ISE 5113 Advanced Analytics and Metaheuristics
Homework #6

Instructor: Charles Nicholson

Due: See course website for due date

Requirement details

1. Homeworks are to be completed in teams of one, two, or three. You must set up your team in Canvas correctly as an official group. If team members disagree on an answer, you can record solutions corresponding to each member (please clearly mark which solution belongs to which team member).
2. Your primary submission will be a single Word or PDF document and must be of professional quality: clean, clear, concise, yet thoroughly responsive.
3. Any code (e.g., Python) must also be submitted separately. Your code MUST be well-documented. Failure to submit the files will result in a penalty.
4. You cannot use preexisting Python packages for heuristics or metaheuristics. In fact, other than `numpy, copy, random, itertools`, and other basic utilities, you should seek specific permission from the instructor if you have any doubt. That is, *you* are responsible for creating the logic yourself.

You will develop Python code to implement a Particle Swarm Optimization (PSO) algorithm for the Schwefel minimization problem.

The Schwefel problem is a generalizable benchmark problem than can be formulated in $n > 0$ dimensions.

Figure 1 presents the formula and a 2D representation of the solution landscape. The feasible region is an n-dimensional hybercube centered at the origin with possible values ranging from -500 to 500 for each dimension. The optimal objective and solution is known.

$$f(x) = 418.982887272433n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$$

Dimensions:	n
Domain:	$-500.0 \leq x_i \leq 500.0$
Global Optimum:	$f(x) \approx 0.0$ at $x = (420.9687, 420.9687, \ldots, 420.9687)$

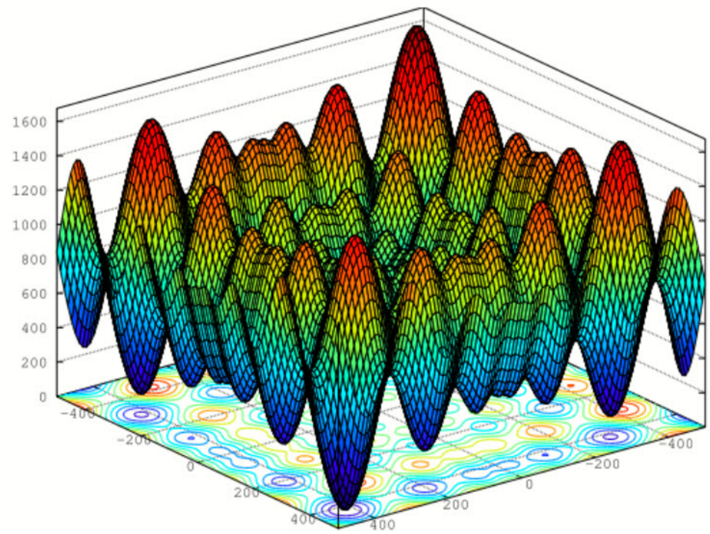

Figure 1: Schwefel function in 2D

Question 1: Particle Swarm Optimization (100 points)

Some basic Python code is available to help you begin a PSO implementation for the Schwefel minimization problem.

(a) (45 points) Complete the basic PSO implementation based on "particle best" and "global best" (as represented in Figure 2).

Basic PSO Update Functions

$$V_i^{t+i} = wV_i^t + \phi_1 r_1 \left(P_i - X_i^t\right) + \phi_2 r_2 \left(P_g - X_i^t\right)$$
$$X_i^{t+i} = X_i^t + V_i^{t+i}$$

where V_i^t and X_i^t are the velocity and position of the i^{th} particle at time step t; w is the inertial weight; ϕ_1 and ϕ_2 are constant multipliers for the "cognitive" and "social" components, respectively; P_i is the best historical position for the i^{th} particle and P_g is the best historical position ever found by swarm; r_1 and r_2 are random numbers uniformly distributed between 0 and 1 that are randomized at each iteration.

Figure 2: Velocity and position update functions for the basic PSO implementation

Make sure to code the following elements (and include code excerpts in your PDF submission):

i. (5 points) correct tracking of particle best position and value

ii. (5 points) correct tracking of global best position and value

iii. (20 points) correct implementation of the velocity and position update functions

iv. (5 points) limits or gudiance on on feasible particle position (e.g., particles should *primarily* remain within the feasible region)

v. (5 points) limits on particle velocity

vi. (5 points) one or more stopping criteria

(b) (15 points) Using your code from Part (a), create a swarm of size 5 and solve the 2D Schwefel problem.

i. (5 points) Record and list in a table the first 5 positions and velocities of each particle.

ii. (5 points) Determine and highlight the particle that represents the "global best" particle position in each of these iterations.

iii. (5 points) Plot the first 5 positions of each of the 5 particles (you can use Python, R, Excel, or even draw it by hand)

(c) (15 points) Using your code from Part (a), create a swarm to best and solve the 200D Schwefel problem. Try different swarm sizes, inertial weights, and values for ϕ_1 and ϕ_2. Record your PSO performance results in a table. (You must report on at least 5 distinct variations of PSO parameters).

(d) (15 points) Implement a PSO algorithm that uses the "local best" in place of the global best and explain which topology you are using. Make sure to explicitly explain your logic in the PDF writeup.

(e) (10 points) Solve the 200D Schwefel problem as best as possible using the code from Part (d). Compare the results with performances in Part (c).