Computational Intelligence Exam 3

This exam is an individual exam. No collaboration, sharing of solutions, or discussion with classmates or anyone else is allowed. **Do the exam yourself**. You may consult any written or existing online sources, but you may not copy code or discuss the exam in any way with anyone, literal or virtual. Any suggestion or evidence to the contrary will be referred to the Dean of Students as an academic integrity violation.

Submit your solution as a nicely-written, professional looking PDF on Canvas by Monday, April 30, at midnight. No late submissions will be accepted.

Produce a particle swarm optimization code with the following function syntax:

[xBest, fBest] = myPSO(fitnessfunc, xrange)

- xBest is a [1 x d] vector of the best output
- fBest is a scalar output of the best fitness value
    - Note: using xBest as input to the fitness function should result in fBest
- fitnessfunc is the function handle of the fitness function to use (just like your GA)
- xrange is a [d x 2] vector, where each entry is the range in that dimension of x, where the range is [xrange(:,1) to xrange(:,2)]. Each row of xrange is the interval on which that dimension is evaluated. That is, if the first row of xrange is [-10 2] then dimension 1 of the fitness function would be evaluated between -10 and 2.
    - This also determines the dimensionality of the problem to be optimized by the number of rows of xrange.

(10 points) Your code must be commented. Submit the final version of your myPSO.m function. You will receive up to 10 points for your submitted code.

a) (10 points) Describe which version of the PSO you coded up. Use an equation editor where necessary to present your algorithm neatly.

b) (30 points) Experiment with the parameters of your PSO to determine what works best overall for the 6 benchmark functions. What parameter values did you choose? (Only use one set of parameter values for all benchmark functions). Using the parameter values you selected, do the following:

    i. Present a table that gives the mean and standard deviation of fBest over 100 different runs of your PSO on each benchmark function.
    ii. Produce a plot of the 100 fBest values rank-ordered from least to greatest for each benchmark.
    iii. Present the xBest vector for the single best of the 100 runs (the lowest fBest found) and the single worst of the 100 runs (the highest fBest found).

Use the following for xrange in each test.

Ackley: xrange = [-32.768*ones(3,1) 32.768*ones(3,1)]
Branins: xrange = [-5, 10; 0, 15]
Dejong: xrange = [-5.12*ones(10,1) 5.12*ones(10,1)]
Rosenbrock: xrange = [-5*ones(5,1) 10*ones(5,1)]
Rastrigin: xrange = [-5.12*ones(4,1) 5.12*ones(4,1)]
Sum of Powers: xrange = [-ones(5,1) ones(5,1)]

Your table for part a)i. should look something like this:

Benchmark	Mean of fBest	Standard Deviation of fBest
Ackley		
Branins		
Dejong		
Rosenbrock		
Rastrigin		
Sum of Powers		

c) (10 points) Show that your PSO works by plotting the fitness of the global best found by the swarm versus iteration of the PSO for each of the 6 benchmark functions. You can essentially do this for one run of each your benchmark functions you ran in part b). Your plot should show that the global best decreases (is optimized) over iterations.

d) (20 points) Consider each of the benchmark functions individually and see if you can improve the convergence speed of the PSO. That is, can you adjust the parameters of your PSO such that it will find the minimum faster. You can use different parameter values for each benchmark for this problem. This question is asking whether it is possible to tune the PSO to work better if you only consider one benchmark at a time.

   i. Show this by plotting the fitness of the global best found by the swarm versus iteration for each of the benchmark functions. Plot this on the same plot you produced for part c) so that you can show that your new parameter choices speed up the convergence.

   NOTE: You may not be able to find parameters that make things faster and better, in terms of fitness, at the end of the iterations. That's ok. I just want to see that you adjusted the parameters and produced a faster convergence, perhaps only in the beginning iterations.

   ii. What were the six sets of parameters you used for each of the plots in part d)i. Show them in a table.

This is what your table should look like:

Benchmark	Parameters Used in Part d)i.
Ackley	
Branins	
Dejong	
Rosenbrock	
Rastrigin	
Sum of Powers	

e) (5 points) Go to https://www.sfu.ca/~ssurjano/optimization.html and choose one benchmark function (different from those above). Code up this benchmark function in a similar style as the ones I have given you. Comment the code similarly and submit your .m file for this benchmark. Describe the benchmark in your report, giving all the information necessary to reproduce it. A plot of the benchmark would be very helpful too. Cite sources where necessary.

f) (15 points) Repeat the experiment from part b) for your new benchmark function using the same parameter values you used in part b) for the other six benchmarks. Present the same results for your new benchmark. That is, mean and standard deviation of the fBest over 100 runs, a plot of the 100 fBest values rank-ordered from least to greatest, and the xBest vectors for the best and worst of your 100 runs.

BONUS: (20 points)
Propose a significant alteration to the PSO you coded up above. You don't need to do something new, you may find something in the literature that someone else has proposed (cite sources). For example, you could choose Roach Infestation Optimization.

Describe the algorithm to sufficient detail such that someone else could reproduce your results. Code up this "new" PSO and rerun the experiments on the six benchmarks plus your chosen benchmark for part b) above.

Submit your .m file and include the necessary results in your report.

You may get up to 20 additional bonus points, depending on how different your new PSO is from what you did above. Small changes will only be awarded small bonuses, big changes and well-described algorithm will be awarded big bonuses (up to 20 points).