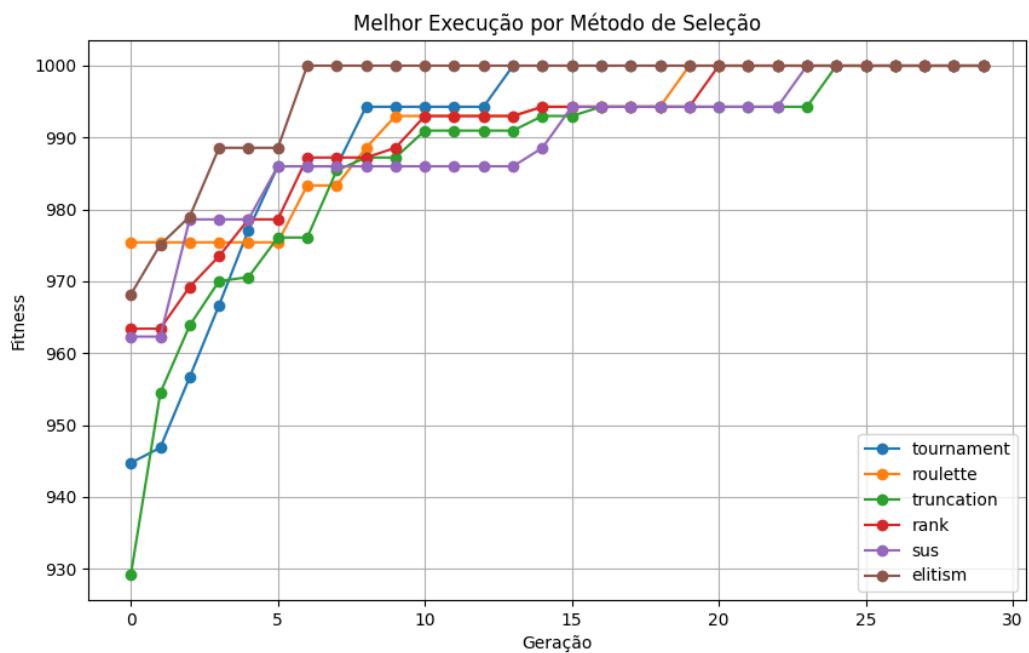


Figure 1: Best Comparation Per Selection

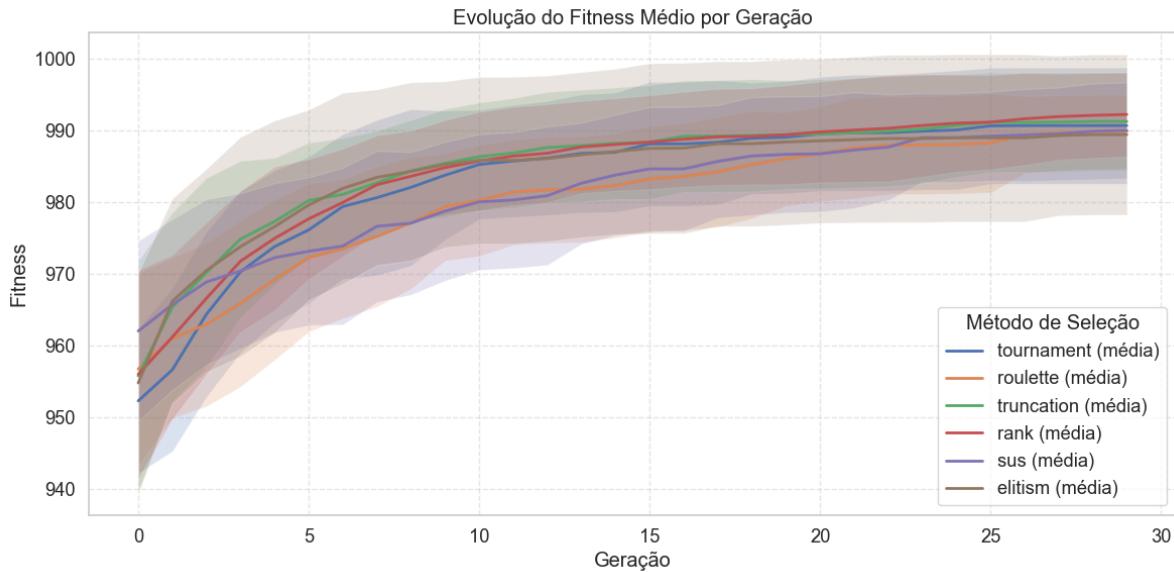


Figure 2 - Average Fitness Evolution Per Generation

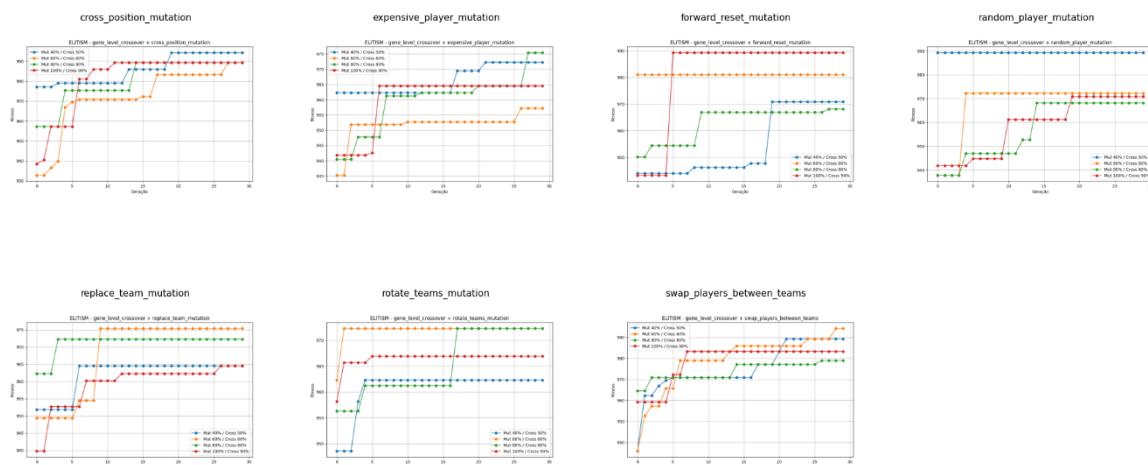


Figure 3 - Elitism Selection: Gene Level Crossover Combinations

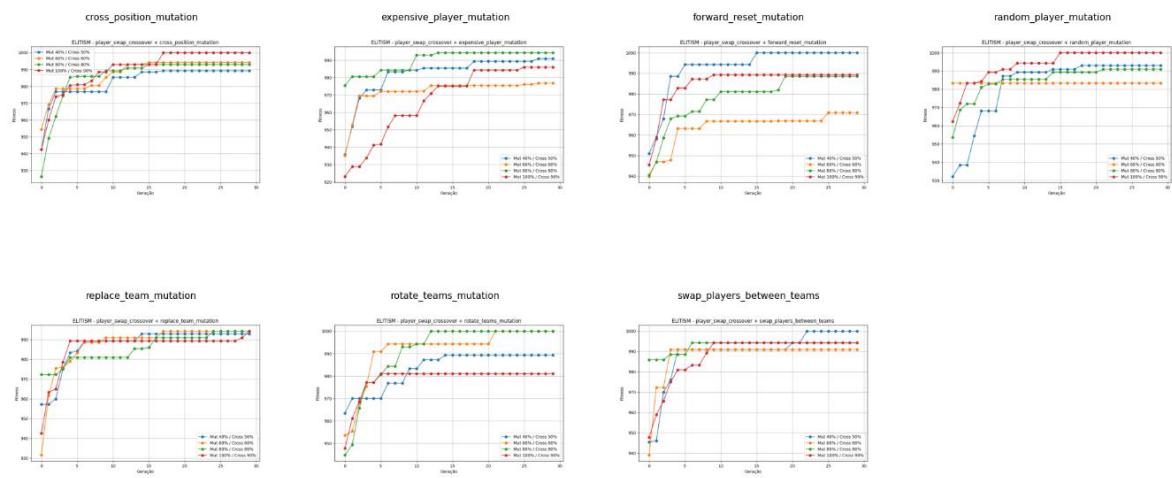


Figure 4 - Elitism Selection: Player Swap Crossover Combinations

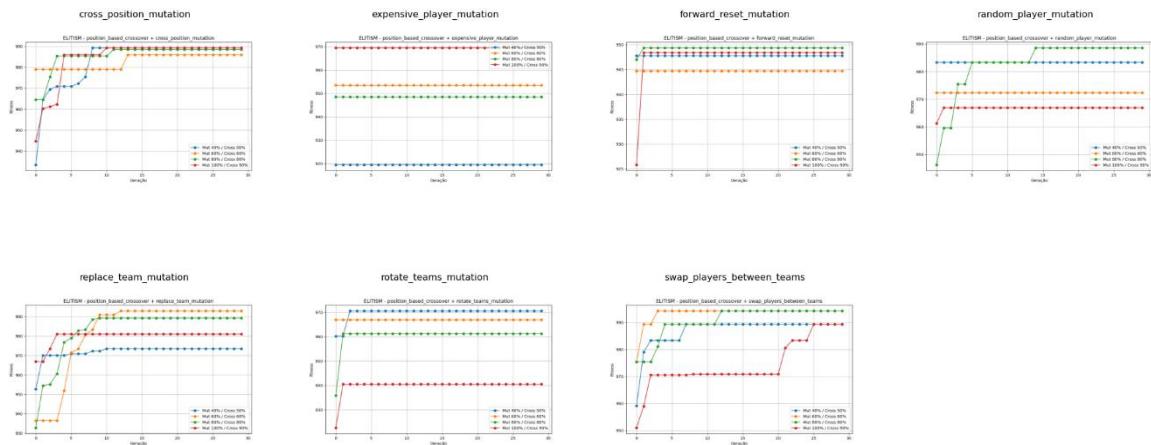


Figure 5 - Elitism Selection: Position Based Crossover Combinations

