



Metaheuristic Chess Artificial Intelligence

Maciej Borkowski
195968@student.pwr.edu.pl

Mariusz Waszczyński
xxxxxxx@student.pwr.edu.pl

Paweł Pałus
zzzzzz@student.pwr.edu.pl

Abstract

Index Terms: *chess, metaheuristics, artificial intelligence, ant colony, genetic, simulated annealing*

1 Introduction

Hello

2 Optimization Problem

The problem describes a standard game of chess, with a square board of 64 fields. Two players have to consecutively move a piece the board onto another field according to complex, well-defined rules. Our task is to find the series of movements in a game of chess that gives the best chance of winning the game in the end. The starting position of pieces can be arbitrary.

2.1 Mathematical model

From card, expand on it

3 Experimentation system

About the application

3.1 UCI

3.1.1 Firenzina

3.2 GUI

About the GUI

4 Algorithms

4.1 Ant colony

4.1.1 Idea

For ant colony search algorithm a mapping is created between a placement of chess pieces on a chess board and a list of possible moves the current player is able to do, when provided such board. Each move on this list is additionally annotated with a real value, which describes the fitness of the move. Moves with higher fitness ought to yield us better

results. Such mapping is called a pheromone and a set of them pheromones.

//pheromone graphic here

Ant is defined here as a chess player, that uses pheromones to choose when its it's time to make a choice of movement. Ant can be a part of a colony, in which case the colony provides the pheromones or it can be independent (used for Greedy Mode). Pheromones can be saved to and loaded from a file. When an ant finds itself on a board that has not yet been added to pheromones a new pheromone is created (possible moves for the board are computed and assigned equal real values).

Ant can work in one of two modes:

- Adventurous Mode

Used for learning. In this case the ant works for the betterment of its colony. It chooses moves randomly, according to the values of pheromones. This strategy improves the pheromones, by visiting a wide range of possible boards, which results in frequent updates and addition of new pheromones. // How to choose randomly

- Greedy Mode

Used for testing and real games. In this case the ant plays for the best end result in its game. Ant chooses a move from the pheromone with the highest value to choose the best move in each turn.

The process of learning consists of many iterations of ants in Adventurous Mode working as a colony. Each iteration amounts to a few phases:

1. Start new games and wait for them to end

Each ant plays one game of chess and remembers all boards it has run across, all moves it has chosen to do and the the cost function of the series of movements (value of cost function for last board).

2. Update pheromones

For each ant the pheromones connected to visited boards are updated by a fraction of the value of cost function of the whole series of movements. //Update formula here

3. Dissipate pheromones

Pheromone for each of the boards that has been visited at least once by any ant in this game is decreased by multiplying the value by a parameter //Dissipation formula here

Pheromones can be saved to a file. The file consists of a list of pheromones, each is described with two lines:

1. String representation of a board, left to right, bottom to up, where # means no chessman, upper case letters mean white chessmen and lower case letters mean black chessmen
2. A list of moves. Each move is described by five integer values. First two are the coordinates of chessman to move, third and fourth where to move the chessman to, the fifth is a special value used for promotion (when a pawn becomes another chess piece) and the sixth a real value of pheromone describing its effectiveness.

4.1.2 Experiments

4.1.3 Result

4.2 Genetic algorithm

4.2.1 what?

4.2.2 gui/experiment

4.2.3 result

4.3 Simulated Annealing

4.3.1 what?

4.3.2 gui/experiment

4.3.3 result

5 Conclusion

It was fun / not fun.

References

- [1] VECEK, N. ; CREPINSEK, M. ; MERNIK, M. ; HRNCIC, D., A comparison between different chess rating systems for ranking evolutionary algorithms
- [2] DORIGO, M. ; MANIEZZO, V. ; COLORNI, A., Ant system: optimization by a colony of cooperating agents
- [3] DAVID, O.E. ; VAN DEN HERIK, H.J. ; KOPPEL, M. ; NETANYAHU, N.S. , Genetic Algorithms for Evolving Computer Chess Programs
- [4] S. KIRKPATRICK; C. D. GELATT; M. P. VECCHI., Optimization by Simulated Annealing
- [5] R. HUBER, S. MEYER-KAHLEN, Universal Chess Interface, <http://www.shredderchess.com/chess-info/features/uci-universal-chess-interface.html>, 2015/06/01