REVIEW

MIMICKING THE BLACK BOX
Genetically evolving evaluation functions and search algorithms

Genetic Algorithms Based Learning for Evolving Intelligent Organisms
Ph.D. thesis, Bar-Ilan University, Israel
Eli Omid David-Tabibi
Reviewed by Dap Hartmann

When you are interested in computer-chess research, and if, alphabetically, your name falls in between Charles Darwin and Richard Dawkins, you must be predestined to work on genetic algorithms (GA). That is the case for Eli Omid David-Tabibi, and *Genetic Algorithms Based Learning for Evolving Intelligent Organisms* is the title of the Ph.D. thesis he submitted to Bar-Ilan University in Israel. In his preface, David-Tabibi notes that 2009 (the year in which he obtained his Ph.D.) marks the 150th anniversary of the publication of *The Origin of Species*, and that he has just spent four years "employing Darwin's idea in an attempt to solve several of the most difficult problems in computer chess, which were considered as tough nuts to crack for over half a century." That certainly wetted my appetite, and so I plunged into this thesis with great expectations.

Since the first ideas about chess-playing computers were published, numerical values have been assigned to the relative worth (strength) of the chess pieces. That was not really a new notion, as many classic introductory chess books already did the same thing to teach beginning chess players the relative strengths (values) of the various chess pieces. For example, Euwe and Kramer in *The Middle Game, Vol.1* (1944) valued the pieces as follows: P=1, N=3.5, B=3.5, R=5.5, Q=10. In his seminal paper in *Philosophical Magazine* (1950), Claude Shannon suggested these values: P=1, N=3, B=3, R=5, Q=9, which corresponds to what Capablanca proposed in *Chess Fundamentals* (1921). Shannon also pointed out the importance of other factors besides the material balance, such as "Rooks should be placed on open files" and "Backward, isolated and doubled pawns are weak". He expressed some of these factors quantitatively (in terms of 'equivalent material value') so that they could be added to the evaluation function. A weak Pawn (backward, isolated or doubled) received a penalty of 0.5 (equivalent pawn value), and every legal move was worth a bonus of 0.1. On the rational choice of these values, Shannon commented: "The coefficients 0.5 and 0.1 are merely the writer's rough estimate."

With the emergence of chess-playing programs came the challenge of constructing an evaluation function in which non-material positional features such as king safety, pawn structure and mobility were given accurate weights. This made computers more 'aware' of the disadvantages of weak Pawns and the dangers of weak king safety, and stimulated more humanlike play in that material might be sacrificed in exchange for superior mobility or a strong attack on the opponent's King. The problem was how to determine these weights more accurately than "the writer's rough estimate". My only (minor) criticism on David-Tabibi's thesis is that it lacks an extensive overview of the relevant literature on this topic. For example, there is no mention of how the most famous chess machine to date (DEEP BLUE) tuned its evaluation function. From *Comparison Training of Chess Evaluation Functions* by Gerald Tesauro (Chapter 6 in *Machines that Learn to Play Games*, edited by J. Fürnkranz & M. Kubat, 2001) we learn that: "DEEP BLUE's evaluation function contains several thousand features, each of which had an associated weight […] representing a score in units of centipawns. Due to the large number of terms in the evaluation function, it was decided to focus tuning on a relatively small subset of features." The weights of DEEP BLUE's evaluation function were tuned using a dataset of grandmaster games. The crucial move Be4 in game 2 of the rematch with Kasparov in 1997 is attributed to the automated tuning of the king-safety weights: "with the original hand-tuned weights, DEEP BLUE plays Qb6 instead."

David-Tabibi applied genetic algorithms to determine the weights of the components in the evaluation function of MAESTRO, a clone of his chess program FALCON (rated at 2700+). MAESTRO and FALCON differ only in their evaluation functions: FALCON's evaluation function has more than 100 components, MAESTRO's has only 35. David-Tabibi created a set of test positions from a database of 10,000 games by grandmasters with a rating above 2600. From each game, (only) one position was picked at random. The 'correct answer' (move and evaluation score) for each position was supplied by FALCON on the basis of (merely) a 2-ply search. Next, the genetic optimization process tried to match these 'correct answers' over several generations of genetically evolving evaluation functions. David-Tabibi calls this approach 'Mentor-Assisted Learning': observing only the mentor's behaviour without any access to (knowledge of) the way in which the mentor operates. In effect, it is a way of reverse-engineering the more complex evaluation function by observing only its behavior (input and output).