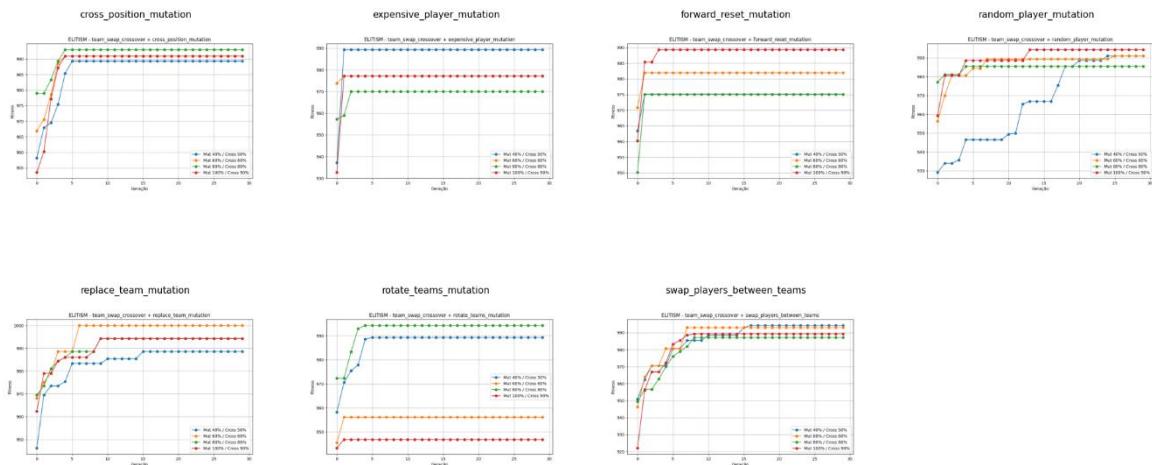


Figure 6 - Elitism Selection: Team Swap Crossover Combinations

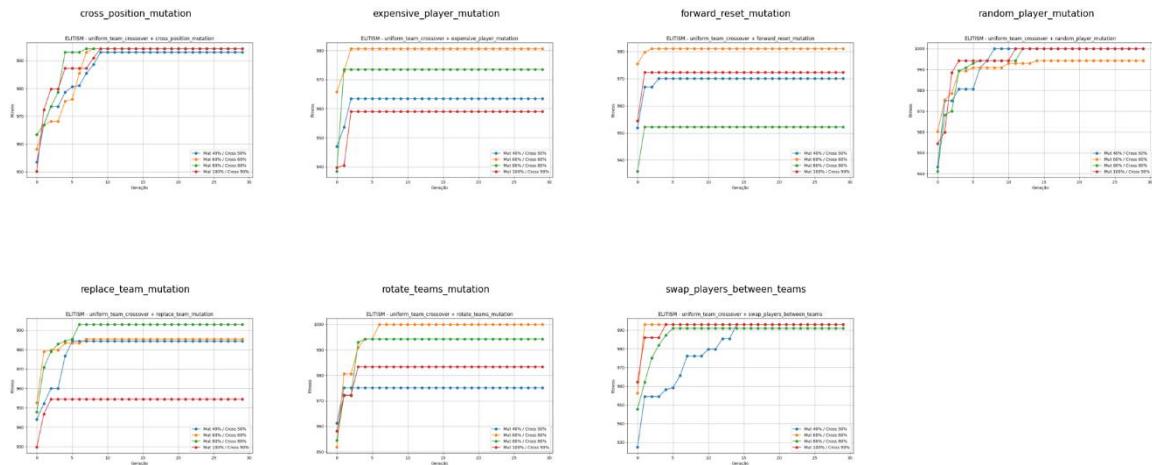


Figure 7 - Elitism Selection: Uniform Team Crossover Combinations

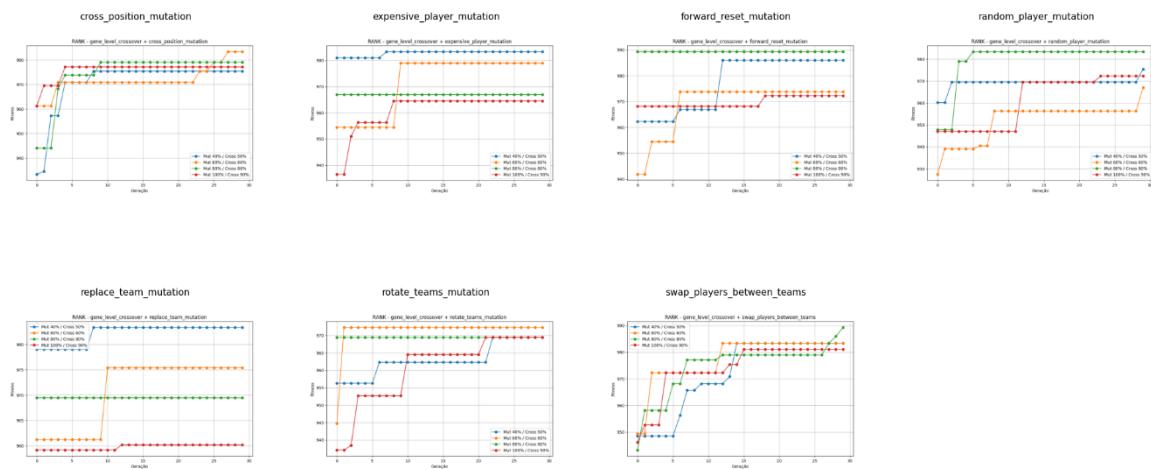


Figure 8 - Rank Selection: Gene Level Crossover Combinations

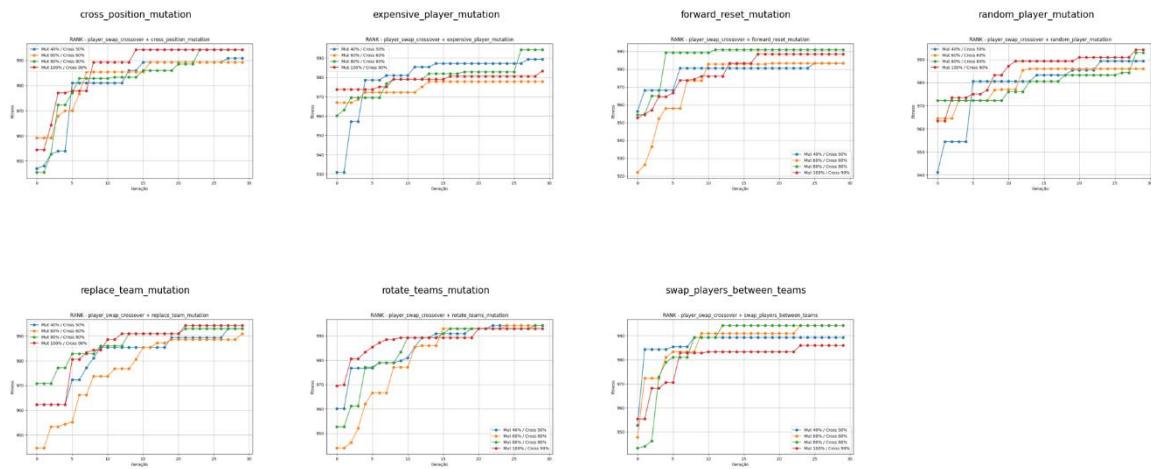


Figure 9 - Rank Selection: Player Swap Crossover Combinations

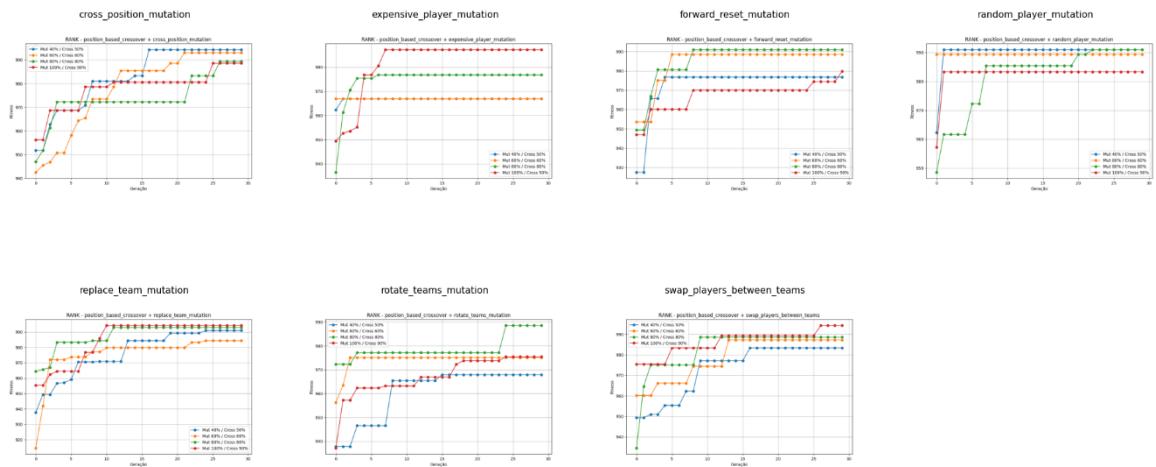


