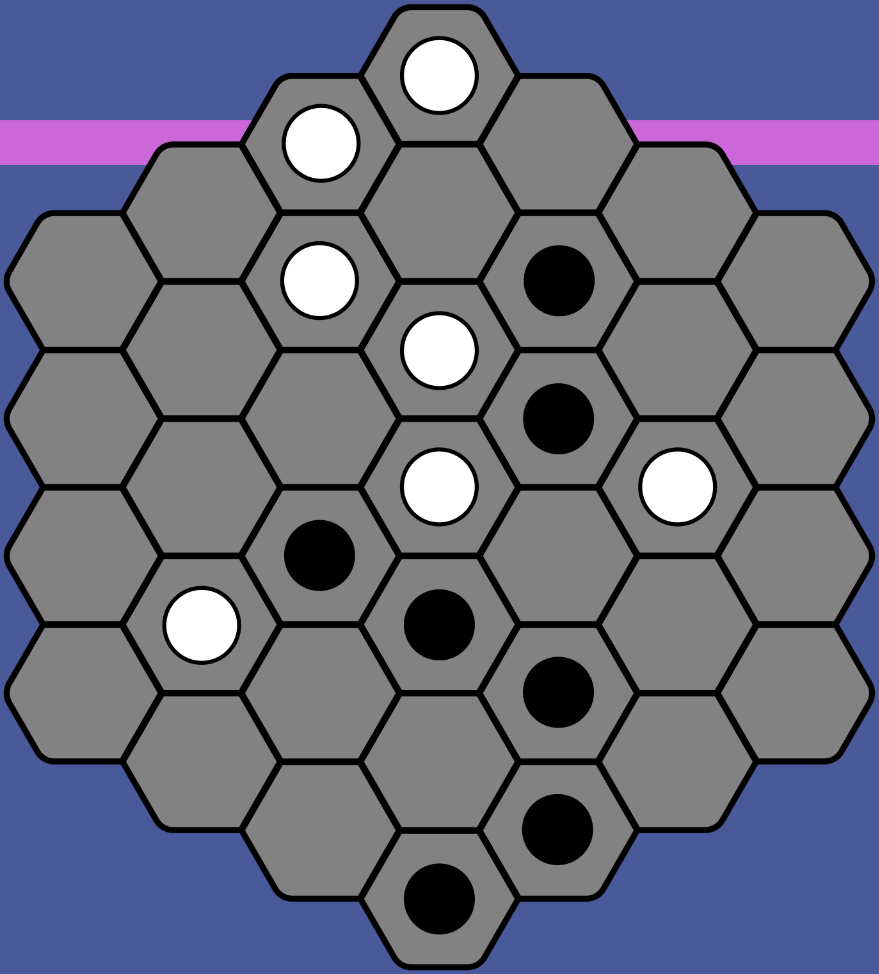
Milestone: Developing an AI for a Novel Game

Connor McCarthy
Corban Chiu
Sunny Chavan



Contents

Keywords:

Minimax tree search, alpha-beta pruning, state evaluation heuristics, hyper-heuristic search, genetic algorithms, neural networks, fitness functions.

Application Setting:

We developed a game-playing AI for a novel board game called *Milestone*, developed by a Cornell Math Ph.D. student Mark Schachner.

External Study Participants:

Anisha Saini, Claudia Arredondo Zayas, Elizabeth Ehl, Jay Sangwan, Jeevan Deol, Lavanya Pinnepalli, Lucia Pannunzio, Ruchitha Rajaghatta, Samarth Desu, Sanjana Shanmugavel, Sylvia Bayrakdarian, Tyler Carreja.

I. Introduction

Milestone, a nacent board game recently developed by Cornell Ph.D student Mark Schachner, is played on a hexagonal grid with 4 hexes a side, oriented such that the two players face opposite vertices. Players start with 10 pieces, either white or black, in the first four rows of their side of the board. The player with the black pieces starts first. Players alternate moving pieces to one of three forward-adjacent hexes: the hex directly in front, front-right, or front-left. Figure 1 depicts the starting arrangement of the board, with one piece highlighted in purple and its possible moves in blue.

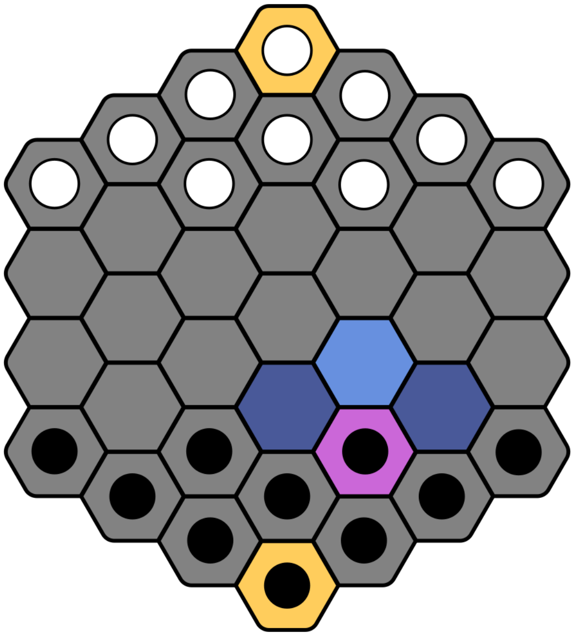


Figure 1: *The piece highlighted in purple can move to the light & dark blue hexes, but can only ever capture directly forward to the light blue hex. Their goal is eventually to reach the opposing, top-most home square, highlighted in yellow.*

Capturing other pieces is possible, but a piece may only capture when moving directly forward, at which point the captured piece is removed from the board. The object of the game is to move one of your pieces into your opponent's *home space*, i.e. the hex closest to your opponent. Alternatively, you may also win if your opponent has no moves remaining, whether they have no pieces left or simply cannot make a legal move.

A. Objectives

We had two primary goals when approaching this project. The first was to learn more about the gameplay and theory behind Milestone. Being a novel game, formal theory for Milestone is extremely limited and the existence of high-level gameplay is a promising vector of discovery. The second was to develop an AI that could play Milestone at a proficient level. This meant that the AI would beat beginner and intermediate players with relative ease, and split games with experts. The relative ranking of players and measure of experience is vaguely defined in the context of this game, since Milestone is not publicly available. Only about 25 people have played Milestone and there is little formal research into strategy besides that studied by Mark. This novelty made the project more intriguing as we had to decide what strategies and moves were superior to others.

In addition, we wanted to create a playable game that allowed players to challenge AI and human opponents, while outputting the status of the game on an associated user interface. As the objective of the project was to develop the best AI, we planned for the interface to be relatively barebones – its primary purpose was to allow a human to follow and play the game (appendix figure 9). The majority of our project complexity would exist in the design and training of our AI.

