Evolutionary computation

Tasks 2024/2025

3 Implementation of an Evolutionary Algorithm

3.1 Evolutionary Algorithm: Differential Evolution

In this task, you will implement the Differential Evolution algorithm (Differential Evolution - DE). DE is a population-based optimization algorithm designed to find the minimum or maximum of a function by iteratively improving candidate solutions. It is known for its simplicity and effectiveness in solving complex optimization problems in large solution spaces. The pseudocode for the DE algorithm is as follows:

Algorithm 1 Pseudocode of Differential evolution with rand/1/bin strategy.

Input: $Problem$: contains the fitness function to optimize
Input: $maxFes$: maximum number of evaluations
Output: $bestSolution$: the best solution found
Define parameters $NP \geq 4$, $CR \in [0, 1]$, and $F \in [0, 2]$
Initialize population $x_i (i = 1, 2, ..., NP)$ with random positions in the search space
Evaluate all solutions
while termination criteria is not met **do**
    for each solution x_i **do**
        Pick random solutions $a \neq b \neq c \neq x_i$
        Create mutant vector v: $v_j = a_j + F \cdot (b_j - c_j)$ for each dimension j
        Select random index $R \in \{1, \ldots, D\}$, where D is the number of dimensions
        for $j \leftarrow 1$ **to** D **do**
            if $rand() < CR \parallel j == R$ **then** ▷ rand() generates a random number in $[0, 1]$
                $y_j = v_j$
            else
                $y_j = x_{ij}$
            end if
        end for
        if $f(y) \leq f(x_i)$ **then**
            Replace solution x_i with trial solution y
        end if
    end for
end while
Return the best solution from population

The algorithm starts with a population of candidate solutions (of size NP), where each individual represents a possible solution to the problem. The main steps of the DE algorithm are mutation, crossover, and selection:

- **Mutation**: For each current solution (x_i) in the population, a mutated vector (v) is created by combining randomly selected solutions from the population (a, b, and c). This is done by taking a weighted difference of two vectors and adding it to a third vector. The parameter that determines the size of the mutation step is called the mutation factor (F).
- **Crossover**: Based on the mutated vector and the current solution, a trial vector (y) is generated. The crossover rate (CR) parameter determines the probability that a component of the vector comes from the mutated vector.
- **Selection**: The trial vector is evaluated, and if it has a better objective function value than the current solution, it replaces the current solution in the population. In this way, the population gradually moves toward better solutions.

The algorithm repeats these steps until the termination criterion is met, usually the maximum number of evaluations (maxFes). In the pseudocode, solutions a, b, c, and x_i represent the parents,

while the trial vector y represents the offspring. The DE algorithm offers multiple strategies for mutation and crossover. You will implement the **rand/1/bin** strategy, which is one of the most commonly used. The notation **rand** indicates that three different vectors are randomly selected from the population for mutation. The number **1** specifies the number of differential vectors used in the mutation process. In this strategy, a single differential vector is created by taking the difference between two selected vectors. The notation **bin** refers to binomial crossover, where the components of the mutated vector and the current vector are combined based on the crossover rate (CR). For each component, a random value is generated to determine whether the component of the trial vector comes from the mutated vector or from the current solution.

3.1.1 Experiment

As in the previous task, you will run the algorithm multiple times on all problems (except Bukin and Carrom Table) on 10, 20 and 30 dimensions.

To ensure a fair comparison, all of you will use the same parameters for each run:

- Number of runs: 50
- Population size (NP): 20
- Crossover rate (CR): 0.5
- Mutation factor (F): 0.6
- maxFes: 3000 * d

For each problem prepare a text file (**.txt**) containing the best results from all 50 repetitions. Each result from a single run should be on its own line (the file will have 50 lines). The file name should be formatted as follows: DE-(your surname without special characters)_problem name with the number of dimensions. Example file name: **DE-Ravber_SphereD10.txt** (the complete list of file names can be found in the appendix at the bottom). Make sure to use a decimal point instead of a comma, and do not include invalid characters (e.g., infinity, NaN, . . .) or any other unnecessary strings in the file! It is crucial that all file names follow the same naming convention! Submit all .txt files in the folder **results**. **The submission deadline for the files is 3 days before the task deadline!**

Example of file content:

```
0.00000007624825970609
0.00000011409994682410
0.00000007894761067107
0.00000007649058186843
0.00000012118417513918
0.00000024030007628184
0.00000020355454166676
0.00000014964383510119
...
0.00000008033081844161
```

---

* Deadline: **December 19, 2024**.
* The task is **mandatory** and worth **25 points**.

Appendix: list of file names for the DE algorithm

```
DE-Ravber_SphereD10.txt
DE-Ravber_SphereD20.txt
DE-Ravber_SphereD30.txt
DE-Ravber_AckleyD10.txt
DE-Ravber_AckleyD20.txt
DE-Ravber_AckleyD30.txt
```

DE-Ravber_GriewankD10.txt
DE-Ravber_GriewankD20.txt
DE-Ravber_GriewankD30.txt
DE-Ravber_RastriginD10.txt
DE-Ravber_RastriginD20.txt
DE-Ravber_RastriginD30.txt
DE-Ravber_Schwefel26D10.txt
DE-Ravber_Schwefel26D20.txt
DE-Ravber_Schwefel26D30.txt
DE-Ravber_RosenbrockD10.txt
DE-Ravber_RosenbrockD20.txt
DE-Ravber_RosenbrockD30.txt
DE-Ravber_TridD10.txt
DE-Ravber_TridD20.txt
DE-Ravber_TridD30.txt
DE-Ravber_StyblinskiTangD10.txt
DE-Ravber_StyblinskiTangD20.txt
DE-Ravber_StyblinskiTangD30.txt
DE-Ravber_LevyD10.txt
DE-Ravber_LevyD20.txt
DE-Ravber_LevyD30.txt
DE-Ravber_MichalewiczD10.txt
DE-Ravber_MichalewiczD20.txt
DE-Ravber_MichalewiczD30.txt