Universiteit Leiden
The Netherlands

Opleiding Informatica

AI Methods in Checkers Board Variations

Jasper Kooij

Supervisors:
Matthias Müller-Brockhausen & Mike Preuss

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

14/08/2025

Contents

Abstract

This study examines how AI methods can be used to analyze variations in Checkers board configurations. An experimental setup is successfully developed, enabling exhaustive testing across multiple board setups using a range of AI agents. The outcomes of these experiments are captured and visualized through tables and graphs. These visualizations provide meaningful insights into the underlying game mechanics and agent performance across different scenarios.

A key finding is the validation of a hypothesis in which deliberate asymmetries in the initial board configuration result in observable imbalances in gameplay outcomes. Furthermore, in cases where the advantageous position is not immediately apparent, the experiments still produce consistent performance advantages for one player.

These findings highlight the potential of using AI-based methods as a tool in the process of developing and testing video games, especially in the early stages of development.

1 Introduction

Video game development is a complex process spanning across a variety of areas of expertise, such as design, art, programming and testing. Before a great game can hit the shelves, multiple iterations of designing, implementing and testing will have had to have happened. One specific aspect of this game development process is testing. Testing is done to ensure the concept is fun to play, gameplay is smooth and bugs are identified in time. Traditionally, testing is only done by human players, often as part of the development team, sometimes as part of a program for interested gamers outside of the team. It is an expensive and time-consuming process.

In recent time, artificial intelligence (AI) has become more and more integrated in almost every field, also in the field of game production [1][2]. AI methods can aid developers in some aspects of game testing, reducing cost and time in the overall game development process.

In particular, classic AI methods allow developers to simulate and play hundreds of games in a short time, something that would take human players much longer. Specifically, using a Monte Carlo approach will be a suitable solution in the early stages of game development, as that method does not necessarily require knowledge of the game's inner workings. This type of AI play-testing is a strong tool for developers to use in combination with regular human play-testing.

The game of Checkers is a well-known board game, and therefore also widely researched [3]. When introducing changes to the game rules or the board setup, that research might no longer be accurate.
This thesis aims to contribute to the field of game development by finding out how AI methods used in testing perform when introduced to changes to the game of Checkers.

Specifically, this research seeks to find the answer to the following research questions:

1. To what extent can AI-vs-AI gameplay outcomes reveal game mechanics or attributes across different board configurations?

2. How can intentional modifications to board configurations create new and interesting gameplay scenarios, while preserving game balance?

3. Which gameplay metrics are most useful for assessing balance in different board configurations?

To answer the research questions, we do an empirical analysis of novel initial board configurations and let a variety of AI methods play on them to judge their fairness [4]. The thesis uses the board game Checkers as a starting point, making implementing and modifying an easy part of the research. This allows for the creation of unique game states and variations of the base game, which are then tested using certain AI methods, like Monte Carlo Tree Search [5]. These AI methods are described in Section 4.2.

2 Related Work

The use of artificial intelligence in the field of game development has gained traction in recent years. This section discusses related work for the topic of this thesis, with a focus on the use of AI in game testing and the use of AI methods in Checkers.

The use of AI in Checkers has been studied in Gill [6]. In this study, AI has been used to intelligently determine optimal parameters in a board state evaluation function. In their approach, AI agents are evaluated using evolutionary strategies, which simulate matches between bots and iteratively improve their evaluation functions. This study shows the effectiveness of automated parameter tuning using AI, creating a system that is capable of intelligently playing checkers by using the optimized evaluation function.

Another relevant study is Goodman et al. [7], where AI agents are used to test successive design iterations of a newly designed analogue board game. In this study, the AI agents play through iterations of the game to help the designer in identifying issues and evaluating balance. Notably, the authors emphasize that the goal is not to fully automate game balancing, but to provide actionable insights to the designer, who remains in control of the design process. The study successfully demonstrates how AI testing can complement traditional human-based play-testing.

This thesis draws inspiration from both studies. Similar to Gill, AI methods are used to evaluate performance in the game of Checkers, but instead of optimizing evaluation functions, this research focuses on how well different AI models perform in modified versions of the game.
Similar to Goodman et al., AI is also used as a support tool for developers, giving insights in game mechanics as they change because of changes in the game.

The distinction between this thesis and previous work lies in the experimental focus. While Goodman et al. study the use of AI in the design of a new game, this research expands on the well-known rules of Checkers and studies the effect of controlled variations on the base game.

The `TAG` framework [8] is used to carry out the experiments. It provides an environment for testing AI methods in custom games, making it well-suited for the research done in this thesis. See Section 3.1 for a more detailed explanation of the framework and the incorporation of it in this thesis.

3 Background

To fully understand the context of this thesis, this section introduces the tools and concepts used in the research. This includes an overview of the TabletopGames (`TAG`) framework and the base game of Checkers.

3.1 Tabletop Games framework

`TAG`: A Tabletop Games Framework[8] is a Java-based framework designed for research in game AI in modern tabletop games [1]. It offers an environment where developers can implement custom games, AI players and analytical metrics or use existing components to carry out experiments.

`TAG` includes readily available AI methods such as Monte Carlo Tree Search (`MCTS`) and simple heuristics, making it a powerful tool for researchers. Developers can configure custom agents and test setups using the framework's configuration options, which are well documented.

An important component of `TAG` is the `RunGames` program, which allows for extensive testing of AI agents. It can be configured to run many games automatically while collecting various game metrics, such as win/loss ratio, game duration and more. For this thesis, a custom metrics class called `CheckersMetrics` is used to gather information specific to the Checkers game. The specifics of metrics used in this thesis are explained further in Section 4.3.1.

3.2 Checkers

The game of Checkers is a well-known turn-based strategy game played between two players on an 8×8 board. Each player begins with twelve regular pieces and attempts to win by following a fixed set of rules. A player wins by either removing all pieces of the opponent from the board, or by making the opponent unable to move.

King pieces can be obtained by moving a regular piece onto the top row of the board. King pieces can move in all directions and are able to skip over any number of squares, in contrast to regular Man pieces. Figure 1b in the next section shows the standard 8×8 Checkers layout.

Checkers is well-suited for this research for several reasons. First, the rules are simple and deterministic, making it easy to simulate and analyze. Second, the board game allows for lots of freedom in modifying attributes, such as board size, movement rules, and piece count or positions. Variations like these are used in this thesis to study AI agent performance, and to answer the research questions presented in Section 1.

By implementing Checkers within the `TAG` framework, this research creates a controlled environment for evaluating AI-based testing methods. It enables researchers to observe AI agent behavior in classic Checkers configurations, as well as modified configurations.

[1] See `https://gaigresearch.github.io/projects/TAG`

4 Methods

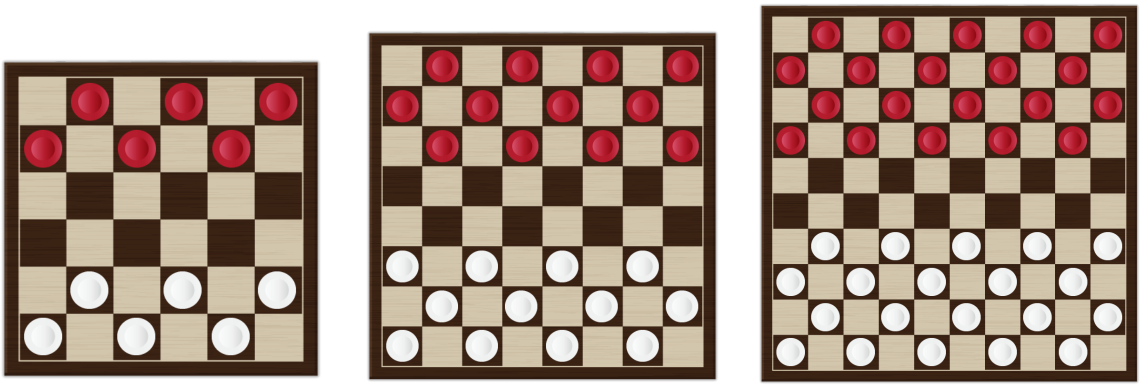
The implementation for this project is developed in Java, using the `TAG` framework. The game of Checkers is implemented following the official rules of Checkers according to the World Checkers Draughts Federation [9]. Board configurations are specified in external files, enabling modification of the initial setup by editing the file content.
The AI methods used in this thesis are pre-existing components in the `TAG` framework, so implementation of these components is not necessary.
Graph and plot generation is performed in Python using the Matplotlib library [10], along with the SciencePlots style package [11], to ensure clear visualizations.

4.1 Board configurations

To investigate the behavior of AI methods, experiments were carried out on different board setups. The aim for these setups is to firstly learn how to correctly implement and test AI methods, and secondly to understand the effects of changing the playing field on AI gameplay outcomes.

Research Question 1. To what extent can AI-vs-AI gameplay outcomes reveal game mechanics or attributes across different board configurations?

In the case of Checkers, playing field variations include variations in the size and shape of the board, and variations in the number and positions of pieces on the board.



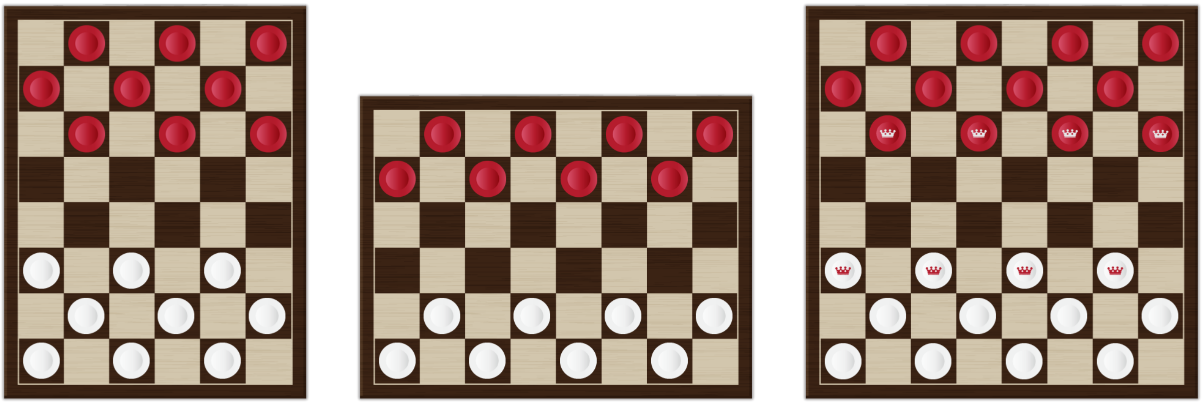
(a) 6×6 setup (b) 8×8 setup (c) 10×10 setup

Figure 1: Standard board layouts for Checkers on 6×6, 8×8 and 10×10 boards. The 8×8 board corresponds to the official Checkers layout defined by the World Checkers Draughts Federation[9]. The 10×10 board is a larger variant played in different parts of the world, including the Dutch game "Dammen", where each player begins with 20 pieces.

4.1.1 Symmetric setups

To establish a baseline for the experiments conducted in this study, the standard board layout for Checkers is first implemented. This standard configuration represents the setup found in the official rules[9] and is shown in Figure 1b. The board consists of an 8-by-8 grid with alternating dark and light squares, beginning with a dark square in the lower-left corner. Each player is assigned 12 pieces of their respective color. These pieces are arranged on the dark squares in the first three rows closest to the player, leaving the two center rows unoccupied. Additional board configurations of varying sizes are created, trying to keep close to the principles of official Checkers rules. For example, a 6-by-6 board shown in Figure 1a includes 6 pieces per player, while a 10-by-10 board shown in Figure 1c includes 20 pieces per player. In all variants, at least two central rows are kept empty.

Three additional variations of the standard Checkers layout are examined in this study. Two of these configurations deviate from the conventional square board format and instead utilize rectangular dimensions. Despite this change in shape, the initial game conditions remain symmetrical, as both players are presented with identical board states. The 6-by-8 and 8-by-6 rectangular configurations are illustrated in Figures 2a and 2b. The third configuration deviates from the conventional board setup by introducing King pieces at the start of the game. As mentioned in Section 3.2, King pieces are stronger variations of regular pieces on the Checkers board. Figure 2c shows that symmetry is maintained by assigning identical King placements to both players.



(a) 6×8 setup (b) 8×6 setup (c) 8×8 setup with King rows

Figure 2: Symmetrical variations on the standard board configuration.

4.1.2 Asymmetric unbalanced setups

The next experiments involve a setup using asymmetric board configurations. The goal of this part of the study is to investigate how different initial conditions, based on unequal starting positions for each player, influence AI agent performance and gameplay outcomes.

