

A Genetic Algorithm for Developing Strategy for Dobo

DOBO - Dobo is a game in which you have an $N \times M$ sized board, with each partition of the board containing an object, normally a stone. The game is played turn by turn, with two players. A player can remove a horizontal or vertical line of stones, where a line of stones consists of one or more contiguous stones. The player that must remove the last stone loses.

The Genetic Algorithm - A genetic algorithm will be composed to find a strategy to play DOBO. A game base will be composed of all possible game states, simplified by symmetry. This game base will be used in the genetic algorithm as follows: each game state will occupy a cell of an individual's property list. The index of the property will represent the row in the game base that is representative of that game state, the value inside of that property will be the game state to which the current game state correlates to. The game base will also have a final parameter in each row that does not directly correspond to the game state, being the "level" of that game state, as to help with integrating this concept into a genetic algorithm. The initial population must not be completely random, but maintain at least similar sections corresponding to the game state levels of the genetic code .

Fitness: The fitness operator will be a result of n games, being the amount of games lost divided by the amount of games played.

Selection: The selection operator will either be a binary tournament operator or be a stochastic uniform selection (SUS) operator along with a niching concept to maintain diversity of solutions in a GA population.

Crossover: Crossover will be fairly simple as we can just follow the standard meaning, selecting some traits (properties) from parent a and some from parent b.

Mutation: The Mutation operator must choose a valid game state accessible from the game state that the index of the property correlates to.

<http://www.egr.msu.edu/~kdeb/papers/k2007002.pdf>