Combinatorial Optimization
Lab No. 5
Applications of the Shortest Path Problem

Industrial Informatics Research Center
http://industrialinformatics.fel.cvut.cz/

March 19, 2019

Abstract

In this lab, we present an interesting application of shortest paths and we remind ourselves, that SPT problem is generally $\mathcal{NP}$-hard.

1 Shortest paths with negative cycles

In this example, we demonstrate that Bellman-Ford's algorithm fails to produce correct answer on the following instance, wheres an ILP formulation of SPT yields correct solution.

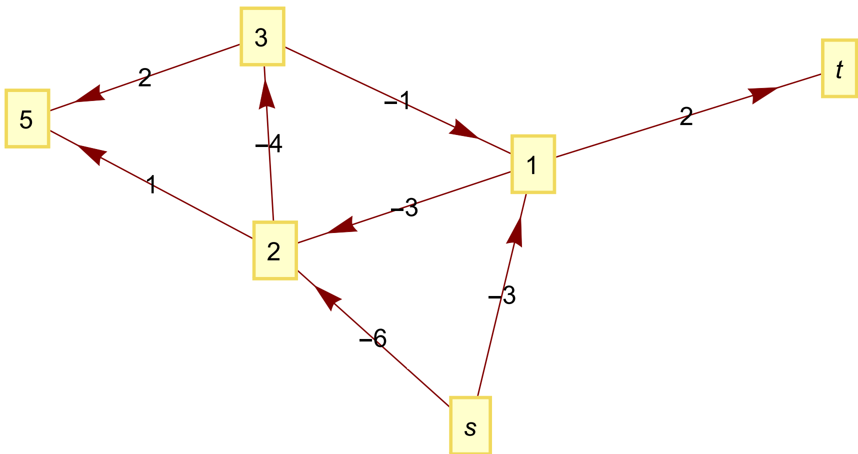

Figure 1: A graph with a negative cycle.

2 Approximation of piecewise-linear functions

Approximation of piecewise linear functions is lesser-known, however quite an interesting application of the shortest path problem [1]. As an input a set of n points is given, i.e. $\{(x_i, f(x_i))\}$.

```
x = [   0,  1.26,  2.51,   3.77,   5.03,  6.28]
f = [0.01,  1.16,   0.7,  -0.34,   -0.8,  0.21]
```

The goal is to select the smallest possible number of points on condition that approximation error is reasonable (see Figure 2).

