Tuning Chess Evaluation Function Parameters using Differential Evolution Algorithm

Borko Bošković
University of Maribor, Faculty of Electrical Engineering and Computer Science
Smetanova 17, 2000 Maribor, Slovenia
E-mail: borko.boskovic@uni-mb.si
http://dkum.uni-mb.si/IzpisGradiva.php?id=14106

Thesis Summary

This article is an extended abstract of a doctoral dissertation on chess evaluation function tuning with differential evolution (DE) algorithm. DE is adopted for efficient chess evaluation function tuning, extended with an opposition-based optimization and a new history mechanism. Experimental results show that the algorithm is efficient and can be applied to the chess evaluation function tuning that has more or less knowledge.

Povzetek: Prispevek predstavlja razširjen povzetek doktorske disertacije s področja uglaševanja šahovske ocenitvene funkcije. Pri uglaševanju so uporabljeni algoritem diferencialne evolucije, mehanizem nasprotij in mehanizem zgodovine.

1   Introduction

Almost all chess knowledge of chess programs is defined in the chess evaluation function. This knowledge is presented with many arithmetic expressions and parameters. The problem is how to set the values of these parameters. With conventional approaches this becomes a very challenging task and, as such, requires a lot of time and work. To solve this problem, researchers used automated tuning or "learning". The most successful methods were temporal difference learning [1, 2] and evolutionary algorithms [3].

In this paper we present an approach which is based on differential evolution (DE) algorithm [4]. We adopted this algorithm for efficient chess evaluation function tuning. It was extended with an opposition-based optimization [6] and a new history mechanism [7].

2   Tuning algorithm

This section contains a short description of our tuning algorithm. First, it describes differential evolution algorithm as the basis of our algorithm. Then follows a description of an opposition-based optimization and the history mechanism which are integrated into our algorithm.

DE is a simple yet powerful evolutionary algorithm for global optimization and it has recently been used in a wide variety of real-world applications with impressive results [8]. This is the reason why we used DE for the chess evaluation function tuning. In our problem individuals are vectors of chess evaluation function parameters. The DE algorithm employs mutation and cross-over operations to generate new individuals and selection operation to select individuals that will survive into next generation. Before selection operation is employed, individuals have to be evaluated. In our case individuals are evaluated according to the games they have played. Therefore individuals play a specific number of games in each generation and individuals with greater efficiency survive into the next generation.

The opposition-based optimization was used because it improves efficiency of the DE for noise problems. The main idea of this optimization is to estimate a certain individual and its opposite individual at the same time, in order to achieve a better approximation for the candidate solution. The efficiency of the tuning process depends on the distance between the solution and the individuals of the initial population. Therefore, initialization operation generates opposite individuals according to the randomly initialized individuals. Then the selection operation is performed and selected individuals are probability closer to the solution which accelerates convergence. In the similar way this mechanism is with some probability performed in the rest of the evolutionary process [6, 5].

History mechanism is a new mechanism that enables potentially good individuals to remain within the evolutionary process and it reduces noise in the evaluation of those individuals. It also reduces the possibility of overfitting and it consequently improves the efficiency of the whole tuning process. This mechanism consists of two parts: history update and history injection. The first part adds a potentially good individuals into the history population or updates its

information about the efficiency. The second part injects individuals from the history population, according to the efficiency, back to the evolutionary process [7].

3 Experiment

With the proposed tuning approach, we tuned the chess evaluation function of BBChess chess program. Tuning was done with and without expert knowledge. When we tuned the parameters without expert knowledge, because the search space was huge, we tuned only a few parameters. After 500 generations of the evolutionary process, the value of parameters convergence to the values that relationship were approximately equal as known from the chess theory. When we tuned the parameter values with expert knowledge, the tuning intervals of parameter values were set around the approximate values and the number of tuned parameters was 190. The obtained results show that, our approach was successful. Efficiency of the tuning is dependent on the number of games that were played into the evolutionary process and on the defined tuning intervals. With larger tuning intervals obtained improvements were better but tuned individuals had smaller rating. We did some experiments without the history mechanism. The obtained results show that the history mechanism improves the tuning process by 155.2 rating points, on average.

4 Conclusion

This paper proposes an approach for the tuning of a chess evaluation function based on a DE algorithm, which includes an opposition-based optimizations and a new history mechanism. The tuned approach was tested and the obtained results show that our tuning approach was efficient and that the introduced history mechanism improves the efficiency of the tuning process.

References

[1] Baxter J, Tridgell A, Weaver L (1998) Experiments in Parameter Learning Using Temporal Differences, *International Computer Chess Association Journal* 21(2):84–99

[2] Baxter J, Tridgell A, Weaver L (2000) Learning to Play Chess Using Temporal Differences, *Machine Learning* 40(3):243–263

[3] Fogel DB (2006) *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 3rd edn. Wiley-IEEE Press

[4] Feoktistov V (2006) *Differential Evolution: In Search of Solutions (Springer Optimization and Its Applications)*. Springer-Verlag New York, Inc, Secaucus, NJ, USA

[5] Rahnamayan S, Tizhoosh HR, Salama MMA (2006) Opposition-Based Differential Evolution Algorithms. In: The 2006 IEEE Congress on Evolutionary Computation CEC 2006, pp 7363–7370

[6] Rahnamayan S, Tizhoosh HR, Salama MMA (2008) Opposition-Based Differential Evolution. *IEEE Transactions on Evolutionary Computation* 12(1):64–79

[7] Bošković B, Brest J, Zamuda A, Greiner S, Žumer V. History Mechanism Supported Differential Evolution for Chess Evaluation Function Tuning, *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, in press

[8] Chakraborty, Uday K. (Ed.) (2008) Advances in Differential Evolution, Studies in Computational Intelligence, Vol. 143, Springer