# Differential evolution (DE)

## A very brief introduction

Radoslav Harman

FMFI UK Bratislava

Note 1: This text is used as a classroom handout. Please, do not cite it. If you need to cite a real paper on DE, see, e.g., https://doi.org/10.1016/j.swevo.2016.01.004 and the references summarized in the survey paper.

Note 2: It may help if the student is familiar with the principles of the simulated annealing algorithm (SAA), the genetic algorithm (GA), and the particle swarm optimization algorithm (PSOA).

As usual, we will consider a real-valued *objective* (or *fitness*) *function* f defined on a set X of *feasible solutions*. For the standard version of DE, X is required to be the space of m-dimensional vectors composed of real numbers. The reason of this requirement is that the fundamental principle of DE is based on shifting feasible solutions by vectors. However, we do not need to assume anything about the properties of f. In this text, we assume the aim is to *minimize* f.

The classical version of DE has 3 parameters: DE is a population metaheuristic, and its first parameter is the population size $Np \geq 4$. The second parameter is F ($0 \leq F \leq 2$), which can be viewed as a mutation intensity parameter and is usually called *differential weight* in the context of DE. We also need a third parameter CR ($0 \leq CR \leq 1$), which is called the *crossover probability*.

The initial population is usually generated at random over a box inside X. Given the actual population $\mathbf{x}_1, \ldots, \mathbf{x}_{Np}$, DE forms the new population as follows:

For each i=1, … ,Np: randomly generate a *base vector* **a** and two *shift vectors* **b**, **c**. The vectors **a**, **b**, **c** must all be members of the current population and all four vectors $\mathbf{x}_i$, **a**, **b**, **c**, must be distinct. Use **a**, **b**, **c** to form the *donor* vector

$\mathbf{y} = \mathbf{a} + F * (\mathbf{b} - \mathbf{c})$,

which can be viewed as a *mutation* of **a**; note that this mutation is implicitly respecting the size and shape of the population. Next, compute a *challenger* **z** via the *uniform* (or *binomial*) *crossover* of $\mathbf{x}_i$ and **y**, that is, each component of **z** is independently *inherited* from $\mathbf{x}_i$ with probability 1-CR and from **y** with probability CR. To ensure that **z** is not the same as $\mathbf{x}_i$, one random component of **y** is assigned to **z** with probability 1. Replace $\mathbf{x}_i$ with **z** if $f(\mathbf{z})<=f(\mathbf{x}_i)$; in the opposite case, discard **z** as a failed attempt to improve the position of $\mathbf{x}_i$.

DE exhibits some similarities to other heuristic optimization methods.

SAA:

DE resembles the SAA in the sense that both algorithms generate candidate solutions that may or may not be accepted, depending on the relation of the objective function values of the current feasible solution and a “candidate” solution. However, DE “greedily” accepts an improving solution, while the SAA uses a subtler principle of acceptance. Also, DE modifies the entire population of feasible solutions, while in the SAA we only have a single feasible solution at each iteration.

PSOA:

DE is like the PSOA in the sense that both algorithms can be viewed as sequential vector shifts of the positions of each member of the population. However, in DE the rules for shifts are simpler and substantially different than in the PSOA; for instance, DE does not utilize the historically best positions of the particles but the actual positions of randomly selected members, as well as the actual size and shape of the entire population. (Note that some versions of DE *do* utilize the best of all available feasible solutions.)

GA:

As for similarities with the GA, DE uses operations that can be viewed as mutation and recombination. However, the mutation in DE adaptively changes with the properties of the population (this is sometimes called *self-referential mutation*), which is not usual in GAs. Moreover, DE first performs the mutation of a base vector, and then recombines the mutated base vector with the feasible solution that we consider for update. In a sense, this is the opposite order compared to the GAs.

The behaviour of DE strongly depends on the choice of Np, F and CR. The usual initial choices are Np=10*n, F=0.5 and CR=0.5, but it often helps to run the optimization with various combinations of the parameters, because some combinations can be much more effective than others, and the best combination of parameters may depend on the problem.