Università di Pisa

Department of Computer Science

Data Science and Business Informatics

Logistics Project

PrivateOnDemand

Simone Di Luna 544322

Academic year 2021/2022

Contents

1 Problem Statement

PrivateOnDemand S.p.a. wants to design a communication network to send private information "on demand". It has the set of nodes $N = \{\mathrm{A, B, C, D, E, F, G}\}$, which can be interconnected through the following set of oriented arcs $A = \{(\mathrm{A, B}), (\mathrm{A, C}), (\mathrm{C, E}), (\mathrm{B, D}), (\mathrm{C, D}), (\mathrm{D, E}), (\mathrm{D, F}), (\mathrm{E, G}), (\mathrm{E, F}), (\mathrm{F, G})\}$.[1]

The fixed costs (in euro) related to the activation of the network arcs are stored in the following vector $\boldsymbol{f} = [10\,000, 15\,000, 30\,000, 18\,000, 20\,000, 12\,000, 16\,000, 16\,000, 12\,000, 10\,000]^T$. The variable costs (in €/Mbps) associated to data routing along the network links are estimated to be $\boldsymbol{c} = [300, 150, 500, 300, 500, 350, 200, 250, 300]^T$. Once activated, the network arcs have the following capacities in terms of Mbps $\boldsymbol{u} = [10, 20, 10, 10, 20, 7, 8, 20, 17, 15]^T$.

The database containing the information is located in node A, while the client to be served is located in node G. The connection request from A to G is 15 Mbps.

PrivateOnDemand S.p.a. wants to evaluate what is the minimum cost required to connect the client in G to the database in A. Namely, it wants to design the communication network by deciding which arcs to activate and how to route data along the network so as to meet the demand for connectivity while respecting capacity constraints, and in such a way to minimize the total cost incurred.

The oriented graph $G = (N, A)$ depicted in Figure 1 summarizes all the information described above. The network contains only one demand node, G, having a positive request. The origin node A does not have an explicitly defined supply, however, since it has to serve only one client, we can assume that the supply matches the demand. All the other vertices are transshipment nodes, hence they have a null demand.

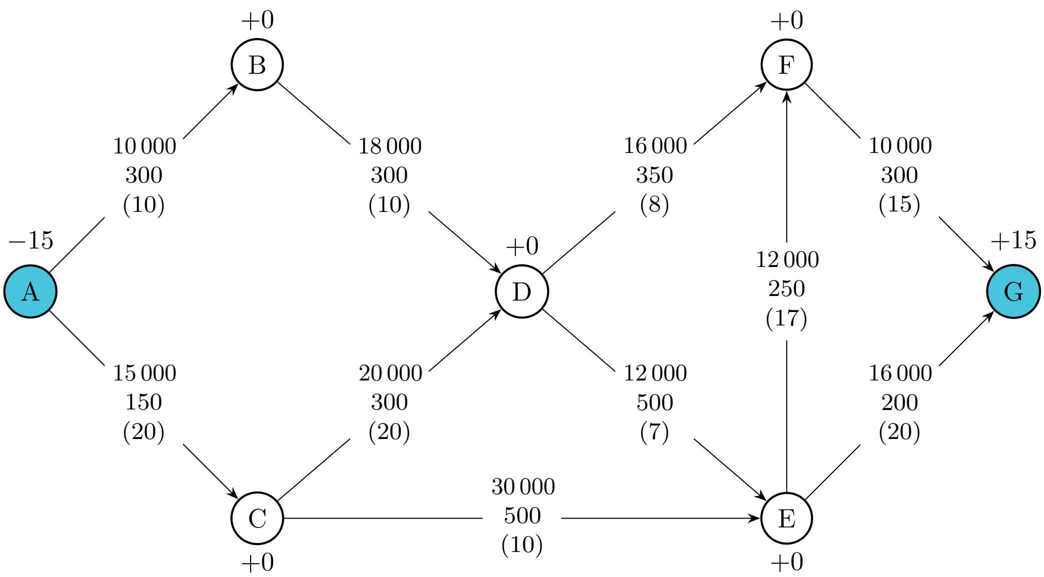

Figure 1: Logistics network of PrivateOnDemand. The database and client are colored in cyan. Over each link of the network are listed the related fixed activation cost, the variable cost and the capacity constraint (in parentheses).