Figure 10 - Rank Selection: Position Based Crossover Combinations

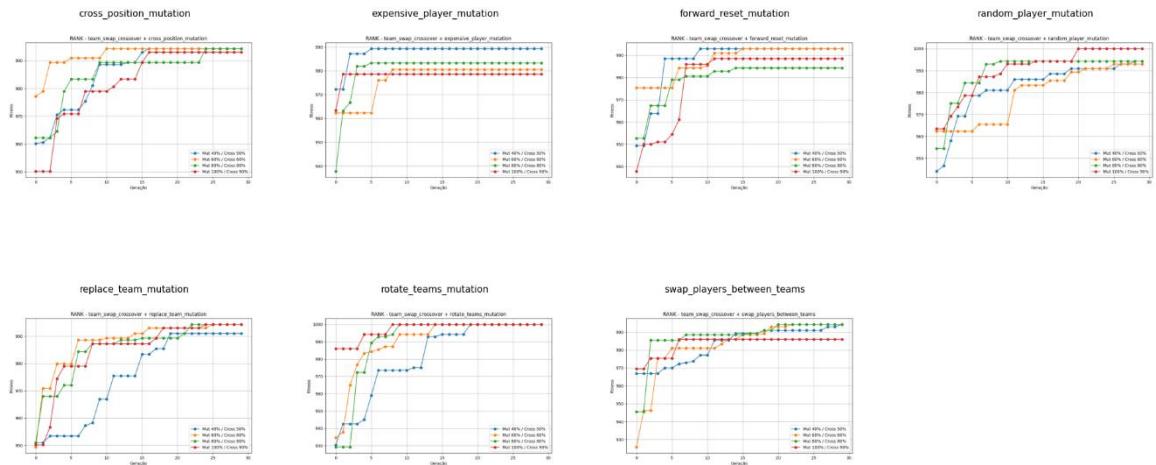


Figure 11 - Rank Selection: Team Swap Crossover Combinations

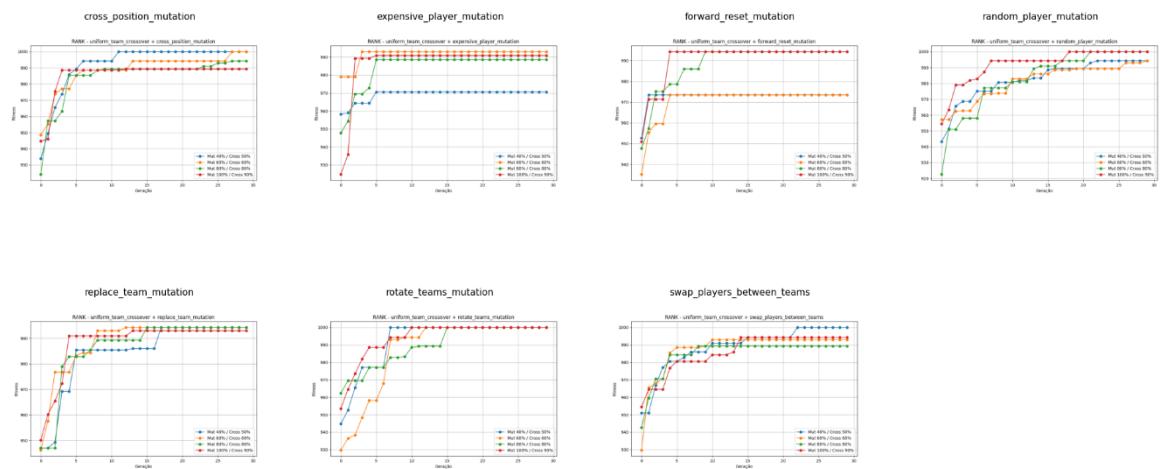


Figure 12 - Rank Selection: Uniform Team Crossover Combinations

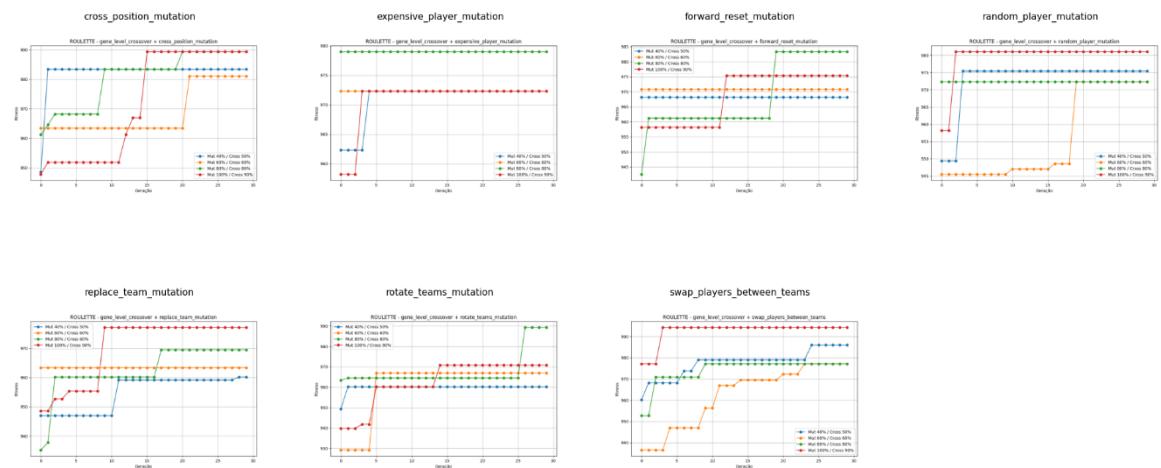


Figure 13 - Roulette Selection: Gene Level Crossover Combinations

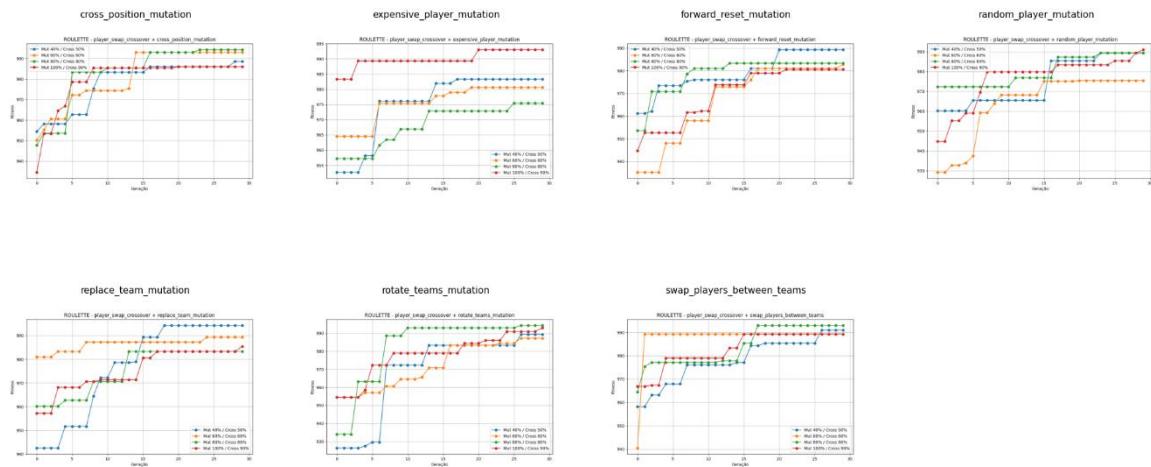


Figure 14 - Roulette Selection: Player Swap Crossover Combinations

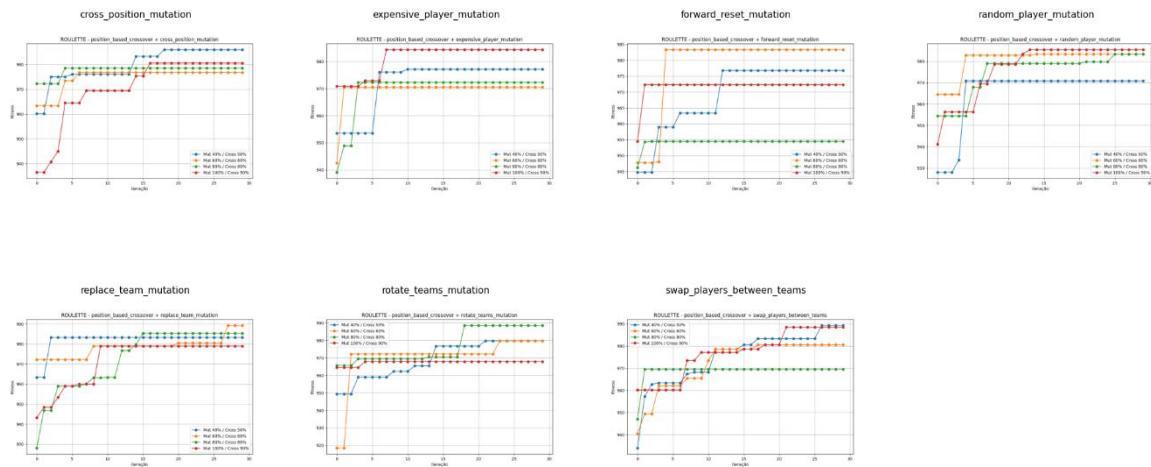


