Differential Evolution with Competitive Control-parameter Setting

Problem Specification

The global optimization problem with box constraints is formed as follows:
For a given objective function $f : D \rightarrow \mathbb{R},\ D \subset \mathbb{R}^d$
the point $\mathbf{x}^*$ is to be found such that $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in D} f(\boldsymbol{x})$.
The point $\boldsymbol{x}^*$ is called the global minimum, D is the search space defined as $d-$dimensional box, $D = \prod_{i=1}^{d}[a_i, b_i]$, $a_i < b_i$, $i = 1, 2, \ldots, d$.

The problem of the global optimization is hard and plenty of stochastic algorithms were proposed for its solution, see e.g. [1], [5]. The authors of many such stochastic algorithms claim the efficiency and the reliability of searching for the global minimum. The reliability means that the point with minimal function value found in the search process is sufficiently close to the global minimum point and the efficiency means that the algorithm finds a point sufficiently close to the global minimum point at reasonable time. However, when we use such algorithms, we face the problem of the setting their control parameters. The efficiency and the reliability of many algorithms is strongly dependent on the values of control parameters. Recommendations given by authors are often vague or uncertain. A user is supposed to be able to change the parameter values according to the results of trial-and-error preliminary experiments with the search process. Such attempt is not acceptable in tasks, where the global optimization is one step on the way to the solution of the user's problem or when the user has no experience in fine art of control parameter tuning. Adaptive robust algorithms reliable enough at reasonable time-consumption without the necessity of fine tuning their input parameters have been studied in recent years.

The differential evolution (DE) [6] has become one of the most popular algorithms for the continuous global optimization problems in last decade years, see [3]. But it is known that the efficiency of the search for the global minimum is very sensitive to the setting of its control parameters. That is why the self-adaptive DE was proposed and included into this library.

Description of Implemented Algorithms

The differential evolution (DE) works with two population P and Q of the same size N. A new trial point $\boldsymbol{y}$ is composed of the current point $\boldsymbol{x_i}$ of old population and the point $\boldsymbol{u}$ obtained by using mutation. If $f(\boldsymbol{y}) < f(\boldsymbol{x_i})$ the point $\boldsymbol{y}$ is inserted into the new population Q instead of $\boldsymbol{x_i}$. After completion of the new population Q the old population P is replaced by Q and the search continues until stopping condition is fulfilled. The DE algorithm can be written as follows:

generate $P = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$; ($N$ points in D)
repeat
  for $i := 1$ **to** N **do**
    compute a mutant vector $\boldsymbol{u}$;
    create $\boldsymbol{y}$ by the crossover of $\boldsymbol{u}$ and $\boldsymbol{x}_i$;
    if $f(\boldsymbol{y}) < f(\boldsymbol{x}_i)$ **then** insert $\boldsymbol{y}$ into Q
      else insert $\boldsymbol{x}_i$ into Q
    endif;
  endfor;
  $P := Q$;
until stopping condition;

There are several variants how to generate the mutant point $\boldsymbol{u}$. One of the most popular (called *rand*) generates the point $\boldsymbol{u}$ by adding the weighted difference of two points

$$\boldsymbol{u} = \boldsymbol{r}_1 + F\,(\boldsymbol{r}_2 - \boldsymbol{r}_3)\,, \tag{1}$$

where $\boldsymbol{r}_1, \boldsymbol{r}_2$ and $\boldsymbol{r}_3$ are three distinct points taken randomly from P (not coinciding with the current $\boldsymbol{x}_i$) and $F > 0$ is an input parameter. Another variant called *best* generates the point $\boldsymbol{u}$ according to formula

$$\boldsymbol{u} = \boldsymbol{x}_{\min} + F\,(\boldsymbol{r}_1 + \boldsymbol{r}_2 - \boldsymbol{r}_3 - \boldsymbol{r}_4)\,, \tag{2}$$

where $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4$ are four distinct points taken randomly from P (not coinciding with the current $\boldsymbol{x}_i$), $\boldsymbol{x}_{\min}$ is the point of P with minimal function value, and $F > 0$ is an input parameter.

The elements y_j, $j = 1, 2. \ldots, d$ of trial point $\boldsymbol{y}$ are built up by the crossover of its parents $\boldsymbol{x}_i$ and $\boldsymbol{u}$ using the following rule

$$y_j = \begin{cases} u_j & \text{if} \quad U_j \le C \quad \text{or} \quad j = l \\ x_{ij} & \text{if} \quad U_j > C \quad \text{and} \quad j \ne l\,, \end{cases} \tag{3}$$

where l is a randomly chosen integer from $\{1, 2, \ldots, d\}$, $U_1, U_2, \ldots, U_d$ are independent random variables uniformly distributed in $[0,\ 1)$, and $C \in [0, 1]$ is an input parameter influencing the number of elements to be exchanged by crossover. Eq. (3) ensures that at least one element of $\boldsymbol{x}_i$ is changed even if $C = 0$.

Several papers deal with the setting of control parameters for differential evolution. Recent state of adaptive parameter control in differential evolution is summarized by Liu and Lampinen [4] and Brest et al.[2].

The setting of the control parameters can be made adaptive trough the implementation of a competition into the algorithm. This idea is similar to the competition of local-search heuristics in evolutionary algorithm [7] or in controlled random search [8]. The competitive control-parameters setting in DE is described in [9], where the numerical comparison with other stochastic algorithms is also presented.

Let us have H settings (different values of F and C used in the statements on line 4 and 5 of Algorithm 1) and choose among them at random with the probability $q_h,\ h = 1, 2, \ldots, H$. The probabilities can be changed according to the success rate of the setting in preceding steps of search process. The h-th setting is successful if it generates such a trial point $\mathbf{y}$ that $f(\mathbf{y}) < f(\mathbf{x}_i)$. When n_h is the current number of the h-th setting successes, the probability q_h can be evaluated simply as the relative frequency

$$q_h = \frac{n_h + n_0}{\sum_{j=1}^{H}(n_j + n_0)} \ , \tag{4}$$

where $n_0 > 0$ is a constant. The setting of $n_0 \geq 1$ prevents a dramatic change in q_h by one random successful use of the h-th parameter setting. In order to avoid the degeneration of process the current values of q_h are reset to their starting values ($q_h = 1/H$) if any probability q_h decreases bellow a given limit $\delta > 0$. The competition provides an self-adaptive mechanism of setting control parameter appropriate to the problem actually solved.

Two most reliable variants of such competitive differential evolution are included into this library. Three values of control parameter C were used in both variants, namely $C = 0$, $C = 0.5$, and $C = 1$.

- DER9 – the mutant vector $\boldsymbol{u}$ is generated according to (1), nine settings of control parameters are all the combinations of three F-values ($F = 0.5$, $F = 0.8$, and $F = 1$) with three values of C given above,
- DEBR18 – 18 settings, aggregation of nine settings used in DER9 and nine setting with the same values of and F and C, but the mutant vector $\boldsymbol{u}$ is generated according to (2).

The search for the global minimum was stopped if $f_{\max} - f_{\min} < my\_eps$ or the number of objective function evaluations exceeds the input upper limit $max\_evals \times d$.

The algorithms were tested on six functions and three levels of the search space dimension. The values of parameters controlling the stopping condition and the competition of the settings used in all the test tasks [9] are also used as default values. They are set up as follows:

- For stopping condition – $my\_eps = 1E-07$, $max\_evals = 20000$
- For competition control – $n_0 = 2$, $\delta = 1/(5\,H)$
- Population size – $N = \max(20, 2\,d)$

In the test tasks DEBR18 and DER9 outperformed standard DE significantly both in the reliability of finding a solution sufficiently close to the global minimum point and in the convergence rate. The reliability of DEBR18 was higher, but not significantly different from the reliability of DER9. DEBR18 worked significantly faster in the case of the most time-consuming Rosenbrock function. The proposed competitive setting of the control parameters F and C proved to be an useful tool for self-adaptation of differential evolution, which can help to solve the global optimization tasks without necessity of fine control parameter tuning.

References

[1] Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York (1996)

[2] Brest, J., Greimer, S., Boškovič, B., Mernik, M., Žumer, V.: Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation (to appear)

[3] Lampinen, J.: A Bibliography of Differential Evolution Algorithm. Technical Report. Lappeenranta University of Technology, Department of Information Technology. http://www.lut.fi/~jlampine/debiblio.htm (2002)

[4] Liu, J., Lampinen, J.: A Fuzzy Adaptive Differential Evolution Algortithm. Soft Computing **9** (2005) 448 – 462

[5] Spall, J. C.: Introduction to Stochastic Search and Optimization, Wiley-Intersience (2003)

[6] Storn, R., Price, K.: Differential evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. J. Global Optimization **11** (1997) 341 – 359

[7] Tvrdík, J., Mišík, L., Křivý, I.: Competing Heuristics in Evolutionary Algorithms. 2nd Euro-ISCI, Intelligent Technologies - Theory and Applications (Sinčák P. et al. eds.), IOS Press, Amsterdam (2002) 159 – 165

[8] Tvrdík, J.: Generalized controlled random search and competing heuristics. MENDEL 2004, 10th International Conference on Soft Computing, (Matoušek R. and Ošmera P. eds). University of Technology, Brno (2004) 228 – 233

[9] Tvrdík, J.: Competitive Differential Evolution. MENDEL 2006 , 12th International Conference on Soft Computing (Matoušek R. and Ošmera P. eds). University of Technology, Brno (2006) 7 – 12