In this context, asymmetry refers to scenarios where each player begins the game in different situations, not necessarily in an unfair manner. Asymmetry is common in many well-balanced games. For example, in Catan, players start with differently placed roads and settlements, and in Ticket to Ride, players receive unique route objectives through randomly drawn cards.
These games demonstrate that asymmetry can coexist with game balance, with players having an unequal playing field, but fair chance of winning.

In Checkers, where the objective is well-defined, it is simple to introduce unbalanced starting conditions. For instance, reducing the number of starting pieces for one player introduces a disadvantage. These deliberately unbalanced setups are used in this part of the thesis to evaluate AI performance and potentially infer game mechanics.
The underlying hypothesis is that visualized results will provide insight into the extent to which such asymmetries influence gameplay outcomes and agent performance.

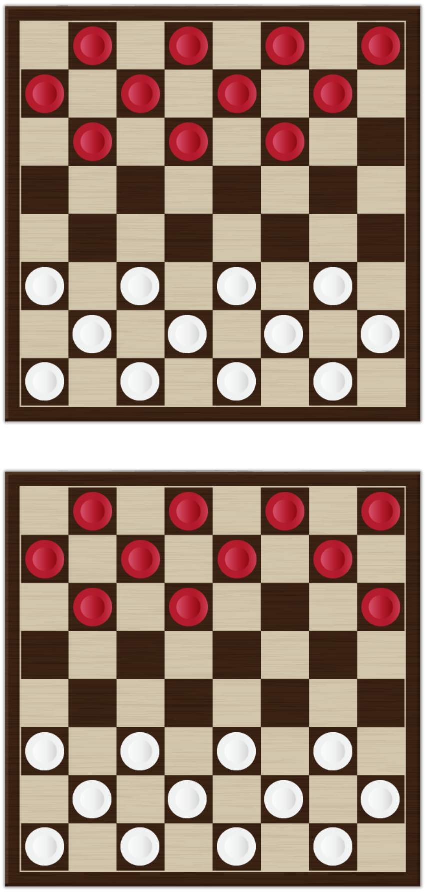


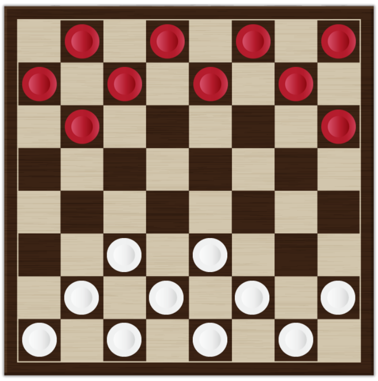
Figure 3: Asymmetrical unbalanced board configurations. The players have an unequal number of pieces, with one player missing a piece in the center row.

4.1.3 Asymmetric balanced setups

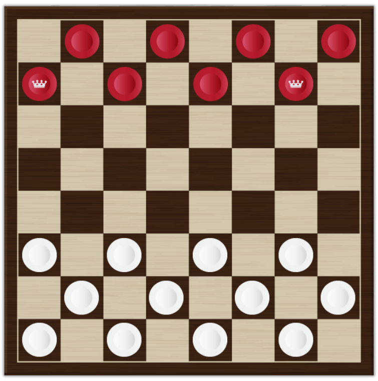
In the final part of the study, previously examined asymmetric and unbalanced board configurations are revisited with the objective of introducing balance through secondary variations. For instance, a player starting with fewer pieces may receive a King piece, creating a more balanced scenario for analysis.

The objective of this section is to see how AI agent performance metrics can be used to evaluate the results on asymmetrical boards, and how this information can be used to create new and interesting configurations.

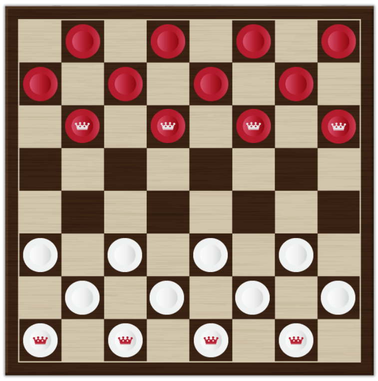
Research Question 2. How can intentional modifications to board configurations create new and interesting gameplay scenarios, while preserving game balance?



(a) Both players have an equal number of pieces, but spacial arrangement differs



(b) One player compensates for a missing row of pieces with a row of King pieces



(c) Both players have an equal number of pieces, but placement of King pieces differ

Figure 4: Asymmetrical yet nominally balanced board configurations. Mirror configurations, used to account for the fixed starting player (Red), are not shown here but are included in the experimental evaluation.

In the first set of nominally balanced asymmetric board configurations, the total number of pieces per player is kept constant, while their positions differ. Specifically, two pieces from the center rows are removed from each player. For one player, the two central pieces are removed, whereas for the other player, the two outermost pieces are excluded. This configuration is illustrated in Figure 4a. The mirror configuration, where the Red player misses their outermost pieces and the White player misses their center pieces, is not shown here in Figure 4 but is included in the experiments.

The second set of configurations as seen in Figure 4b introduces the use of King pieces as compensation for unequal piece counts. In this setup, one player begins with the standard three full rows of regular pieces, while the other player starts with one row fewer but is compensated with two King pieces substituting regular pieces in the second row.

A third set of configurations keeps equality in both the total number of pieces and the number of King pieces assigned to each player. Each player receives a full row of King pieces; however, asymmetry is introduced through their placement. One player's Kings are positioned on the back row, while the other's occupy the central row. As shown in Figure 4c, the King pieces are visualized with a crown.

4.2 AI agents

To allow for autonomous gameplay, this study uses several AI methods, capable of playing Checkers against one another. These AI methods are referred to as AI player or AI agents.

4.2.1 `Random` player

To establish a baseline level of play with no strategic reasoning, a `Random` agent is used. This agent selects available actions at random, having no understanding of game strategy. Its role in the experiments is to serve as a reference point for more advanced agents. It represents a completely uninformed player.

4.2.2 `OSLA` player

To represent a player with basic strategic knowledge, a simple greedy algorithm known as the One-Step Look-Ahead (`OSLA`) agent is implemented. This agent simulates each available action in a copy game state, and evaluates the resulting state using a heuristic function. The action resulting in the highest heuristic value is then executed in the original game state. The custom heuristics function used is described in Section 4.2.4.

The `OSLA` algorithm is called "greedy", because it selects the action that appears best in the immediate moment, without considering longer-term consequences. While this can lead to suboptimal decisions over the course of a game, this AI method is computationally efficient and can be quite effective when paired with a strong heuristic function.

4.2.3 MCTS player

To simulate a more advanced level of play, the Monte Carlo Tree Search (MCTS) method is used. Unlike OSLA, MCTS considers future game states by constructing a tree of game states and performing simulations (rollouts) from leaf nodes to terminal states. Based on the outcome of these simulations, values are backpropogated through the tree in order to select an action in the current state.

In this thesis, the standard implementation of MCTS provided by the TAG framework is used, with heuristic guidance as described in the following section. The internal mechanics of the MCTS algorithm are not covered in detail here. For explanations and variations, see Browne et al. [5].

Heuristic	Weight distribution
Number of pieces	0.5
Capture actions	0.3
Available actions	0.2

Table 1: Weight distrubution of heuristic evaluation

4.2.4 Heuristic function

A key component of greedy AI methods is the heuristic evaluation function, which serves as a guideline or rule of thumb for decision-making. The heuristic function evaluates the current game state and computes a value that reflects how favorable the position is for the current player. This value is then used by the algorithm to determine which action to take.

In order for the OSLA player to operate effectively, and for the MCTS player to perform well, a heuristic function must be implemented. Implementing such a function requires an understanding of the specific game being played. In the case of Checkers, general strategies are already well-established [3]. All implemented strategies within the heuristic function must be appropriately weighted to ensure optimal performance, which introduces a balancing problem that must be addressed.

In this project, three heuristics were used to estimate the heuristic value of a given game state. These heuristics were selected based on the researcher's intuitive assessment of their relevance to gameplay and were found to be sufficient for the objectives of this thesis. The selection involved a brief process of trial-and-error, during which the greedy algorithm was manually examined over the course of ten games against a Random agent. The algorithm's decision-making was analyzed on a step-by-step basis, ultimately finding a weight distribution for the heuristics that effectively guided the agent's actions.

The first heuristic is based on the number of pieces on the board. Specifically, it considers the ratio between the number of pieces belonging to the current player and those of the opponent. A higher ratio results in a higher heuristic value. Individual heuristic values are clamped between -1.0 and 1.0 before calculating the final heuristic value, which is done by averaging the three values following the weight distribution shown in Table 1.

The second heuristic evaluates the presence of potential capture actions. If the player can execute a capture action in the next turn, the state receives a score of $+1.0$. Conversely, if the opponent can execute a capture action in their next turn, the state is penalized with a score of -1.0. In cases where both players have captures available, the player's own captures are weighted at $+0.2$, while opponent's captures remain at -1.0. This is because it is the opponent's turn in the evaluated state and they will have the first oppertunity to execute their action.

The third heuristic is based on the number of actions available to the player, referred to as action size. A larger action size indicates a greater freedom and control over the progression of the game. More available actions than the opponent result in a higher heuristic value, at $+0.1$ per extra move, and vice versa.

In the field of hyperparameter optimization, various automated techniques have been developed to determine optimal parameter values for heuristic evaluation functions [12]. An example of such an approach is provided by Gill et al.[6], who focus specifically on the game of Checkers. In their paper, they present a novel approach to computer-based checkers play by employing a coevolutionary structure, specifically designed to intelligently determine the optimal parameters in a board state evaluation function. The result of the study included a weight distribution for the heuristic function, with high values for piece count and king pieces, as well as nine other heuristics specific for piece positions.

These results correspond to the heuristics used in this thesis, thereby affirming their validity.

4.3 Data collection and visualization

To facilitate testing AI methods on the Checkers variants, `TAG` includes a helpful program that is used to run games and collect data effectively.

4.3.1 Tournaments and game metrics

The `RunGames` program provides a framework for automated tournament-style testing of AI agents. In this research, it is utilized to perform exhaustive testing across all agents pairings and Checkers board configurations. The provided `GameMetrics` class is used to record data during experiments.

> **Research Question 3.** Which gameplay metrics are most useful for assessing balance in different board configurations?

In addition to the general metrics collection system provided by the `TAG` framework, this thesis introduces custom game-specific metrics tailored to the game of Checkers. The data collected during each experiment is divided into two categories: per-turn metrics and per-game metrics, as shown in Table 2. Per-turn metrics record the specific action taken in each turn, the number of pieces remaining on the board, and the number of legal moves available to each player. Per-game metrics track the outcome of each game (win, loss, or draw) and the total number of turns taken to reach that outcome.

Category	Metric
Per turn	Action type (move / capture)
	Number of pieces
	Number of available actions
Per game	Win/draw/loss outcome
	Total game duration

Table 2: Overview of game metrics collected during experiments. Per-turn metrics are recorded for each turn, while per-game metrics summarize the overall outcome.

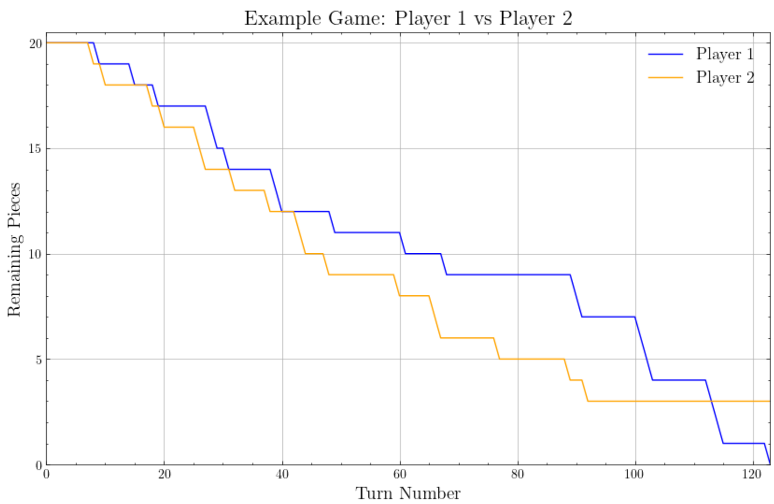


Figure 5: Example of a per-turn piece-count plot, showing the evolution of the board state over time. This type of graph is used to visualize the progression of individual games.

4.3.2 Graphs generation

Data collected during the experiments is used to generate various graphs and plots. To start, results from 200 games played between two AI agents in a specific board configuration are analyzed. The primary metrics visualized include the percentage of games won, lost or drawn by each agent and the average game duration. This is done for every combination of agents and board configurations, and these summary statistics provide a quick comparison of agent performance within each setup.

To examine the progress of individual games, a second type of plot is generated. In these graphs, the x-axis represents the number of turns, while the y-axis indicates the number of remaining pieces for each player. A drop in a player's line corresponds to a capture action by the opponent.

