Homework #5: Gradient-free Optimization — ME 575

due 3/18/2015 before class via Learning Suite — 50 possible points

---

5.1 Improve your project problem from the last homework and optimize it using an existing *gradient-free* algorithm of your choice. You may use any of the optimizers available in Matlab, SciPy, pyOpt, etc. Discuss your findings. You might like to try multiple optimizers, and even compare results using both gradient-based and gradient-free methods (comparing across different optimizers is purely optional, but I would recommend it).

5.2 Implement your own gradient-free optimization algorithm (e.g., Nelder-Mead, Genetic Algorithm, Particle Swarm,). Apply it to your project problem, or if you project problem is overly complex apply it to a simpler test problem. Compare your results and performance as compared to a similar existing algorithm (e.g., Matlab global optimization toolbox).

5.3 Study the effect of increased problem dimensionality using the n-dimensional Rosenbrock function (Eq. (1)).

$$f(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - {x_i}^2)^2 + (1 - x_i)^2\right) \tag{1}$$

Solve the problem using all three different approaches:

(a) Gradient-free

(b) Gradient-based with finite differencing

(c) Gradient-based with exact gradients

You may use existing optimizers, or your own implementations that you developed. In each case repeat the minimization for $n = 2, 4, 8, 16, \ldots$ up to the highest number you can manage. Plot the number of function calls required as a function of dimension size for all three methods on one figure. Discuss any differences in optimal solutions found by the various algorithms and dimensions. Compare and discuss your results.