This week's homework https://mycourses.aalto.fi/mod/assign/view.php?id=361337 is due no later than **Thursday 27.09.2018 23:55**.

Exercise 1.1 Extreme points of a univariate function

Characterize the stationary points of the function $f(x) = 2x^4 - 5x^3 - x^2$. Plot the function to verify your analysis.

Solution. We have

$$f'(x) = 8x^3 - 15x^2 - 2x \quad \text{and} \quad f''(x) = 24x^2 - 30x - 2$$

Setting $f'(x) = 0$ and solving for x, we find the following 3 stationary points

$$x_1 = -1/8, \quad x_2 = 0, \quad \text{and} \quad x_3 = 2.$$

Examining $f''(x_i)$ for $i \in \{1, 2, 3\}$, we get

$$f''(x_1) = 17/8, \quad f''(x_2) = -2, \quad \text{and} \quad f''(x_3) = 34$$

Therefore:

1. As $f''(x_1) > 0 \Rightarrow x_1 = -\frac{1}{8}$ is a local minimum
2. As $f''(x_2) < 0 \Rightarrow x_2 = \quad 0$ is a local maximum
3. As $f''(x_3) > 0 \Rightarrow x_3 = \quad 2$ is a local minimum (also global in this case)

Exercise 1.2 Extreme points of a bivariate function

Characterize the stationary points of the function $f(x, y) = (y - x^2)^2 - x^2$. Plot the surface and contour plots of the function to verify your analysis.

Solution. We have

$$\nabla f(x, y) = \begin{pmatrix} -4xy + 4x^3 - 2x \\ 2y - 2x^2 \end{pmatrix} \quad \text{and} \quad \nabla^2 f(x, y) = \begin{pmatrix} -4y + 12x^2 - 2 & -4x \\ -4x & 2 \end{pmatrix}.$$

Setting $\nabla f(x, y) = 0$ and solving for x and y, we find out that the only stationary point is

$$(x, y) = (0, 0) \quad \text{with} \quad \nabla^2 f(0, 0) = \begin{pmatrix} -2 & 0 \\ 0 & 2 \end{pmatrix}.$$

We can solve the eigenvalues $\lambda \in \mathbb{R}^2$ of the Hessian $\nabla^2 f(0, 0)$ from the eigenvalue equation

$$(\nabla^2 f(0, 0) - \lambda I)v = 0$$

which has a solution if and only if

$$\mathbf{det}(\nabla^2 f(0, 0) - \lambda I) = 0 \quad \Leftrightarrow \quad (-2 - \lambda)(2 - \lambda) = 0$$

The eigenvalues are $\lambda_1 = 2$ and $\lambda_2 = -2$, so the Hessian is indefinite and the stationary point $(x, y) = (0, 0)$ is neither a local minimum nor a local maximum (it is a saddle point).

Exercise 1.3 Newton's method for a univariate problem

Consider the following unconstrained optimization problem where $f(x)$ is a univariate function.

$$\text{minimize} \;\; f(x) \tag{1}$$

Solve the problem (1) with different functions $f(x)$ using Newton's method. In the univariate case, Newton's method starts with an initial starting point $x_0 \in \mathbb{R}$ and updates the solution as follows:

$$x_{n+1} = x_n - f''(x_n)^{-1} f'(x_n)$$

Try different starting points and observe if the method converges to a stationary point or diverges without producing a solution. Plot the functions $f(x)$ and show the path taken by Newton's method. You can try, for example, the following functions and starting points:

$f(x) = x^4 - x^3 - 8x^2$	try different values for x_0
$f(x) = -x^2$	$x_0 = 1$
$f(x) = \arctan(x)$	$x_0 = 1$
$f(x) = (1/4)x^4 - x^2 + 2x$	$x_0 = 0$

Solution.

See Julia code.

Exercise 1.4 Newton's method for a bivariate problem

Consider the following unconstrained optimization problem

$$\text{minimize } (x_1 - 2)^4 + (x_1 - 2x_2)^2 \tag{2}$$

Let $f(x) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$ denote the objective function. Solve the problem (2) using Newton's method. Newton's method starts with an initial starting point $x_0 \in \mathbb{R}^2$ and updates the solution as follows:

$$x_{n+1} = x_n - \nabla^2 f(x_n)^{-1} \nabla f(x_n)$$

Try different starting points and observe if the method converges to a stationary point or diverges without producing a solution. Plot the contours of $f(x)$ in (x_1, x_2) plane and show the path taken by Newton's method.

Solution.

See Julia code.

Exercise 1.5 Pooling Problem

Consider the following simplified example of a pooling problem presented in Figure 1. This problem arises, for example, in gas transportation and oil refinery blending problems.