Figure 15 - Roulette Selection: Position Based Crossover Combinations

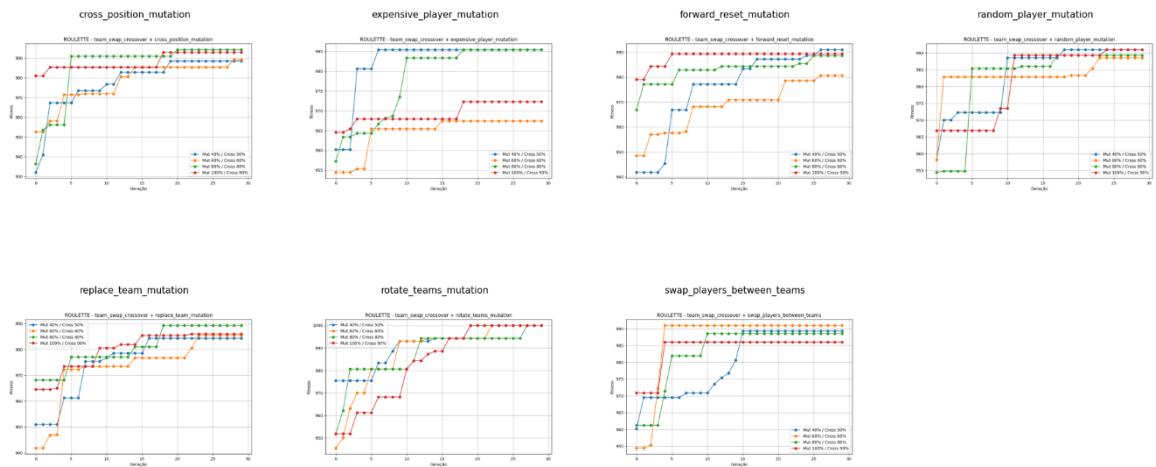


Figure 16 - Roulette Selection: Team Swap Crossover Combinations

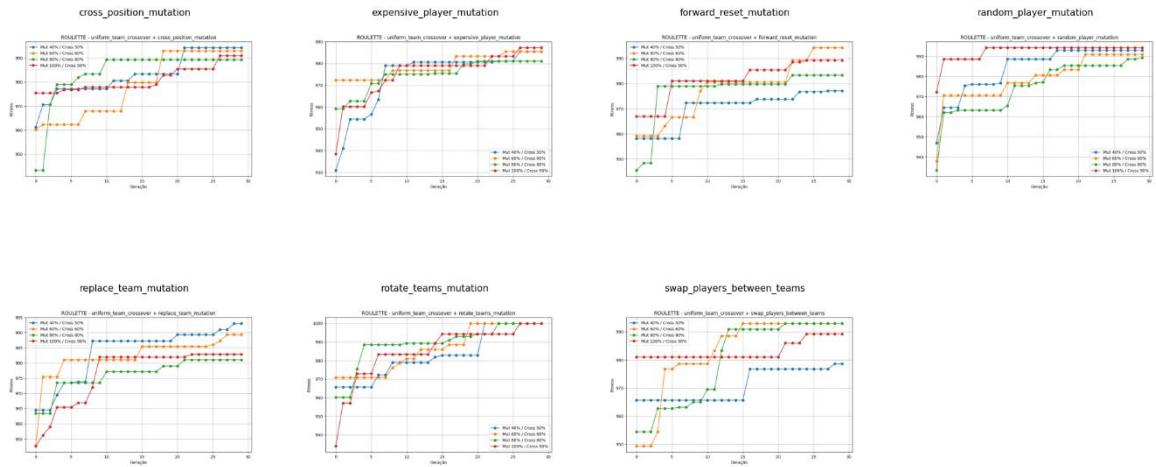


Figure 17 - Roulette Selection: Uniform Team Crossover Combinations

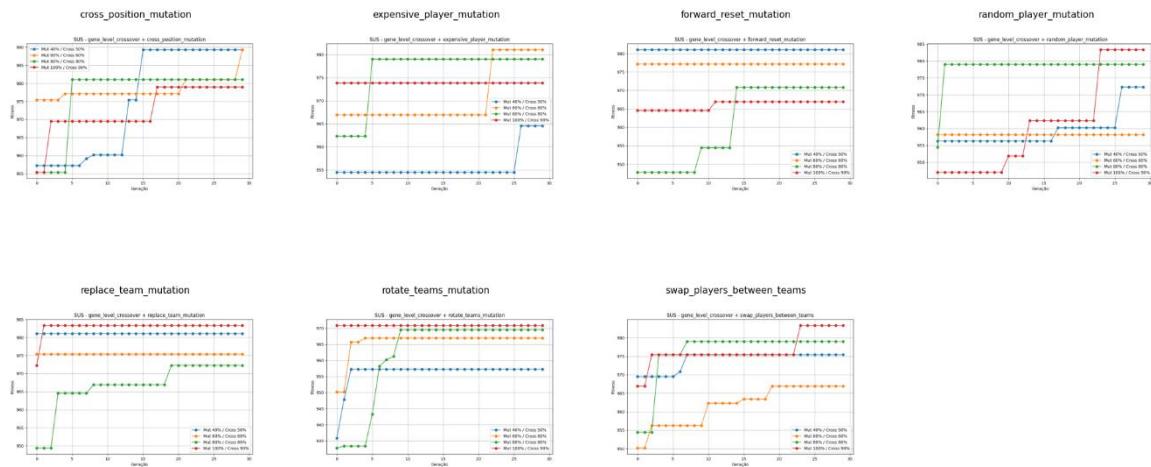


Figure 18 – SUS Selection: Gene Level Crossover Combinations

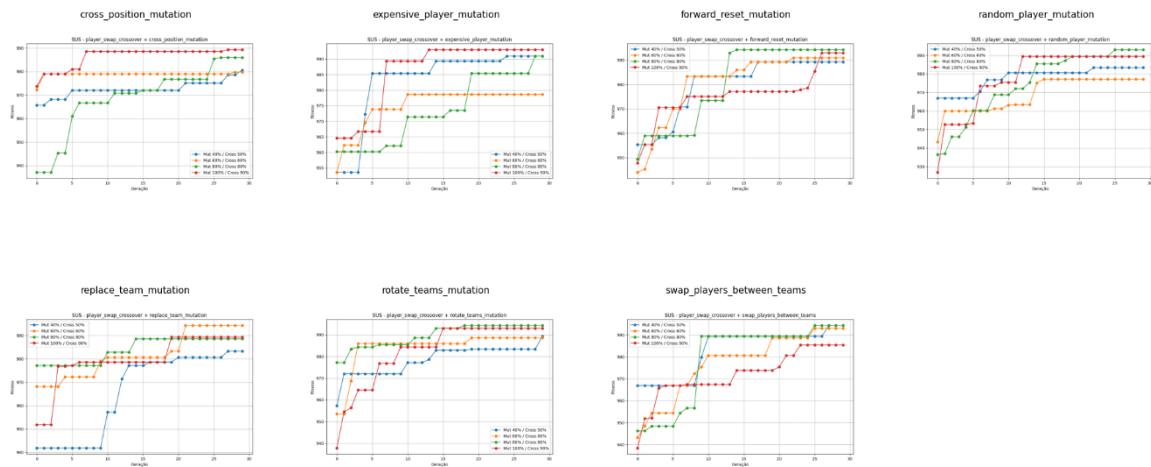


Figure 19–SUS Selection: Player Swap Crossover Combinations