[1] According to best practices and international standards of mathematical typography, we will use the italic typeface for sets, roman type for vertices, and boldface type for vectors.

2 Mathematical Formulation

To serve client G at the minimum cost, PrivateOnDemand needs to design the telecommunication network and organize the transportation of the information as efficiently as possible. Therefore, the problem can be considered an instance of the *fixed-charge network design problem*. Specifically, it is a *single-commodity capacitated fixed-charge network design problem*. A mixed-integer linear programming formulation of the problem is the following:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} f_{ij} y_{ij} + \sum_{(i,j)\in A} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = b_i & \forall i \in N \\
& 0 \le x_{ij} \le u_{ij} y_{ij} & \forall (i,j) \in A \\
& y_{ij} \in \{0,1\} & \forall (i,j) \in A
\end{aligned}
\tag{1}
$$

where the flow variables x_{ij} indicates the amount of information in terms of Mbps to be pushed along the arc (i, j), $\forall (i, j) \in A$. The design variables y_{ij} are defined as:

$$
y_{ij} = \begin{cases} 1 & \text{if arc (i,j) is activated} \\ 0 & \text{otherwise} \end{cases} \qquad \forall (i,j) \in A
$$

The first family of restrictions concerns flow conservation constraints: for each node $i \in N \setminus \{\mathrm{A}, \mathrm{G}\}$, it is imposed that the amount of incoming information net of outgoing one is equal to 0. On the other hand, nodes A and G have a net flow equal to the supply and the demand, respectively. In general, the balance of a node i of the network is defined as:

$$
b_i = \begin{cases} -15 & \text{if } i = \mathrm{A} \\ 15 & \text{if } i = \mathrm{G} \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in N
$$

The second set of constraints encapsulates three distinct types of restrictions: non-negativity, capacity, and linking constraints. Specifically, they ensure that the amount of information passing through each arc of the network is positive and does not exceed the capacity of the considered link. Moreover, when the edge (i, j) is not activated, the binary variable y_{ij} is equal to 0 and the flow variable x_{ij} must also be equal to 0, thus guaranteeing that neither fixed or variable costs are paid.

Eventually, the last family of constraints requires that the variables y_{ij} be binary.

Given the fixed f_{ij} and variable c_{ij} costs associated to each link of the network, for any feasible directed path T from A to G (meant as a sequence of edges respecting the problem constraints), the objective function computes the total cost as the summation between the telecommunication network design costs $\sum_{(i,j)\in T} f_{ij} y_{ij}$ and the information transmission costs $\sum_{(i,j)\in T} c_{ij} x_{ij}$.

By instantiating the model 1 with the PrivateOnDemand input data we obtain the following optimization problem:

$$
\begin{aligned}
\min \quad & 10\,000y_{\mathrm{AB}} + 300x_{\mathrm{AB}} + 15\,000y_{\mathrm{AC}} + 150x_{\mathrm{AC}} + 30\,000y_{\mathrm{CE}} + 500x_{\mathrm{CE}} + 18\,000y_{\mathrm{BD}} + \\
& + 300x_{\mathrm{BD}} + 20\,000y_{\mathrm{CD}} + 300x_{\mathrm{CD}} + 12\,000y_{\mathrm{DE}} + 500x_{\mathrm{DE}} + 16\,000y_{\mathrm{DF}} + 350x_{\mathrm{DF}} + \\
& + 16\,000y_{\mathrm{EG}} + 200x_{\mathrm{EG}} + 12\,000y_{\mathrm{EF}} + 250x_{\mathrm{EF}} + 10\,000y_{\mathrm{FG}} + 300x_{\mathrm{FG}} \\
\text{s.t.} \quad & -x_{\mathrm{AB}} - x_{\mathrm{AC}} = -15 \\
& x_{\mathrm{AB}} - x_{\mathrm{BD}} = 0 \\
& x_{\mathrm{AC}} - x_{\mathrm{CD}} - x_{\mathrm{CE}} = 0 \\
& x_{\mathrm{BD}} + x_{\mathrm{CD}} - x_{\mathrm{DE}} - x_{\mathrm{DF}} = 0 \\
& x_{\mathrm{CE}} + x_{\mathrm{DE}} - x_{\mathrm{EF}} - x_{\mathrm{EG}} = 0 \\
& x_{\mathrm{DF}} + x_{\mathrm{EF}} - x_{\mathrm{FG}} = 0 \\
& x_{\mathrm{EG}} + x_{\mathrm{FG}} = 15 \\
& 0 \le x_{\mathrm{AB}} \le y_{\mathrm{AB}} \\
& 0 \le x_{\mathrm{AC}} \le y_{\mathrm{AC}} \\
& 0 \le x_{\mathrm{CE}} \le y_{\mathrm{CE}} \\
& 0 \le x_{\mathrm{BD}} \le y_{\mathrm{BD}} \\
& 0 \le x_{\mathrm{CD}} \le y_{\mathrm{CD}} \\
& 0 \le x_{\mathrm{DE}} \le y_{\mathrm{DE}} \\
& 0 \le x_{\mathrm{DF}} \le y_{\mathrm{DF}} \\
& 0 \le x_{\mathrm{EG}} \le y_{\mathrm{EG}} \\
& 0 \le x_{\mathrm{EF}} \le y_{\mathrm{EF}} \\
& 0 \le x_{\mathrm{FG}} \le y_{\mathrm{FG}} \\
& y_{\mathrm{AB}}, y_{\mathrm{AC}}, y_{\mathrm{CE}}, y_{\mathrm{BD}}, y_{\mathrm{CD}}, y_{\mathrm{DE}}, y_{\mathrm{DF}}, y_{\mathrm{EG}}, y_{\mathrm{EF}}, y_{\mathrm{FG}} \in \{0, 1\}
\end{aligned} \tag{2}
$$

3 Model Implementation with AMPL

Model 1 was implemented in AMPL using the following code contained in the `.mod` file:

```
/* .mod */

set nodes;
set links within (nodes cross nodes);

param fixed_costs {links};
param var_costs {links};
param capacities {links};
param balances {nodes};

var flows {links} >= 0;
var activated {links} binary;

minimize tot_cost:
   sum {(i,j) in links} fixed_costs[i,j] * activated[i,j] +
```

```
    + sum {(i,j) in links} var_costs[i,j] * flows[i,j];

subject to flow_cons_constr {i in nodes}:
    sum {(j,i) in links} flows[j,i] - sum {(i,k) in links} flows[i,k] +
    - balances[i] == 0;

subject to capacity_constr {(i,j) in links}:
    flows[i,j] - capacities[i,j] * activated[i,j] <= 0;
```

Next, the model was instantiated with the PrivateOnDemand input data using the following block of code contained in the `.dat` file:

```
/* .dat */

set nodes := A B C D E F G;
set links :=
    (A, B)
    (A, C)
    (C, E)
    (B, D)
    (C, D)
    (D, E)
    (D, F)
    (E, G)
    (E, F)
    (F, G)
;

param balances :=
    A -15
    B 0
    C 0
    D 0
    E 0
    F 0
    G 15
;

param:     fixed_costs, var_costs, capacities :=
    A, B       10000        300        10
    A, C       15000        150        20
    C, E       30000        500        10
    B, D       18000        300        10
    C, D       20000        300        20
    D, E       12000        500         7
    D, F       16000        350         8
    E, G       16000        200        20
    E, F       12000        250        17
    F, G       10000        300        15
;
```

