

Computational mixed-integer programming 2nd set of programming exercises: ILP modeling

The package `exercise02.tgz` contains instances to test and compare your models and template `zpl` files for the exercises.

In this set of exercises, we will consider the *Steiner tree problem* and two more realistic network design problems stemming from applications in telecommunication network design.

Exercise 1 – The Steiner Tree Problem

Task: Write a ZIMPL model for each of the following formulations of the Steiner tree problem. Then compare how the different formulations perform by solving them in SCIP.

The data parsing part is already implemented in the file `sndparser.zpl`. Use the sets and functions provided in `sndparser.zpl`. A detailed description of these can be found in the beginning of the file `sndparser.zpl`. The cost l_{ij} for installing edge $ij \in E$ is given by the function `l(i, j)`.

Use the parser `sndparser.zpl` to read the test instances as follows:

```
zimpl -o MyMILP -D file=Data/instance.txt sndparser.zpl MyModel.zpl
```

The above ZIMPL command will construct a `MyMILP.lp` file containing the MIP formulation that can then be passed to SCIP to solve the problem.

- a) The first model to implement is the formulation via undirected cuts. Use the file `ucut.zpl` as a template for this implementation and then test it with the test set `Data/pdh.txt`.

$$\begin{aligned} \min \quad & c^T x && \text{(UCUT)} \\ \text{s.t.} \quad & x(\delta(S)) \geq 1 && \forall S \subset V : \emptyset \neq S \cap T \neq T \\ & x_e \geq 0 && \forall e \in E \\ & \mathbf{x} \in \mathbb{Z}^E \end{aligned}$$

- b) Open the file `dcut.zpl` and model the formulation via directed cuts. The parser `sndparser.zpl` already constructs a set `S1 = {r}` containing a single-element $r \in T$ as the root node. Therefore, you only need to create the set $T' := T \setminus \{r\}$. Test your model with `Data/pdh.txt` and `Data/polska.txt` and compare with the first model.

$$\begin{aligned} \min \quad & c^T x && \text{(DCUT)} \\ \text{s.t.} \quad & y(\delta^+(S)) \geq 1 && \forall S \subset V : r \in S, S \cap T' \neq T' \\ & y_{(i,j)} + y_{(j,i)} \leq x_{ij} && \forall ij \in E \\ & y_{(i,j)} \geq 0 && \forall (i,j) \in A \\ & x_{ij} \geq 0 && \forall ij \in E \\ & \mathbf{x} \in \mathbb{Z}^E \end{aligned}$$

c) Now implement the (directed) multi-commodity flow formulation using the file `mcf.zpl`. Solve all given test instances using this model and check your solutions.

$$\begin{aligned}
 & \min c^T x && \text{(MCF)} \\
 \text{s.t.} \quad & \sum_{(i,j) \in \delta^+(i)} f_{(i,j)}^t - \sum_{(j,i) \in \delta^-(i)} f_{(j,i)}^t = \begin{cases} 1 & i = t \\ 0 & \text{otherwise} \end{cases} && \forall t \in T', i \in V \setminus \{r\} \\
 & f_{(i,j)}^s + f_{(j,i)}^t \leq x_{ij} && \forall s, t \in T', ij \in E \\
 & f_{(i,j)}^t \geq 0 && \forall t \in T', (i, j) \in A \\
 & x_{ij} \geq 0 && \forall ij \in E \\
 & \mathbf{x} \in \mathbb{Z}^E
 \end{aligned}$$

Optimal solution values

atlanta	4.5550000000e+06
cost266	5.6727000000e+05
di-yuan	1.8490000000e+05
france	1.0608000000e+04
nobel-eu	6.9620000000e+04
nobel-germany	2.2050000000e+04
nobel-us	5.1290000000e+04
pdh	4.8463500000e+05
polska	1.8420000000e+03

Exercise 2 – The Survivable Network Design Problem

Task: Formulate the Survivable Network Design Problem with edge connectivity requirements 1 and 2. Implement and solve your model using ZIMPL and SCIP for the given problem instances.

Formally, the Survivable Network Design Problem is given as follows:

{1, 2}-Survivable Network Design Problem

Given	undirected graph (V, E) with edge costs l_e for the edges $e \in E$ together with the connectivity requests $\rho_v \in \{1, 2\}$
Solution	edge set $E' \subseteq E$ such that (V, E') contains $\min\{\rho_s, \rho_t\}$ edge disjoint (s, t) -paths for each pair of distinct nodes $s, t \in V$.
Goal	minimize the total edge cost, i.e. $\min_{e \in E'} l_e$

All necessary sets and functions are constructed in `sndparser.zpl`. Assume that

$$\rho_v := \begin{cases} 2 & v \in T \\ 1 & v \in V \setminus T \end{cases} \quad \text{for all } v \in V.$$

The cost l_{ij} for installing edge ij is again given by the function `l(i, j)`.

