Fundamental Evolutionary Algorithm for Real Domain Optimization

KHOA TRAN, University of Information Technology, Vietnam and Vietnam National University, Vietnam

The problem of this report is about different evolutionary algorithms to solve **the single-objective optimization problem** (minimization specifically) for some benchmark functions in **real domain variable**, formulated as:

$$f : \mathbb{R}^D \to \mathbb{R} \tag{1}$$

$$\arg\min_{x} {}_x f(x) \quad \text{subject to} \quad x_i \in [B^i_{\min}, B^i_{\max}], \quad \forall i = 1, \ldots, D \tag{2}$$

where $B^i_{\min}$ and $B^i_{\max}$ are the lower and upper bounds of the variable x in the ith dimension.

The code of these algorithm can be found

1 GENETIC ALGORITHM (GA)

Algorithm 1 Genetic Algorithm with Tournament Selection

Input: f (the function to be minimized), D (dimension of the variable), B (the bounds of each dimension), C_r (the probability to mutate a candidate), N (Population size - **even number**),
Output: $\mathbf{x}$, $f(\mathbf{x})$ (The solution of problem)
Initialize population $X \leftarrow \{x_0, x_1, x_2, ..., x_{N-1}\}$
while stop condition not met **do**
    $P \leftarrow X$
    while $i = 0 \rightarrow 1$ **do**
        $p_1, p_2 \leftarrow \text{RandomChoice}(X)$
        $o_1, o_2 \leftarrow \text{Crossover}(p_1, p_2)$
        if Random(0, 1) < C_r **then**
            $o_1 \leftarrow \text{Mutate}(o_1)$
        end if
        if Random(0, 1) < C_r **then**
            $o_2 \leftarrow \text{Mutate}(o_2)$
        end if
        $o_1, \leftarrow \text{clip}(o_1, B)$
        $o_2, \leftarrow \text{clip}(o_2, B)$
        $P = P \cup \{o_1, o_2\}$
    end while
    $X \leftarrow \text{TournamentSelection}(P, N, f)$
end while
$\mathbf{x} = \arg\min_x(f(X))$
return $\mathbf{x}$, $f(\mathbf{x})$

Author's address: Khoa Tran, University of Information Technology, Ho Chi Minh City, Vietnam and Vietnam National University, Ho Chi Minh City, Vietnam, 22520691@gm.uit.edu.vn.

Algorithm 2 Tournament Selection

Input: P (the pool of candidates), N (population size), f (the function to be minimized), D (dimension of the variable)
Output: X **(The selected population)**
$X \leftarrow \{\}$
while $i = 0 \rightarrow 1$ **do**
    $P \leftarrow \text{Shuffle}(P)$
    while $j = 0 \rightarrow 2N$ **do**
        $x \leftarrow \arg\min_x \{f(P_k) \mid P_k \in P[0:4]\}$
        $X \leftarrow X \cup x$
        $j \leftarrow j + 4$
    end while
end while
return X

2 DIFFERENCIAL EVOLUTIONARY (DE)

Algorithm 3 Differencial Evolution

Input: f (the function to be minimized), D (dimension of the variable), B (the bounds of each dimension), C_r (the probability to crossover a candidate), N (Population size - **even number**), F (Scaling Factor)
Output: $\mathbf{x}$, $f(\mathbf{x})$ (The solution of problem)
Initialize population X $\{x_0, x_1, x_2, ..., x_{N-1}\}$
while stop condition not met **do**
    while $i \rightarrow N$ **do**
        $r_1, r_2, r_3 \leftarrow \text{RandomChoice}(X)$
        $v \leftarrow x_1 + F(x_2 - x_3)$
        $v \leftarrow \text{clip}(v, B)$
        $j \leftarrow \text{Random}(0, D)$
        $u \leftarrow X[i]$
        while $k \rightarrow D$ **do**
            if $\text{Random}(0, 1) < C_r$ or $k = j$ **then**
                $u_{(k)} = v_{(k)}$
            end if
        end while
    end while
end while

3 PARTICAL SWARM OPTIMIZATION (PSO)

Algorithm 4 Partical Swarm Optimization

1: **Input**: f (the function to be minimized), D (dimension of the variable), B (the bounds of each dimension), C_r (the probability to crossover a candidate), N (Population size - **even number**)
2: **Output:** $\mathbf{x}$, $f(\mathbf{x})$ (The solution of problem)
3: Initialize population X $\{x_0, x_1, x_2, ..., x_{N-1}\}$
4: Initialize velocity V $\{v_0, v_1, v_2, ..., v_{N-1}\}$
5: $w \leftarrow 0.5, c_1 \leftarrow 1.5, c_2 \leftarrow 1.5$
6: $P^* \leftarrow X, p^* \leftarrow \arg\min_x(f(X))$
7: **while** stop condition not met **do**
8: $\quad r_1, r_2 \leftarrow \text{RandomChoice}(X)$
9: $\quad$ **while** $i \rightarrow N-1$ **do**
10: $\quad\quad V[i] \leftarrow wV[i] + c_1r_1(P^*[i] - X[i]) + c_2r_2(p^* - X[i])$
11: $\quad$ **end while**
12: $\quad X \leftarrow \text{clip}(X + V, B)$
13: $\quad$ **while** $i \rightarrow N-1$ **do**
14: $\quad\quad$ **if** $f(X[i]) < f(P[i])$ **then**
15: $\quad\quad\quad P^*[i] = X[i]$
16: $\quad\quad$ **end if**
17: $\quad$ **end while**
18: $\quad$ **if** $\min(f(X)) < f(p^*)$ **then**
19: $\quad\quad p^* \leftarrow \arg\min_x f(X)$
20: $\quad$ **end if**
21: **end while**
22: **return** $p^*, f(p^*)$