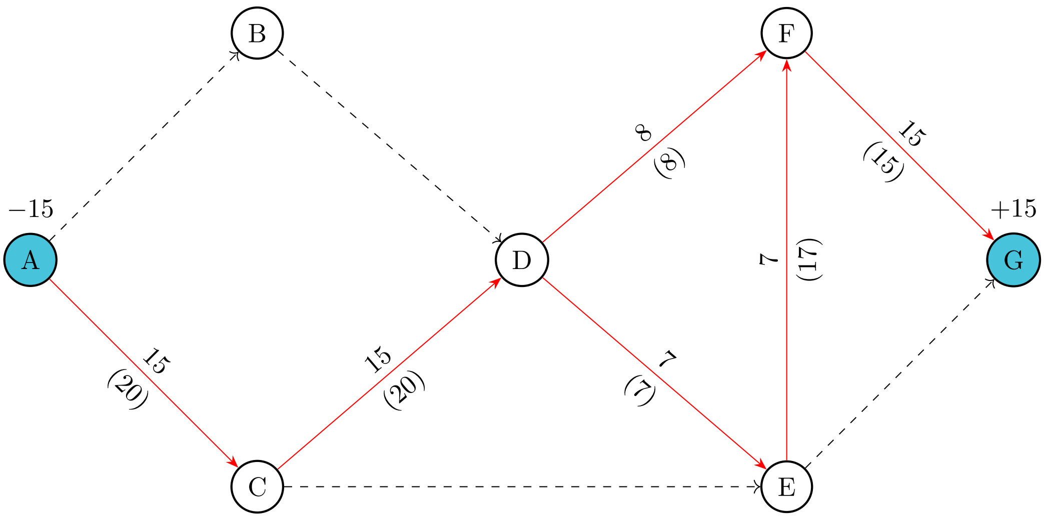

Figure 2: For each activated link of the optimal path (in red), the amount of flow passing through it and its capacity (in parentheses) is indicated. On the other hand, the dashed black lines represent the non-activated arcs.

The output produced was the following:

```
solve_message = 'CPLEX 20.1.0.0: optimal integer solution; objective 104300\
10 MIP simplex iterations\
0 branch-and-bound nodes'

:   flows activated  :=
A B    0       0
A C   15       1
B D    0       0
C D   15       1
C E    0       0
D E    7       1
D F    8       1
E F    7       1
E G    0       0
F G   15       1
;
```

The CPLEX solver obtained an optimal integer solution after 10 simplex iterations. The minimum cost that PrivateOnDemand has to pay in order to serve client G is €104 300. As suggested by the design variables, this is achieved by activating the following subset of arcs: $\{(A, C), (C, D), (D, E), (D, F), (E, F), (F, G)\}$. Moreover, Figure 2 shows the Mbps passing through each arch of the optimal path. It is interesting to note that the arcs $(D, E), (D, F), (F, G)$ are completely saturated. In particular, both the outgoing links from node D have a limited capacity and thus, taken individually, cannot handle the entire information flow required by the client in G. Put alternatively, the optimal solution does not provide a single path from A to G. We will address this "problem" in the next section.

4 Unique Path: Mathematical Formulation

In this section, we propose a possible modification of model 1 with the goal to find a unique optimal path from A to G. However, it should be noted that such a path cannot exist for the PrivateOnDemand problem. Indeed, to reach node G, we are forced to pass from one of the following three arcs: $(\mathrm{D}, \mathrm{E}), (\mathrm{D}, \mathrm{F}), (\mathrm{C}, \mathrm{E})$. As Figure 1 suggests, these links have a limited capacity of 7, 8, and 10 Mbps, respectively, hence individually they cannot handle all the 15 Mbps demanded by client G.

