Chess AI improvement through an evolutionary approach

Cédric Guillot

CPSC 565 - Winter 2013
cpguillo@ucalgary.ca

Abstract. The purpose of this project is to provide an enhanced strategy for chess by using an evolutionary approach. It would aim at improving a pre-existing strategy by tuning its parameters.

1 Introduction

AI in chess has been of a particular interest for a very long time. Even before the computer could be used, some algorithms had already come to life to challenge humans. For a long time and still today, the brute force method has been widely used in order to improve the chess AIs. But given the number of possible games, the brute force method can only be applied under a certain number of plies. That is why applying the evolutionary approach to that area may be beneficial and smarter. The vastness of the solution domain also makes it a nice candidate for that purpose.

In my project, I would like to start from a simple strategy and progressively improving it in order to beat a standard player (I will consider myself a standard player ie knowing how to play but no special training).

2 Related Work

Several article caught my attention and made me feel confident that such a project could be undertaken ([1], [2], [3]).

They all relied on a pre-existing strategy with parameters assigned to the different chess pieces. The algorithms they implemented provided better results than the original ones and they could challenge commercial software.

3 Project Details

The basic idea is to have a look at a given strategy and try to improve its parameters by using an evaluation function that gives us an idea of how favorable to us the chess plate currently is.

The core of my project would be to build a simple strategy inspired on the available strategies and then improve it by interation. The goal would be to beat a standard player by reaching a good enough strategy after a certain number of iterations.
If this is successful soon enough, I would go on and try to improve a more complexe solution that can be found with open source chess engines [4].

4 Software Tools

The chess engine Crafty [4] (written in C) will be used for performance purposes, as a lot of games will have to be played in order to compare two strategies.
Additionaly, I may use a couple of other engines as an inspiration for developing my original solution.

5 Time line

The timeline will be as follows :

1. Week 1 and 2 : getting familiar with Crafty and the strategies,
2. Week 3 and 4 : building my own strategy based on the previous work,
3. Week 5 and 6 : implementing the evolution algorithm and testing the result solution against the simple solution and a human.

References

1. Hallam Nasreddine, Hendra Suhanto Poh and Graham Kendall : Using an Evolutionary Algorithm for the Tuning of a Chess Evaluation Function Based on a Dynamic Boundary Strategy
http://red.cs.nott.ac.uk/ gxk/papers/ieeecis2006.pdf
2. DAVID B. FOGEL, FELLOW, IEEE, TIMOTHY J. HAYS, SARAH L. HAHN, AND JAMES QUON : A Self-Learning Evolutionary Chess Program
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1360168&tag=1
3. Graham Kendall, Glenn Whitwell : An Evolutionary Approach for the Tuning of a Chess Evaluation Function using Population Dynamics
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=934299
4. Crafty : http://www.craftychess.com/