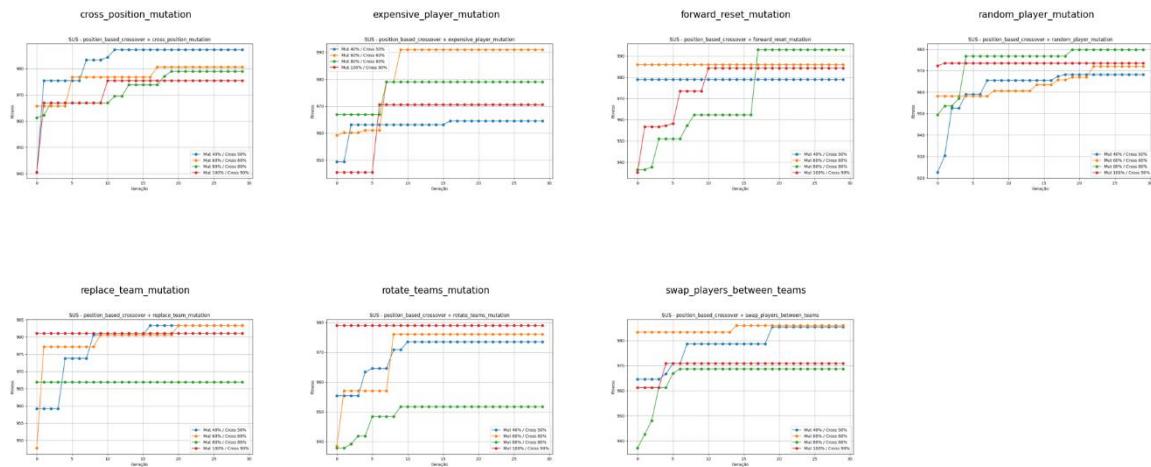


Figure 20–SUS Selection: Position Based Crossover Combinations

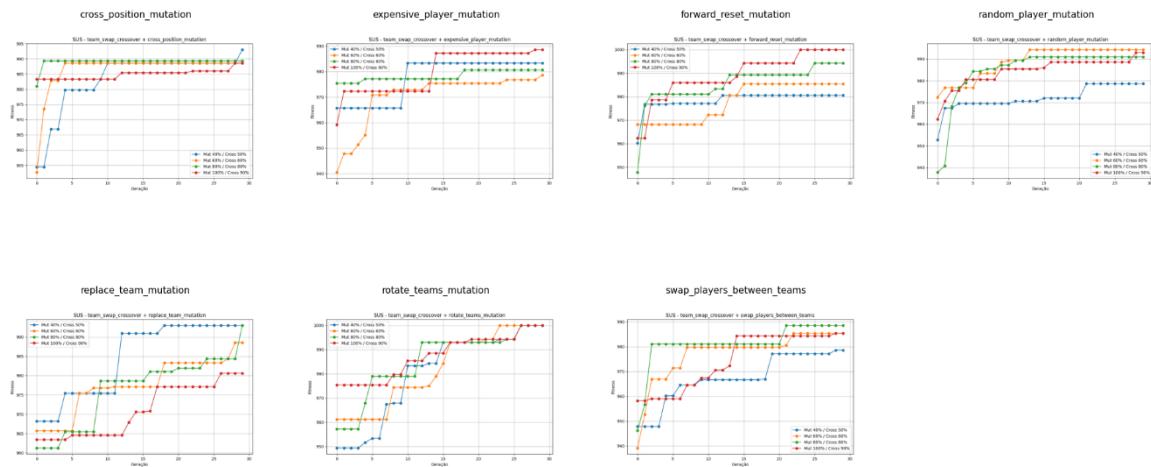


Figure 21–SUS Selection: Team Swap Crossover Combinations

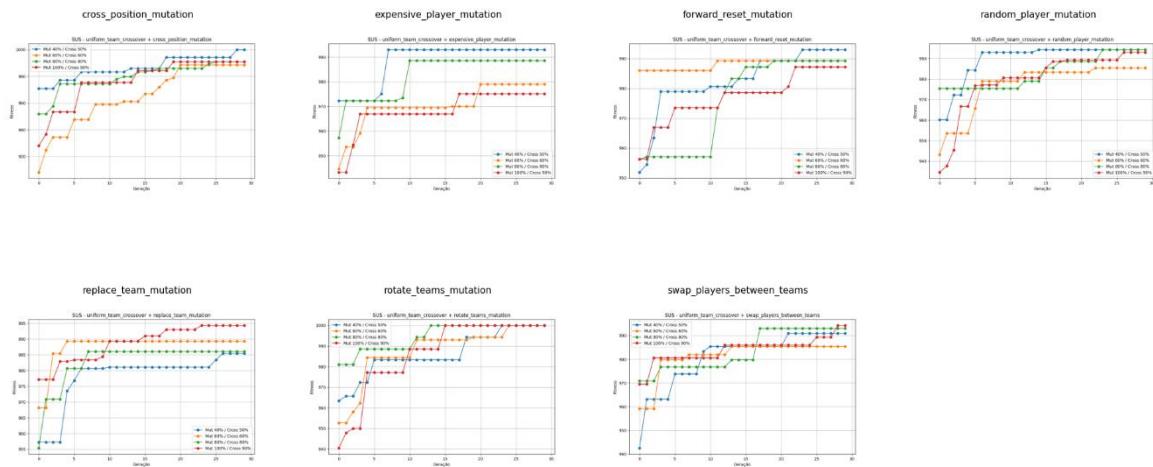


Figure 22– SUS Selection: Uniform Team Crossover Combinations

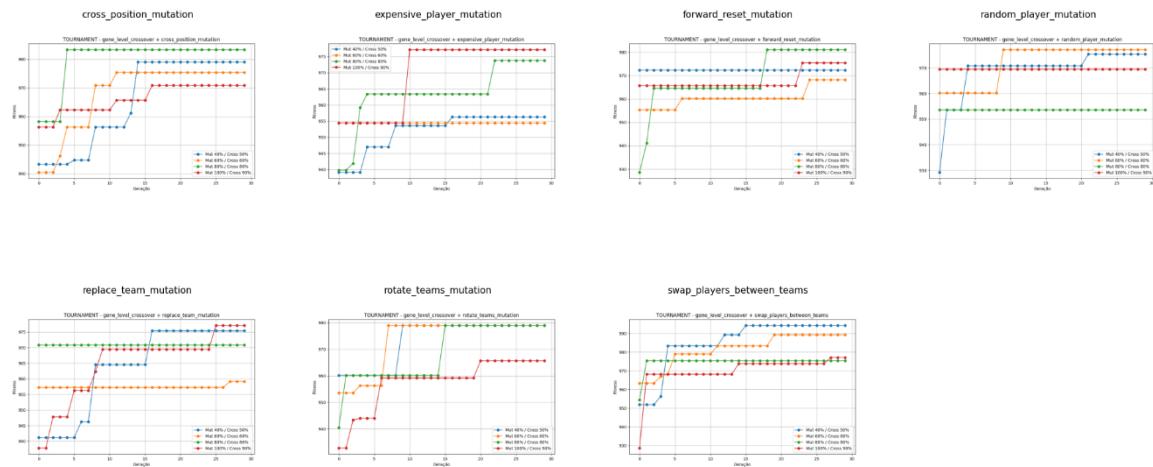


Figure 23– Tournament Selection: Gene Level Crossover Combinations

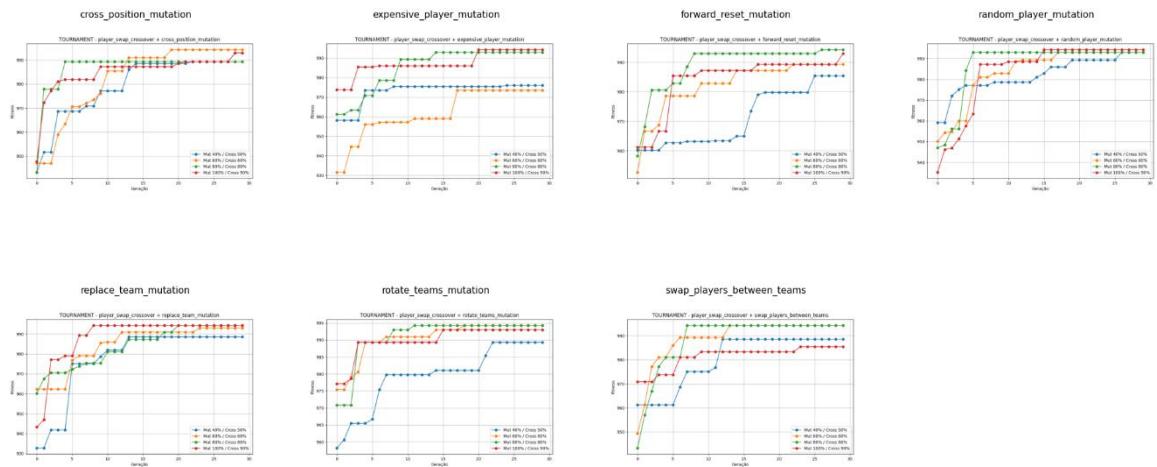