Anyway, generally speaking, to search for a unique path we have to add the following two family of constraints to model 1:

$$\begin{aligned} \sum_{(j,i)\in BS(i)} y_{ji} &\leq 1 \qquad \forall i \in N \\ \sum_{(i,j)\in FS(i)} y_{ij} &\leq 1 \qquad \forall i \in N \end{aligned} \tag{3}$$

that is, we force each vertex of the graph to have at most one incoming arc and at most one outgoing arc. Note that, since the origin node has an in-degree equal to zero, the first constraint of model 3 does not apply to A. Similarly, since the out-degree of the destination node is zero, the second constraint does not apply to G.

To instantiate the model we just need to add the following constraints to the optimization problem 2:

$$\begin{aligned} y_{\mathrm{AB}} &\leq 1 \\ y_{\mathrm{AC}} &\leq 1 \\ y_{\mathrm{BD}} + y_{\mathrm{CD}} &\leq 1 \\ y_{\mathrm{CE}} + y_{\mathrm{DE}} &\leq 1 \\ y_{\mathrm{DF}} + y_{\mathrm{EF}} &\leq 1 \\ y_{\mathrm{EG}} + y_{\mathrm{FG}} &\leq 1 \\ y_{\mathrm{AB}} + y_{\mathrm{AC}} &\leq 1 \\ y_{\mathrm{BD}} &\leq 1 \\ y_{\mathrm{CD}} + y_{\mathrm{CE}} &\leq 1 \\ y_{\mathrm{DE}} + y_{\mathrm{DF}} &\leq 1 \\ y_{\mathrm{EF}} + y_{\mathrm{EG}} &\leq 1 \\ y_{\mathrm{FG}} &\leq 1 \end{aligned} \tag{4}$$

5 Unique Path: Model Implementation with AMPL

To implement Model 3 in AMPL we used the following code:

```
/* .mod */

set nodes;
set origin_nodes within {nodes};
set destination_nodes within {nodes};
```

```
set not_origin_nodes = nodes diff origin_nodes;
set not_destination_nodes = nodes diff destination_nodes;
set links within (nodes cross nodes);

param fixed_costs {links};
param var_costs {links};
param capacities {links};
param balances {nodes};

var flows {links} >= 0;
var activated {links} binary;


minimize tot_cost:
    sum {(i,j) in links} fixed_costs[i,j] * activated[i,j] +
    + sum {(i,j) in links} var_costs[i,j] * flows[i,j];

subject to flow_cons_constr {i in nodes}:
    sum {(j,i) in links} flows[j,i] - sum {(i,k) in links} flows[i,k] +
    - balances[i] == 0;

subject to capacity_constr {(i,j) in links}:
    flows[i,j] - capacities[i,j] * activated[i,j] <= 0;

subject to unique_incoming_constr {i in not_origin_nodes}:
    sum {(j,i) in links} activated[j,i] - 1 <= 0;

subject to unique_outgoing_constr {i in not_destination_nodes}:
    sum {(i,j) in links} activated[i,j] - 1 <= 0;
```

To not have redundant restrictions, we did non applied the in-degree constraint to the origin node. For the same reason, we did not applied the out-degree constraint to the destination node. The instructions to instantiate the model are exactly the same described in the previous section with only the addition of specifying the origin and destination nodes:

```
set origin_nodes := A;
set destination_nodes := G;
```

As expected, the CPLEX solver was not able to find a feasible solution. Indeed it returned in output the following message:

```
presolve, constraint capacity_constr['D','F']:
    all variables eliminated, but upper bound = -8 < 0
Infeasible constraints determined by presolve.
solve_message = 'Infeasible constraints determined by presolve.'
```

If we truly want to find a unique path from A to G, we need to change the input data of the problem. As suggested by AMPL, the optimal unique path pass through (D, F). Therefore, one possible solution is to increase the capacity of the arc (D, F) to 15 Mbps. However, doing so, both models 1 and 3 will find the same optimal solution, that is, the unique path composed of the edges $\{(\mathrm{A}, \mathrm{C}), (\mathrm{C}, \mathrm{D}), (\mathrm{D}, \mathrm{F}), (\mathrm{F}, \mathrm{G})\}$. This is because the fixed costs are too high compared to the variable ones, thus resulting not convenient for the company to activate two more arcs—i.e.,