B. AI Development

The creation of our AI involved three distinct stages that built upon each other. In the first stage, we created a set of nineteen heuristics (see appendix table C) to evaluate a position and used a minimax tree search to find the best move. Due to there being no prior research on optimal game strategy, we developed these heuristics by playing many games amongst ourselves, and discussing strategies we noticed performed better. For example, we noticed that controlling the middle of the board seemed important, so we implemented a location mapping called *Middle Proximity* which gave pieces more value the closer they were to the middle. In general, we sought to create a larger number of potentially viable heuristics in an effort to provide the largest possible search space for our AI to explore. Each heuristic provides a score for each given game state and these scores were averaged as our AI's evaluation function. By convention, heuristics were developed from the perspective of the player with black pieces and they were designed such that they were zero sum. By the end of this stage we had an AI which played

Milestone at a rudimentary level and served as the baseline for future improvements.

The second stage of our AI development involved the creation of a genetic algorithm[(1)]. Our goal with such an algorithm was to explore the space composed of all possible weightings of our 19 heuristics. To do so, we developed a system that allowed us to compare batches, or *populations*, of various AIs, defined by their weightings, and repopulate those that were most successful into a subsequent generation.

To begin, we initialize 36 AI agents, each with randomized weight values, providing us a starting point for our search. From there, we chose to select a random subsampling of possible matchups between AI, in what resembles a partial round-robin match schedule. Each match scheduled consists of two games, one for each player with the black and white pieces. This design allows us a large sample of game results from which we can evaluate our AI, without having to enumerate all possible matchups. We track the performance of these AIs throughout their matches using an ELO system – a fitness function designed to represent the skill-levels of players, with large differences in ELO indicating large skill gaps.

When all the matches have been played, we are able to select our most fit agents – those with the highest ELO. Each selected AI will be allowed to proliferate into the next round through children which are generated by adding or subtracting a small random perturbation from each of the weights of the parent. In this manner we can create a new batch of AI, representative of the best-performing AI of the previous generation. To this population we also add a number of randomly initialized AI. This addition is designed to mitigate the possibility of convergence on non-equilibrium strategies.

From this newly constructed population, we can repeat this process – once for each generation. We conducted two experiments, A and B, each with 100 generations of this process (appendix table E). Experiment A was designed to prioritize slow, steady progress towards an equilibrium by maintaining lower amounts of randomness between generations and having a less-strict selection criteria. In contrast, experiment B introduced more randomness and variation, attempting to ensure we located a global maximum, at the risk of slower convergence. This stage was critical in improving our AI's ability to make more accurate and effective decisions, allowing it to narrow in on proper heuritic weights.

The final stage of our AI development involved training a neural network on a database of the nearly 200,000 games played in the previous stage. The database included every state reached and the respective winner (black or white). We created two neural networks – one from each respective experiment. Our neural networks were trained to minimize the mean squared error of black win percentage for a game state, allowing it to predict the "quality" of a position. Then, our neural network AIs would select the move which maximized their chances of winning. We felt this stage was important because it allowed us to explore beyond the 19-D space of heuristics that our genetic algorithm searched in. These neural networks were able to build their own concept of depth and served as a benchmark to evaluate our genetic algorithm. Ultimately, these three stages combined to help us develop a game-playing AI that provides a challenging opponent for human players.

II. Findings

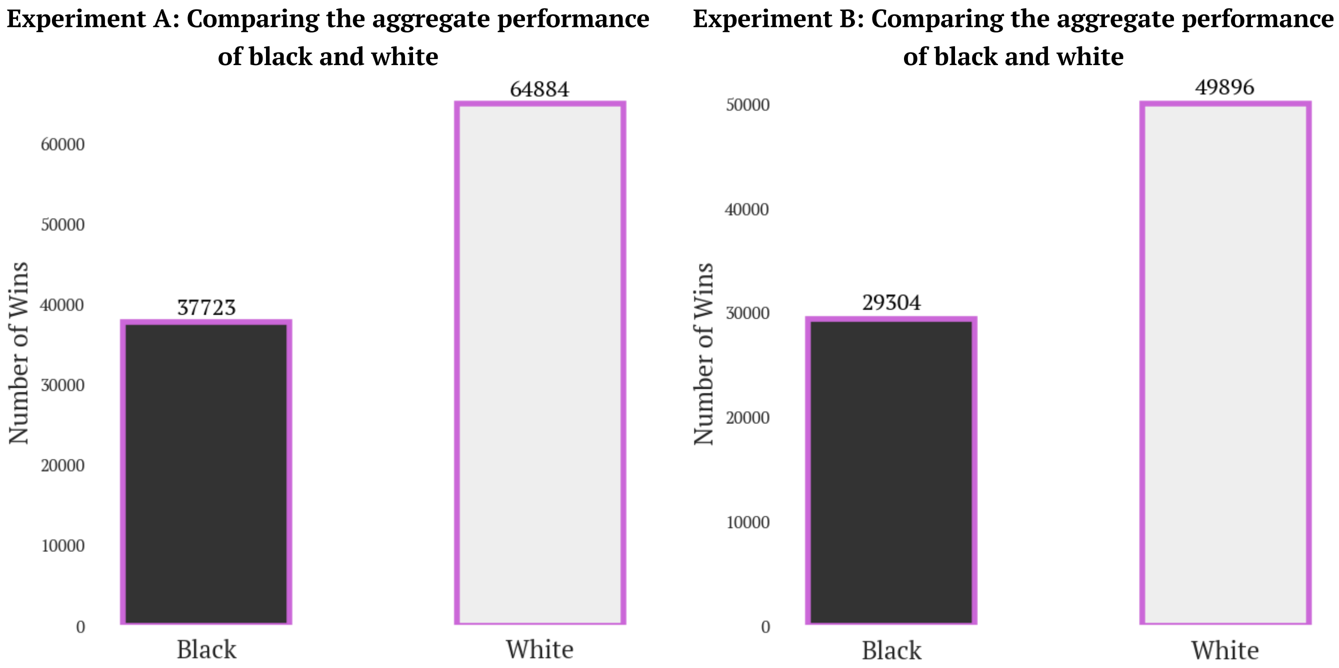
In addition to creating a proficient AI, another primary goal of our project was to gain more insight into the theory of Milestone. The data from the two genetic algorithm experiments serve as the basis for the following analysis.

A. Black vs. White Performance

The first area we explored is whether the player with the first move, playing with the black pieces, has an inherent advantage. We expected games to either be evenly matched or in favor of the black AI as taking the initiative appeared to be favored in our gameplay, an attribute commonly known as *first mover advantage*. The data shown in figures 2i and 2 actually illustrates the opposite. In both experiments, AIs playing with the white pieces won ~63% of the total games played.

There are a few reasons we might consider for this outcome. Firstly, as white has the opportunity to react to black's initial move, it can establish a defensive presence in the correct area of the board, allowing it to strategically counter black's moves throughout the game. Secondly, Milestone presents numerous situations where there are no more favorable moves available. In such instances, an experienced player would recognize this scenario and execute stalling moves until their opponent is forced to make a bad move, similar to the chess concept of zugzwang. Since white is one turn behind black, black may consistently be the first player to run out of good or stalling moves.