To visualize aggregated performance across multiple games, the x-axis is normalized to represent the percentage of game completion rather than absolute turn counts. Since game lengths vary, this normalization allows data from games of different durations to be aligned and averaged. A confidence interval is included in these graphs to indicate the variability of the data.

5 Results

The experiments in this thesis were conducted by evaluating each board layout described in Section 4.1 across all player combinations defined in Section 4.2. For every board configuration, each of the nine AI player matchups were executed 200 times, resulting in 1800 games per configuration.

Game metrics were collected using the procedure described in Section 4.3. During execution, a dedicated CSV file was generated for each board configuration, recording per-turn data for all matches played. These files served as the data source for the plots and tables presented in this thesis.

In total, 16 board configurations were examined, resulting in a total of 28.800 games played. All relevant board configurations, visualizations and tables can be found in Appendices A and B.

It should be noted that the relatively low sample sizes in the results introduce a greater margin of error, sometimes preventing definitive conclusions to be made. This limitation is recurrent throughout this thesis. Although a large number of games were played, subdividing 200 games per matchup into three outcome categories occasionally resulted in small counts. Further discussion on this issue is provided in Section 6.1.

5.1 Symmetrical board setups

The experiments began with the standard 8×8 Checkers configuration, as shown in Figure 1b of Section 4.1. This setup served as a baseline for evaluating AI performance on variations of the Checkers board game. In this configuration, the `MCTS` and `OSLA` agents were put up against the `Random` agent to provide an initial comparison. Figure 6 displays the outcomes of these two matchups: `MCTS` versus `Random`, and `OSLA` versus `Random`. In these plots, and in all subsequent plots in this thesis, the `Random` agent is represented in green, `MCTS` in red, and `OSLA` in blue. The trajectory of the green line, moving almost linearly from the upper left to the lower right, indicates a steady reduction in the number of pieces controlled by the `Random` agent throughout the game. Conversely, the `MCTS` and `OSLA` agents maintained a relative stable number of remaining pieces, displaying their dominance in the matchup.
These visualizations suggest that both `MCTS` and `OSLA` consistently outperform the `Random` agent.

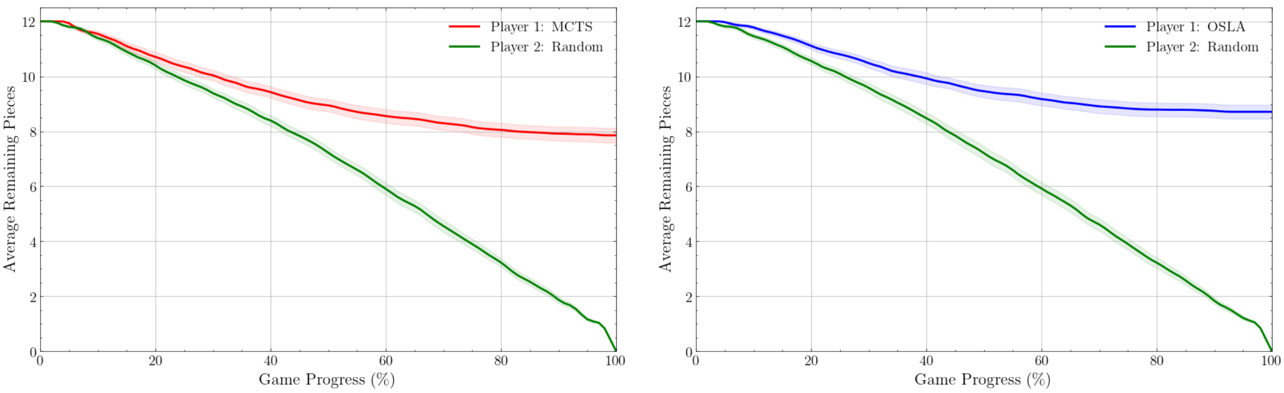


Figure 6: Performance comparison on the standard 8×8 board configuration. Plots show `MCTS` vs `Random` and `OSLA` vs `Random`.

Self-play on the standard board configuration produces important results regarding the evaluation of balance. These are the matchups where both players are controlled by the same AI agent. `MCTS` versus `MCTS`, `OSLA` versus `OSLA` and `Random` versus `Random`. An important conclusion can be made, namely that self-play produces even results for both players, showing balance in fairness for the standard configurations. Figure 7 displays `MCTS` and `OSLA` results, showing even outcomes.

In addition to the graphical visualizations, the outcomes of each agent matchup are also summarized in a table format. Table 3 presents the win, loss and draw distributions for all pairwise combinations of the three AI agents across the three standard Checkers board configurations. These distributions provide an overview of perfomance dynamics between agent pair, which allows for direct comparison of outcomes across different matchups.

The AI agents were pitted against one another and the resulting outcomes were captured in plots. Figure 8 illustrates the matchup between `MCTS` and `OSLA` on the standard 8×8 board configuration. Once again, `MCTS` is represented by the red line, and `OSLA` by the blue line. During the first 60% of the game, both agents maintained comparable piece counts. However, after this point, `MCTS` maintained a higher average piece count, resulting in most wins for `MCTS` in this matchup.

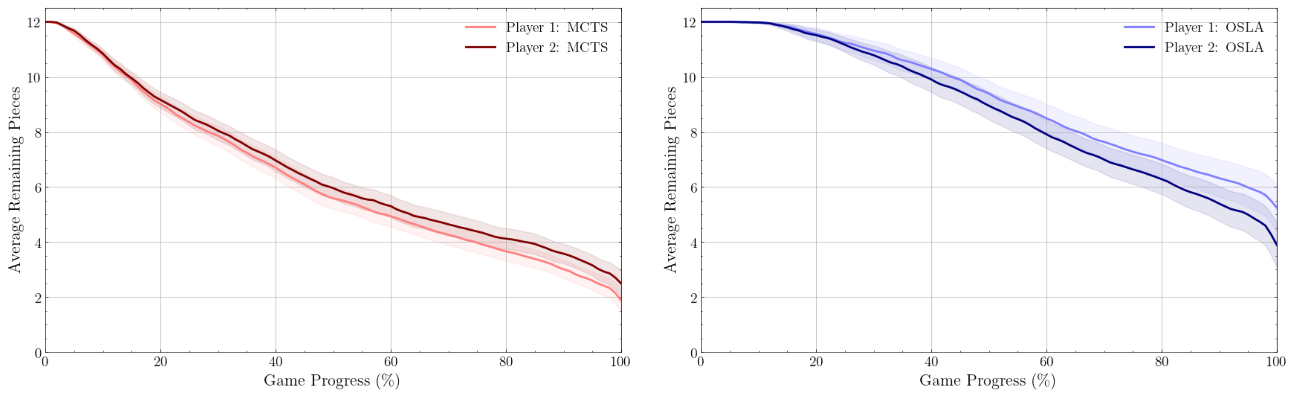


Figure 7: Performance comparison on the standard 8×8 board configuration. Plots show self-play results of `MCTS` and `OSLA`.

second		MCTS			OSLA			Random		
first		6×6	8×8	10×10	6×6	8×8	10×10	6×6	8×8	10×10
MCTS	win	53	74	65	181	130	93	200	200	200
	loss	59	70	68	1	34	70	0	0	0
	draw	88	56	67	18	36	37	0	0	0
OSLA	win	1	15	74	46	85	88	193	200	200
	loss	180	169	87	134	86	68	7	0	0
	draw	19	16	39	20	29	44	0	0	0
Random	win	0	0	0	4	0	0	95	103	97
	loss	200	200	200	196	200	199	105	97	103
	draw	0	0	0	0	0	1	0	0	0

Table 3: Game outcomes of standard 6×6, 8×8 and 10×10 board configurations

The results for this specific matchup are summarized in Table 5, highlighted with thick borders. Tables 4, 5 and 6 present statistics for all games played on standard Checkers configurations of sizes 6×6, 8×8 and 10×10. These include game outcomes, average game duration and average remaining piece counts. Several observations can be made from this data.

Firstly, across all board configurations, the `Random` agent was consistently defeated by both the `MCTS` and `OSLA` agents, losing nearly every game. This outcome aligns with expectations, as the `Random` agent lacks any strategic decision-making capabilites.

Another notable observation is that games ending in a draw tend to have significantly longer durations. This can be attributed to the nature of Checkers, where a draw may be forced if both players avoid capture actions. When both agents play defensively, the number of captures decreases, often triggering the draw condition after 40 consecutive moves without a capture.

Additionally, the distribution of wins, losses and draws between `MCTS` and `OSLA` varies across the board configurations. On the 6×6 board, `MCTS` achieved a higher number of wins against `OSLA` compared to the 8×8 and 10×10 configurations. Game durations also differed across board sizes. Matches on the 6×6 board ended significantly faster than those on the 10×10 board.
This is expected, as larger boards contain more pieces, which increases the time required to capture or immobilize all opponent pieces to reach the end conditions.

Further analysis focuses on the experiments conducted on the 6×6 board configuration, the smallest among those tested in this study. In this setup, each player begins with only six pieces, which is half the amount used in the standard 8×8 board and substantially fewer than the 20 pieces per player on the 10×10 board.

This reduced number of pieces results in several notable patterns. Most obviously, the average number of moves per game is significantly lower compared to the larger board configurations. More interestingly, the results displayed in Table 4 reveal that the second player got more wins than the first player when `OSLA` played against itself. In other words, being the starting player appears to reduce the chance of winning on the 6×6 board, but

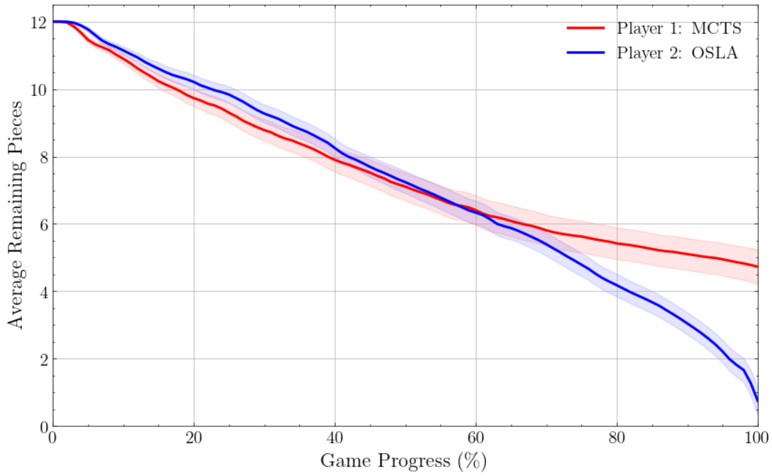


Figure 8: MCTS vs OSLA on a standard 8×8 board.

Standard 6×6

first \ second		MCTS			OSLA			Random		
		games	duration	pieces	games	duration	pieces	games	duration	pieces
MCTS	win	53	31.38	2.6	181	28.81	2.98	200	20.7	3.9
	loss	59	37.34	2.73	1	25.0	4.0	0	–	–
	draw	88	74.22	3.28	18	70.22	3.72	0	–	–
OSLA	win	1	69.0	2.0	46	30.96	4.7	193	23.36	4.42
	loss	180	27.6	3.09	134	24.92	4.95	7	22.71	3.57
	draw	19	74.53	3.68	20	72.75	5.0	0	–	–
Random	win	0	–	–	4	21.75	4.5	95	27.14	2.85
	loss	200	19.73	4.25	196	23.16	4.52	105	24.5	3.27
	draw	0	–	–	0	–	–	0	–	–

Table 4: Summary statistics for games played on a 6×6 board setup.

only in case of OSLA self-play. This asymmetry in outcomes is not observed in different agents self-play results, suggesting that the heuristic evaluation function used by OSLA may contribute to this effect.

In addition to the standard square boards, two rectangular symmetrical configurations were also tested: the 6×8 and 8×6 boards. For these experiments, focus is placed on games in which each AI agent played against itself. Table 7 presents the game outcomes for these self-play matches. In this table, a win corresponds to the first starting player defeating the second, while a loss indicates the second player winning over the first.

Across all three compact symmetrical configurations, a substantial part of matches resulted in a draw. When a game did not end in a draw, outcomes generally favored the second player. The only exception to this occurred in games played by the Random agent on the 6×8 board, where the first player won more than the second. The largest disparity is seen in the OSLA self-play results on the 6×6 board, where the second player won 134 games compared to just 46 wins by the first player.