Figure 24– Tournament Selection: Player Swap Crossover Combinations

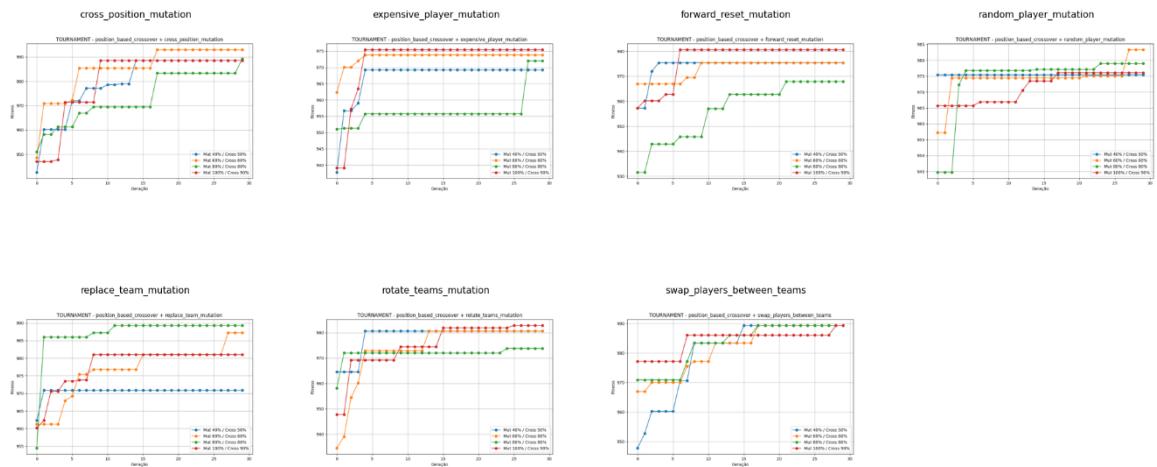


Figure 25– Tournament Selection: Position Based Crossover Combinations

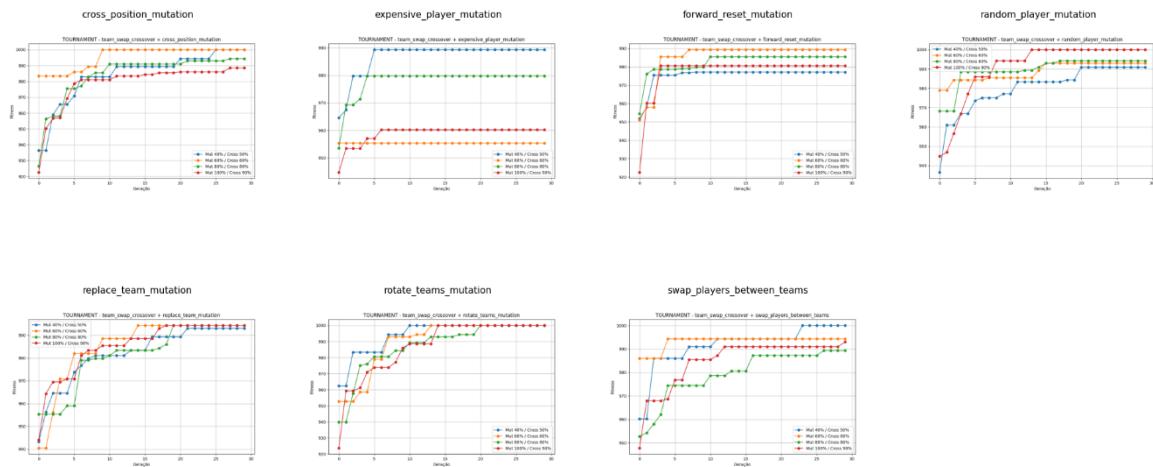


Figure 26– Tournament Selection: Team Swap Crossover Combinations

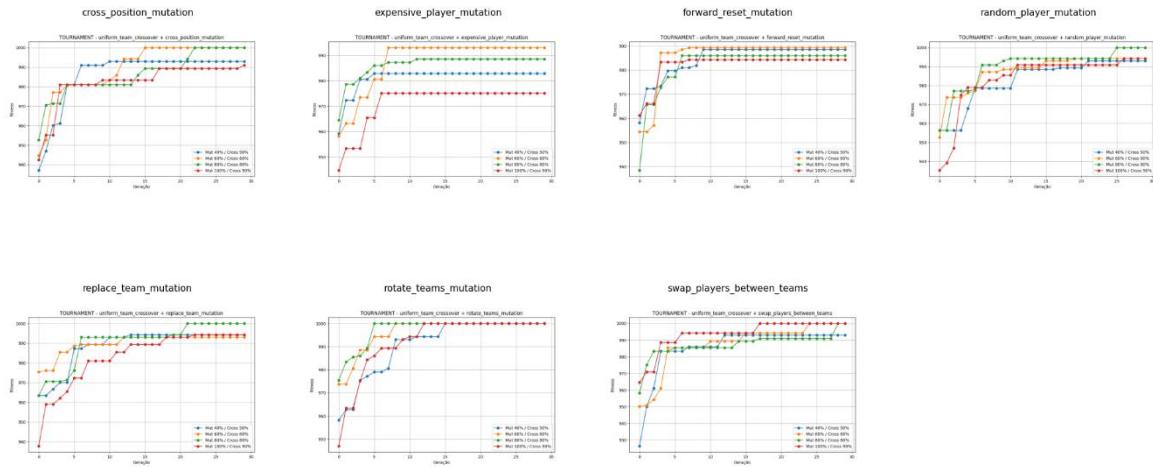


Figure 27– Tournament Selection: Uniform Team Crossover Combinations

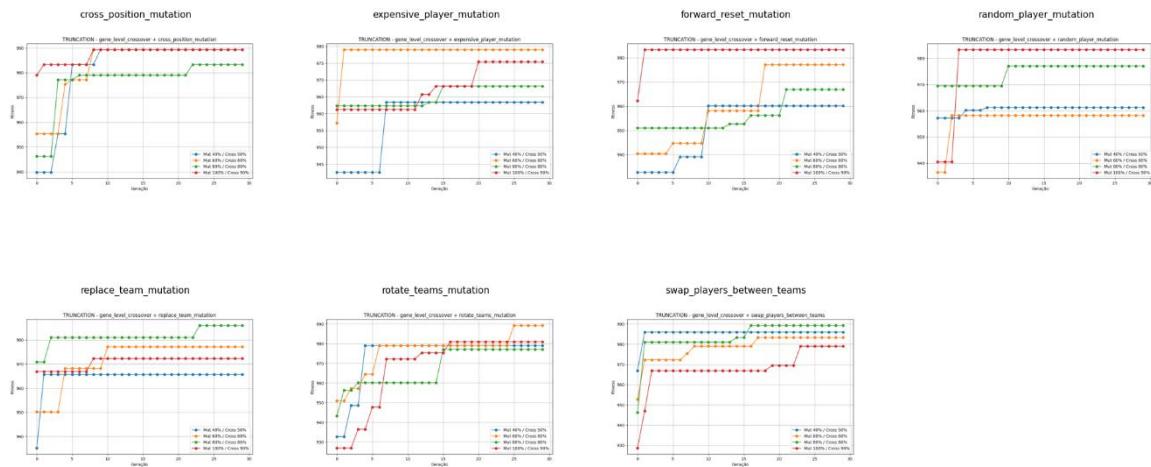


Figure 28–Truncation Selection: Gene Level Crossover Combinations