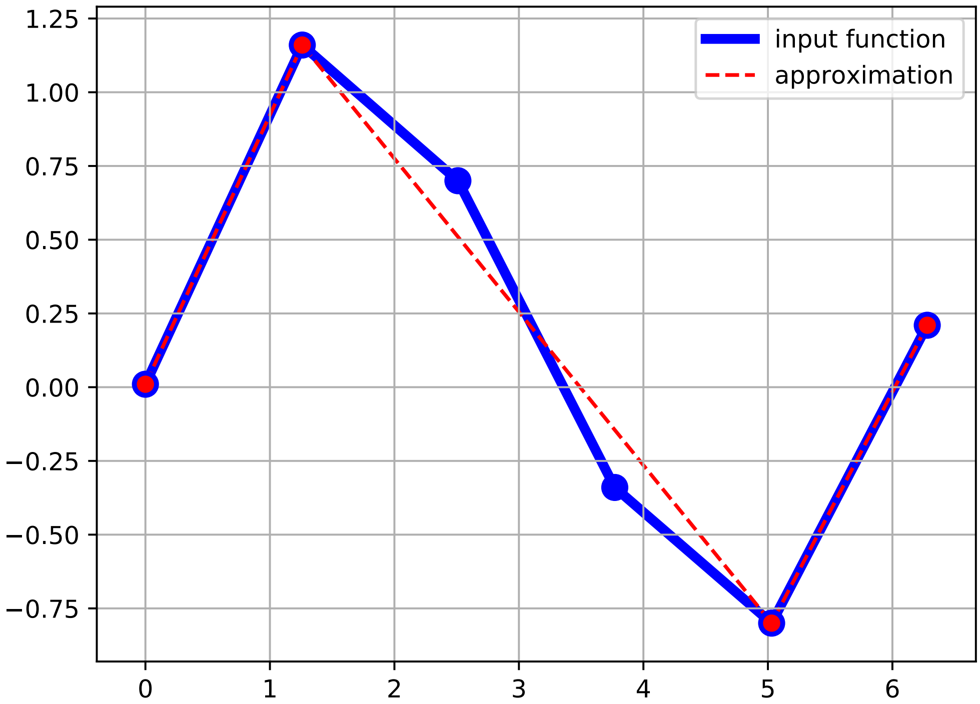

Figure 2: The input function (blue) and the approximated function (red).

This problem can be solved using the shortest path problem applied to graph G where each node v_i corresponds to a data point $(x_i, f(x_i))$ and each edge $e = (i, j)$ (such that $i < j$) has weight $c_{i,j}$ which corresponds to a penalty if the part of the function from $(x_i, f(x_i))$ to $(x_j, f(x_j))$ is approximated by a line segment. The penalty consists of the approximation error (weighted by β) and the number of required samples (the 1 term). An example of such graph can be seen in Figure 2 and penalty $c_{i,j}$ is calculated as follows.

$$c_{i,j} = 1 + \beta \left[\sum_{k=i}^{j} \left(f(x_k) - f'(x_k, x_i, x_j) \right)^2 \right] \tag{1}$$

$f'(x_k)$ is a precomputed value of the approximating function.

$$f'(x, x_i, x_j) = f(x_i) + (x - x_i) \cdot \frac{f(x_j) - f(x_i)}{x_j - x_i} \tag{2}$$

In a similar way the shortest path problem can be applied to some vector pictures which are composed of points (x_i, y_i). The difference is that weights $c_{i,j}$ are computed as a sum of squared distances from points (x_k, y_k) to the line segment defined by (x_i, y_i) and (x_j, y_j) points where $i < k < j$.

$$c_{i,j} = 1 + \beta \left[\sum_{k=i}^{j} \frac{|(y_j - y_i)x_k - (x_j - x_i)y_k + x_j y_i - y_j x_i|^2}{(y_j - y_i)^2 + (x_j - x_i)^2} \right] \tag{3}$$

An example of two-dimensional approximation, where the frontiers of the Czech Republic are input data, is depicted in Figure 3.

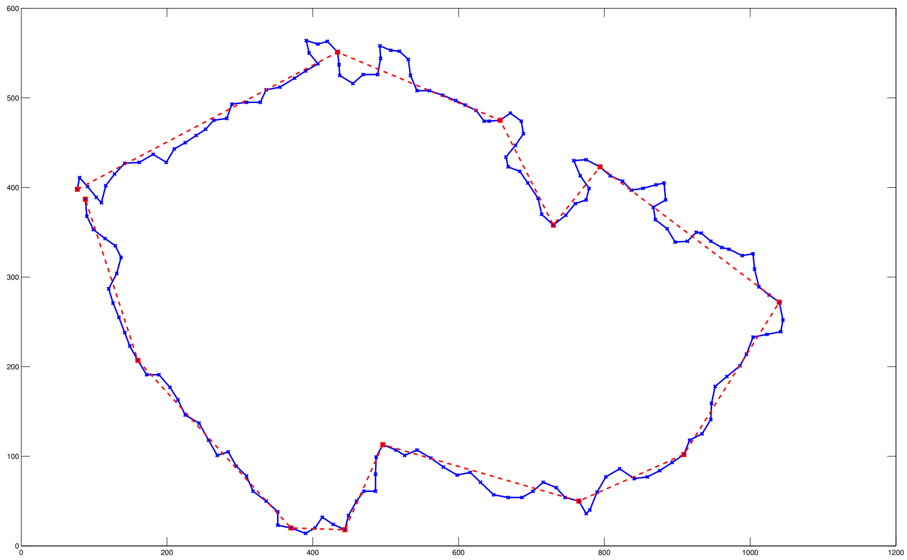

Figure 3: Entered points (blue) and selected points (red).

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall; United States Ed edition, 1993.