Standard 8×8

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	74	68.15	4.26	130	70.17	4.39	200	43.07	7.76
	loss	70	65.96	4.33	34	59.03	7.76	0	–	–
	draw	56	134.18	3.46	36	113.64	5.73	0	–	–
OSLA	win	15	64.33	6.47	85	54.88	9.41	200	42.08	8.72
	loss	169	61.8	5.4	86	54.42	9.43	0	–	–
	draw	16	127.25	4.5	29	107.9	10.03	0	–	–
Random	win	0	–	–	0	–	–	103	54.54	4.81
	loss	200	43.69	7.61	200	42.98	8.7	97	53.02	5.26
	draw	0	–	–	0	–	–	0	–	–

Table 5: Summary statistics for games played on an 8×8 board setup. Results from the matchup shown in Figure 8 are highlighted.

Standard 10×10

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	65	124.94	5.57	93	134.0	4.61	200	85.45	10.73
	loss	68	121.13	6.0	70	97.29	10.96	0	–	–
	draw	67	192.42	3.88	37	179.0	6.0	0	–	–
OSLA	win	74	94.54	11.12	88	100.23	13.78	200	74.98	13.38
	loss	87	134.55	4.49	68	107.09	12.66	0	–	–
	draw	39	172.82	6.54	44	147.73	16.95	0	–	–
Random	win	0	–	–	0	–	–	97	99.64	7.24
	loss	200	85.61	10.88	199	74.25	13.6	103	96.95	7.78
	draw	0	–	–	1	125	12	0	–	–

Table 6: Summary statistics for games played on a 10×10 board setup.

Small board configurations

AI agent self-play		6×6 (6 pieces) games	6×6 (6 pieces) duration	6×6 (6 pieces) pieces	8×6 (8 pieces) games	8×6 (8 pieces) duration	8×6 (8 pieces) pieces	6×8 (9 pieces) games	6×8 (9 pieces) duration	6×8 (9 pieces) pieces
MCTS	win	53	31.38	2.6	61	43.84	3.21	69	55.99	3.43
	loss	59	37.34	2.73	70	46.59	3.56	73	52.53	3.6
	draw	88	74.22	3.28	69	96.58	3.54	58	98.95	3.64
OSLA	win	46	30.96	4.7	57	37.88	6.47	64	36.31	7.8
	loss	134	24.92	4.95	110	34.59	6.86	93	37.45	7.59
	draw	20	72.75	5.0	33	86.45	6.42	43	94.0	6.67
Random	win	95	27.14	2.85	86	34.37	3.77	111	41.73	3.79
	loss	105	24.5	3.27	114	32.75	3.94	89	39.42	4.51
	draw	0	–	–	0	–	–	0	–	–

Table 7: Summary statistics for games played on small board configurations, showing AI agents self-play.

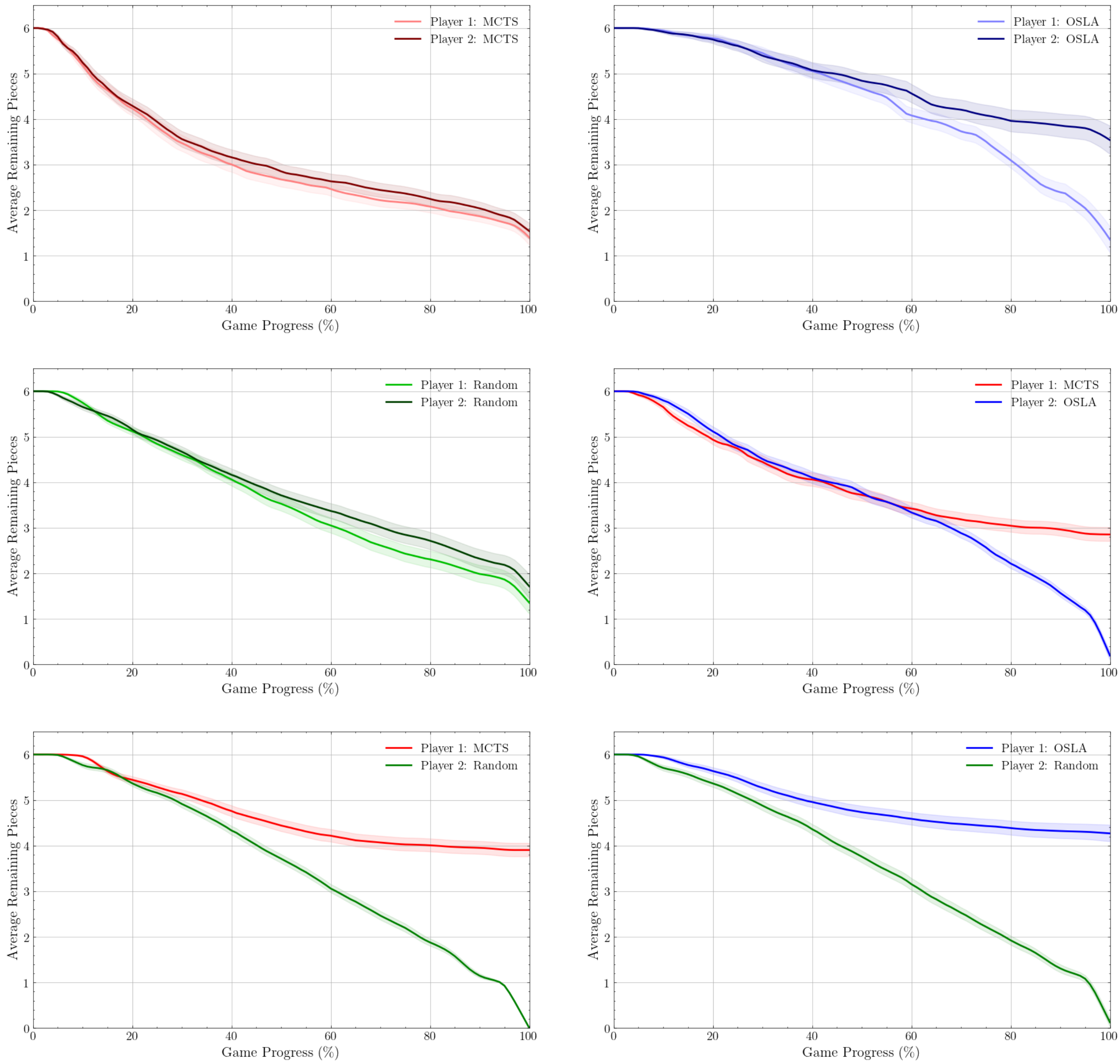
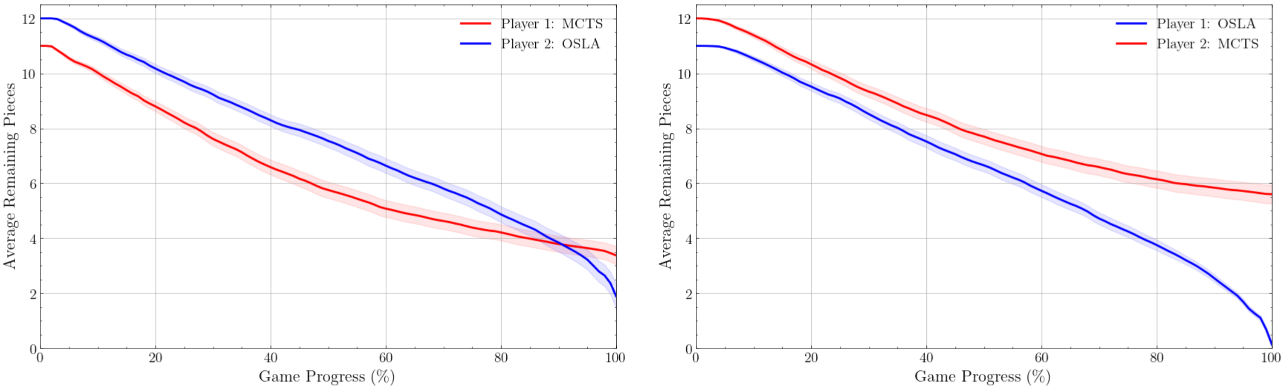


Figure 9: Average remaining piece count per player across normalized game progress on a 6×6 board. Each subfigure corresponds to a different agent pairing, as labeled.

5.2 Asymmetrical unbalanced board setups

This section presents the results from experiments conducted on asymmetrical board configurations with deliberate imbalance. These unbalanced setups, as described in Section 4.1.2, were designed to give one player a clear disadvantage, for instance by reducing their starting piece count. Figure 10 illustrates matchups between `MCTS` and `OSLA` on one such asymmetrical board configuration, specifically the second variant shown in Figure 3. The only asymmetry in this setup is that the first player begins with 11 pieces, while the second player starts with 12.



(a) `MCTS` vs `OSLA` (b) `OSLA` vs `MCTS`

Figure 10: Matchups between `MCTS` and `OSLA` on an asymmetrical, unbalanced board configuration. (a) `MCTS` starts; (b) `OSLA` starts.

Figure 10a displays results from games in which `MCTS` begins first. Despite starting with a piece disadvantage, `MCTS` managed to overtake `OSLA` as the game progresses and won a majority of the matches. In contrast, Figure 10b shows `MCTS` starting second and achieving an even higher win count. These outcomes are quantified in Table 8, where `MCTS` wins 138 games as first player and 190 as the second player out of 200 games per matchup.

These finding clearly reflect the asymmetry of the initial configuration, and the performance difference between the agents. `MCTS` performed better than `OSLA` , being able to overcome the one-piece deficit. This performance difference was also evident in matchups against the `Random` agent, where both `MCTS` and `OSLA` consistently won despite starting with fewer pieces.

An additional observation is that `MCTS` produced a higher number of draws compared to `OSLA` when playing against itself. This pattern is consistent across board configurations.

These results affirm the designed imbalance of the test configurations. The ability of `MCTS` to outperform `OSLA` under disadvantageous conditions invites further experimentation. In particular, identifying the threshold at which such asymmetries lead to balanced probabilities is a promising direction. The next section addresses this line of investigation by evaluating nominally balanced yet asymmetrical configurations.

Asymmetrical unbalanced

	second	`MCTS`			`OSLA`		
first		games	duration	pieces	games	duration	pieces
`MCTS`	win	36	68.72	4.19	138	68.15	4.43
	loss	121	60.99	5.38	29	57.34	8.14
	draw	43	129.44	3.84	33	112.36	6.27
`OSLA`	win	2	67.5	5.5	58	51.84	9.72
	loss	190	58.48	5.78	117	49.85	9.96
	draw	8	125.75	4.75	25	107.16	8.16

Table 8: Summary statistics for `MCTS` vs. `OSLA` on an asymmetrical, unbalanced board setup. `Random` player matchups are omitted.

5.3 Asymmetrical balanced board setups

In the final phase of the experimental analysis, asymmetrical board configurations are examined that have been intentionally designed to be more balanced. Here, ”balanced” is to be interpreted loosely, as it is not evident before testing whether the configurations are truly balanced. Rather, the term indicates that an effort was made to counteract the asymmetry by introducing compensatory elements. The goal is not to achieve perfect balance, but rather to explore whether the experimental results can reflect the balance present in new board configurations, where any advantage might not be immediately obvious.

First variation: piece positioning

The first variation investigates whether the initial positions of an equal number of pieces affects gameplay outcomes. Figure 11 display such a setup, where both players begin with ten pieces. The asymmetry in this configuration arises from the spacial distribution. One player's pieces are placed in the central columns of the board, while the other player's pieces occupy the outer columns.

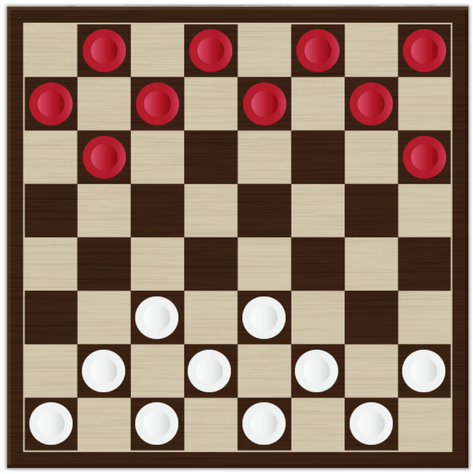


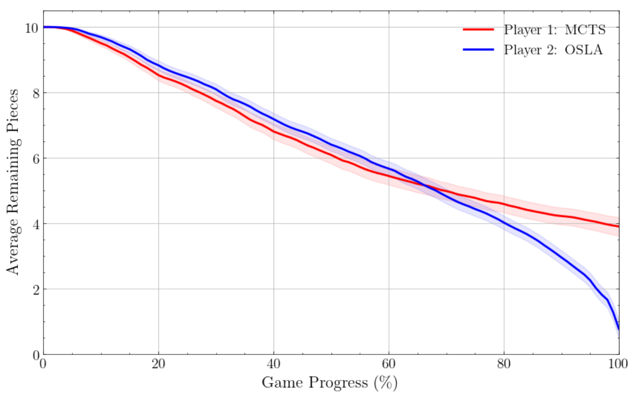
Figure 11: First variation of an asymmetric, nominally balanced setup.