Transform the undirected problem into a problem involving directed paths between the terminals and use a directed multi-commodity flow formulation similar to that for the Steiner tree problem. There are several variants to do this. A very good model is obtained as follows (you can use `snd.zpl` as a template for this exercise):

1. Pick a root node $r \in T = \{v \in V \mid \rho_v = 2\}$ (in ZIMPL, use the set $S1 = \{r\}$ created by `sndparser.zpl`, again) and create a unit-demand commodity from r to each node in $V \setminus \{r\}$. These commodities form the first commodity set. Add arc-flow variables and flow conservation constraints for these commodities.
2. Create a second commodity from r to each node in $T \setminus \{r\}$. These commodities form the second commodity set. Add extra arc-flow variables and flow conservation constraints for these commodities, too.
3. Add inequalities to ensure that, for each node $t \in T \setminus \{r\}$, the two commodities from r to t are routed along edge disjoint paths.
4. Create binary variables for each edge and inequalities ensuring that an edge is installed whenever some commodity flows across the corresponding arcs.
You get a stronger (but also larger) formulation by adding certain flows on anti-parallel arcs.

Optimal solution value

atlanta	1.6095000000e+07
cost266	1.7801100000e+06
di-yuan	3.5250000000e+05
france	3.8932000000e+04
nobel-eu	1.5270000000e+05
nobel-germany	2.2050000000e+04
nobel-us	2.2050000000e+04
pdh	1.0468340000e+06
polska	5.2120000000e+03

Exercise 3 – The Capacitated Network Design Problem

In this exercise, we consider an undirected version of the Capacitated Network Design Problem with modular edge capacities and edge setup costs given as follows:

Capacitated Network Design Problem

Given	undirected graph $G = (V, E)$ with edge setup costs l_e capacity u_{ek} and cost c_{ek} for all edges $e \in E$ and all capacity modules $k \in K$ directed commodities $C \subset V^2$ with demands d_{st} for each $(s, t) \in C$
Solution	integers y_{ek} for all $e \in E$ and $k \in K$ such that the edge capacities $x_e := \sum_{k \in K} u_{ek} y_{ek}$ permit a (fractional) multi-commodity flow routing all commodities, i.e., the edge capacity x_{ij} is at least as large as the sum of the directed flows on the two arcs (i, j) and (j, i) .
Goal	minimize the total edge setup cost plus the total capacity installation cost, i.e. $\min \sum_{e \in E: x_e > 0} l_e + \sum_{e \in E, k \in K} c_{ek} y_{ek}$

Task: Formulate the Capacitated Network Design Problem as a mixed-integer linear program. Implement and solve your model using ZIMPL and SCIP for the given problem instances.

All necessary sets and functions are constructed in `sndparser.zpl`.

To create a correct model, the following steps might be helpful:

1. Add (directed) arc-flow variables and flow conservation constraints for all commodities. Since these flows are not going to be aggregated we can interpret every commodity as a unit-demand

commodity.

2. Add inequalities to link the directed flow and the undirected capacity installation variables. If you used unit-demand commodities as above, you now have to multiply the flow value on each edge with the corresponding demand and then compare it with the corresponding $u_{ek}y_{ek}$.
3. Introduce extra binary variables z_e for all edges $e \in E$, expressing whether an edge is set up or not.

These variables can then be used to determine the total edge setup cost.

Some solutions

atlanta	1.1717500000e+08
di-yuan	2.4304000000e+06
pdh	1.2435812000e+07
polska	3.1024000000e+04

Exercise 4 – A realistic extension*

Now adapt your formulation of the Capacitated Network Design Problem in such a way that the underlying flow problem no longer permits an arbitrary fractional multi-commodity, but enforces a single-path confluent flow.

In a single-path confluent flow, flow towards a destination can only travel on one arc out of a node. So, if flows with different origins and a common destination merged in some intermediate node, then they can never split again and they must continue along the same path towards their common destination. For each destination, all flows to this destination must form a tree directed towards this destination.

Formulate your model using a disaggregated formulation for the confluent flow.

Remark: The single-path confluency requirement arises in applications, where the traffic routing is controlled in an automatic way by (decentralized) routing protocols that, for the sake of simpler network management, permit only a single outgoing arc towards each destination. Variants of such protocols are used in telecommunication networks, in postal and logistics networks, and in the freight car network of Deutsche Bahn, for example.