UNIVERSITÄT BERN

Applied Optimization
Exercise 5 - Line Search Methods

Heng Liu Nicolas Gallego

Handout date: 31.10.2019

Submission deadline: 6.11.2019, 23:59 h

Hand-in instructions:

Please hand-in **only one** compressed file named after the following convention: `Exercise`n`-GroupMemberNames.zip`, where n is the number of the current exercise sheet. This file should contain:

- **Only** the files you changed (headers and source). It is up to you to make sure that all files that you have changed are in the zip.
- A `readme.txt` file containing a description on how you solved each exercise (use the same numbers and titles) and the encountered problems.
- Other files that are required by your `readme.txt` file. For example, if you mention some screenshot images in `readme.txt`, these images need to be submitted too.
- Submit your solutions to ILIAS before the submission deadline.

Exact line search for the convex quadratic function (2 pts)

Consider the convex quadratic function

$$f(x) = \frac{1}{2}x^T Q x + q^T x + c,$$

where $x \in \mathbb{R}^n$, and constant parameters Q a square matrix $n \times n$ symmetric positive semi-definite, q a vector in $\mathbb{R}^n$, and $c \in \mathbb{R}$.

Given a search direction Δx, compute the exact line search parameter t for and arbitrary point x in the domain of f.

$$t := \arg\min_{s \geq 0} f(x + s\Delta x)$$

Gradient descent with exact line search (2 pts)

Consider the unconstrained minimization problem:

$$\text{minimize } \frac{1}{4}x_1^2 + x_2^2$$

starting from point $x^{(0)} = (2,1)$ perform one iteration of the gradient descent algorithm with exact line search. Sketch the function, the line and the update. [1]

What is the value of $\|\nabla f(x^{(1)})\|^2$ after the update $x^{(1)} := x^{(0)} + t\Delta x$?

Programming Exercise: Constrained Mass Spring System (6 pts)

In Exercise2 and 3, we setup the mass spring system problem with two different element functions:

$$E_{a,b} = \frac{1}{2}k_{a,b}||\mathbf{x_a} - \mathbf{x_b}||^2$$

and

$$\hat{E}_{a,b} = \frac{1}{2}k_{a,b}(||\mathbf{x_a} - \mathbf{x_b}||^2 - \mathbf{l_{a,b}^2})^2.$$

Now let's introduce constraints to the system by fixing certain nodes in the spring graph. Let $n \in C$ be the constrained node in the constraint set and p_n be the target position of node n. The energy function of the mass spring system are:

$$E = \sum_{e(a,b)} E_{a,b} + \sum_{n \in C} \frac{1}{2} penalty_n||(\mathbf{x_n} - \mathbf{p_n})||^2$$

and

$$\hat{E} = \sum_{e(a,b)} \hat{E}_{a,b} + \sum_{n \in C} \frac{1}{2} penalty_n||(\mathbf{x_n} - \mathbf{p_n})||^2,$$

where *penalty* is a constant that controls the effect of the constraints. Here we set it to 1000000. You need to implement the gradient descent method with backtracking line search and find the optimum solutions for the following two constraint conditions:

1. Fix the four corner nodes in a $m \times n$ spring graph at target points (0,0), (m, 0), (0, n) and (m, n) respectively.
2. Fix the nodes on the side $N_{(i,0)}$ to the point with coordinates (i, 0) and nodes $N_{(i,n)}$ to their initial coordinates (i, n), where $i \in [0, m]$.

Regarding the code that is missing, you are requested to fill in:

- Complete the `f(...)` and `grad_f(...)` in the `NodeConstraintElement.hh` file.

[1] here the link to our interactive plotting system https://slides.cgg.unibe.ch/aopt19/plots/plot2d_create.html

- Modify the `eval_f(...)` and `eval_gradient(...)` in the `MassSpringProblem2D.hh` file because we add node constraints.
- Complete the `setup_problem(...)` function to add spring elements and node constraint elements to the problem.
- Implement the `solve(...)` function in `GradientDescent.hh` and `backtracking_line_search(...)` in `LineSearch.hh`.

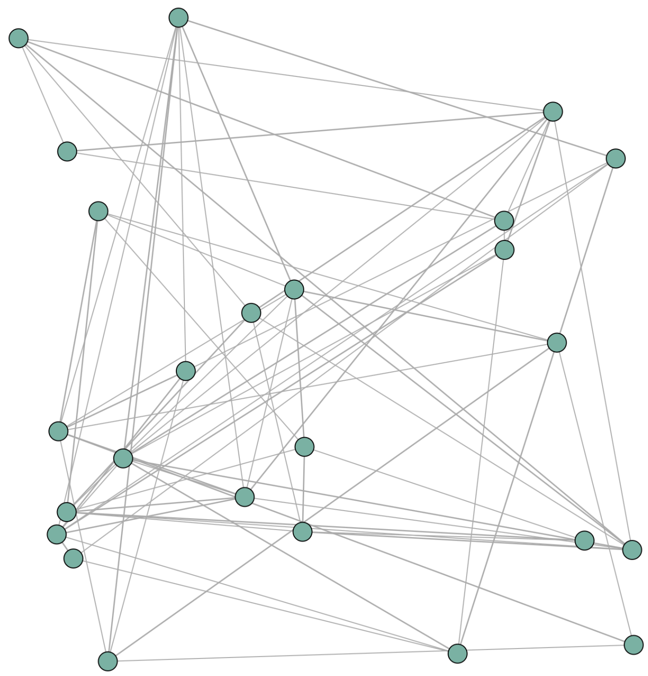

Figure 1: random 4×4 spring graph

For the usage of the executable, please see the `main.cc` in ConstrainedMassSpringSystem folder. You should provide the filename as a parameter so that the program can save the optimized spring graph to the specified files which you can then visualize on this website. Figure 1 is a 4×4 random spring graph. Try the mass spring system of two different spring element functions and different dimensions. Submit screenshots of the spring graphs (initial and optimized). Give some comments on how the gradient descent method and back tracking line search behave in this problem.