This configuration is intended to evaluate whether a centrally concentrated or edge-oriented piece distribution is more advantageous, or if the starting positions give similar results.

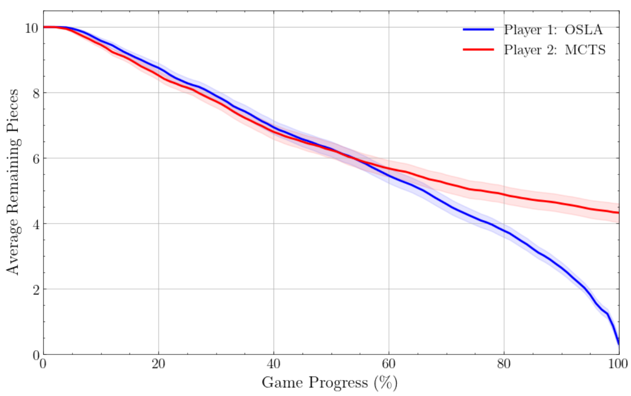
The figures below present the experimental results. Figure 12 shows the matchup between `MCTS` and `OSLA`. In Figure 12a, `MCTS` plays first with centrally aligned pieces, while in Figure 12b, `OSLA` takes the first move with the central arrangement. Figure 13 displays self-play results, where `MCTS` and `OSLA` play against themselves.

Several observations were drawn from these results. First, `MCTS` consistently outperformed `OSLA` regardless of the side from which it started. This outcome was expected, given that `MCTS` was the stronger of the two agents and the board configuration was designed to be balanced. The consistency of `MCTS`'s performance suggested that the initial piece positioning had a limited influence on the outcome when there was a significant difference between the players.

In contrast, the self-play matchups revealed subtle effects of piece positioning. When `MCTS` played against itself, performance remained nearly identical regardless of which player started. However, in `OSLA` self-play matchups, a more noticable difference is present between the first and second starting players. This suggests that `OSLA` was more sensitive to initial board asymmetries or to first-move advantages.



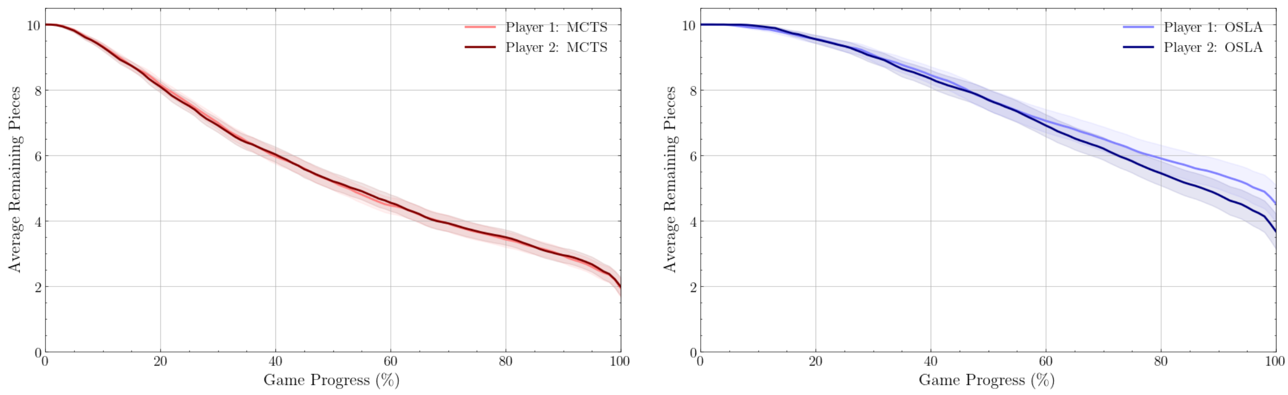
(a) `MCTS` vs `OSLA`



(b) `OSLA` vs `MCTS`

Figure 12: Matchups between `MCTS` and `OSLA` on the asymmetrical board configuration shown in Figure 11. (a) `MCTS` starts; (b) `OSLA` starts.

It seemed that the starting `OSLA` player won more often than the second `OSLA` player. When examining the board configuration, the first player – shown with red pieces – started with the two center pieces missing. One



(a) MCTS vs MCTS

(b) OSLA vs OSLA

Figure 13: Self-play results of MCTS and OSLA on the asymmetrical configuration shown in Figure 11.

possible conclusion from this result is that beginning with pieces closer to the edge of the board, rather than the center, may provide a slight early-game advantage.

Of course, it must be acknowledged that OSLA is a greedy algorithm and does not necessarily reflect the behavior of human players or more advanced strategies. It is possible that the observed advantage in the OSLA versus OSLA matchup is party a result of the agent's greedy nature. The absence of such effects in MCTS matchups further undermines the validity of any advantage attributed to piece positioning.

Asymmetrical balanced setup 1

second		MCTS			OSLA		
first		games	duration	pieces	games	duration	pieces
MCTS	win	71	64.97	3.96	169	61.7	4.4
	loss	77	67.78	3.86	11	55.27	7.04
	draw	52	125.17	4.02	20	111.85	5.7
OSLA	win	4	53.0	6.25	85	50.92	8.58
	loss	180	61.47	4.63	70	54.47	8.11
	draw	16	115.12	4.38	45	107.38	7.64

Table 9: Summary statistics for MCTS vs. OSLA on the asymmetrical, nominally balanced board setup, shown in Figure 11. Random player matchups are omitted.

As shown in Table 9, the results indicated that in the self-played games of MCTS, the first and second player were evenly matched. A slight difference in performance of MCTS was observed in matchups against the OSLA agent, when comparing the starting player configurations. Specifically, with OSLA as the starting player, MCTS won 180 games and drew 16, leaving 4 wins for OSLA , which started with missing center pieces. This is indicated by the red pieces in the board image. Conversely, when MCTS was the starting player, it won 169 games and drew 20, with OSLA winning 11. These findings suggest that starting with missing pieces in the center of the board may be less advantageous than starting as second player while possessing center pieces but missing edge pieces.
Regarding the OSLA self-play matchup, 85 wins were recorded for the starting player, compared to 70 wins for the second player, with 45 draws. This implies that only OSLA appeared to experience an advantage when starting with the red pieces.

However, as mentioned in the introduction of Section 5, the low sample size might skew the results. Therefore, the above-mentioned conclusion cannot be stated with certainty.
To address potential asymmetry related to the starting player, the experiment was repeated using a mirrored board configuration. The results generally exhibited similar ratios. However, the limited number of games contributed to some variance in the data.

Asymmetrical balanced setup 1 – mirror

first \ second		MCTS			OSLA		
		games	duration	pieces	games	duration	pieces
MCTS	win	76	61.67	4.21	175	57.79	4.86
	loss	82	63.04	4.33	6	63.17	6.5
	draw	42	123.67	3.67	19	104.26	6.32
OSLA	win	12	49.17	7.42	63	59.78	7.38
	loss	172	61.1	4.76	89	53.19	7.9
	draw	16	122.12	4.56	48	109.6	8.19

Table 10: Summary statistics for the mirror version of the asymmetrical configuration shown in Figure 11. See Figure 21 in Appendix A for the exact setup.

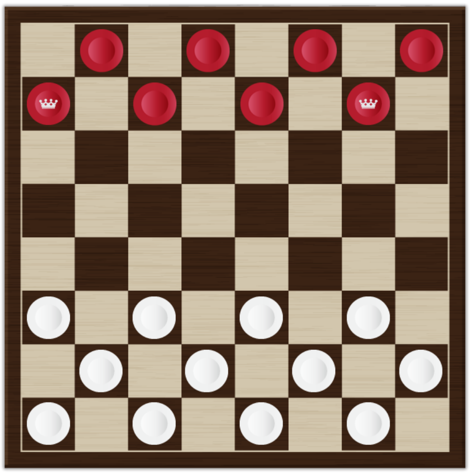


Figure 14: Second variation of an asymmetric, nominally balanced setup.

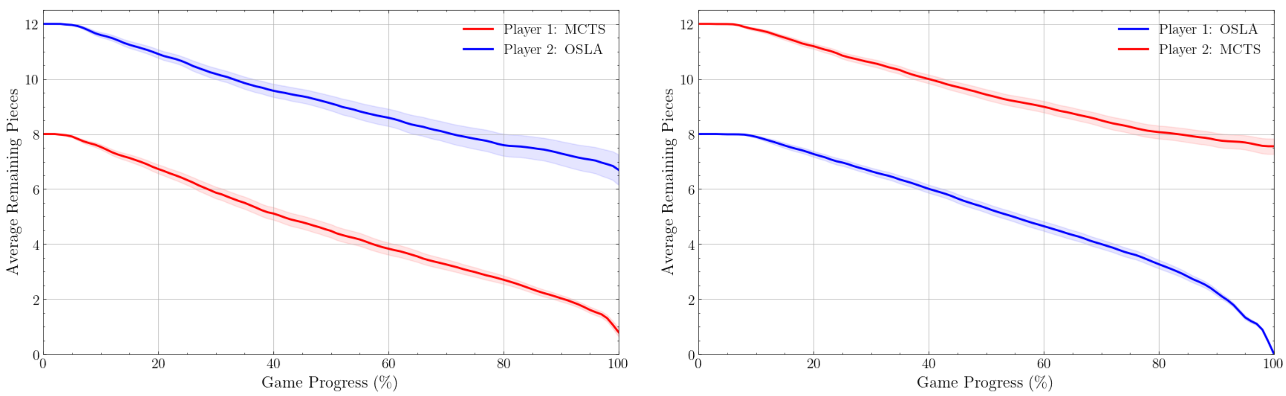
Second variation: piece count and King pieces

The next analysis examined the experimental results on a different asymmetric board configuration. Here, King pieces were introduced at the start of the game. This variation was included to explore an unconventional scenario, as standard Checkers games do not begin with King pieces. Such a setup presents an interesting variation of typical gameplay, where conventional Checkers strategies might be less applicable.

Figure 15 presents the performance plots of MCTS versus OSLA on the board configuration depicted in Figure 14. In this setup, one player starts with all 12 pieces, while the opponent starts with only 8 pieces, including 2 King pieces.

The plots in Figure 15 clearly illustrate the imbalance present in this board configuration. In these setups, the player who started with fewer pieces was compensated with two King pieces. However, the experimental results showed that this compensation was insufficient to offset the initial disadvantage.

In previous experiments involving more balanced configurations, MCTS consistently outperformed OSLA, even when starting with a slight disadvantage. In contrast, the current experiment showed MCTS losing most of the matches against OSLA. When MCTS was assigned the disadvantaged starting position, it won only 38 out of 200 games and drew 41. OSLA, starting with a full set of twelve regular pieces, won the remaining 121 games. This outcome demonstrates an advantage for the player starting with a higher piece count.



(a) MCTS vs OSLA

(b) OSLA vs MCTS

Figure 15: Matchups between MCTS and OSLA on the asymmetrical board configuration shown in Figure 14. (a) MCTS starts; (b) OSLA starts

Asymmetrical balanced setup 2

first \ second	MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces
MCTS win	2	74.0	2.5	38	76.16	2.45
MCTS loss	192	45.49	7.47	121	44.4	9.31
MCTS draw	6	123.67	4.33	41	106.73	6.78
OSLA win	0	–	–	11	78.55	4.91
OSLA loss	198	45.63	7.61	170	44.52	10.38
OSLA draw	2	117.0	3.5	19	119.68	8.68

Table 11: Summary statistics for MCTS vs. OSLA on the board setup shown in Figure 14. Random player matchups are omitted.

These findings suggest that the balancing attempt of crowning two pieces Kings was ineffective. The advantage provided by the King pieces did not compensate for the reduced number of total pieces.

This result highlights the kind of insight developers could obtain through experimentation. Before testing, it was unclear whether the strategic value of two King pieces could outweigh the reduced number of pieces.

Third variation: King piece position

The final configuration, illustrated in Figure 16, introduced variation solely through the positioning of King pieces on the board. Both players started with an equal number of pieces, including a full row of King pieces. The asymmetry in this setup comes from the location of the King row. One player's King pieces were placed in the center row, while the other player's King pieces were positioned on the back row. Appendix 22 provides a view of these configurations, including a mirrored version with the King piece positions swapped.

This setup was of particular interest due to its potential implications for early-game versus late-game strategy. One player had immediate access to their King pieces from the start of the match, while the other needed to clear space before their King pieces could be used in play.

To assess whether either player was in an advantageous position, this experiment focused on self-play matchups. The outcomes of these matchups are summarized in Table 12. The board shown in Figure 16 is labeled "Original", and the mirrored version is also present in the table.