$\underline{\overline{q}}_s$: value of propery q at node $s \in S$
$\overline{q}_t$: upper bound of property q at node $t \in T$
c_i: unit cost at node $i \in N$
$\overline{b}_i$: upper bound of flow to node $i \in N$

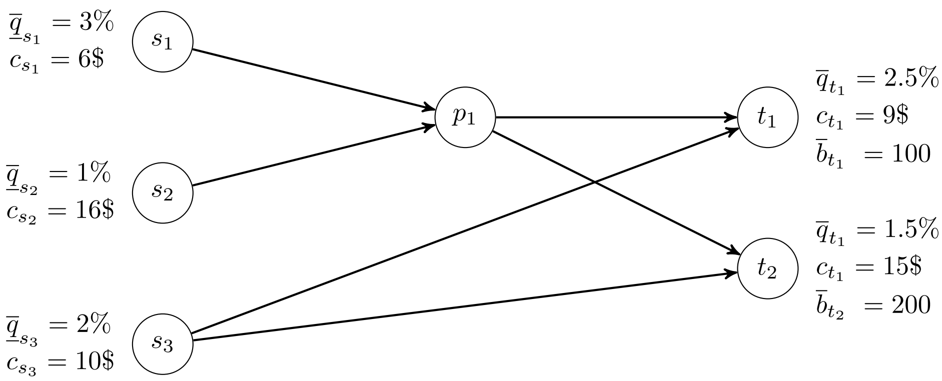

Figure 1: Pooling problem network.

The problem considered here is defined on a directed graph $G = (N, A)$ as follows. We have a set of source nodes $S = \{s_1, s_2, s_3\}$, a set of intermediate pool nodes $P = \{p_1\}$, and a set of target nodes $T = \{t_1, t_2\}$. The node set is thus

$$N = S \cup P \cup T$$

Each source $s \in S$ has a unit cost c_s to transport oil and each target $t \in T$ has a unit cost c_t which represents revenue of receiving oil. Each node $i \in N$ has a property value q_i which corresponds to oil sulfur content in this example. The property values at sources $s \in S$ are constants $q_s = \underline{\overline{q_s}}$ and the values at targets $t \in T$ have upper bounds $q_t \leq \overline{q}_t$. The property values q_p at pool nodes $p \in P$ and q_t at target nodes $t \in T$ are unknown.

The arcs A represent pipes transporting oil between the nodes. From each source $s \in S$, crude oil with a property value q_s flows to target nodes $t \in T$ either directly or via pool nodes $p \in P$. When two or more oil streams with different properties flow to a pool node $p \in P$ or a target node $t \in T$, the properties q_p or q_t of the oil at that node change due to blending.

The objective is to maximize profit by sending oil from sources $s \in S$ to targets $t \in T$. We can use the following variables to formulate the problem:

$$\begin{aligned} x_{ij} &\geq 0 \qquad \text{amount of oil flowing through each arc } (i,j) \in A \\ q_i &\geq 0 \qquad \text{propery (sulfur content) at each node } i \in N \end{aligned}$$

Let us further define

$$N_i^- = \{j \in N : (j,i) \in A\} \quad \text{and} \quad N_i^+ = \{j \in N : (i,j) \in A\}$$

The problem can be formulated as follows.

$$\begin{aligned}
\underset{x,q}{\text{maximize}} \ & \sum_{t \in T} c_t \sum_{j \in N_t^-} x_{jt} - \sum_{s \in S} c_s \sum_{j \in N_s^+} x_{sj} && && (3)\\
\text{subject to} \ & \sum_{j \in N_p^-} x_{jp} = \sum_{j \in N_p^+} x_{pj}, && \forall p \in P && (4)\\
& \sum_{j \in N_t^-} x_{jt} \leq \overline{b}_t, && \forall t \in T && (5)\\
& \sum_{j \in N_p^-} q_j x_{jp} = q_p \sum_{j \in N_p^+} x_{pj}, && \forall p \in P && (6)\\
& \sum_{j \in N_t^-} q_j x_{jt} = q_t \sum_{j \in N_t^-} x_{jt}, && \forall t \in T && (7)\\
& q_t \leq \overline{q}_t, && \forall t \in T && (8)\\
& q_s = \underline{\overline{q}}_s, && \forall s \in S && (9)\\
& q_i \geq 0, && \forall i \in P \cup T && (10)\\
& x_{ij} \geq 0, && \forall (i,j) \in A && (11)
\end{aligned}$$

The objective (3) maximizes revenue - cost. (4) maintains flow balance, and (5) defines upper bound of flow to target nodes. (6) and (7) determine the property values at pool nodes and target nodes, respectively. (8) imposes upper bounds for property values at target nodes and (9) sets the initial property values at source nodes.

The problem is non-convex due to the constraints (6) and (7) which are called *bilinear*. Typically, there are more pools and more than one property q_i for each node i. This can be modeled by introducing a set K of different properties so that q_i^k denotes the value of property k at node i.

One strategy to obtain a solution is to first fix the values of unknown property values $q_i, i \in P \cup T$ and solve the remaining Linear Programming problem. You can find different solutions by trying different value combinations. Another option is to set the smallest lower bounds on $q_i, i \in P \cup T$ and try to solve the problem with a non-linear programming solver.

Model and solve the problem (3) – (11) with Julia using JuMP using the data shown in Figure 1.

Solution.

See Julia code.