Assignment 1: Crew Scheduling

Transportation and Scheduling (FEM21043) – January 2022

January 4, 2022

After having determined a timetable and rolling stock schedule, a railway operator aims to schedule the crew. Each crew member operates from one of the *crew depots* in the railway network. Denote the set of depots by K. A crew member from depot $k \in K$ should start and end his/her day of work at depot k. The work to be scheduled is represented by a set of tasks. Each *task* has a start station, an end station, a start time, and an end time. We assume that each depot $k \in K$ corresponds to a station where tasks can start and end.

The aim of the crew scheduling stage is to assign the tasks to the crew of the railway operator. The tasks that are assigned to a specific crew member define a *duty* for that crew member. Such a duty is a sequence of tasks, which are to be operated consecutively by the crew member. In particular, this means that for each pair of consecutive tasks in a duty, the start station of the second task is equal to the end station of the first task, and the start time of the second task is strictly later than the end time of the first task. Furthermore, each duty should end at the depot where it has started and must respect a maximum duty duration. An example of a duty is provided in Figure 1 below. This duty starts and ends at the depot at station A.

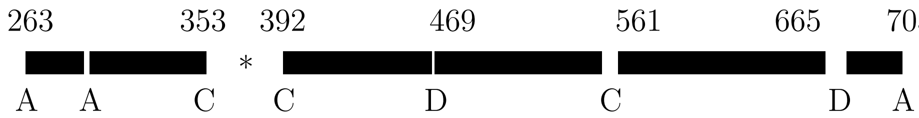

Figure 1: An example of a duty. The rectangles represent the tasks in the duty. The start and end stations of the tasks are given below the tasks, and the start and end times (in minutes after midnight) are given above the tasks. This duty contains a break, which is indicated by the $*$

We impose the additional requirement that the average duration of all duties must be at most 7.2 hours. The crew scheduling problem can then be formulated mathematically as a set covering problem with one side constraint. Here, we assume that the set of duties, D, is given. Each duty $d \in D$ has a cost coefficient c_d and a binary parameter a_{id} for each task $i \in I$ that specifies whether the task is covered by the duty ($a_{id} = 1$) or not ($a_{id} = 0$). Additionally, the parameter ℓ_d denotes the duration of the duty. Defining the decision variable x_d that equals 1 if duty d is selected in the solution and 0 otherwise, the crew scheduling problem (CSP) can be formulated as follows.

$$\text{(CSP)} \quad \min \sum_{d \in D} c_d x_d$$

such that

$$\begin{aligned} \sum_{d \in D} a_{id} x_d \geq 1 & \qquad \forall i \in I, \\ \sum_{d \in D} \ell_d x_d \leq 7.2 \sum_{d \in D} x_d, & \\ x_d \in \mathbb{N}_0 = \{0, 1, \ldots\} & \qquad \forall d \in D. \end{aligned}$$

The first constraint ensures that each task is covered by at least one duty. The second constraints imposes the maximum average duration. The third constraint sets the range of the decision variables.

We impose the following requirements on the duties in D. First of all, a duty can last at most 9.5 hours. Thus, the difference of the end time of the last task in the duty and the start time of the first task in the duty can be at most 9.5 hours. (The difference is allowed to equal exactly 9.5 hours.) Furthermore, each duty should contain a break of at least 30 minutes. To ensure that the break is roughly in the middle of the duty, the duration before the break and after the break can be at most 5.5 hours. The cost of the duty is given by a constant cost of 900 and a variable cost of 60 per hour.

Two instances of the crew scheduling problem are provided: a small and a large instance. The small instance contains 400 tasks and 2 depots (corresponding to stations A and B), the large one contains 1000 tasks and 3 depots (corresponding to stations A, B, and C).

In this assignment, you are asked to solve the crew scheduling problem, using a column-generation-based heuristic. In particular, you are asked to answer the following questions.

a. First, discard the requirement for duties to contain a break. Consider the LP-relaxation of the CSP and assume it is solved by column generation. Describe the pricing problem as a shortest path problem with additional constraints. Develop an algorithm to solve this pricing problem. What is the running time of your algorithm? (Is it polynomial?) Also discuss how many columns (or: duties) are found when the pricing problem is solved by your algorithm, and how many you would add to the master problem.

b. Implement the column generation algorithm to solve the LP-relaxation of the CSP. Report the solution value and computation time for the small and large instance. Also measure and report the computation time spent in the master and in the pricing problem, respectively. For both instances, provide a text file that describes the solution of the LP-relaxation. (See the requirements on the text file below.)

c. Suggest a heuristic to find an integer solution for the CSP. Describe and implement this heuristic and report the solution value and computation time for the small and large instance. For both instances, provide a text file that describes the solution of the CSP. (See the requirements on the text file below.)

d. Now, do consider the requirement for duties to contain a break. Answers questions a., b., and c. again, now incorporating the break requirement in the pricing problem of the column generation algorithm.

e. We now aim to improve the time required to solve the LP-relaxation of the CSP. Explain some possibilities to solve the LP-relaxation faster, implement them, and report whether they indeed improve the overall computation time.

Your answers have to be handed in before Wednesday, January 12, 2022, at 18:00 via Canvas. You should hand in the following files.

- 1 PDF-file of at most 4 pages with the answers to the above questions.
- 1 zip-file containing all the code that you have used.
- 8 text files, each with a solution of the LP-relaxation of the CSP or the CSP itself. Each line in these text files should describe a duty that is selected in the solution. In particular, it should contain the value of x_d in the solution, the costs c_d, the duration ℓ_d, and a list of the indices of tasks in the duty, for each duty $d \in D$ with $x_d > 0$. Thus, each line is of the following form:

  ```
  0.45 1340 440 4 13 53 80 116 157
  ```

  This line represents a duty that has costs 1340 and lasts for 440 minutes. It covers the tasks 4, 13, 53, 80, 116, and 157. Its solution value is 0.45. This duty corresponds to the small instance and is depicted in Figure 1. The following files should be submitted.

  – `small_noBreak_lp.txt`
  – `small_noBreak_ip.txt`
  – `large_noBreak_lp.txt`
  – `large_noBreak_ip.txt`
  – `small_break_lp.txt`
  – `small_break_ip.txt`
  – `large_break_lp.txt`
  – `large_break_ip.txt`