Initially, the weights of the 35 features in the MAESTRO evaluation function were random numbers, rendering the program an absolute beginner without even the knowledge that a Queen is more valuable than a Pawn. But after 300 generations (each generation being tested for 'fitness' by measuring its performance on the test set against the evaluation of FALCON) this retarded evaluation function evolved into an excellent evaluation function. So excellent, in fact, that it outperforms the original MAESTRO evaluation function (with hand-picked weights). David-Tabibi provides ample evidence that is indeed the case. So what weights resulted from this mentor-assisted learning? Survival of the fittest taught MAESTRO these material values (renormalized to P=1): P=1, N=3.9, B=3.9, R=5.8, Q=11.5. The penalties for backward, isolated and doubled Pawns came to be 0.04, 0.12 and 0.37, respectively.

In a further refinement of his method, David-Tabibi used the expert judgement of human grandmasters (who played the games in his test set of 10,000 positions) to pre-tune the evaluation function. However, the grandmasters did not provide a quantitative measure (evaluation value) but only the best move to play in a given position. To overcome this problem, David-Tabibi developed a hybrid approach (more exactly, a two-stage approach that he calls 'coevolution') in which the genetic algorithm first optimizes for the choice of the best moves (as played by the grandmasters). The 50 best (fittest) generations are then further refined using the evaluation scores of a strong chess program.

David-Tabibi also applied this procedure to genetically evolve a set of 18 parameters associated with the search process, including four parameters related to null-move pruning and five parameters related to multi-cut pruning. The seed of this new evolutionary experiment was a clone of FALCON with randomized search parameters. The results of the best evolved organism, EVOL*, are impressive albeit not superior to the hand-tuned parameters used in FALCON. Nevertheless, as David-Tabibi describes it: "the evolved parameters of EVOL* perform on par with those of FALCON, which have been manually tuned and refined for the past eight years. Note that the performance of FALCON is by no means a theoretical upper bound for the performance of EVOL*, and the fact that the automatically evolved program matches the manually tuned one over many years of world championship level performance, is by itself a clear demonstration of the capabilities achieved due to the automatic evolution of search parameters."

A GA-based version of FALCON (genetically 'grown' from a retarded clone of the original FALCON) running on modest hardware, achieved second place in the 2008 World Computer Speed Chess Championship, and finished sixth in the 2008 World Computer Chess Championship, in a field dominated by the strongest computer-chess programs in the world (all running on superior hardware).

David-Tabibi believes that this Mentor-assisted learning approach to genetic algorithms can also be applied to other games, specifically to improve Diplomacy-playing programs (which, according to reviewers, currently exibit "terrible artificial intelligence") by using databases which contain thousands of Diplomacy games played by humans. Outside the scope of game-playing computer programs, internet search engines could be 'trained' to become as good as Google. Furthermore, he points out, that by using this method "it might be possible to evolve the successful mechanism without violating the patent (in cases where the patent protects the underlying 'mechanism', and not the resulting 'behavior')." In other words, the Mentor-assisted learning approach to genetic algorithms could emulate the behaviour of a patented decision-making process without infringing the patent. A computer program could be trained to invent around such patent.

This is wonderful thesis in many ways. The subject is fascinating, the approach is crystal clear, and the results are spectacular. The superb writing makes reading this thesis a joyous experience. And last, but certainly not least, I believe there is great potential in applying this method in other domains. A prime example of how computer chess is indeed the drosophila of artificial intelligence.

As far as I am aware, David-Tabibi's thesis has not been officially published. Maybe if you send him an email (mail@omiddavid.com) he will be kind enough to send you the PDF version. Highly recommended!