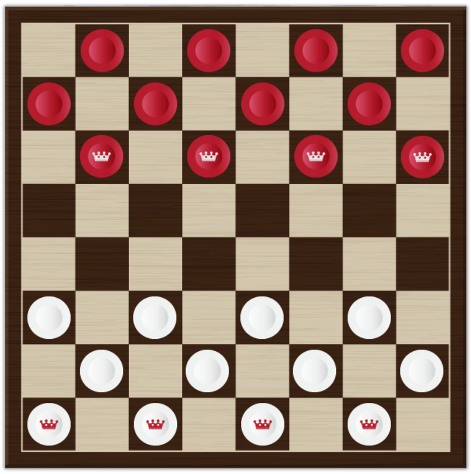


Figure 16: Third variation of an asymmetric, nominally balanced setup.

The results of the self-play experiments on this configuration indicated that both MCTS and OSLA agents were relatively evenly matched. A notable number of games resulted in draws for both AI agents, suggesting that neither the first nor second player consistently dominated under this setup.

A particularly interesting observation can be made when comparing the original configuration to its mirrored counterpart. The initial hypothesis assumed that any observed performance difference in the original configuration would be reversed in the mirrored version. However, this was not the case in the experimental outcomes. For the MCTS self-play experiments, the red player, with King pieces in the center row, recorded 68 wins and 79 losses in the original setup. In the mirrored version, where the red player had King pieces in the back row, the outcomes shifted to 58 wins and 86 losses. In both cases, the white, second player appeared to perform better. This stands in contrast to the standard 8×8 configuration, where no significant first-player disadvantage was observed in MCTS self-play.

The OSLA agent produced different trends. In the original setup, the red player with the center row of King pieces won 93 games and lost 75. In the mirrored configuration, with King pieces in the back row, the red player won 85 games and lost 79.

Third variation of asymmetric, nominally balanced configuration

AI agent self-play		Original games	duration	pieces	Mirror games	duration	pieces
`MCTS`	win	68	72.01	4.35	58	69.62	4.31
	loss	79	65.7	4.61	86	68.35	4.4
	draw	53	122.91	3.72	56	126.18	3.88
`OSLA`	win	93	50.04	10.12	85	51.31	9.95
	loss	75	54.0	9.37	79	54.87	9.62
	draw	32	115.47	8.47	36	104.58	9.36
`Random`	win	112	54.12	5.15	109	55.49	4.67
	loss	87	54.49	4.97	91	50.81	5.53
	draw	1	110.0	2.0	0	–	–

Table 12: Summary statistics for AI self-play of `MCTS` and `OSLA` on the configuration from Figure 16, including both the original and mirror layout.

Unlike `MCTS`, `OSLA` showed a consistent advantage for the starting player in both configurations. Similarly, the `Random` agent also favored the first player in both configurations.

These observations suggest the possibility that the apparent imbalances in outcomes could be due to random variation, given the low number of games per configuration. Alternatively, they may indicate lower level interactions between agent behavior and board layout that call for further investigation. In any case, the positioning of King pieces appeared to have a smaller influence on game outcomes than the effect of the starting player in this set of experiments.

6 Discussion & Conclusion

6.1 Limitations

This study has several limitations. First, the `MCTS` agent used was used directly from the `TAG` framework, without any modifications or tuning. Even though it performed better than the `OSLA` agent, it likely does not represent optimal play and should not be compared to a skilled human player. That's why conclusions drawn from AI versus AI gameplay in this thesis do not necessarily reflect outcomes in human versus human games. In practice, human player testing is still required. This is in line with the conclusion of the study by Goodman et al. [7], where it's stated that "AI game testing cannot remove the need for human playtesting."

The experimental evaluation consisted of simulating 200 games per player pair, per board configuration. In total, that resulted in 28.800 games played. Although a large number of games were played, subdividing 200 games per matchup into three outcome categories occasionally resulted in small counts. The relatively low sample sizes in the results introduce a greater margin of error, thereby preventing definitive confirmation of hypotheses.

This research involved a limited selection of board configurations. A more thorough experiment, using more variations and iterations, could provide even more insight. Even so, most of the configurations used in this study were useful examples and sufficient for the aim of the thesis.

6.2 Future work

The findings in this thesis point in several directions for future research. Expanding the experiments to increase the number of simulations and to include a wider variety of board configurations could give more insights, possibly revealing more nuanced game mechanics or strategies. Combining features from multiple configurations may lead to new game mechanics worth investigating.

Expanding on the visualizations is another direction for future work. While the graphs and tables in this thesis are informative, a more advanced presentation could make experimental results easier to interpret. In the future, a dashboard could be developed to display automated test results in a concise manner. Different kinds of visualizations could presented, enriching the insights in various game mechanics.

6.3 Conclusion

This study shows that AI-driven gameplay analysis can reveal meaningful insights into game balance and mechanics. Variations in board configuration lead to different gameplay patterns, often attributing an advantage to one player over the other. By using multiple AI agents with varying performance, researchers can compare different levels of play. This is accomplished by exhaustively running hundreds of games across all agent pairings.

Several board configurations were tested, grouped into three categories: one set of symmetric configurations, one set of deliberately unbalanced asymmetric configurations, and one set of nominally balanced asymmetric configurations. In the last category, balance was attempted by introducing secondary variations, to counteract initial asymmetries.

To answer the research questions presented in Section 1, AI-vs-AI gameplay outcomes reveal the balance present in different board configurations, showing which player might have a more advantageous position from the start of the game. New configurations can be tested in this way to analyze the balance, and figure out whether or not the scenario might be fair for both players. Most of the metrics used to gather data in this thesis were useful for analyzing balance. For example, game result (win/loss/draw) and piece count during gameplay showed to be the most useful metrics. Game duration indicated how close the games were. The remaining piece count at the end of the game did not give a clear indication on any game mechanic in particular.

To highlight one of the findings in this study: the configuration shown in Figure 14 shows an imbalance. In this setup, the developer introduced a secondary variation to an asymmetric board, attempting to compensate the disadvantage caused by a reduced number of pieces by assigning two King pieces to the disadvantaged player. However, the data in Table 11 and visualization in Figure 15 showed that this compensation is not sufficient to level the playing field.

A logical next step for a game developer would be to increase the number of King pieces and test the configuration again to assess whether this change improved the overall balance of the asymmetric board setup.

Although this thesis did not end up producing a balanced game, these results confirm that this AI-based approach does give insights into game balance, which support developers in making more balanced game variations.

References

[1] Mark Owen Riedl and Alexander Zook. "AI for game production". In: *2013 IEEE Conference on Computational Inteligence in Games (CIG)*. ISSN: 2325-4289. Aug. 2013, pp. 1–8. DOI: 10.1109/CIG.2013.6633663. URL: https://ieeexplore.ieee.org/abstract/document/6633663 (visited on 08/13/2025).

[2] Tommy Thompson. "The Changing Landscape of AI for Game Development". In: *Game AI Uncovered*. Num Pages: 11. CRC Press, 2024. ISBN: 978-1-003-32410-2.

[3] Jonathan Schaeffer et al. "Checkers Is Solved". In: *Science* 317.5844 (2007), pp. 1518–1522. DOI: 10.1126/science.1144079. eprint: https://www.science.org/doi/pdf/10.1126/science.1144079. URL: https://www.science.org/doi/abs/10.1126/science.1144079.

[4] Jesse Schell. "Balance Type #1: Fairness". In: *The Art of Game Design: A Book of Lenses*. 3rd ed. In section *Game Mechanics Must Be in Balance*. Taylor Francis Group Limited, 2019. Chap. 13, pp. 212–217. ISBN: 9781138632059.

[5] Cameron B. Browne et al. "A Survey of Monte Carlo Tree Search Methods". In: *IEEE Transactions on Computational Intelligence and AI in Games* 4.1 (Mar. 2012), pp. 1–43. ISSN: 1943-0698. DOI: 10.1109/TCIAIG.2012.2186810. URL: https://ieeexplore.ieee.org/abstract/document/6145622 (visited on 07/07/2025).

[6] Barnaby Gill. "Machine Learning in Checkers: Using Coevolution to Drive Game Strategy". en. In: (2023). Publisher: Unpublished. DOI: 10.13140/RG.2.2.23824.17928. URL: https://rgdoi.net/10.13140/RG.2.2.23824.17928 (visited on 06/23/2025).

[7] James Goodman et al. "A case study in AI-assisted board game design". en. In: *2023 IEEE Conference on Games (CoG)*. Boston, MA, USA: IEEE, Aug. 2023, pp. 1–4. ISBN: 979-8-3503-2277-4. DOI: 10.1109/CoG57401.2023.10333138. URL: https://ieeexplore.ieee.org/document/10333138/ (visited on 05/12/2025).

[8] Raluca D. Gaina et al. "TAG: A Tabletop Games Framework". In: *Experimental AI in Games (EXAG), AIIDE 2020 Workshop*. 2020.

[9] World Checkers Draughts Federation. *Rules of Draughts (Checkers)*. URL: https://wcdf.net/rules/rules_of_checkers_english.pdf.

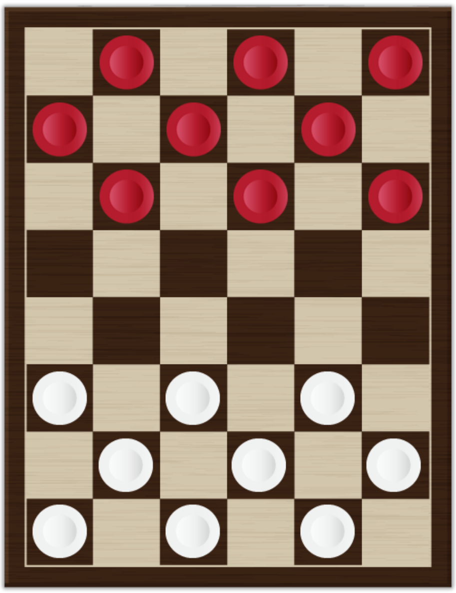
[10] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[11] John D. Garrett. "garrettj403/SciencePlots". Version 1.0.9. In: (Sept. 2021). DOI: 10.5281/zenodo.4106649. URL: http://doi.org/10.5281/zenodo.4106649.

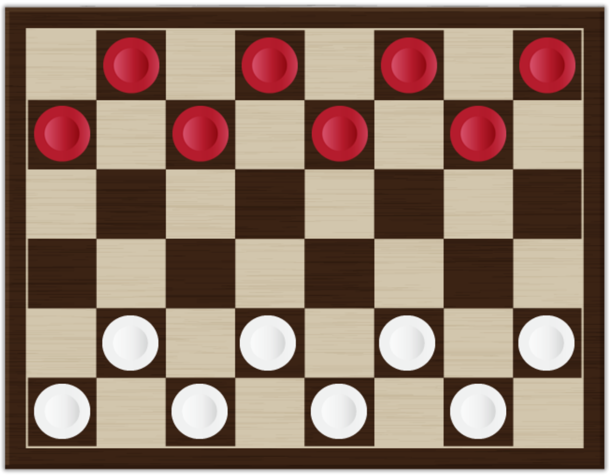
[12] Matthias Feurer and Frank Hutter. "Hyperparameter Optimization". In: *Automated Machine Learning: Methods, Systems, Challenges*. Ed. by Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. Cham: Springer International Publishing, 2019, pp. 3–33. ISBN: 978-3-030-05318-5. DOI: 10.1007/978-3-030-05318-5_1. URL: https://doi.org/10.1007/978-3-030-05318-5_1.

