DIFFERENTIAL EVOLUTION: GLOBAL, EVOLUTION STRATEGY BASED OPTIMIZATION METHOD

This chapter introduces another evolutionary method – differential evolution – for predictive gene selection. The choice was dictated by the fact that differential evolution is one of the newest members in the family of evolutionary optimization methods.

Differential evolution is akin to the evolutionary algorithms in (Jirapech-Umpai & Aitken, 2005), (Deutsch, 2003). It is the tool for global optimization.

You can naturally suspect that finding a globally optimal solution can be much harder than getting a locally optimal solution. In many cases, the exact solution of the global optimization problem is impossible to find. In the context of combinatorial problems, such problems are called NP-hard, where the frightening ‘NP-hard’ stands for non-deterministic polynomial-time hard. By providing this meaning and

DOI: 10.4018/978-1-60960-557-5.ch015
halting here, let me avoid of venturing further into the broad and misterious realm of computational complexity theory.

For those of you who would like to know more about global optimization, I can recommend the book of Zhigljavsky and Žilinskas (Zhigljavsky & Žilinskas, 2008). The history and, what is more important, the development of differential evolution is well presented in (Feoktistov, 2006)². I will rely on this book when explaining the basic differential evolution algorithm, despite the fact that it was invented by Kenneth Price and Rainer Storn (Price & Storn, 1997), (Storn & Price, 1997). As another view of differential evolution, the book of these authors (Price, Storn, & Lampinen, 2005) is highly recommended in addition to (Feoktistov, 2006). I preferred the book of Feoktistov, because it is thin (yes, yes, I must confess that like many of you I like thin books) and contains MATLAB® and C code that turned to be extremely useful in understanding differential evolution. I, of course, do not want to say that the inventors of differential evolution left people starving: on the contrary, the inventors turned out to be polyglots as can be found at the following web page: http://www.icsi.berkeley.edu/~storn/code.html. This web page hosts code in C, Java, MATLAB, C++, Python, R and some other programming languages. In addition, there is an advice (under ‘Practical Advice’ title) on how to choose values for the parameters of differential evolution. Such guidance is invaluable as we all are aware of headache caused by tedious parameter tuning.

**Main Idea**

According to Wolfram MathWorld http://mathworld.wolfram.com/DifferentialEvolution.html, differential evolution is a stochastic parallel direct search based on evolution strategy optimization. It can handle non-differentiable, nonlinear and multimodal objective functions. It starts from random initialization of a population of individuals (vectors). At each iteration, called a generation, new vectors are obtained by combining vectors randomly chosen from the current population (mutation operation). These vectors are then mixed with a predetermined target vector to create a trial vector (recombination operation). Finally, the trial vector is accepted for the next generation if and only if it yields a reduction in the objective (fitness) function value (selection operation). Again, as with other evolutionary algorithms we considered, the value of the fitness function is the accuracy of a trained classifier, attained on the validation set.

As you can see, these operations resemble those of evolutionary algorithms considered earlier in our book. However, there is an important distinction that differential evolution owes its success to: the manner of the trial individual creation. I will talk more about this point below in text.