Combinatorial Optimization
CoContest Semester Project Assignment:
Sewer Design Problem

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

February 17, 2022

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 2023 with the help of structural EU funds is now in poor shape due to the constantly growing number of people living in cities. Once the United Kingdom showed that leaving the EU is actually possible, all rich western countries left soon after and established the EU 2.0 (*Faster European Union*). Meanwhile, the eastern countries are not able to renovate the sewer network for all the citizens on their own.

Therefore, European Parliament (1.0) in Prague decided that only important people will be connected to the renovated sewer network. The parliament asked the members of Industrial Informatics Research Center to find the cheapest way to build this new renovated network on the base of the old one. Hence, options for the segments connecting different sites are limited and given by the old network. The rebuilding of each sewer segment has a different price, and the individual segments are connected together at the sites. The task is to select segments of the sewer network to rebuild so that each site with important people is connected to the new network with minimum overall expenses. Moreover, the whole renovated sewer network should be connected, as it is planned to install just a single wastewater treatment plant.

2 Formal Problem Statement

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

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

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

such that

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

$$G' \text{ is connected} \tag{3}$$

3 Rules

If you choose the contest as your semestral project, you are expected to implement a correct solver for the Sewer Design Problem. BRUTE https://cw.felk.cvut.cz/brute/ will be used to evaluate it automatically. The number of submissions is not limited. The grading combines the ability to find optimal solutions for testing instances and the achieved rank relative to other students (w.r.t. the objective function) on competition instances. Therefore, you can acquire some points even if your solver is not very efficient relative to other students.

In BRUTE, you will find 2 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 a specific solver for each task.

1. `SP_CC_O`: You have to implement an exact MILP solver for the problem. If your solver solves all the instances optimally in this task, then you will get 4 points for this task. If the solver returns a 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_R`: The goal is to find the best possible feasible solution within the specified time limit, i.e., optimal solutions are not required, and you are encouraged to implement clever heuristics solving these instances. 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 (5 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 their own code. However, sharing ideas and discussion about the problem is encouraged.

4 Input and Output Format

In `SP_CC_O`, your solver will be called as

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

whereas in `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 a 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 0 points. Hence, the output of your solver is considered only if your program exits with status code 0 before it timeouts.

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

$$
\begin{array}{cccc}
n & m & & \\
i_1 & j_1 & c_1 & \\
i_2 & j_2 & c_2 & \\
i_3 & j_3 & c_3 & \\
\vdots & \vdots & \vdots & \\
i_m & j_m & c_m & \\
t_1 & t_2 & \dots & t_{|T|}
\end{array}
$$

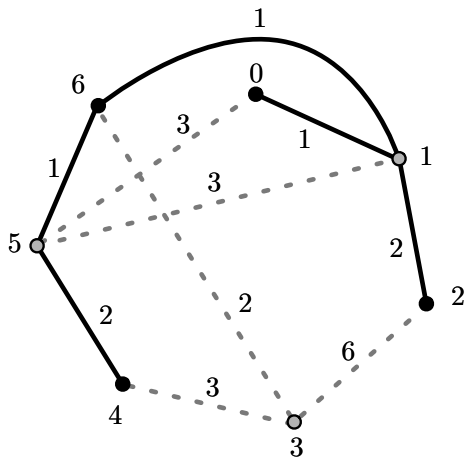

Figure 1: Example instance and an optimal solution.

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

The output file has the following format

obj
$i'_1 \quad j'_1$
$i'_2 \quad j'_2$
$\vdots \quad \vdots$
$i'_k \quad j'_k$

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

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
```