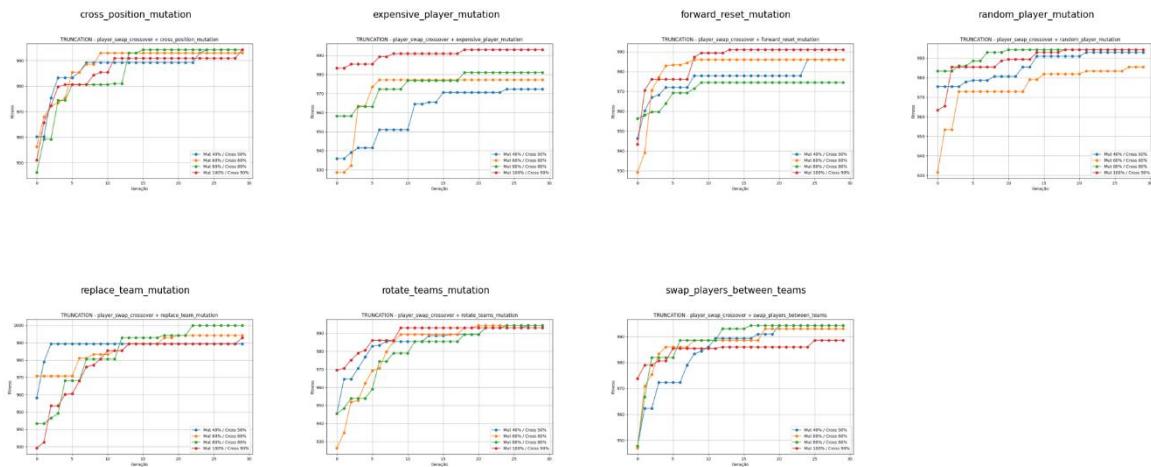


Figure 29–Truncation Selection: Player Swap Crossover Combinations

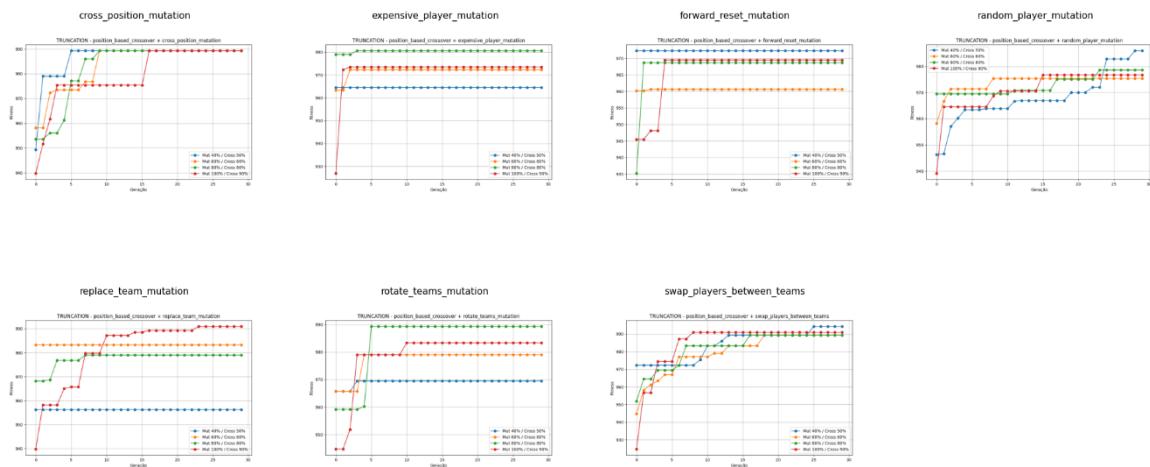


Figure 30– Truncation Selection: Position Based Crossover Combinations

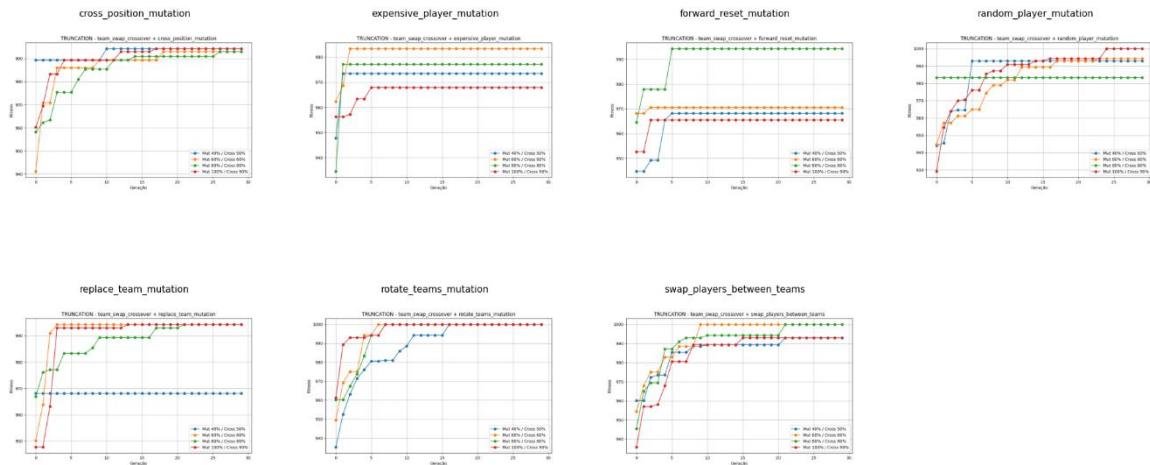


Figure 31– Truncation Selection: Team Swap Crossover Combinations

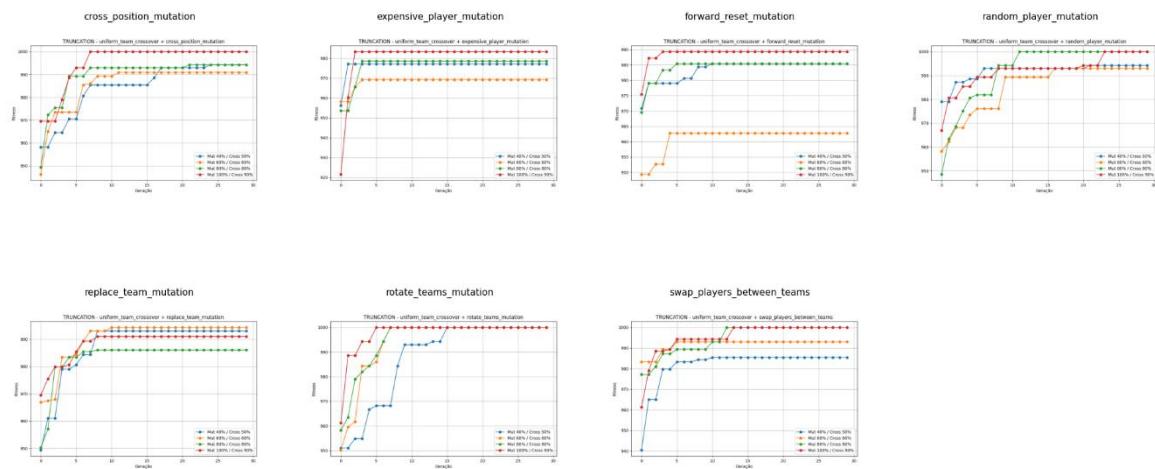


Figure 32– Truncation Selection: Uniform Team Crossover Combinations

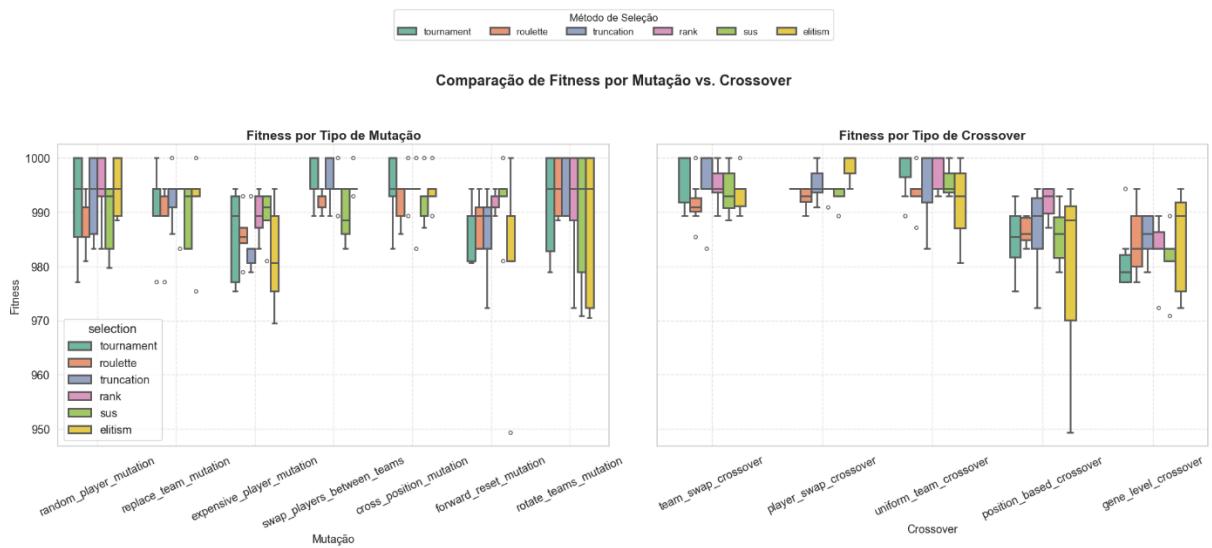