[(1)] appendix figure 10 depicts the running of one generation of this genetic process



(i) *The White AI won 63% of the ~100,000 games played.*

(ii) *The White AI won 63% of the ~80,000 games played.*

Figure 2: *Number of Games Won: Black vs. White*

B. Heuristics

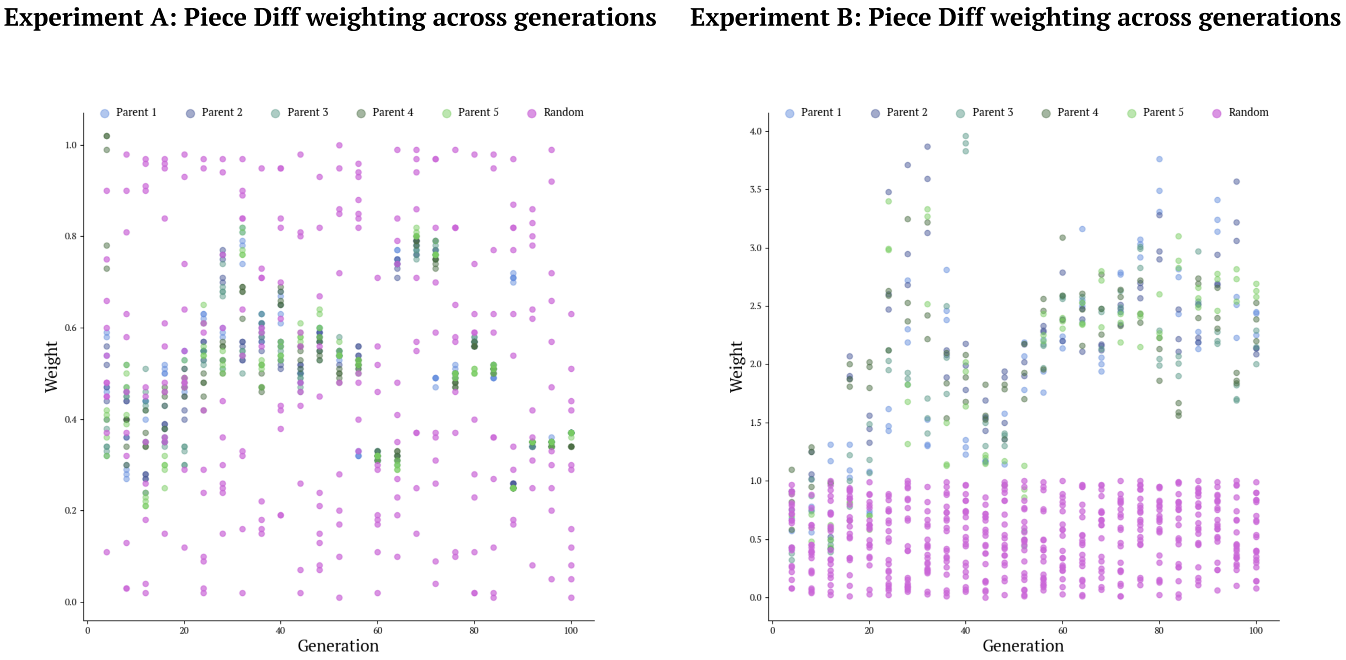
A key component of our AI development involved the selection of heuristics. To mitigate the introduction of our own biases, we wanted our pool to include a wide variety of what we thought were strong strategies, and allow the genetic algorithm to identify which of these were important or not. To add complexity to our heuristics, we also created location maps which gave pieces higher value depending on factors such as distance to the center hex or distance to the middle dividing line. For example, the heuristic *Aggr Pieces Anti Centrality* takes the *Aggr Pieces* heuristic, which is simply a count of aggressive pieces, and adds weight to those further from the central hex. The rating a heuristic produces for a position is actually a difference between the heuristic's evaluation of black and white pieces. In doing so, we can reflect the zero-sum nature of Milestone where a good position for one player is an equally poor position for the other. The enumeration of heuristics and their descriptions can be found in table C.

Some of the heuristics we felt were most important, from our experience, included: *Piece Diff*, *Straight Line Middle Prox*, and *Attack In-sync*. *Piece Diff* counts the number of pieces a player has left; in a game where pieces are limited, it is crucial to prevent one's opponent from building too much of an advantage. *Straight Line Middle Prox* counts the number of pieces stacked behind each other, with greater value placed on those closer to the middle line. Possessing lines of pieces allows a player to have a stronger defensive position while simultaneously opening up opportunities for a more consolidated attack. *Attack In-sync* measures how synchronized a player's attacking pieces are with respect to each other. This is a pivotal concept as Milestone is a game where multiple avenues of coordinated timed attacks are crucial to strong gameplay. Forcing a well-timed trade can often be beneficial if you have another piece nearby. If not, it may be a waste of pieces and/or time.

i. Heuristic Distribution

With regards to the heuristics described above, we analyzed the distribution of their weights over generations as shown in figures 3, 4, & 5. Every 4 generations, we plot the weight each AI has for the heuristic in question. Within these figures, AIs that are children from a previous generation are represented as green and blue dots and random AIs as purple dots. Across both experiments and all three heuristics, we see different levels of convergence which provide valuable insights.

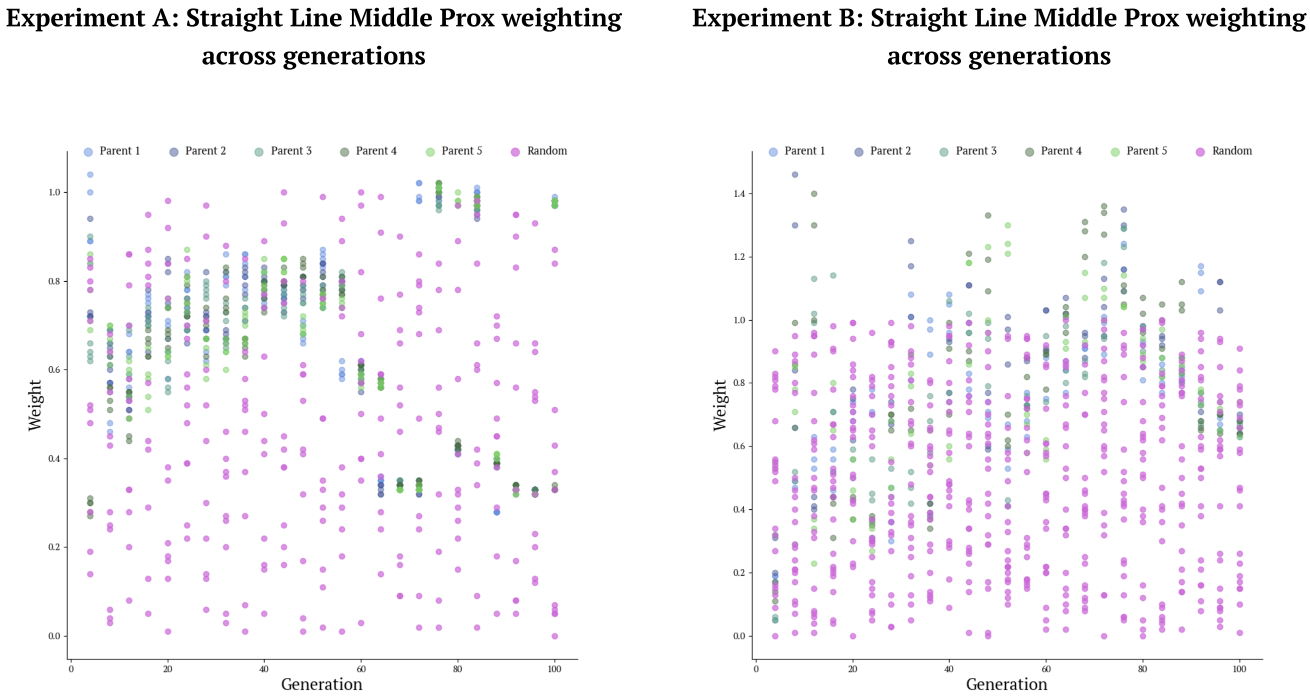
When considering the *Piece Diff* heuristic, we notice exploration in the middle generations of Experiment A where weighting it more heavily was favored. However, as the generations progressed, its weighting returned to its starting



(i) *The Piece Diff heuristic is valued more highly in middle generations but converges to a more typical weight in the end.*

(ii) *The Piece Diff heuristic is valued very highly from the onset and remains that way across generations.*

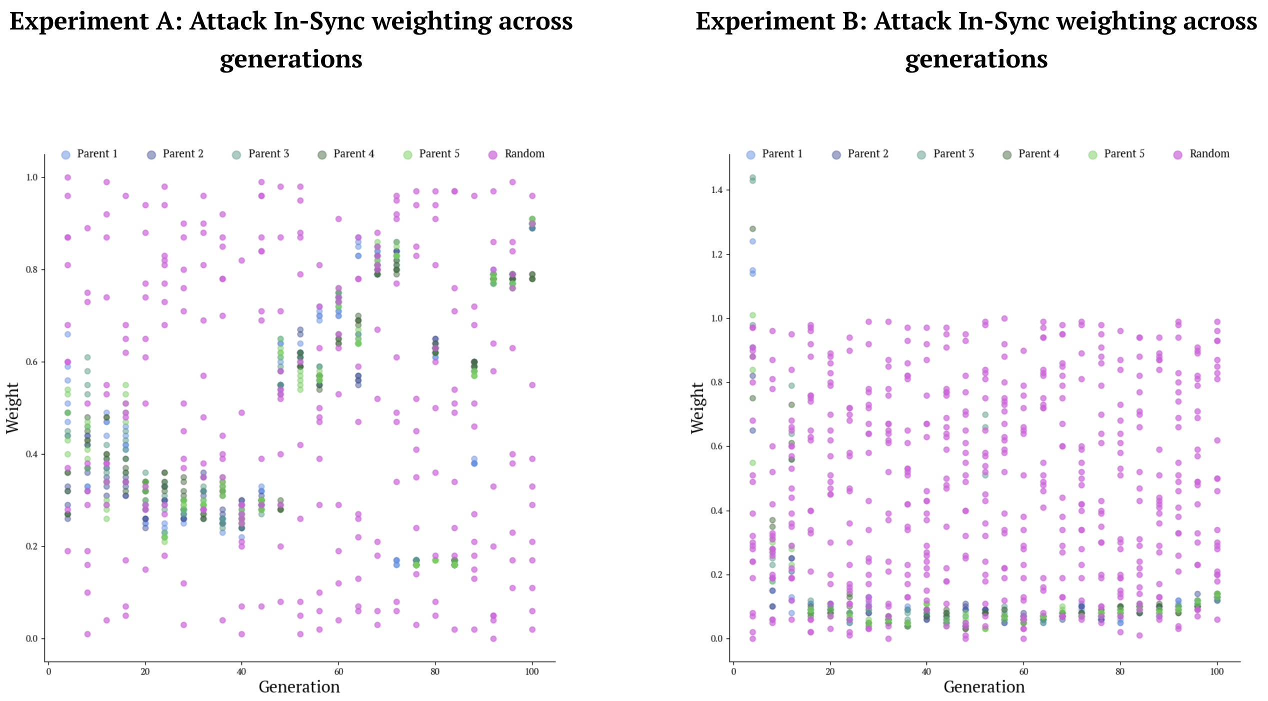
Figure 3: *Distribution of Piece Diff Heuristic Across Generations*



(i) *The Straight Line Middle Prox heuristic is valued more highly in middle generations but converges to an average weight in the end.*

(ii) *The Straight Line Middle Prox heuristic is initially valued highly but its importance decreases with later generations.*

Figure 4: *Distribution of Straight Line Middle Prox Heuristic Across Generations*


Experiment A: Attack In-Sync weighting across generations

Experiment B: Attack In-Sync weighting across generations

(i) *The Attack In-Sync heuristic is valued weakly from the onset but its importance suddenly increases in later generations.*

(ii) *The Attack In-Sync heuristic is valued weakly from the onset and stays unimportant throughout the experiment.*

Figure 5: *Distribution of Attack In-Sync Heuristic Across Generations*

level. In Experiment B, the heuristic was strongly favored early on and stayed that way throughout the generations. The separation between weights assigned by children and random AIs was surprising because we did not expect any one heuristic to be so heavily emphasized. As a result, we only generated random AIs to have heuristic weights between 0 and 1.

With the *Straight Line Middle Prox* heuristic, we observe exploration across both experiments. In Experiment A, there seems to be a sense of convergence for most generations until later generations suddenly adopt a lower weighting. The change in weight is far more gradual in Experiment B, where the weight of the heuristic slowly increased until it finally receded to a weighting of middling importance.

The *Attack In-sync* heuristic was rather unique. Experiment A is an example of a successful exploration where various weights are considered, with higher weights ultimately proving more beneficial. In contrast, Experiment B is an example of an early convergence where by generation 10, the weight of the heuristic is minimal and varies very little until the end of the experiment.

Across all three heuristics we also saw convergence to a different weight in both experiments, which hints that there is no one strategy that dominates Milestone.

ii. Heuristic ELO Correlation

Another area of interest is the relationship of each heuritic with the performance of an associated AI. If a given heuristic has a high correlation to ELO, it may suggest that the heuristic is associated with better performance. The heatmaps depicting this correlation for both experiments are shown in figure 6.

One noteworthy finding is the positive correlation (≥ 0.3) of the heuristic *Defended Hexes Middle Prox* in both experiments. This suggests that in playing Milestone, it is crucial not only to control the middle of the board but also to maintain control over specific middle hexes. On the other hand, the heuristic *Aggr Pieces Anti Centrality* exhibits a significantly negative correlation (≤ -0.6) in both experiments. This heuristic measures how close your pieces are to your opponent's side, with greater weighting assigned to pieces farther from the center hex. This observation aligns with our experience in Milestone, where advancing pieces towards the outskirts of the board often limits their future potential and weakens the position in the middle.

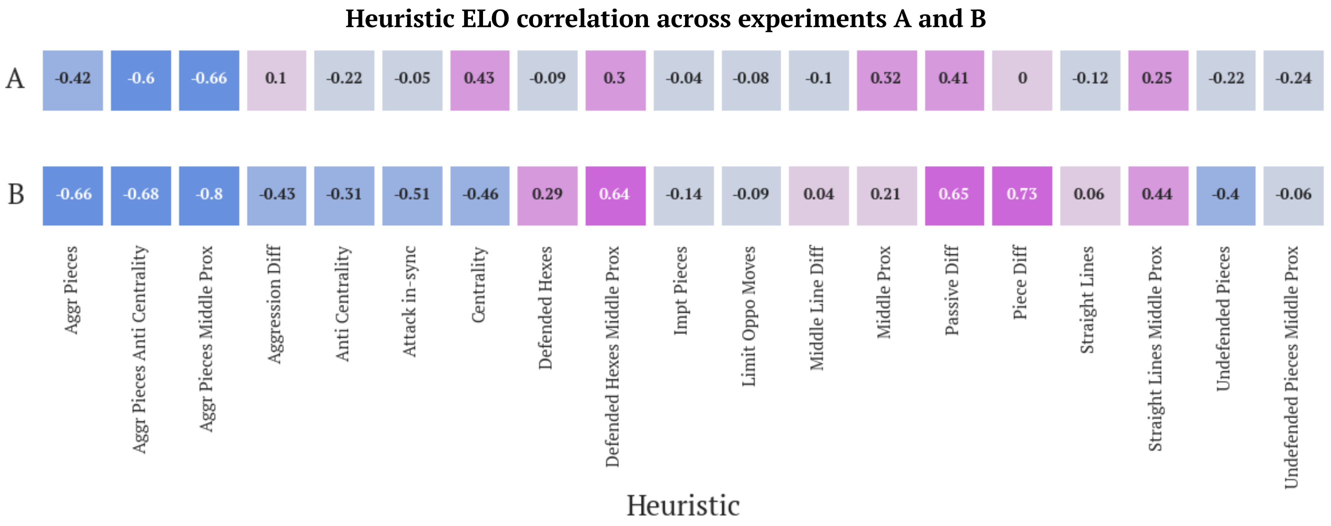


Figure 6: *This illustrates the correlation of each heuristic and ELO. A dark purple indicates a strong positive correlation whereas a dark blue indicates a strong negative correlation.*

It is also important to note that for many of the heuristics across experiments, the correlation values are much different. This is likely because the two experiements do not follow the same search paths. This analysis is made under the assumption that all heuristics are independent of one another, which is surely not the case and therefore this analysis should loosely be used as a guideline for general strategies that should continue to be tested.

III. Project Evaluation

Due to the novelty of Milestone, designing a rigorous evaluation of our AI proved challenging, prompting us to divide this process into two categories: one conducted during our ongoing experiments and one conducted with the help of outside participants.

A. Internal Evaluation

We broke this category down into three main areas: progress and evaluation during the genetic algorithm stage, during the neural network stage and between both stages.

i. Genetic Algorithm Evaluation

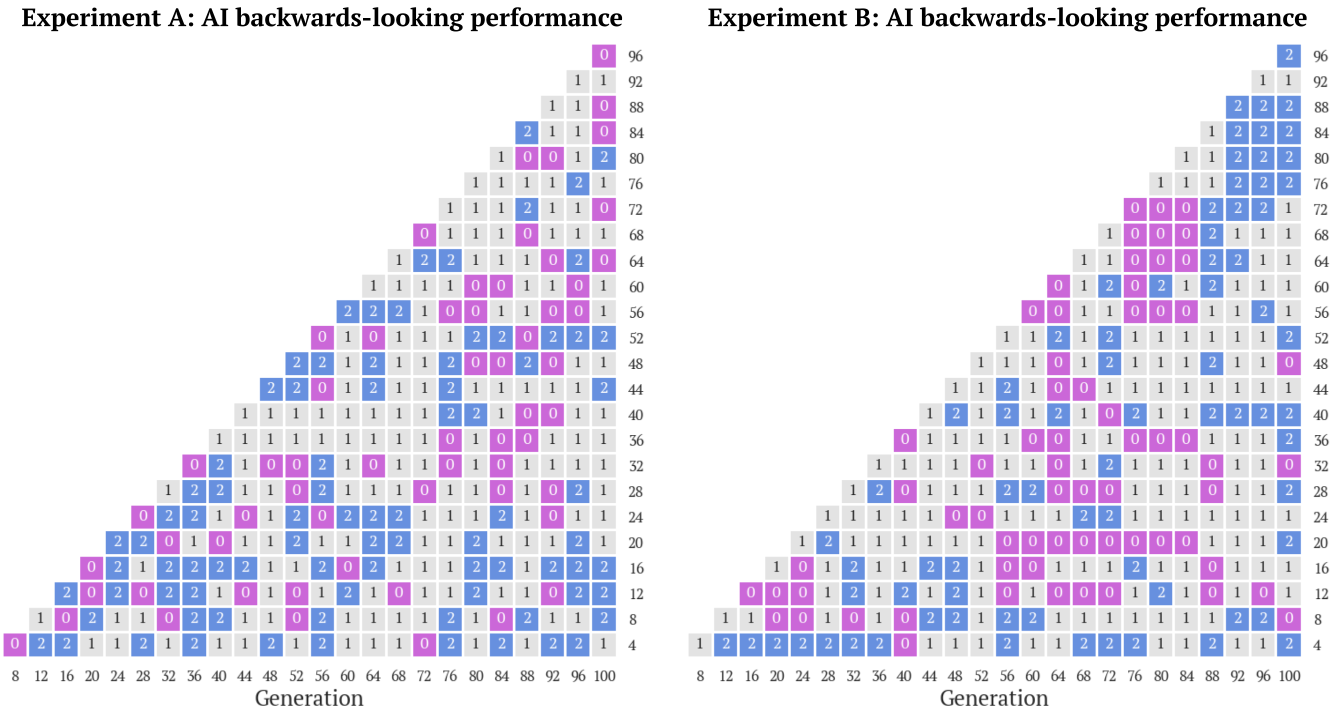
Throughout this stage of development, the AI was continuously trained and evolved to improve its performance in gameplay. The majority of this evaluation process involved having the latest generations of the AI play against previous versions, which provided us valuable insight into the progress made during each generation.

The matrices shown in figure 7 depict the progress across generations for each experiment. Each cell contains the outcome of a match played between the best AI of a later generation and the best AI of an earlier generation. The value within each cell denotes the number of wins of the later-generation, hopefully more-advanced, AI in a 2-game match.

Of particular interest is the consistency of AI development in Experiment A (figure 7i). AIs from later generations very rarely encounter earlier generation AIs that beat it consistently – ties, losses and wins are evenly distributed throughout the figure. On the other hand, in Experiment B (figure 7ii), our later generation AIs often encounter those of earlier generations with significant relative strength; note the groupings of losses to the AIs from generation 20 and generations 64 to 72. However, by the last 10 generations, the best AIs were able to convincingly defeat nearly every previous generation's AI, indicating significant improvement.

ii. ELO Distribution

We can also evaluate progress over time through the distribution of ELOs across generations. When comparing Experiments A and B, we observe that the increased the number of randomly generated AIs added to each generation led to a much clearer separation between good and bad agents. Starting from about generation 40 in Experiment B, the ELOs of AIs tend to be split between a low range and a high range; random AIs mostly populate the lower range



Experiment A: AI backwards-looking performance

	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100
96																								0
92																							1	1
88																						1	1	0
84																					2	1	1	0
80																				1	0	0	1	2
76																			1	1	1	1	2	1
72																		1	1	1	2	1	1	0
68																	0	1	1	1	0	1	1	1
64																1	2	2	1	1	1	0	2	0
60															1	1	1	1	0	0	1	1	0	1
56														2	2	2	1	0	0	1	1	0	0	1
52													0	1	0	1	1	1	2	2	0	2	2	2
48												2	2	1	2	1	1	2	0	0	2	0	1	1
44											2	2	0	1	2	1	1	2	1	1	1	1	1	2
40										1	1	1	1	1	1	1	1	2	2	1	0	0	1	1
36									1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1
32								0	2	1	0	0	2	1	0	1	1	0	1	0	1	1	1	1
28							1	2	2	1	1	0	2	1	1	1	0	1	1	0	1	0	2	1
24						0	2	2	1	0	1	2	0	2	2	2	1	1	1	2	1	0	1	1
20					2	2	0	1	0	1	1	2	1	1	2	2	1	1	2	1	1	1	2	1
16				0	2	1	2	2	2	2	1	1	2	0	2	1	1	1	2	2	1	2	2	2
12			2	0	2	0	2	2	1	0	1	0	1	2	1	0	1	1	2	1	1	0	2	2
8		1	0	2	1	1	0	2	2	1	1	0	2	1	1	1	1	2	1	0	2	1	1	2
4	0	2	2	1	1	2	1	2	1	1	2	1	2	1	1	1	0	2	1	2	1	2	2	1

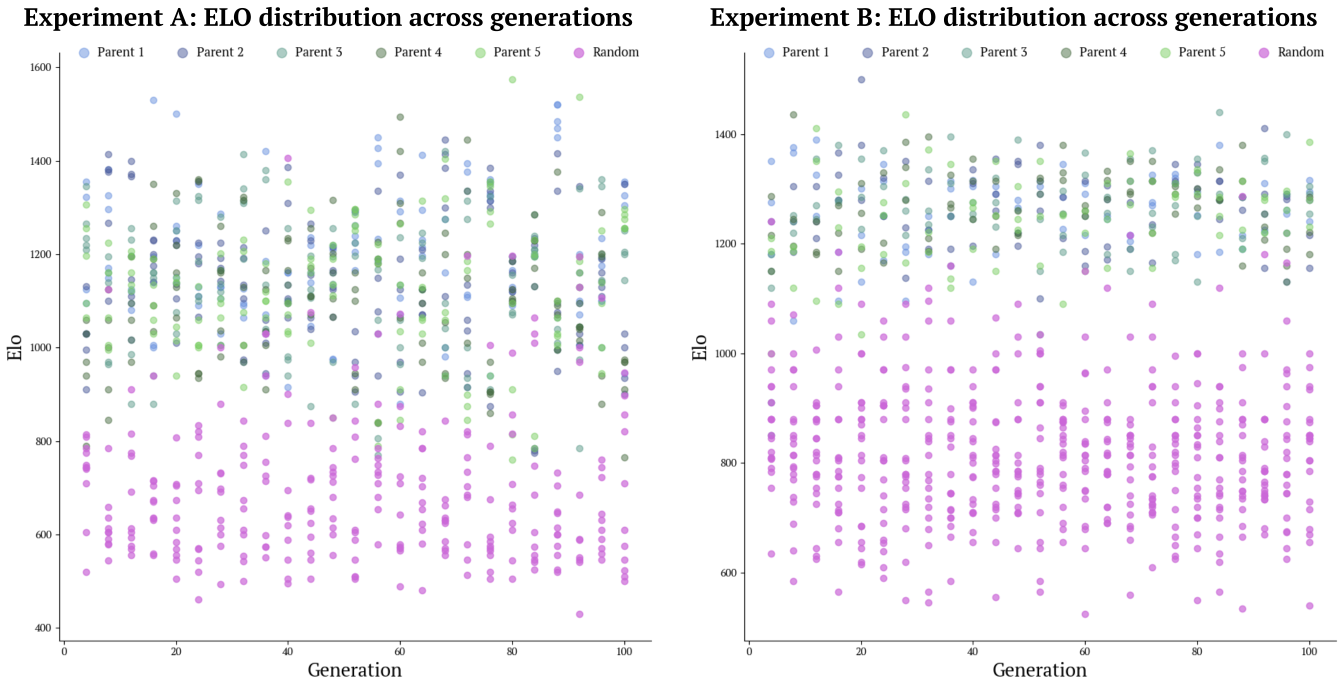
Experiment B: AI backwards-looking performance

	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100
96																								2
92																							1	1
88																						2	2	2
84																					1	2	2	2
80																				1	1	2	2	2
76																			1	1	1	2	2	2
72																		0	0	0	2	2	2	1
68																	1	0	0	0	2	1	1	1
64																1	1	0	0	0	2	2	1	1
60															0	1	2	0	2	1	2	1	1	1
56														0	0	1	1	0	0	0	1	1	2	1
52													1	1	2	1	2	1	1	1	1	1	1	2
48												1	1	1	0	1	2	1	1	1	2	1	1	0
44											1	1	2	1	0	0	1	1	1	1	1	1	1	1
40										1	2	1	2	1	2	1	0	2	1	1	2	2	2	2
36									0	1	1	1	1	0	0	1	1	0	0	0	1	1	1	2
32								1	1	1	1	0	1	1	0	1	2	1	1	1	0	1	1	0
28							1	2	0	1	1	1	2	2	0	0	0	1	1	1	0	1	1	2
24						1	1	1	1	1	0	0	1	1	1	2	2	1	1	1	1	1	1	1
20					1	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	2
16				1	0	1	2	1	1	2	2	1	0	0	1	1	1	2	1	1	0	1	1	1
12			0	0	0	1	2	1	2	1	2	1	0	1	0	0	0	1	2	1	0	1	0	1
8		1	1	0	0	1	0	1	0	2	2	1	2	2	1	1	1	1	1	1	1	2	2	0
4	1	2	2	2	2	2	2	2	0	1	1	1	2	1	1	2	2	2	1	1	2	1	1	2

(i) *A: AIs from later generations very rarely encounter earlier generation AIs that beat it consistently – ties, losses and wins are evenly distributed.*

(ii) *By the last 10 generations, the best AIs were able to convincingly defeat nearly every previous generation's AI, indicating significant improvement.*

Figure 7: *Comparative Progress Against Previous Generations*



Experiment A: ELO distribution across generations

Experiment B: ELO distribution across generations

(i) *The worse children from the previous generation performed similarly to the best randoms added to the current generation.*

(ii) *Children from the previous generation significantly outperformed the randoms added to the current generation.*

Figure 8: *Distribution of ELO Across Generations*

Number of Genetic AI Wins	**Experiment A NN**	**Experiment B NN**
Experiment A AI	2	2
Experiment B AI	2	1

Table A: *Comparing genetic AI with neural networks, across both experiments.*

and children of the previous generations mostly populate the higher range. In Experiment A, it is harder to observe any clear split, as an increased number of children inevitably led to worse performing AIs. Perhaps if more generations had been played out, we might see a similar convergence of ELOs. One notable takeaway from these figures is that every so often, a randomly generated AI has one of the highest ELOs in its generation. This is significant because it indicates that our injection of randomness at each generation is contributing as planned – it prevents our genetic algorithm from settling on a local maximum within this 19-D space, instead forcing it to continue searching for a global maximum. This gives us confidence that the final AIs produced by the genetic algorithm are indeed playing the game at a high level.

iii. Experiment Comparison

Following the completion of both experiments, we looked to compare AI performance. In our first comparison, we played a match between the highest ranked AI in the final generation from both experiments. This resulted in a tie, indicating some form of equal gameplay. Surprisingly, when we repeated this test by conducting a round-robin between the top 5 AIs from each experiment, we found a similar result: each experiment won and lost the same number of games. However, while playing games against both experiment's best AI, we noticed that the one produced from Experiment B appeared to be slightly stronger and was seemingly more adaptable to a wider variety of positions and opponent strategies.

iv. Neural Network Evaluation

We next trained a neural network on our database of games played during the genetic algorithm to provide a point of comparison for how well our AIs were performing. By learning to predict the black win percentage of a given game state, this AI could develop its own intuition about Milestone. Not only would it be able to explore a feature space beyond the one defined by our heuristic pool, but more advanced concepts such as playing for a future position would be inherently built into that calculation.

Each experiment contained approximately 3 million unique game states which were split into train and test sets. To minimize mean squared error (MSE), we varied hyperparameters such as the size of hidden layers and the regularization parameter, *alpha*. Our best neural network had an MSE of approximately 8.5% on our test set. It is important to highlight that our neural network was trained on data provided by the genetic algorithm – this may mean that we were unable to train on certain gameplay styles not covered by the scope of our 19 heuristics. However, due to the novelty of Milestone, no other database of games exists and thus we elected to use our own data.

v. Comparative Evaluation of Genetic Algorithm and Neural Network

With the evaluation of the genetic algorithm and neural network stages completed, we looked to arrive at a final decision about which AI was strongest. We matched the best performing AI from each of the genetic algorithm experiments against the two neural network AIs, arriving at the results depicted in table A. We considered the best performing AI for a given experiment to be the AI with the highest ELO in the final generation. The genetic AIs won a majority of the games, indicating that using a heuristic search space likely can accurately represent strong gameplay in Milestone.

B. External Evaluation

i. Human Evaluation Methodology

For human evaluation, we challenged our best AI to play multiple human participants. Due to the results from our experiment comparison above, we selected the best AI from Experiment B to play against our human participants.

For our participants, we enlisted 5 beginner, 5 intermediate, and 5 advanced players to compete against our final AI. Beginner players had played one game to grasp the rules and gameplay, while intermediate players had played 5 or more games, allowing them to form their own strategies. Advanced players, having played more than 15 games, possessed deeper intuitions and could discern optimal moves in different situations. All evaluators were Cornell undergraduate students. To ensure accuracy, we met with each player prior to testing, verifying their skill level. We encouraged them to share strategies they observed during gameplay, or moves they found interesting. The AI played one match against each player in every skill level: this process resulted in 10 games per skill level, 30 games overall.

ii. Human Evaluation Results

Win / Loss Record for AI	**AI Wins**	**AI Losses**
Beginners	9	1
Intermediates	7	3
Advanced	2	8

Table B: *AI performance against human opponents.*

The results against human opponents are shown in table B. Our AI fared consistently well against beginners and intermediates and won 2 games against advanced players. The main difference we identified between skill levels was that advanced players were much more situationally aware than their counterparts – they showcased a comprehensive understanding of the different phases of Milestone from opening to middlegame to endgame and played accordingly. Thus, one potential area of improvement for the AI is adapting its play style to each phase of the game – this could mean constructing different evaluation functions for each phase or including a time variable that affects the weighted combination of heuristics as the game progresses.

During this process, we also took note of some aspects of the AI's gameplay. Many times, the AI would play a clean game until it would make one strategic misstep. Since Milestone is a rather short game with a small number of pieces and provides no opportunity for moving backwards, these mistakes proved consequential in some situations. Against beginners and most intermediates, these mistakes did not make much difference as they could not capitalize on a possible theoretical advantage. However, advanced players were able to identify these missteps early on and respond effectively, which would often lead to an AI loss.

IV. Further Exploration

Reflecting on this semester-long project, we believe there are specific areas within our AI development that could be further explored.

The first area is as it relates to searching within the heuristic space. With better hardware, we could increase the number of games played in parallel[2], enabling larger experiments. This would allow us to delve deeper into the 19-D heuristic space previously established and also include additional new heuristics, using our current resulsts to help generate ideas. We would also increase the number of agents per generation and tinker with the perturbation amount to delay convergence until the longer experiment ends. This data could be analyzed for gameplay findings and also be added to the neural network dataset.

Another avenue of exploration is increasing the complexity of our chosen fitness function. Currently, the best agents of a generation are mutated on to create children, but the actual best agent is not carried onto the next generation, a technique known as *elitist selection*. By including the best agents in addition to their children, we can provide stronger guarantees about performance improvement. This would also allow us to maintain their ELOs throughout a given experiment, providing more data to track the performance of given agents between generations.

One last opportunity would be to provide and extract more information from the neural network. Currently, our only input feature is the game state; adding additional input features to the neural network might be tested. Changing the target variable to predict attributes beyond the black win percentage may prove beneficial to NN performance.

[2] as it stands, we were able to play ~16 games in parallel at a time, completing, on average, 10 games in ~8 seconds

