

Group 23

Karl Meakin & Ben Maxwell

Our approach

We used Alpha-Beta search guided by heuristics:

1. Difference between player's score and opponent's score
2. Difference between number of stones player can capture and number of stones opponent can capture
3. Difference between player's chaining potential (moves that allow the player to have a second turn) and opponent's chaining potential
4. Difference in seeds on each side (avoid starvation)

These heuristics are used to evaluate the board at the depth limit, and to decide the order to search child nodes.

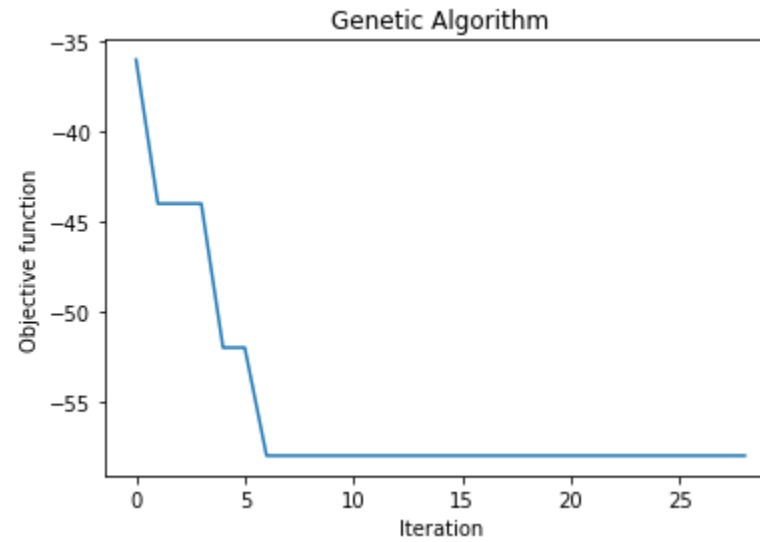
Unique aspects

Rust

We chose to implement our bot in Rust (a systems programming language with compile time memory safety checks) to guarantee high performance vs Java or Python.

Weights finding

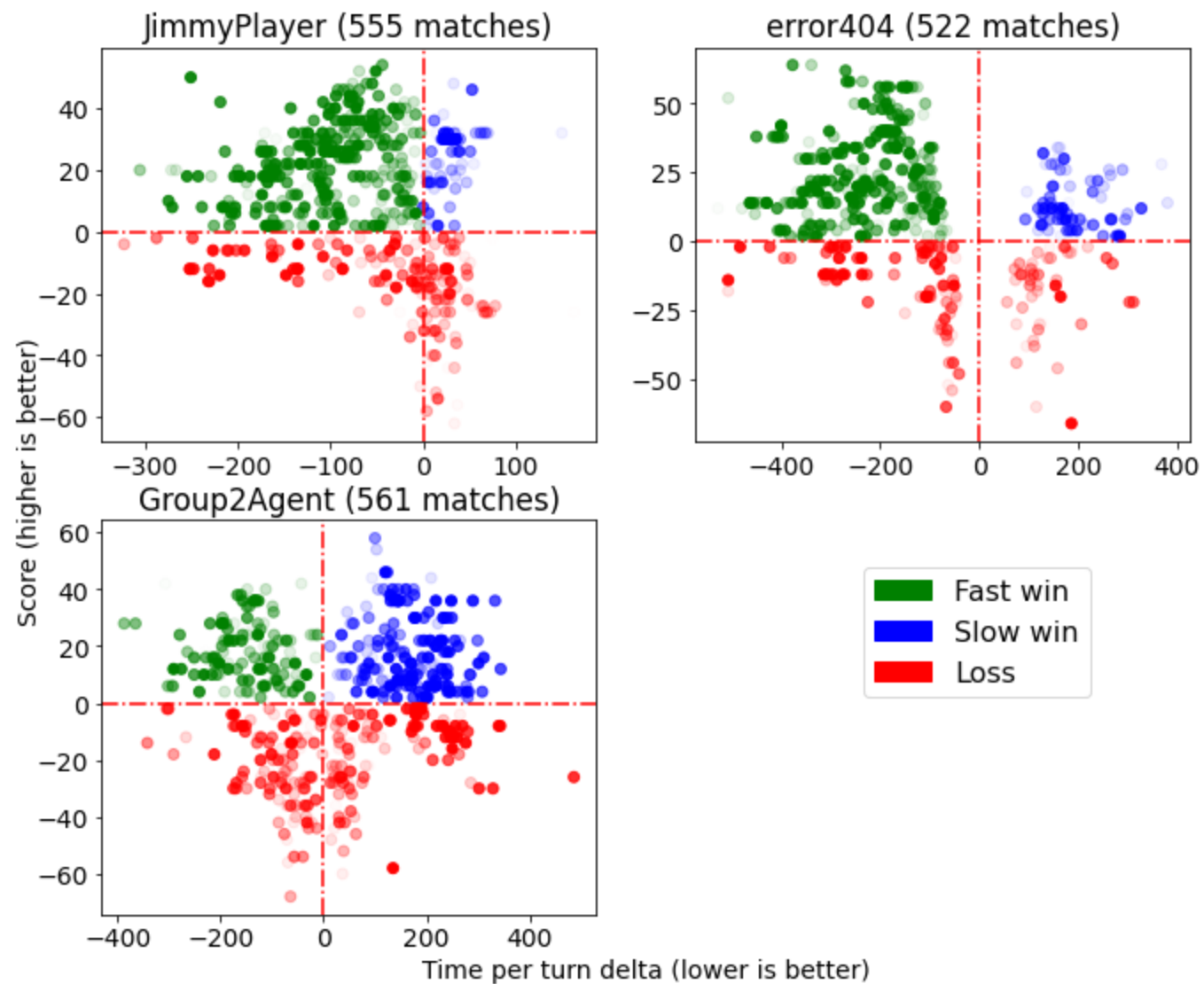
We used a genetic algorithm library to find weights for our heuristic to maximise win margin when playing against the provided test bots.



The best solution found:
[REDACTED]

Objective function:
-58.0

Performance



Performance