Figure 33– Crossover vs Mutation: Fitness Comparation

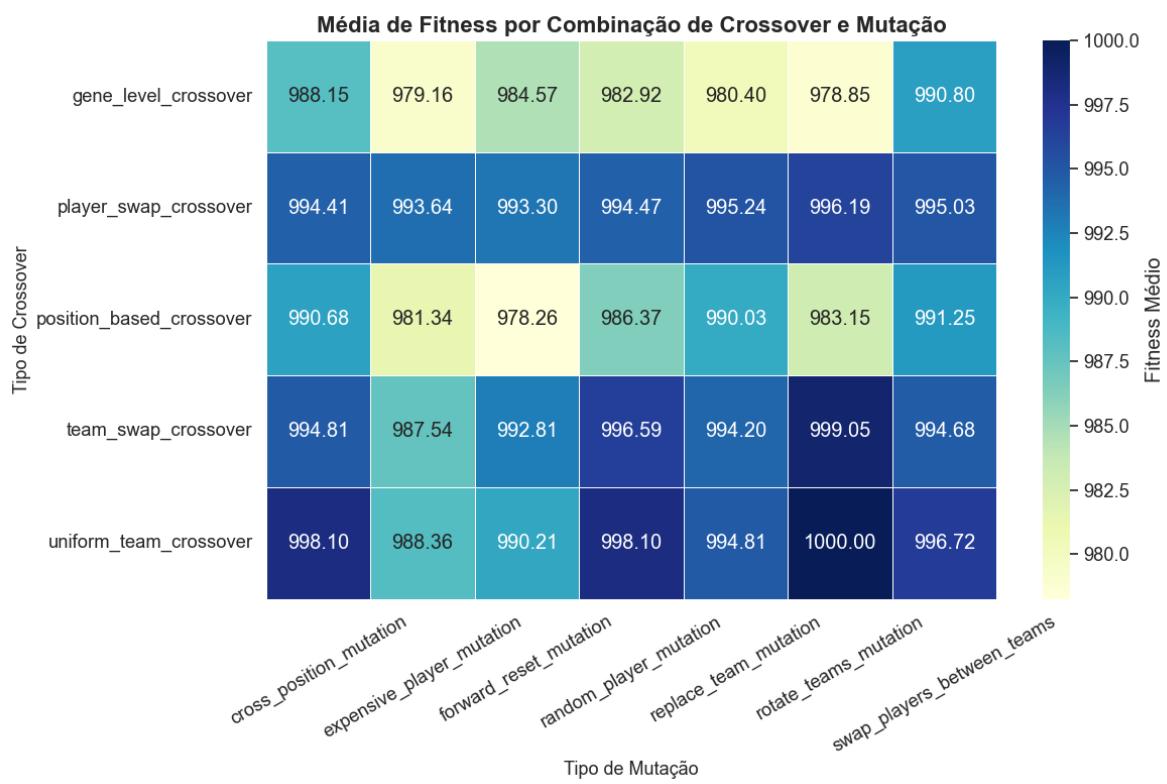


Figure 34 - Mutation vs Crossover Fitness HeatMap

Relações entre Fitness e Taxas Genéticas

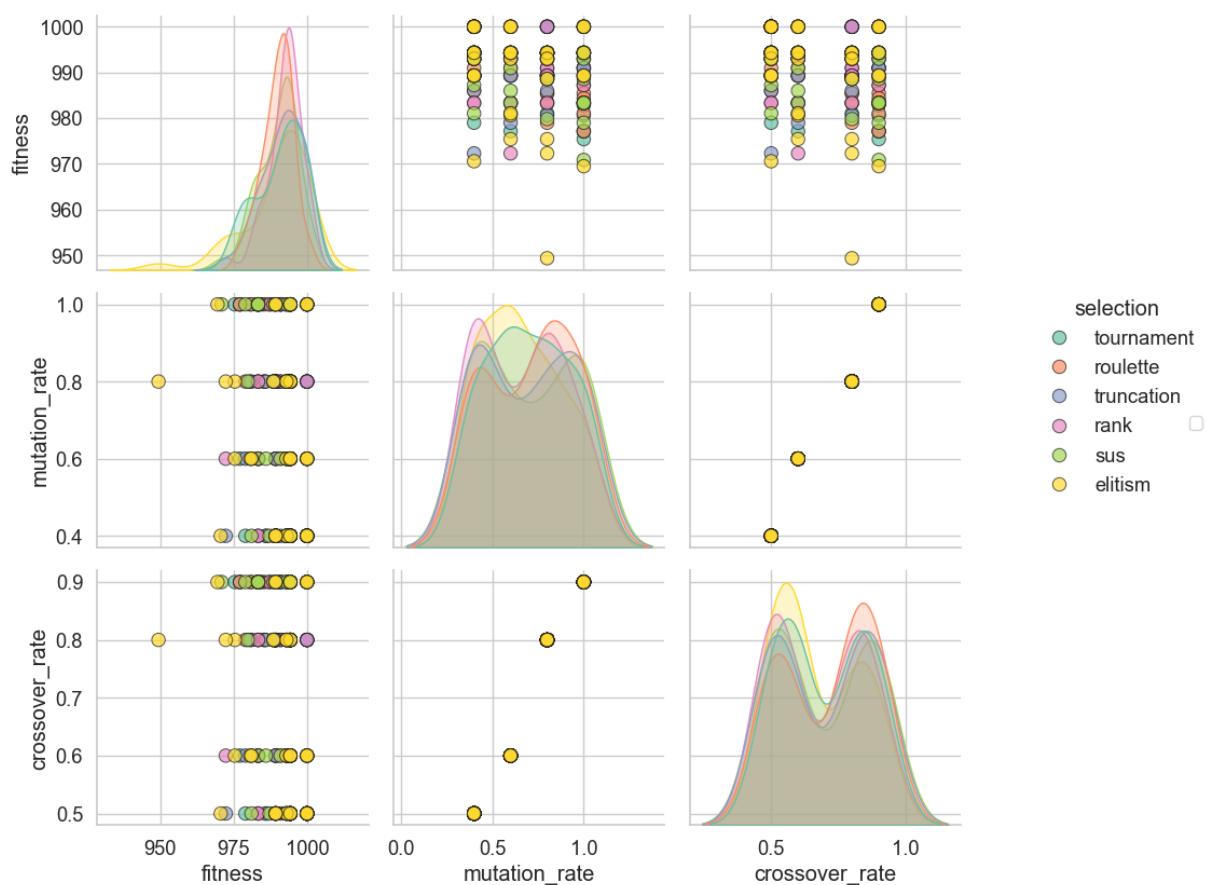


Figure 35 - Genetic Rate Fitness Pairplot

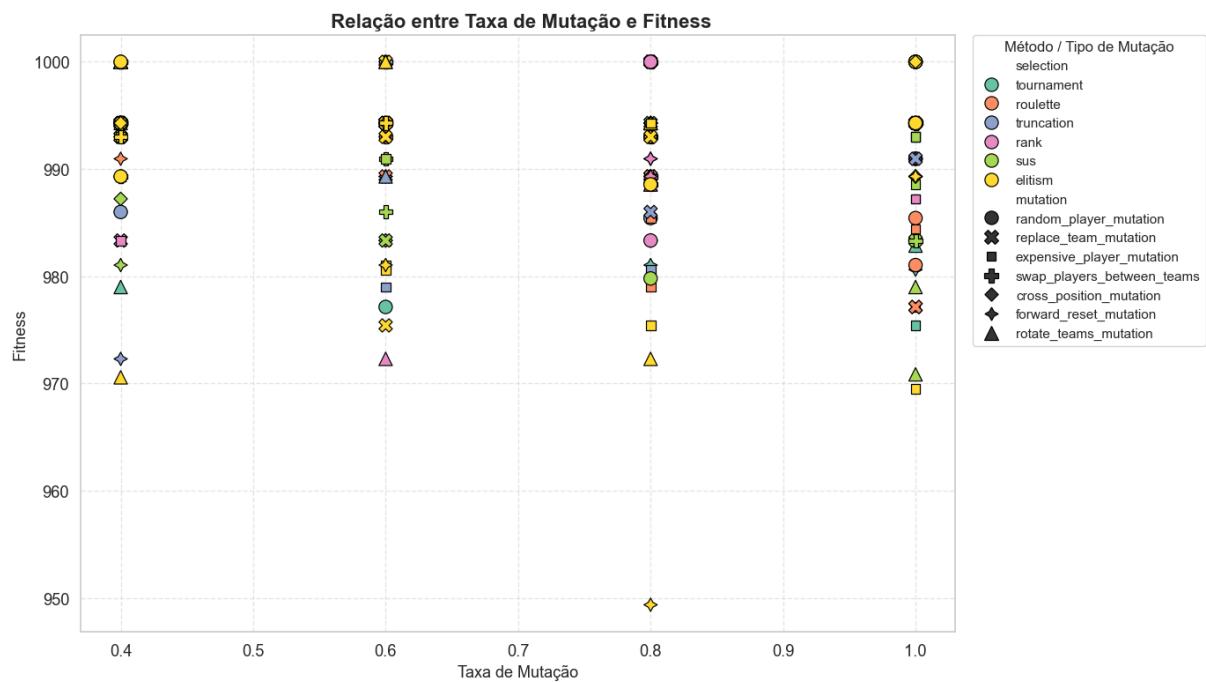


Figure 36 - Mutation Rate vs Fitness Scatter

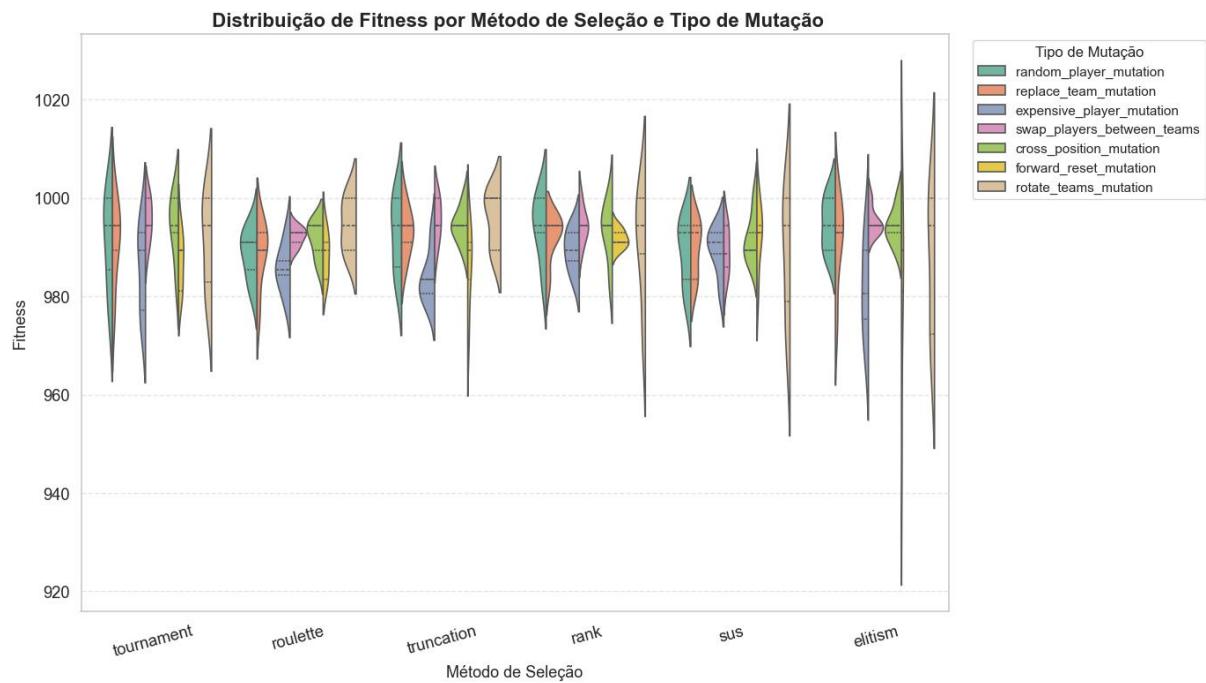


Figure 37 - Selection vs Mutation Fitness ViolinPlot



NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa