Fall 2023 COSC 3P71 Artificial Intelligence: Assignment 3

Instructor: Prof. Beatrice Ombuki-Berman

Teaching Assistants: Alanna McNulty, Zachary McGovarin, Tom Wallace

Available: Sunday, November 12, 2023

Due: Monday, November 27, 2023 at 4:00 pm (No lates!)

You can pick one of two options for this assignment, both of which are given below. Read through both options carefully before deciding which you would like to complete.

Option 1: Particle Swarm Optimization (PSO) implementation

This option will be worth 10% of your final grade. The PSO algorithm is a stochastic search technique that is applicable to real-valued optimization problems. However, it's performance is rather sensitive to the values assigned to its control parameters, namely the inertia weight (ω), the cognitive acceleration coefficient (C_1), and the social acceleration coefficient (C_2). In fact, poor parameter values can lead to performance that is even worse than random search!

For this option, your task is to demonstrate your understanding of the Vanilla PSO algorithm discussed in lecture by implementing and testing your implementation on two well-known fitness functions, and comparing the results to random search. Random search, in this context, refers to a uniform random sampling of the search space. This can be accomplished by sampling and evaluating random (feasible) points uniformly within the search space. You must ensure that fairness is maintained for the comparison. For example, if your Vanilla PSO has 30 particles and runs for 1000 iterations, it is evaluating 30,000 positions – a random sampling should then also uniformly sample 30,000 positions within the search space.

Recall that the main components that need to be considered when implementing Vanilla PSO are:

- **Initialization:** Construct a random initial population of particles, ensuring that each has a valid initial position and velocity. Typically, a swarm will consist of 20–40 particles.
- **Neighbourhood Topology:** You can assume a global-best neighbourhood such that the neighbourhood best refers to the best solution found by any particle throughout the course of the search.
- **Boundary Constraints:** Assume that particles are permitted to exit the feasible region, but you should only allow updating of the personal best when a solution is feasible.
- **Iteration Strategy:** You can assume a synchronous iteration strategy.

• **Termination Criterion:** You are free to determine your own termination criterion. The most straightforward way would be to fix the maximum number of iterations. However, remember that you must ensure a fair comparison between PSO and random search.

As a reminder, the Vanilla PSO pseudocode is as follows:

```
function PSO
     Initialize the particles randomly across the search space
     Initialize each particle's velocity to zero
     while termination criterion is not met:
          For each particle in the swarm:
               Evaluate fitness
               Update the personal best position
          End For
          Update the neighbourhood best
          For each particle in the swarm
               Calculate the new velocity
               Update the particle's position
          End for
     End while
     Return the global best particle
End function
```

Remember that each particle should know its current position, personal best position, and velocity. The particle's position and velocity should be represented using 'd'-dimensional vectors, where 'd' is the number of problem dimensions.

The velocity equation is given by:

$$v_i(t+1)=\omega v_i(t)+C_1 r_1(y(t)-x_i(t))+C_2 r_2(\hat{y}(t)-x_i(t))$$

Where ω is the inerita component, C_1 is the cognitive component, C_2 is the social component, r_1 and r_2 are two random vectors of size 'd' with values in [0, 1], $x_i(t)$ is the current particles's current position, y(t) is the current particle's personal best position, and $\hat{y}(t)$ is the neighbourhood (i.e. global) best position.

The particle's position is updated using:

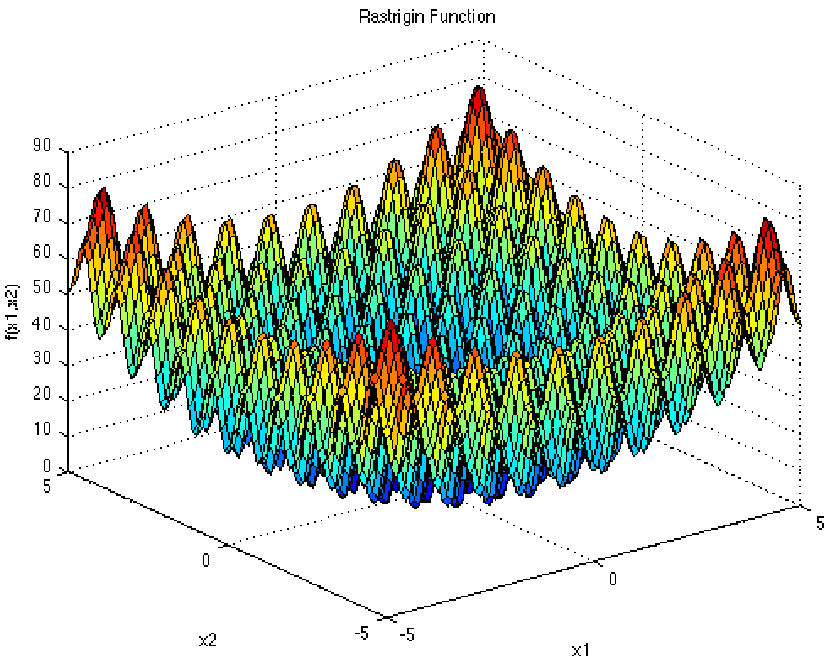
$$x_i(t+1)=x_i(t)+v_i(t+1)$$

There are two fitness functions that you will implement and use for testing both random search and your Vanilla PSO implementation. Both problems are minimization problems, i.e. particles with lower fitness values are better. Note the the fitness of a particle is simply the value of f(x), where 'x' is the particle's position. The two fitness functions required for this assignment are the Rastrigin function and the Eggholder function, the details of which are given as follows:

(a) The Rastrigin function, use a value of d = 10:

$$f(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$$

Domain = $x_i \in [-5.12, 5.12]$ for all i = 1,...,d

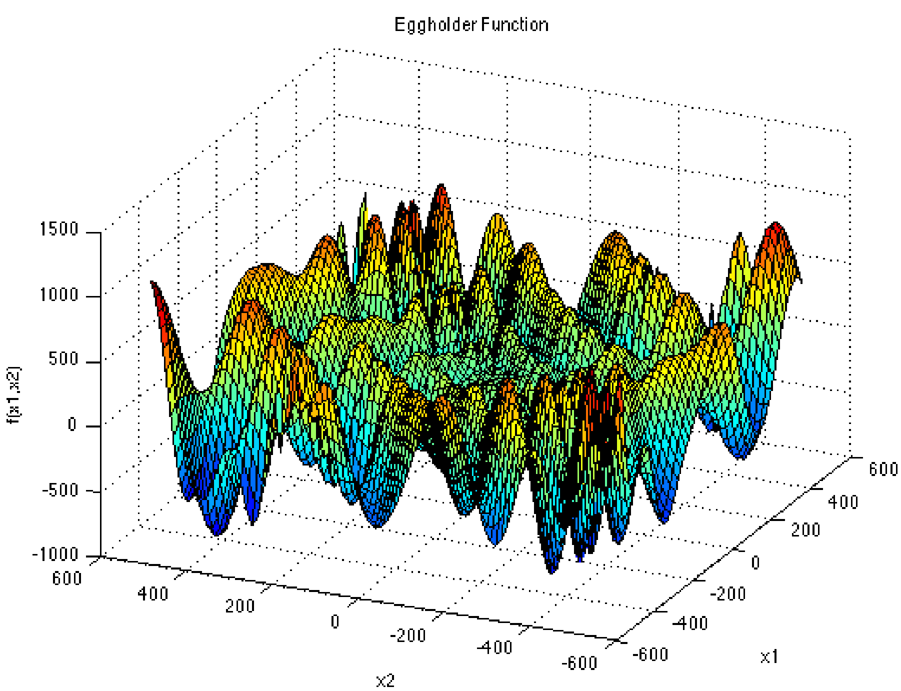


Above is the 2D Rastrigin function, image from: https://www.sfu.ca/~ssurjano/rastr.html

(b) The Eggholder function, use a value of d = 2:

$$f(x) = -(x_2 + 47)\sin\left(\sqrt{\left|\left(x_2 + \frac{x_1}{2} + 47\right)\right|}\right) - x_1 \sin\left(\sqrt{\left|(x_1 - (x_2 + 47))\right|}\right)$$

Domain = $x_i \in [-5.12, 5.12]$ for all i = 1, 2



Above is the 2D Eggholder function, image from: https://www.sfu.ca/~ssurjano/egg.html

At a minimum, at the end of each iteration, you should display the global best position and it's corresponding fitness value.

Use the following four sets of parameter values:

Parameter	Parameter Set 1	Parameter Set 2	Parameter Set 3	Parameter Set 4
ω (inertia component)	0.729844	0.4	1.0	-1.0
C_1 (cognitive component)	1.496180	1.2	2.0	2.0
C_2 (social component)	1.496180	1.2	2.0	2.0
Swarm size	Your choice	Your choice	Your choice	Your choice
Number of iterations	Your choice	Your choice	Your choice	Your choice

It is recommended to allow the user select either random search or PSO, and to input their own parameters for swarm size and iterations, as well as to decide which fitness function to use.

Repeat each experiment at least 5 times (but more is always better!) and provide a short report that highlights your findings. The report does not need to be nearly as extensive as Assignment 2 but should clearly describe the random search and the Vanilla PSO algorithm and what observations you made from the experiments.

Submission Details:

Your electronic submission should include:

- Your source code
- An executable
- Instructions for compiling/running your program and changing parameters

Submission will be done electronically using Brightspace. Any questions regarding submission can be directed to Alanna at am17xy@brocku.ca or Zach at zm19hc@brocku.ca. Please note that the virtual COMMONS is available to all students at Brock. You are not required, but if you prefer to (or need to) you can use the virtual COMMONS instead of a personal computer. Machines in the virtual COMMONS have IDEs for Java, Python, and C#. 10 Any questions/concerns regarding the grading of any assignment MUST be raised within 7 days of graded assignment hand-back.

In this case, please send your concerns/questions to one of the course coordinators: Alanna at am17xy@brocku.ca or Zach at zm19hc@brocku.ca.. To better serve you, please don't send multiple queries on the same topic to the Professor and other TAs. The course coordinators will be the point of contact for any such queries and the Professor will receive them all at once from them.

Feel free to use any language (with reason) as long as it can be opened and executed on the lab computers. Examples include Java, C#, C++, and Python. If you are unsure about the suitability of a particular language, ask before proceeding. No matter your choice of language, ensure you have provided sufficient comments such that your program can be understood. At minimum, include a comment describing each function/method and class/module. Unity projects are not allowed for this assignment.

This assignment is to be completed individually. Plagiarism detection software will be applied in this course for all submitted work. Additionally, a number of assignments will be randomly selected and the authors will be asked to explain their code and submitted documents.

Option 2: Non-Coding Option

The Vanilla PSO implementation is optional if you choose this option. You can implement PSO at your own leisure, and you will self-grade. NO marks will be awarded.

Instead, the 10% of your final grade will be given by re-weighting the marks such that your test, midterm and exam are each worth 30/30/40% in that order.