Benchmark for testing evolutionary algorithms

Csaba Barcsak, Karoly Jarmai

University of Miskolc, Hungary

Evolutionary algorithms are popular for their efficiency, and easy implementation, but these algorithms have drawbacks which we have to consider in order to create a good benchmark. We should find test problems which can bring up these disadvantages. The biggest disadvantage is that these algorithms select the better solutions in every iteration step knowing only the solutions in the previous iteration steps. Because of this it is hard to find the optimum in a noisy test function. The other disadvantage is that the algorithms can't test whether the solution is optimal or not. In some cases the algorithms find the local optimum instead of the global. We can put various exit criterion in the algorithms which can help achieving better results, but it doesn't solve the problem.

A modified PSO algorithm

PSO algorithm has a lot of variants. The efficiency of lots of techniques depends on the random nature of the process. We can work with multiple swarms which communicate, or we can use the Crazy Bird approach which sends some of the particles to random direction. We propose a method that doesn't need more function evaluations than the standard algorithm, and its efficiency doesn't depend on the initial state of the process. This method uses the gradient information in the previous sample points. Using the gradient information we can change the speed of the particles. If there are a lot of positive gradients in a particle's history, we raise its speed, because we conclude that the particle is moving towards the optimum point. This statement is true for a lot of objective functions, but not in all cases.

Benchmark

We have used a number of test problems in our benchmark. A lot of test problems and test results can be found in the literature, but the authors does not mention why they include a given test problem to their tests. The complexity of the test problems depends on the number and distribution of local optimums, and the number of variables. In the design of our benchmark we have taken into account the disadvantages of evolutionary algorithms. In order to create a useful benchmark we have to include some not too complex problems.

Composition test function

We proposed a method that can generate more complex functions from given basis functions. We described a general framework how to create an own method for function generation, which is useful for testing optimization algorithms.