V. Appendix

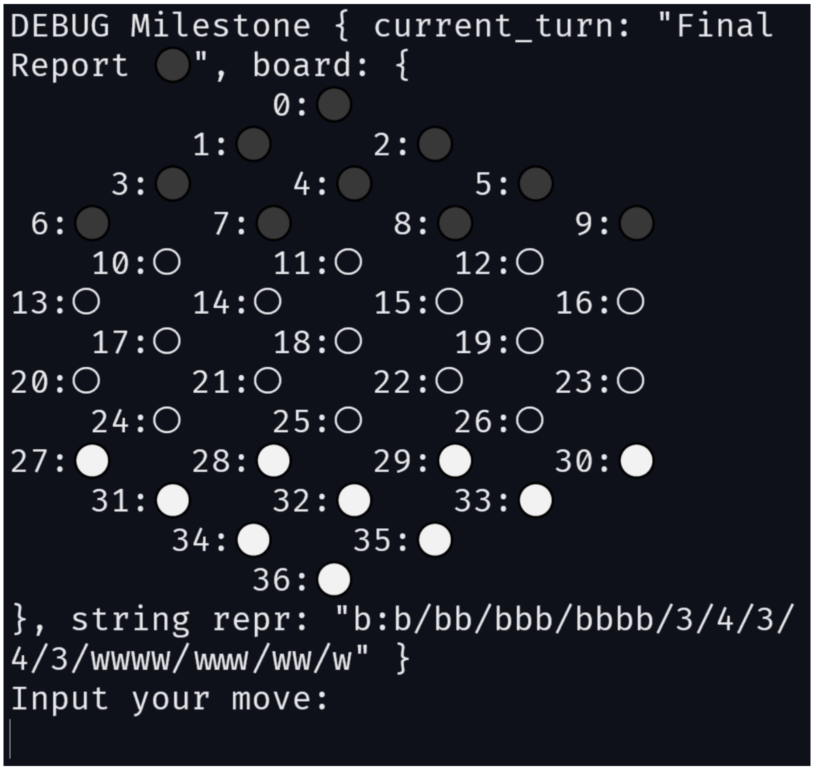
Heuristic	**Description**
Aggr. Pieces	The number of your pieces which have passed the other player's furthest piece i.e. pieces which can no longer impact the other player's furthest piece.
Aggr. Pieces Anti Centrality	An application of the anti centrality mapping to the aggr pieces heuristic.
Aggr. Pieces Middle Prox.	An application of the middle prox mapping to the aggr pieces heuristic.
Aggression Diff.	A mapping which assigns more value to your pieces which are closer to the other player's homespace.
Anti Centrality	An application of the anti centrality mapping to your pieces.
Attack In-Sync	The difference in how close your most aggressive piece in the middle line is to the other player's homespace compared to your most aggressive piece outside the middle line.
Centrality	An application of the centrality mapping to your pieces.
Defended Hexes	The number of empty hexes which your pieces are attacking.
Defended Hexes Middle Prox.	An application of the middle proximity mapping to the defended hexes heuristic.
Impt. Pieces	A mapping which assigns value to your pieces being in your homespace and the respective front-right and front-left hexes.
Limit Oppo. Moves	The number of possible moves you have.
Middle Line Diff.	The number of your pieces in the middle line.
Middle Prox.	An application of the middle proximity mapping to your pieces.
Passive Diff.	A mapping which assigns more value to your pieces which are closer to your homespace.
Piece Diff.	The number of your pieces remaining.
Straight Lines	The number of your pieces which have a friendly piece directly in front of them.
Straight Lines Middle Prox.	An application of the middle proximity mapping to the straight lines heuristic.
Undefended Pieces	The number of your pieces left hanging i.e. pieces which do not have a friendly piece directly behind them (excludes pieces with no hex behind them).
Undefended Pieces Middle Prox.	An application of the middle proximity mapping to the undefended pieces heuristic.

Table C: *Glossary of heuristics*[3].

[3] In our treatment of Milestone as a zero-sum game, all heuristics are measured as the difference of a black and white score i.e. one value is generated for remaining black pieces, another is generated for remaining white pieces and the evaluation is the difference. In this way, a good score for one player is an equivalently bad score for the other.

Location Mapping	Description
Centrality	A mapping which assigns more value to pieces closer to the center hex.
Anti Centrality	A mapping which assigns more value to pieces further from the center hex.
Middle Proximity	A mapping which assigns more value to pieces closer to the middle line.

Table D: *Glossary of location mappings used in calculating heuristic scores.*



```
DEBUG Milestone { current_turn: "Final
Report ⚫", board: {
             0:⚫
         1:⚫      2:⚫
     3:⚫      4:⚫      5:⚫
 6:⚫      7:⚫      8:⚫      9:⚫
    10:○     11:○     12:○
13:○     14:○     15:○     16:○
    17:○     18:○     19:○
20:○     21:○     22:○     23:○
    24:○     25:○     26:○
27:⚪     28:⚪     29:⚪     30:⚪
    31:⚪     32:⚪     33:⚪
        34:⚪     35:⚪
            36:⚪
}, string repr: "b:b/bb/bbb/bbbb/3/4/3/
4/3/wwww/www/ww/w" }
Input your move:
```

Figure 9: *Our CLI that outputs the current board state (as well as a string representation for debugging purposes) and requests a human player to input a move.*

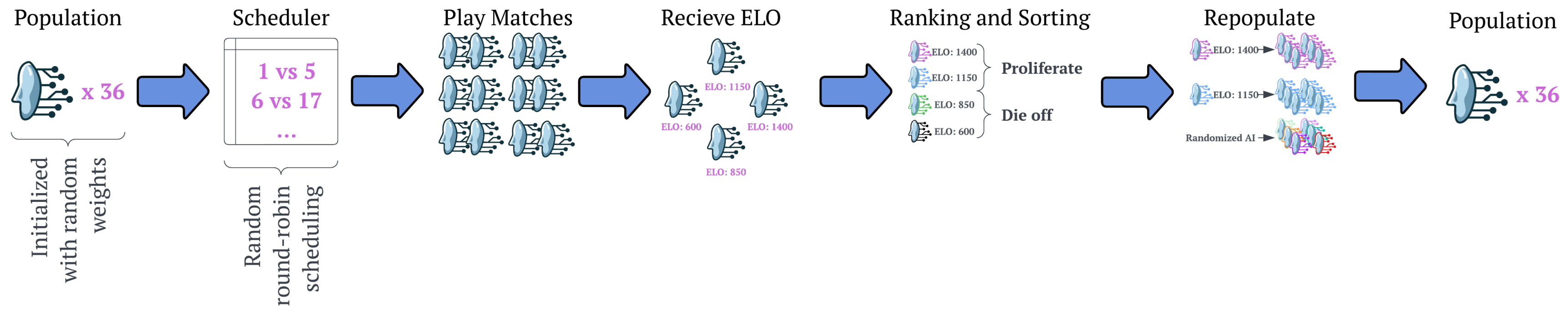


Figure 10: *Each generation undergoes a process consisting of match scheduling and execution; evaluation and ranking; and repopulation. This process is executed for each generation iteratively.*

Parameter	**Definition**	**Exp. A Value**	**Exp. B Value**
Number of generations	the total number of generations run in this experiment	100	100
Number of agents	the number of unique AI agents in each generation	36	36
Number of matches	the total number of (unique) matches that get played per generation	504	396
Retained agents	the number of best agents whose genes persist to the next generation	5	5
Children per agent	the number of agents each retained agent populates	5	3
Random agents	the number of agents introduced each round with random weights	11	21
Max perturbation	the randomness threshold applied to each heuristic of children agents	20%	35%
Perturbation decay	how much the perturbation decreases between rounds	3%	1.5%

Table E: *Between experiment A and experiment B, B had a larger focus on introducing randomness.*

Human Challenger	**Experience Level**	**Result Against AI**
Anisha Saini	Beginner	1-1
Claudia Arredondo Zayas	Beginner	0-2
Elizabeth Ehl	Beginner	0-2
Lavanya Pinnepalli	Beginner	0-2
Lucia Pannunzio	Beginner	0-2
Jeevan Deol	Intermediate	0-2
Ruchitha Rajaghatta	Intermediate	0-2
Samarth Desu	Intermediate	0-2
Sanjana Shanmugavel	Intermediate	1-1
Sylvia Bayrakdarian	Intermediate	1-1
Connor McCarthy	Advanced	2-0
Corban Chiu	Advanced	2-0
Jay Sangwan	Advanced	1-1
Sunny Chavan	Advanced	2-0
Tyler Carreja	Advanced	1-1

Table F: *To test our AI, we challenged people with a variety of skill levels.*