Evolutionary Computing (2023)

Assignment (3):

Multi-dimensional function optimization

Shiraz University

Due date: 22/December (2023)

1 Introduction

Function optimization refers to the process of finding the maximum or minimum value of a mathematical function. In the context of this assignment, optimization involves using evolutionary algorithms to find the optimal solution(s) for a given objective function.

In this assignment you'll have to implement an Evolutionary Algorithm (EA) with adaptive mutation step control, in fact an evolutionary strategy (ES), to solve a classic multi-dimensional function optimization problem.

2 Problem statement

The problem you are attempting to solve is using an EA with adaptive mutation step control, in fact an evolutionary strategy (ES), for each dimension to find the global minimum of the given objective functions. Refer to the lecture slides for a more comprehensive understanding of ES.

2.1 Schwefel function

The generalized Schwefel function is defined as follows:

$$f(x) = \alpha n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$$

$$\alpha = 418.9829$$

Dimensions: n

The Schwefel function is complex, with many local minima. The plot shows the two-dimensional form of the function. Its dimensionality is controlled by the parameter n.

Input Domain:

The function is usually evaluated on the hypercube $x_i \in [-500, 500]$, for all $i = 1, \ldots, n$.

The global minimum: $f(x^*) = 0, at\ x^* = (420.9687, \ldots, 420.9687)$

Figure 1 illustrates the generalized Schwefel function for $n = 2$.

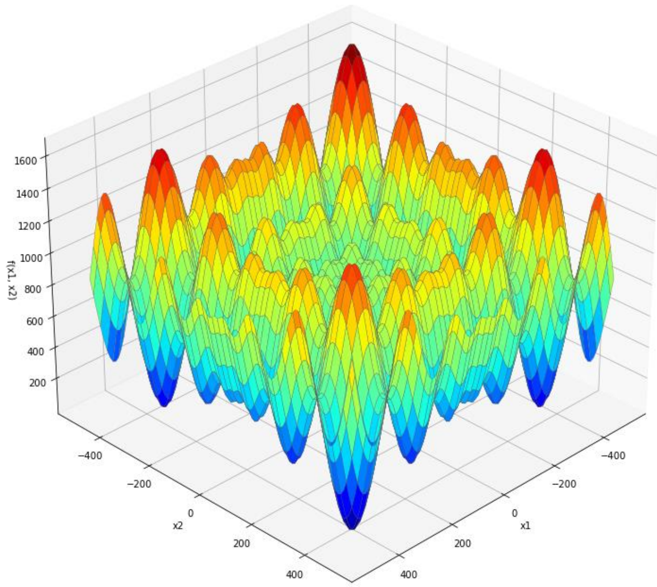

Figure 1: Generalized Schwefel function for n = 2

2.2 Ackley function

The Ackley function is defined as follows:

$$f(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos\left(cx_i\right)\right) + a + \exp\left(1\right)$$

Dimensions: n

The Ackley function is widely used for testing optimization algorithms. In its two-dimensional form, as shown in the Figure 2, it is characterized by a nearly flat outer region, and a large hole at the center. The function poses a risk for optimization algorithms, particularly hill climbing algorithms, to be trapped in one of its many local minima.

Recommended variable values are: a = 20, b = 0.2 and c = 2π.

Input Domain:

The function is usually evaluated on the hypercube $x_i \in [-32.768, 32.768]$, for all $i = 1, \dots, n$, although it may also be restricted to a smaller domain.

The global minimum: $f(x^*) = 0, at\ x^* = (0, \dots, 0)$

Figure 2 illustrates the Ackley function for n = 2.

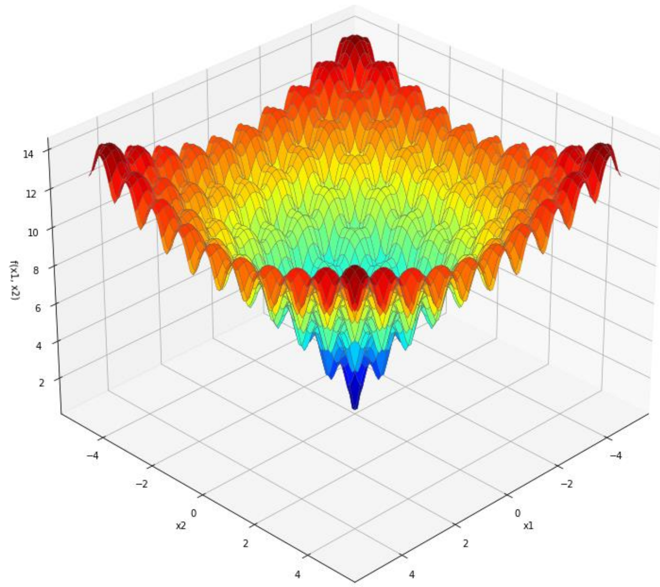

Figure 2: Ackley function for n = 2

3 Tasks

Implement an EA solution for given functions employing a real-valued representation with adaptive mutation step control for each dimension and appropriate reproduction operators. The adaptive mutation step control must be implemented as n real values corresponding to the mutation rates of the n elements in x. Your implementation should include both $(\mu + \lambda)$ and (μ, λ) approaches.

Instead of hard coding all the ES parameters you are to employ a separate configuration file. The ES parameters in your configuration file should include, but are not limited to, parameters for initialization, parent selection, reproduction, competition and termination. It must be possible to enable or disable the adaptive mutation step control in your configuration file. Furthermore, your configuration file should also include experiment parameters including, but not limited to, number of runs, dimensionality of the function in use (n) and logging parameters.

Report the results of applying your EA to the following problem instances:

- Ackley function: (n = 2)
- Ackley function: (n = 5)
- Schwefel function (n = 3)
- Schwefel function (n = 6)

with at least two different parameter sets for each of the problem instances averaged over at minimum 10 runs to compare convergence quality and convergence speed both in terms of the population fitness average and the population fitness max versus number of evaluations. Out of your two different parameter sets only report the one with the best results achieved for each problem instance.

Your report must include a comparison on the effectiveness of using adaptive mutation control by comparing at minimum two parameter sets which are different only in whether or not they use adaptive mutation control.

You should also compare the differences between using $(\mu + \lambda)$ and (μ, λ) selection methods based on your final results.

Notes:

- Allowed programming languages: Python, MATLAB
- Any sign of cheating would result in a zero grade for this assignment.
- You should upload your submissions at:
  https://quera.org/course/add_to_course/course/14736/
  All of the files should be in a ZIP file named in this format: "Lastname-SudentNumber.zip"
  Ex: "Zamani-4023040.zip"
- Your reports should be in a PDF file including: key points of your implementation, explanation of your chosen approach, reports of your final results and answers of assignment questions (if given).