Appendices

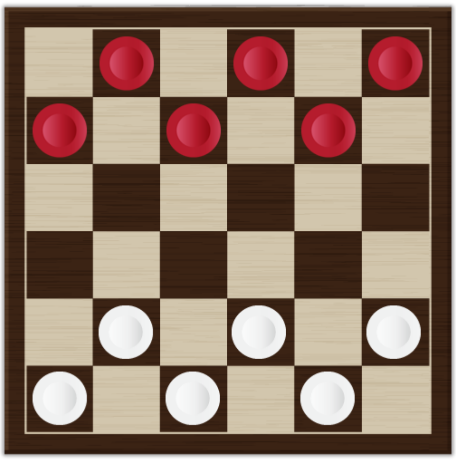
A Board Configurations



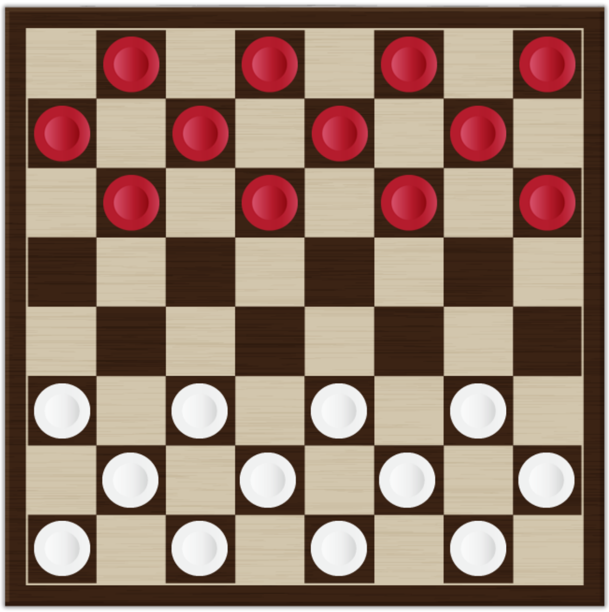
6×8



8×6

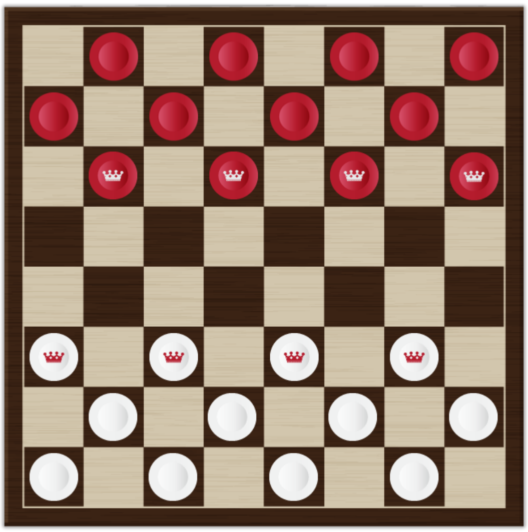


6×6

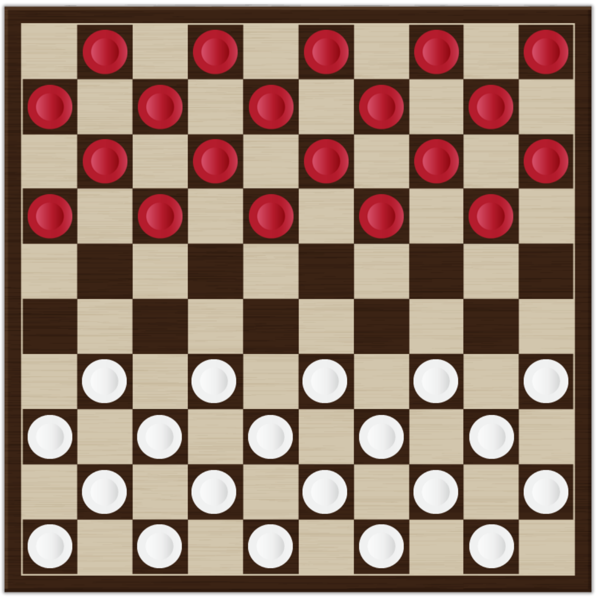


8×8

Figure 17: Symmetrical setups

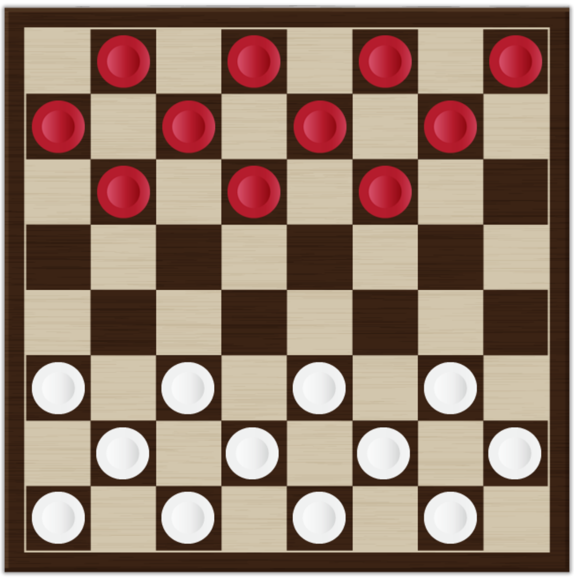


8×8 with kings



10×10

Figure 18: Symmetrical setups



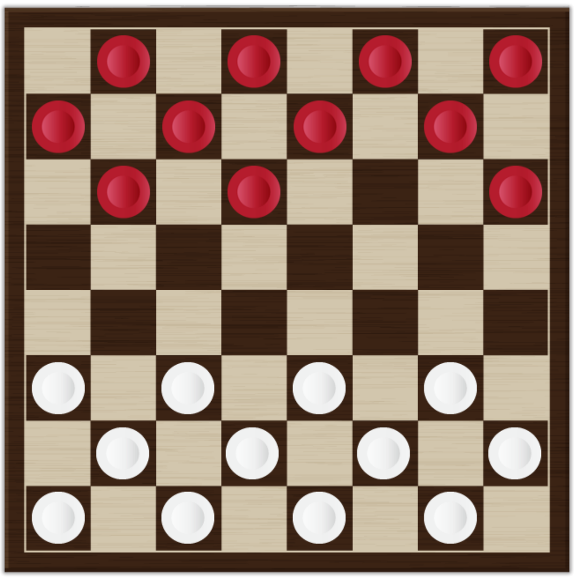


Figure 19: Asymmetrical unbalanced setups: missing piece for red player (variation 1 and 2)

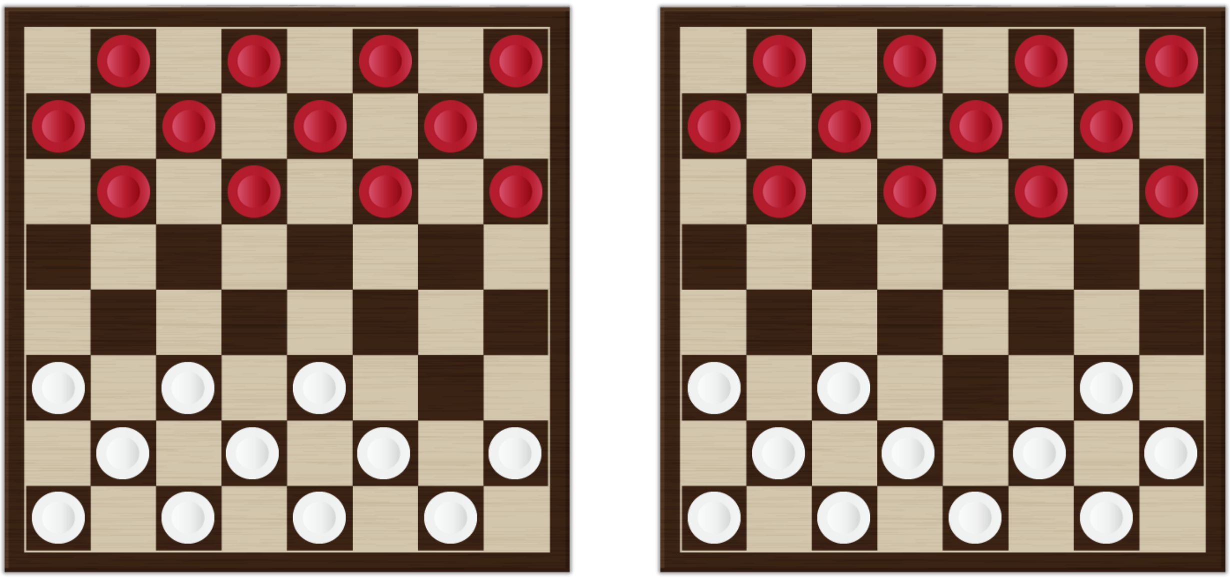


Figure 20: Asymmetrical unbalanced setups: missing piece for white player (variation 3 and 4)

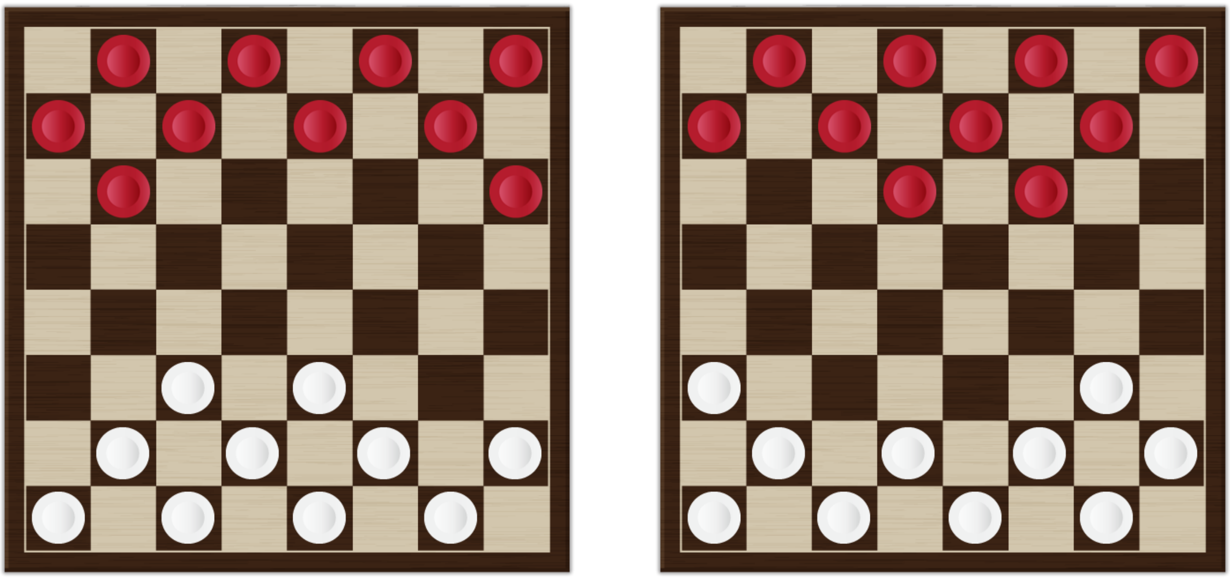


Figure 21: Asymmetrical balanced setups (variation 1 and 2)

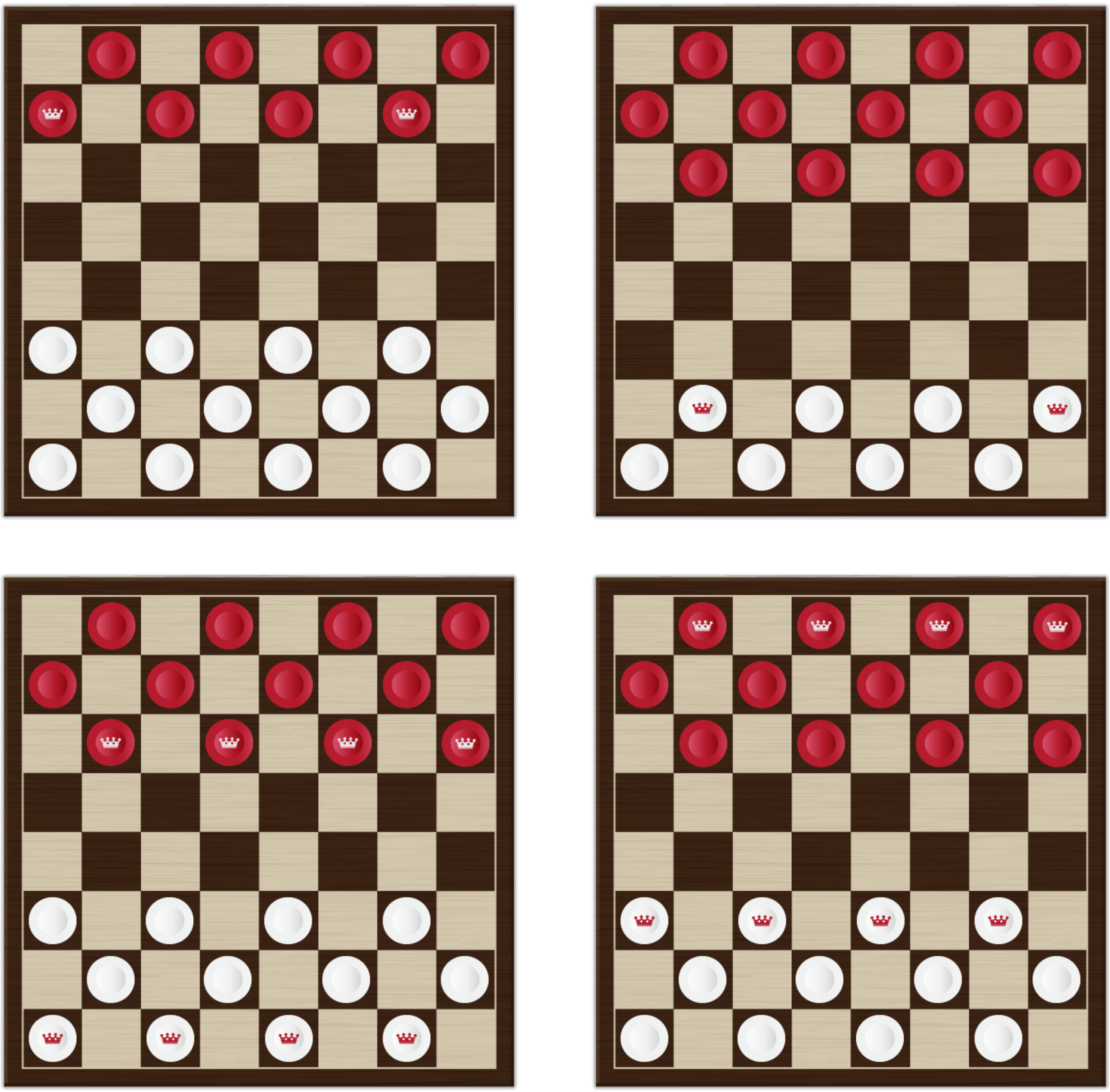


Figure 22: Asymmetrical balanced setups with kings (variation 3 - 6)

