

Combinatorial optimization

CoContest semester project assignment: Sewer Design Problem

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

April 3, 2019

Abstract

This document introduces the assignment for the CoContest semester project.

1 Motivational example

It is the year 2080 and the sewer network that was renewed back in 2020 with the help of structural EU funds is now in a poor shape due to the constantly growing number of people living in cities. All the western countries except for the United Kingdom left the EU to establish the EU 2.0 (*Faster European Union*), whereas the UK is still a part of the EU, being in the process of negotiation of a better exit agreement. Meanwhile, the eastern countries are not able to renovate the sewer network for all the citizens on their own.

Therefore, European Parliament in Prague decides that only noble people who are in favor of the current regime will be a part of the new sewer network. The Parliament asks members of Industrial Informatics Research Center to find the cheapest way to build the new network on the base of the old one. Hence, options for the segments connecting different sites are limited and given. Rebuilding of each sewer segment has different price, and the individual segments are connected together at the sites. The task is to select segments of sewer network to rebuild, so that each site with a noble is connected to the new network with the minimum overall expenses. Moreover, the new sewer network should be connected, as it is planned to install just a single wastewater treatment plant.

2 Sewer Design Problem - formal problem statement

Let $G = (V, E, c)$ be an undirected weighted graph describing the old sewer network. Let $c : E \rightarrow \mathbb{N}_0$ be the cost of rebuilding the sewer segment. Let $T \subseteq V$ be a non-empty set of the noble sites.

The goal is to determine a new sewer network G^* (i.e., a subgraph of the old network G) such that:

$$G^* = \arg \min_{G' \subseteq G} \sum_{e \in E(G')} c(e) \quad (1)$$

such that

$$t \in V(G') \quad \forall t \in T \quad (2)$$

$$G' \text{ is connected} \quad (3)$$

3 Rules

If you decide to choose the contest as your semestral project, then you are expected to implement a correct solver for the Sewer Design Problem. The implementation will be submitted to BRUTE <https://cw.felk.cvut.cz/brute/> where it will be automatically evaluated (number of submissions is not limited). The grading is combination of ability of finding good solutions and the achieved rank relative to other students (w.r.t. the objective function). Therefore, you can acquire some minimum number of points even if your solver is not very efficient relative to other students.

In BRUTE, you will find 3 tasks related to the contest. Each task has specific instances, rules and grading. The contest is split into different tasks so that we avoid re-evaluation of the instances (which is time-consuming) and so that you can implement specific solver for each task.

1. **SP_CC_0**: you have to implement an exact, MILP solver for the problem. If your solver solves optimally all the instances in this task, then you will get 3 points for this task. If the solver returns suboptimal solution for **any** instance in this task, then the evaluation of your solver is stopped and you will get 0 points in this task.
2. **SP_CC_T**: the goal is to find the best possible feasible solution within the specified time limit, i.e., the optimal solutions are not required and you are encouraged to implement clever heuristics solving these instances. For each instance in this task, you will obtain some fraction of the point if the cost of the sewer network in your solution is not worse than our threshold (5 points at max).
3. **SP_CC_R**: similarly as in **SP_CC_T**, in this task we are also interested in finding the best possible feasible solution within the specified time limit. However, the evaluation of your solver will depend on how good your solver is relative to other students' solvers, i.e. the number of points obtained will depend on your rank (3 points at max).

Some general contest rules also apply

1. usage of single-purpose problem-specific solvers is prohibited (i.e., a MILP solver is allowed, but somebody's else code for solving the Sewer Design Problem is not).
2. every participant is required to write its own code. However, sharing ideas and other discussion about the problem is encouraged

4 Input and Output Format

In **SP_CC_0**, your solver will be called as

```
$ ./your-solver PATH_INPUT_FILE PATH_OUTPUT_FILE
```

whereas in **SP_CC_T** and **SP_CC_R** we include a time-limit

```
$ ./your-solver PATH_INPUT_FILE PATH_OUTPUT_FILE TIME_LIMIT
```

- **PATH_INPUT_FILE** and **PATH_OUTPUT_FILE**: similarly as in homeworks, these parameters represent the path to the input and output files, respectively (see below for description of the file formats).
- **TIME_LIMIT**: a float representing the time-limit in seconds given to your solver. Your solver will be killed after the time-limit is reached and you will be awarded with 0 points. Hence, the output of your solver is considered only if your program exits with status code 0 before it timeouts.

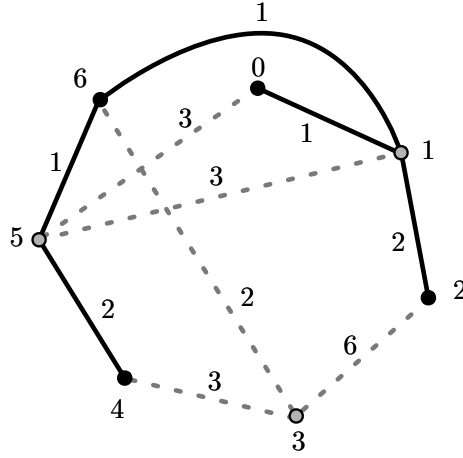


Figure 1: Example instance and an optimal solution.

The input file has the following form (we use one space as a separator between values on one line)

```

n    m
i1 j1 c1
i2 j2 c2
i3 j3 c3
⋮   ⋮   ⋮
im jm cm
t1 t2 ... t|T|

```

where $n = |V|$, $m = |E|$, $e_k = \{i_k, j_k\} \in E$, $c(e_k) = c_k$ and $\{t_1, t_2, \dots, t_{|T|}\} = T$. Furthermore, we ensure that given graph $G = (V, E)$ is connected.

The output file has the following format

```

obj
i'1 j'1
i'2 j'2
⋮   ⋮
i'k j'k

```

where obj is optimal objective value and $\{i'_k, j'_k\} \in E(G^*)$ is an edge of graph G^* .

Example 1

Input:

```

7 10
0 1 1
0 5 3
1 2 2
1 5 3
1 6 1
2 3 6
3 4 3
3 6 2

```

4 5 2
5 6 1
0 2 4 6

Output:

7
0 1
1 2
5 4
1 6
6 5