Project 7 : Derivative Free Optimization Strategies

This project is our only project which does not introduce a specific application. We will see several new applications later during the remainder of the course. The focus here is to provide and test code for some of the strategies we discussed late last Fall in Math 564. You may use any coding language of your choice, though you can obtain the best assistance from me if you use either Matlab or python.

Task 1: Review the course material on dense space samplings on $[0,1]^n$, the Nelder-Mead Simplex Method, continuous-variable Genetic Algorithms, and Simulated Annealing.

The goal is to recall the discussions and the basic concepts. A deep understanding is not necessary here as this will develop as we use these tools. These materials are accessible through the course canvas page.

Task 2: [Due: Jan 15] Compose a function that produces a sampling of points on $[0,1]^n$. Your code should accept as inputs: the spatial dimension n, the type of sampling to compute, and any relevant type-specific parameters. Test your code and show me example representations of the samplings.

You should include at least the following two sampling methods (and parameters).

1. `Random.` Each point is a vector in $[0,1]^n$ with entries which are drawn from a uniform probability function on $[0,1]$. Parameters should include an optional random number seed.
2. `Halton Vector Sequence.` The vector coordinate sequences are distinct Halton Sequences. Parameters should include n Halton seeds and the number of initial vectors to discard.

Optionally, you can also include the method of `Latin Hypercube Sampling`.

Task 3: [Due Jan 22] Compose a function that performs a Nelder-Mead simplex optimization search. Use your code to solve the Function Fitting problem (Project #1 of Math 564) as both an unconstrained problem and a problem with reasonable box constraints of your choosing. Use your space sampling code to construct different initial simplices and compare the solution efficiencies of different approaches. Compose and submit an informal **two page** report outlining your results.

This code can (and maybe should be) separate from your main code from Math 564. However, some design characteristics should be similar or identical to maximize compatibility. Your function input can be a single structure/dictionary/class variable with at least the following information: initial simplex, stopping criteria parameters, function handle to the objective computation. In this case, your objective computation should include any constraints using the extreme barrier approach.

Task 4: [Due Jan 29] Compose a function that performs a global optimization strategy for continuous variables: either `Simulated Annealing` or a `Genetic Algorithm`. Research test functions for global optimization and test your algorithm on one of you choosing that has at least seven decision variables. Again, compose and submit an informal **two page** report outlining your results.

For this task, choose a objective function that has more than one local minimizer and is moderately difficult to solve. In other words, choose a function that is likely to be very challenging for a derivative-based method, but could be reasonably solved with your new code.