B Tables

Symmetric 6×8

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	69	55.99	3.43	178	46.3	4.42	200	32.16	5.92
	loss	73	52.53	3.6	6	40.67	5.17	0	–	–
	draw	58	98.95	3.64	16	103.12	4.38	0	–	–
OSLA	win	7	45.57	5.0	64	36.31	7.8	199	33.67	6.59
	loss	179	43.98	4.61	93	37.45	7.59	1	35.0	4.0
	draw	14	101.07	4.0	43	94.0	6.67	0	–	–
Random	win	0	–	–	0	–	–	111	41.73	3.79
	loss	200	32.13	6.09	200	32.0	6.86	89	39.42	4.51
	draw	0	–	–	0	–	–	0	–	–

Symmetric 8×6

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	61	43.84	3.21	179	37.97	4.24	200	26.77	5.42
	loss	70	46.59	3.56	3	30.0	7.33	0	–	–
	draw	69	96.58	3.54	18	82.78	4.78	0	–	–
OSLA	win	2	74.5	5.0	57	37.88	6.47	196	28.83	5.92
	loss	188	34.95	4.39	110	34.59	6.86	4	26.5	5.5
	draw	10	92.9	4.1	33	86.45	6.42	0	–	–
Random	win	0	–	–	3	32.33	4.67	86	34.37	3.77
	loss	200	26.05	5.56	197	28.28	6.04	114	32.75	3.94
	draw	0	–	–	0	–	–	0	–	–

Symmetric 6×6

first	second	MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	53	31.38	2.6	181	28.81	2.98	200	20.7	3.9
	loss	59	37.34	2.73	1	25.0	4.0	0	–	–
	draw	88	74.22	3.28	18	70.22	3.72	0	–	–
OSLA	win	1	69.0	2.0	46	30.96	4.7	193	23.36	4.42
	loss	180	27.6	3.09	134	24.92	4.95	7	22.71	3.57
	draw	19	74.53	3.68	20	72.75	5.0	0	–	–
Random	win	0	–	–	4	21.75	4.5	95	27.14	2.85
	loss	200	19.73	4.25	196	23.16	4.52	105	24.5	3.27
	draw	0	–	–	0	–	–	0	–	–

Symmetric Standard 8×8

first	second	MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	74	68.15	4.26	130	70.17	4.39	200	43.07	7.76
	loss	70	65.96	4.33	34	59.03	7.76	0	–	–
	draw	56	134.18	3.46	36	113.64	5.73	0	–	–
OSLA	win	15	64.33	6.47	85	54.88	9.41	200	42.08	8.72
	loss	169	61.8	5.4	86	54.42	9.43	0	–	–
	draw	16	127.25	4.5	29	107.9	10.03	0	–	–
Random	win	0	–	–	0	–	–	103	54.54	4.81
	loss	200	43.69	7.61	200	42.98	8.7	97	53.02	5.26
	draw	0	–	–	0	–	–	0	–	–

Symmetric Standard 10×10

first	second	MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	65	124.94	5.57	93	134.0	4.61	200	85.45	10.73
	loss	68	121.13	6.0	70	97.29	10.96	0	–	–
	draw	67	192.42	3.88	37	179.0	6.0	0	–	–
OSLA	win	74	94.54	11.12	88	100.23	13.78	200	74.98	13.38
	loss	87	134.55	4.49	68	107.09	12.66	0	–	–
	draw	39	172.82	6.54	44	147.73	16.95	0	–	–
Random	win	0	–	–	0	–	–	97	99.64	7.24
	loss	200	85.61	10.88	199	74.25	13.6	103	96.95	7.78
	draw	0	–	–	1	125	12	0	–	–

Symmetric 8×8 with Kings

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	62	66.73	4.66	166	61.75	5.4	200	43.51	7.53
	loss	82	66.5	4.5	18	60.94	6.56	0	–	–
	draw	56	121.39	3.62	16	113.62	5.56	0	–	–
OSLA	win	9	55.33	7.0	96	50.83	10.07	200	42.52	8.61
	loss	173	63.72	5.23	71	54.21	9.39	0	–	–
	draw	18	116.11	5.39	33	113.58	8.79	0	–	–
Random	win	0	–	–	0	–	–	91	52.93	5.33
	loss	200	42.91	7.71	200	42.08	8.81	109	52.98	5.28
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Unbalanced – variation 1

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	60	70.45	3.77	133	65.76	4.61	200	44.24	6.84
	loss	101	63.33	5.22	32	54.56	7.44	0	–	–
	draw	39	120.59	3.85	35	117.23	5.31	0	–	–
OSLA	win	2	56.0	5.5	55	56.35	8.78	198	44.57	7.61
	loss	189	60.39	5.76	121	51.1	9.86	2	46.0	10.0
	draw	9	113.0	4.44	24	115.46	7.62	0	–	–
Random	win	0	–	–	0	–	–	85	54.16	4.33
	loss	200	40.62	8.08	200	39.31	9.27	115	49.13	5.83
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Unbalanced – variation 2

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	36	68.72	4.19	138	68.15	4.43	200	43.82	7.02
	loss	121	60.99	5.38	29	57.34	8.14	0	–	–
	draw	43	129.44	3.84	33	112.36	6.27	0	–	–
OSLA	win	2	67.5	5.5	58	51.84	9.72	200	44.7	7.67
	loss	190	58.84	5.78	117	49.85	9.96	0	–	–
	draw	8	125.75	4.75	25	107.16	8.16	0	–	–
Random	win	0	–	–	0	–	–	63	54.86	4.24
	loss	200	39.8	8.53	200	39.42	9.29	137	49.95	5.82
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Unbalanced – variation 3

first	second	MCTS			OSLA			Random		
		games	duration	pieces	games	duration	pieces	games	duration	pieces
MCTS	win	103	62.81	4.98	181	57.82	5.98	200	40.81	8.23
	loss	40	75.35	3.77	7	61.29	7.14	0	–	–
	draw	57	129.18	3.61	12	119.83	4.5	0	–	–
OSLA	win	22	56.09	7.41	119	50.69	9.71	200	40.26	9.04
	loss	155	67.85	4.08	52	60.06	8.25	0	–	–
	draw	23	113.48	6.65	38	109.32	8.84	0	–	–
Random	win	0	–	–	1	73.0	3.0	136	49.65	6.12
	loss	200	44.62	6.75	199	45.78	7.58	64	54.0	4.92
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Unbalanced – variation 4

first	second	MCTS			OSLA			Random		
		games	duration	pieces	games	duration	pieces	games	duration	pieces
MCTS	win	120	64.56	4.94	187	60.77	5.64	200	40.72	8.13
	loss	45	76.51	3.73	4	58.25	6.75	0	–	–
	draw	35	127.94	3.69	9	106.22	5.44	0	–	–
OSLA	win	23	55.26	7.26	101	53.58	9.53	200	39.17	9.21
	loss	147	66.92	4.19	69	56.26	8.77	0	–	–
	draw	30	118.4	5.97	30	118.0	8.27	0	–	–
Random	win	0	–	–	1	71.0	5.0	126	48.57	6.1
	loss	200	44.46	6.76	199	44.38	7.68	74	54.73	4.32
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Balanced – variation 1

first	second	MCTS			OSLA			Random		
		games	duration	pieces	games	duration	pieces	games	duration	pieces
MCTS	win	71	64.97	3.96	169	61.7	4.4	200	39.73	7.04
	loss	77	67.78	3.86	11	55.27	7.09	0	–	–
	draw	52	125.17	4.02	20	111.85	5.7	0	–	–
OSLA	win	4	53.0	6.25	85	50.92	8.58	200	39.35	7.79
	loss	180	61.47	4.63	70	54.47	8.11	0	–	–
	draw	16	115.12	4.38	45	107.38	7.64	0	–	–
Random	win	0	–	–	0	–	–	112	50.55	4.63
	loss	200	39.6	7.18	200	39.13	7.95	88	48.44	4.94
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Balanced – variation 2

second / first		MCTS			OSLA			Random		
		games	duration	pieces	games	duration	pieces	games	duration	pieces
MCTS	win	76	61.67	4.21	175	57.79	4.86	200	38.94	7.17
	loss	82	63.04	4.33	6	63.17	6.5	0	–	–
	draw	42	123.67	3.67	19	104.26	6.32	0	–	–
OSLA	win	12	49.17	7.42	63	59.78	7.38	199	39.49	7.87
	loss	172	61.1	4.76	89	53.19	7.9	0	–	–
	draw	16	122.12	4.56	48	109.6	8.19	1	71.0	8.0
Random	win	0	–	–	0	–	–	102	48.79	5.18
	loss	200	39.25	7.26	200	39.56	7.79	98	49.34	5.09
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Balanced – variation 3

second / first		MCTS			OSLA			Random		
		games	duration	pieces	games	duration	pieces	games	duration	pieces
MCTS	win	2	74.0	2.5	38	76.16	2.45	198	48.48	4.63
	loss	192	45.49	7.47	121	44.4	9.31	2	45.5	7.0
	draw	6	123.67	4.33	41	106.73	6.78	0	–	–
OSLA	win	0	–	–	11	78.55	4.91	191	51.06	4.87
	loss	198	45.63	7.61	170	44.52	10.38	9	45.33	7.78
	draw	2	117.0	3.5	19	119.68	8.68	0	–	–
Random	win	0	–	–	0	–	–	35	56.63	3.11
	loss	200	33.08	9.61	200	31.68	10.57	165	38.85	7.95
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Balanced – variation 4

second / first		MCTS			OSLA			Random		
		games	duration	pieces	games	duration	pieces	games	duration	pieces
MCTS	win	185	48.35	7.22	195	48.31	7.37	200	33.53	9.55
	loss	5	79.4	2.2	1	56.0	6.0	0	–	–
	draw	10	119.0	3.7	4	159.75	3.0	0	–	–
OSLA	win	130	44.62	9.15	162	45.53	10.18	200	31.89	10.35
	loss	31	78.23	2.52	7	81.43	4.71	0	–	–
	draw	39	101.46	7.28	31	119.06	7.48	0	–	–
Random	win	3	41.67	7.67	10	48.2	7.5	161	39.53	7.84
	loss	197	48.97	4.54	190	50.57	4.98	39	52.36	3.64
	draw	0	–	–	0	–	–	0	–	–

Asymmetric Balanced – variation 5

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	68	72.01	4.35	173	63.51	5.21	200	42.2	7.93
	loss	79	65.7	4.61	11	67.0	6.64	0	–	–
	draw	53	122.91	3.72	16	114.75	5.25	0	–	–
OSLA	win	9	64.11	6.78	93	50.04	10.12	197	43.03	8.61
	loss	171	62.83	5.31	75	54.0	9.37	3	47.72	6.67
	draw	20	113.0	6.15	32	115.47	8.47	0	–	–
Random	win	0	–	–	1	46.0	5.0	112	54.12	5.15
	loss	200	42.72	7.91	199	42.62	8.61	87	54.49	4.97
	draw	0	–	–	0	–	–	1	110.0	2.0

Asymmetric Balanced – variation 6

first \ second		MCTS games	MCTS duration	MCTS pieces	OSLA games	OSLA duration	OSLA pieces	Random games	Random duration	Random pieces
MCTS	win	58	69.62	4.31	174	62.83	5.2	200	42.94	7.56
	loss	86	68.35	4.4	11	57.0	7.55	0	–	–
	draw	56	126.18	3.88	15	122.6	5.3	0	–	–
OSLA	win	6	55.67	7.0	85	51.31	9.95	200	42.78	8.7
	loss	178	60.95	5.44	79	54.87	9.62	0	–	–
	draw	16	121.12	5.06	36	104.58	9.36	0	–	–
Random	win	0	–	–	0	–	–	109	55.49	4.67
	loss	200	43.09	7.82	199	42.42	8.93	91	50.81	5.53
	draw	0	–	–	1	107.0	7.0	0	–	–