CI Final Exam

Timothy C. Havens

This is exam is worth 100 points: 20 points will be given for presentation and writing quality. It will count as one of your three exams. The report for this assignment must be written up as a paper. You should clearly introduce the problem, state your solutions, and present results in the form of plots, tables, etc. You will graded on not only technical quality, but also on your presentation and writing, which are vital skills for scientists and engineers.

Problem 1: Particle Swarm Optimization
Assume you want to optimize some fitness function $f(\vec{x})$, $\vec{x} \in \mathcal{R}^d$, where the values of $\vec{x}$ are the in range $[-\texttt{xrange}/2 : \texttt{xrange}/2]$. Develop a Matlab or Python function for particle swarm optimization with the following syntax:

```
function [fxbest,xbest] = myPSO(funchandle, xrange, dim)
```

where `fxbest` is the best fitness function value, `xbest` is a column vector that is the associated best value of the input vector $\vec{x}$, `funchandle` is the handle of the function being optimized (e.g., `funchandle = @dejong`), `xrange` is a scalar, which is the range of the input values, and `dim` is the dimensionality of the fitness function. Note that your internal code should automatically (or deterministically) define the termination criteria or other necessary aspects. No other inputs should be passed to your code.

You will use the three benchmark functions (Dejong, Rastrigin, Ackley) that you used for the GA algorithm as well as two others (Rosenbrock, Branins).

Address the following questions in your report:

(a) (5 points) Give a psuedo-code for your algorithm. Describe the major parameters of your PSO. Someone should be able to reproduce your algorithm perfectly from your description. Note that code should *not* be used to describe your algorithm. Equations, text, and psuedo-code are acceptable.

(b) (10 points) Run your PSO for each of the five benchmark functions (use 3 or more dimensions for at least three of the benchmarks) for a variety of parameter values. For the *cognitive* and *social* parameters, do a grid search over parameter values $[0, 0.2, 0.4, \ldots, 2]$ and fill in the following table for each benchmark, where each cell includes the mean fitness squared error $(f(\vec{x}) - f_{best})^2$ over 10 runs, where $f(\vec{x})$ is the fitness value your PSO found and f_best is the known best fitness value for that benchmark. If you have additional parameters in your PSO, make sure you describe how you chose that parameter value, e.g., the momentum constant or the constriction factor.

Benchmark Name	Cognitive Parameter										
Social Parameter	0	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
0											
0.2											
0.4											
0.6											
0.8											
1											
1.2											
1.4											
1.6											
1.8											
2											

(c) (10 points) Using the best overall parameter values for the *cognitive* and *social* parameters from part (b), explore the performance of your PSO with the five benchmarks by varying the other parameters in your code (choose two parameters). Do a grid search over possible values of these two parameters—e.g., momentum constant (starting at 0) and constriction factor—and make similar tables as you did for part (b).

(d) (10 points) Using the results from parts (b) and (c) and present the squared error for each of the five benchmarks for the best overall parameter choice. That is, if you had to choose one set of parameter values, what would you choose. Report the parameter values and the five squared errors. Note, you can just copy the squared error values from the experiments you did in (b) and (c). Comment on these parameter values. Are they the best for all the benchmarks? If they aren't the best for all the benchmarks, is there something different about the benchmarks that the parameters aren't optimal for?

(e) (10 points) Choose 10 different sets of parameters that work well for the five benchmarks. Make sure the parameter choices are somewhat diverse. Run the PSO for these 10 different sets of parameters and time your code. Report the results. Comment on the results. Is there something you can conclude from this experiment?

(f) (5 points) Choose the slowest and fastest parameter choices from part (e) and plot the squared error of the PSO versus iteration. Note you can do this by plotting the squared error of the global best vs. iteration. Comment on the results.

(g) (20 points) Find a research paper on a modification of the PSO and implement this modification. A good place to look is the *IEEE Trans. Evolutionary Computation*. Describe this modification and run a set of experiments that shows how this compares to your previous results, addressing both squared error performance and algorithm speed. Make sure you run enough experiments to support your conclusion. You don't need to completely reproduce all of the results from parts (b)-(f), but if you write your code in a smart way, it certainly wouldn't be hard to plug in your modification to a script that just runs everything for you.

(h) (10 points) Compare the results of your PSO with your GA results. Which is better and in what way? Comment on the overall quality of the solutions, the speed of the algorithm, and also the complexity of the code.

Submit your Matlab or Python function code (just myPSO function and dependencies) as a separate file along with a PDF of your Problem 1 (you can combine the PDF with other solutions).

Your grade on this assignment will be based on originality, technical soundness, organization, clarity of presentation, grammar and style, as well as the quality of the results. Your report must be nicely written and presented, axes should be labeled in all plots, tables should be easy to understand (do not use more significant figures in your numbers than necessary), all graphics (plots, tables, etc.) should have captions.