C6.2 Continuous Optimization

Doctoral Broadening/CDT Assignment

To be handed in by Monday of Week 9 (to either CDT admin or in my MI pigeonhole). Please choose either a theoretical assignment or a numerical one. Please write up your work and findings in an executive summary of at most 5 pages (including bibliography).

Part I: Reading assignment

Choose one of the following topics and read the suggested bibliography; please feel free to change the suggested bibliography if you wish. Within the topic, either give a brief overview or choose a particular aspect of the topic and discuss it in your report.

1. **Quasi-Newton methods for unconstrained optimization**
   - Chapter 6 (Quasi-Newton methods) in [1].
   - R. Byrd, S.L. Hansen, J. Nocedal and Y. Singer. A stochastic quasi-newton method for large-scale optimization, 2015. `https://arxiv.org/pdf/1401.7020.pdf`
2. **Gauss-Newton methods for data fitting problems**
   - Chapter 10 (Least-squares problems) in [1].
   - S. Gratton, A. Lawless and N.K.Nicholls. Approximate Gauss-Newton methods for nonlinear least squares problems. SIAM Journal on Optimization, vol 18(1):106–132, 2007.
3. **Solving the trust region subproblem**
   - Chapter 4 (Trust region methods) in [1] (up until Section 4.2).
   - N.I.M. Gould, D.P. Robinson and H.S. Thorne. On solving trust region and other regularized subproblems in optimization. Mathematical Programming Computation, vol 2(1):21–57, 2010.
4. **Conjugate gradients method for quadratic problems/linear systems**
   - Chapter 5 (Conjugate gradient methods) in [1].
   - W.W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient linesearch. SIAM Journal on Optimization, vol 16(1):170–192, 2005.
5. **Coordinate descent methods**
   - S.J. Wright, Coordinate descent algorithms, 2015. Appeared in Mathematical Programming, but also available on Arxiv.
   - M.J.D. Powell, On search directions for minimization algorithms, Mathematical Programming, vol 4:193–201, 1973.
6. **Stochastic gradient methods**
   - F. E. Curtis and K. Scheinberg. Optimization Methods for Supervised Machine Learning: From Linear Models to Deep Learning. In INFORMS Tutorials in Operations Research, chapter 5, pages 89—114. Institute for Operations Research and the Management Sciences (INFORMS), 2017.
   - F. E. Curtis, K. Scheinberg, and R. Shi. A stochastic trust region algorithm based on careful step normalization. INFORMS Journal on Optimization, 2019 (available online).

7. **Derivative-free optimization**
   - J. Larson, M. Menickelly, S. Wild. Derivative-free optimization methods. Acta Numerica 28 (2019), pp. 287-404, also available on ArXiv. OR
   - Chapter 9 (Derivative free methods) in [1].

Reference
J. Nocedal and S.J. Wright. *Numerical Optimization.* Second Edition, Springer, 2006. (But first edition is also fine!)

Part II: Numerical assignment

The aim of the numerical component is to explore existing optimization software and apply it to a given constrained optimization problem (the hanging catenary) or some standard test functions with box constraints.

(**Algorithms/codes**) – **choose one**

- (**Knitro - a state-of-the-art optimization solver**) Download and install the *student* version of **Knitro** interfaced with Matlab, from

  `https://www.artelys.com/en/optimization-tools/knitro`

  If you prefer other interfaces, do not hesitate to use them.

  Please consult the documentation and find out about the algorithms that it implements. Learn how to run Knitro (see documentation and try for instance, the nonlinear programming examples that come with the package).

- (**Matlab optimization toolbox**) Explore the Matlab optimization toolbox: find out the built-in solvers it offers and the class of optimization problems that they are designed for. Select an appropriate solver(s) from the toolbox for the problems below and experiment with it.

(**Test functions**) – **choose either the Catena problem or a couple of standard test problems**

1. (**The hanging catenary problem.**) There are continuous versions of this problem that you may have encountered in modelling courses. Here we formulate this problem as a discrete problem (namely with finitely many continuous variables) and solve it using the Knitro solver or some Matlab solver. I give below a (nonconvex) discrete formulation as it appears in the CUTEr collection (problem **catena**); you are welcome to use this or alternative /modified formulations.

   Let the catenary have $n+1$ beams of length L, with the first beam fixed at the origin and the final beam fixed at a fraction $\gamma \in (0,1)$ of the total length of all the beams. Let (x_i, y_i, z_i), $i = 1, \dots, n+1$ be the coordinates of the end of the beams and $(x_0, y_0, z_0) = (0,0,0)$ the coordinates of the first beam (fixed at the origin). The resulting problem minimizes the total downward force as the sum of the gravitational force on each beam, assuming it acts at the middle/centre of each beam, subject to the length of beams being preserved; we obtain

   $$\begin{aligned} \min \quad & mg\left(\tfrac{1}{2}y_0 + y_1 + \cdots + y_n + \tfrac{1}{2}y_{n+1}\right) \\ \text{subject to} \quad & (x_0, y_0, z_0) = (0,0,0), \quad x_{n+1} = \gamma(n+1)L, \quad y_{n+1} = 0, \\ & (x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 = L^2, \ \forall i = 0, \dots, n, \end{aligned}$$

   where the variables are (x_i, y_i, z_i) for all i.

2. For other low-dimensional standard optimization test functions, please see for example, `https://en.wikipedia.org/wiki/Test_functions_for_optimization`.

   You may want to apply your codes to some of these functions, using the (linear) bound constraints given.

Ensure the solver works for your test problem(s) by checking any output messages or error codes. Verify that the minimizer found by the solver is feasible, to within the desired tolerance. What is the optimal objective value/gradient and Lagrange multipliers for the constraints?

Possible avenues of exploration (choose one or two): Explore different starting points and final accuracy tolerances (if possible; note that most solvers do not require a feasible starting point); changing other allowed algorithm parameters is also acceptable. For the catenary problem, you may want to experiment with increasing number of beams. You can also experiment with different algorithms (in Knitro or Matlab) and possibly different input (provide only function values and gradients, or provide both of those and Hessian values).