Assignment 1 (35 pts)

Due February 14 2025

1. (**25pts**) Implement the particle swarm optimization (PSO) algorithm for general constrained optimization problems and apply your code to the portfolio of optimization problems described in the "Case Studies" document on Quercus. Here are some guidelines:

   - Use the 2D bump test function (**P3**) to tune the parameters of your PSO implementation and use the tuned parameters for the other case studies.
   - Use the quadratic penalty function approach to enforce the constraints. Compare the performance of static penalty parameter and the penalty parameter adaptation strategy (see Slides on Quercus) for the 2D bump test function. Based on these studies, you can select the approach that worked best when tackling other constrained test problems.
   - Since PSO is a stochastic search algorithm, carry out *at least* **10** independent runs for each parameter setting. In your report, include a plot containing the statistics of the convergence trends across the independent runs. Report the best solution obtained over these runs in a table.

2. (**2pts**) Consider a regression model for a scalar function of the form

$$\widehat{f}(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{M-1} w_i \phi_i(\mathbf{x}),$$

   where $\mathbf{x} \in \mathbb{R}^D$, $w_i \in \mathbb{R}, i = 0, 1, 2, \ldots, M-1$ are the undetermined weights of the model, and $\phi_i : \mathbb{R}^D \rightarrow \mathbb{R}$ are known basis functions.

   Given a regression dataset $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$, we seek to estimate the undetermined weights by minimizing the ℓ_2 regularized weighted least-squares error function

$$\sum_{i=1}^{N} c_i \; (\widehat{f}(\mathbf{x}^{(i)}, \mathbf{w}) - y^{(i)})^2 + \sum_{i=1}^{M-1} \lambda_i w_i^2,$$

   where $c_i > 0$ and $\lambda_i > 0$ are user-defined regularization parameters and weights, respectively. Derive a system of linear algebraic equations to be solved for the weights of the regression model.

3. (**8pts**) Consider the constrained optimization problem $\min_{\mathbf{x}\in\mathbb{R}^n} \quad f(\mathbf{x})$, subject to the inequality constraints $g_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m$. While solving this optimization problem, it was found that the feasible set is empty and no solution that satisfies all the constraints could be found. One way to resolve this issue would be to rewrite the constraints as $g_j(\mathbf{x}) \leq \delta_j, j = 1, 2, \ldots, m$, where $\delta_j \geq 0$ can be interpreted as a constraint relaxation parameter. Answer the questions below which involve the formulation of *tractable* optimization problems reflecting different approaches to the constraint relaxation problem (numerical implementation is not need).

   (a) (**4pts**) Your goal is to relax the constraints (i.e., find the smallest possible values of $\delta_j, j = 1, 2, \ldots, m$) such that the feasible set is no longer empty. Translate this goal into an optimization problem statement.

   (b) (**2pts**) Consider the case when we want to relax the least number of constraints to make the problem feasible. Formulate an optimization problem statement that captures this requirement. Do you foresee any challenges in solving this problem?

   (c) (**2pts**) In real-world applications, some constraints are more critical than others. Let $c_j > 0$ denote the "cost" associated with relaxing constraint j, i.e., the financial cost associated with relaxing the jth constraint by δ_j is given by $c_j\delta_j$. Formulate an optimization problem to find the minimum-cost constraint relaxation that makes the problem feasible.

   Guidelines: Be mathematically precise in your writeup and follow these guidelines: (1) use only standard mathematical operators (e.g., $\min, \max, \leq, \geq$, etc) in your optimization problem statement, (2) define all optimization variables, constraints, and objectives explicitly, (3) state objectives/constraints without using set notation; more specifically, ensure that the objectives and constraints are written in a format that permits tractable numerical computation, (4) When using min and max operators, clearly define the variables over which the operator is applied, for example $\min_{z\in\mathbb{R}^n} f(z)$.

Bonus question (5pts): Apply your PSO code to the bump test function (**P3**) with $n = 50$. Report the best solution you obtain and the corresponding value of $\mathbf{x}^*$ in an ASCII file appended to your assignment submission (the TA should be able to verify your reported solution). The bonus marks will be awarded using a relative scaling system with the best solution in the class (f^{best}) being awarded 5pts. The points awarded to submissions with an objective function value of f' will be $5(f^{best}/f')$.

Submission guidelines:

- Your assignment must be submitted in `pdf` format to Quercus. Reports submitted in other formats such as Microsoft Word are not acceptable.
- You are permitted to use Python for this assignment. Please mention the Python version you are using in your report.

- Submit MATLAB/Python codes archived in tar, zip or gzip format along with input files used to generate results. Do NOT use proprietary data compression formats such as rar. Please verify the integrity of your tar/zip/gzip file before uploading.
- Include a README file that describes how the codes must be run.
- Clearly explain all working steps in your answers.
- All submissions must comply with the AER1415 Collaboration Policy posted on Quercus.

Marking guidelines:

- Q1: Marks for Q1 will be assigned as follows:
  - PSO implementation and parameter tuning studies [5pts]
  - Results for case studies [20pts]: Marks will be split in the following way for the case studies: **P1** (15%), **P2** (10%), **P3** (15%), **P4** (30%), **P5** (30%)
- Q2 and Q3: You are expected mathematically describe your approach to these problems. Clearly define all variables and explain all working steps.

Suggested format of report:

- PSO algorithm implementation and testing
  - Algorithm description
  - Parameter tuning studies
  - Results for case studies (including formulation of **P4** and **P5**)
  - Conclusions
- Your answer to Q2
- Your answer to Q3