Exercises for Track Allocation

Exercise 1: Use ZIMPL to implement the track allocation problem as an integer program with standard packing constraints, as in model (APP) below. Let

- ▷ J be the set of tracks,
- ▷ I be the set of trains,
- ▷ A be the set of time-expanded arcs representing the track usage of a train,
- ▷ V be the set of time-expanded nodes representing departure or arrival events of a train,
- ▷ $\sigma : A \mapsto J$ a mapping of each time-expanded arc $uv \in A$ to track $j \in J$,
- ▷ $\tau : V \mapsto \{0, 1, \ldots\}$ a map, which maps each time-expanded node $v \in V$ to an event time.

Furthermore, each train has a type, either passenger ("PT") or cargo train ("CT"), an origin s_i and destination station t_i. You will find this data in files *arcs.dat* and *trains.dat*.

Each $uv \in A$ corresponds to exactly one train $i \in I$, which defines a disjunctive partition of the set of arcs $A = \bigcup_{i \in I} A_i$ and the set of nodes $V = \bigcup_{i \in I} V_i$, i.e., the subdigraph $D_i = (V_i, A_i)$ defines the routing possibilities of train $i \in I$. For simplicity we assume that a train occupies a track uv until reaching station v, i.e. for the interval $[\tau(u), \tau(v))$. Hence, we say two trains are in conflict on track $j = uv \in J$, if a pair of arcs u_1v_1, $u_2v_2 \in A$ temporary overlap, i.e. if $\sigma(u_1v_1) = \sigma(u_2v_2) = j$ and $[\tau(u_1), \tau(v_1) - 1] \cap [\tau(u_2), \tau(v_2) - 1] \neq \emptyset$. This defines conflict sets C as some subsets of 2^A.

$$
\begin{array}{lllll}
\text{(APP)} & \text{(i)} & \max \sum_{a \in A} w_a x_a & & \\
 & \text{(ii)} & \sum_{a \in \delta_{out}(v)} x_a - \sum_{a \in \delta_{in}(v)} x_a & = 0, & \forall i \in I, v \in V_i \backslash \{s_i, t_i\} \\
 & \text{(iii)} & \sum_{a \in \delta_{out}(s_i)} x_a & \leq 1, & \forall i \in I \\
 & \text{(iv)} & \sum_{a \in c} x_a & \leq 1, & \forall c \in C \\
 & \text{(v)} & x_a & \in \{0, 1\} & \forall a \in A
\end{array}
$$

By $\delta_{in}(v)$ we denote the set of incoming arcs $uv \in A$ for $v \in V$, by $\delta_{out}(v)$ the set of outgoing arcs, respectively. Finally, additional data to evaluate the train schedule is given, as follows:

- ▷ `revenue_pt := 1000` (revenue of a scheduled passenger train)
- ▷ `revenue_ct := 5000` (revenue of a scheduled cargo train)
- ▷ `cost_per_minute_pt := 1` (cost for track usage per minute for passenger trains)
- ▷ `cost_per_minute_ct := 2.5` (cost for track usage per minute for cargo trains)

(a) Read in all needed data and define the model (APP), see file *app.zpl*.

(b) Formulate the objective function w, which maximizes the total revenue.

(c) In a first step use sets for all pairs of arcs which are in conflict to formulate constraints (iv).

(d) Strengthen the model by constructing only constraints (iv) for the (inclusion-)maximal conflict sets.

(e) Solve both instances using SCIP, what happens (#branch and bound nodes, #variables and #constraints after presolving, ...) ?

(f) Change the model so that passenger trains are always scheduled, optimize again and analyze the timetable changes (number of scheduled cargo trains).

Exercise 2: Implement in Zimpl an extended formulation of the packing model (APP). The set of auxiliary arcs B is given in file *yarcs.dat*. Each arc $b = u_b v_b \in B$ corresponds to exactly one track $j = uv \in J$; the set of all arcs on track j induce a track digraph $D_j = (V_j, B_j)$. Using the set A from exercise 1, the sets A_b are defined as the arcs of A, which have to be coupled. Let for each $b = u_b v_b \in B$, A_b be the subset of time expanded arcs $u_a v_a \in A$ with $\sigma(u_b v_b) = \sigma(u_a v_a)$, $\tau(u_b) = \tau(u_a)$ and $\tau(v_b) = \tau(v_a)$, i.e. which means all arcs in A_b use the same track for the same interval. Finally, the coupling model (ACP) can be formulated as:

$$
\text{(ACP)} \quad
\begin{array}{rrll}
\text{(i)} & \max \sum\limits_{a \in A} w_a x_a & & \\
\text{(ii)} & \sum\limits_{a \in \delta_{out}(v)} x_a - \sum\limits_{a \in \delta_{in}(v)} x_a & = 0, & \forall i \in I, v \in V_i \backslash \{s_i, t_i\} \\
\text{(iii)} & \sum\limits_{a \in \delta_{out}(s_i)} x_a & \leq 1, & \forall i \in I \\
\text{(iv)} & \sum\limits_{b \in \delta_{out}(v)} y_b - \sum\limits_{b \in \delta_{in}(v)} y_b & = 0, & \forall j \in J, v \in V_j \backslash \{s_j, t_j\} \\
\text{(v)} & \sum\limits_{b \in \delta_{out}(s_i)} y_b & = 1, & \forall j \in J \\
\text{(vi)} & \sum\limits_{a \in A_b} x_a - y_b & \leq 0, & \forall b \in A \cap B \\
\text{(vii)} & x_a & \in \{0, 1\} & \forall a \in A \\
\text{(viii)} & y_b & \in \{0, 1\} & \forall b \in B
\end{array}
$$

(a) Read in the additional data and define the model (ACP), see file *acp.zpl*.

(b) Solve again both instances (with or without fixed passenger trains) of exercise 1 using SCIP.

(c) Compare the sizes of all three models (APP (c), APP (d) and ACP) and analyze the SCIP solution process.