$\{(\mathrm{D},\mathrm{E}),(\mathrm{E},\mathrm{F})\}$ or $\{(\mathrm{D},\mathrm{E}),(\mathrm{E},\mathrm{G})\}$. However, if, for example, we assume to also reduce the design and transportation costs of links $\{(\mathrm{D},\mathrm{E}),(\mathrm{E},\mathrm{G})\}$ such that for the former they were respectively 500 and 30, while for the latter 1000 and 100, models 1 and 3 produce different optimal solutions:

```
/*Modified scenario: output model 1*/

ampl: include run_model.run
CPLEX 20.1.0.0: optimal integer solution; objective 77460
3 MIP simplex iterations
0 branch-and-bound nodes
:   flows activated   :=
A B    0       0
A C   15       1
B D    0       0
C D   15       1
C E    0       0
D E    7       1
D F    8       1
E F    7       1
E G    0       0
F G   15       1
;

/*Modified scenario: output model 2*/
Solution determined by presolve;
objective tot_cost = 77500.
solve_message = 'Solution determined by presolve;\
objective tot_cost = 77500.'

:   flows activated   :=
A B    0       0
A C   15       1
B D    0       0
C D   15       1
C E    0       0
D E    0       0
D F   15       1
E F    0       0
E G    0       0
F G   15       1
;
```

In this scenario, the arcs activated by model 1 in the optimal solution are the same obtained without modifying the input data (see Figure 3). Instead, the optimal solution of model 3 produces a unique path from A to G by activating only the links $\{(\mathrm{A},\mathrm{C}),(\mathrm{C},\mathrm{D}),(\mathrm{D},\mathrm{F}),(\mathrm{F},\mathrm{G})\}$ (see Figure 4) and producing an increment of the total cost of €40.

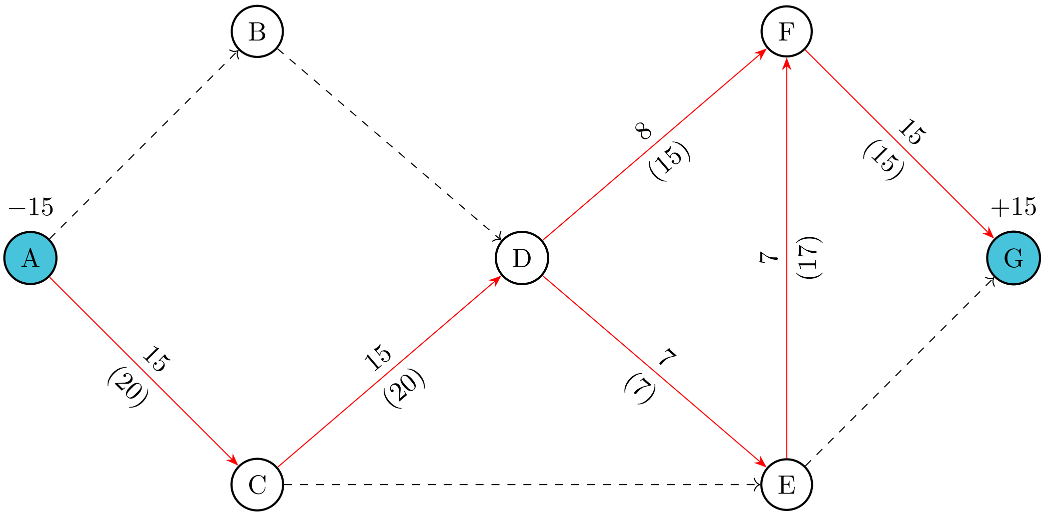

Figure 3: Optimal path obtained in the modified scenario using model 1

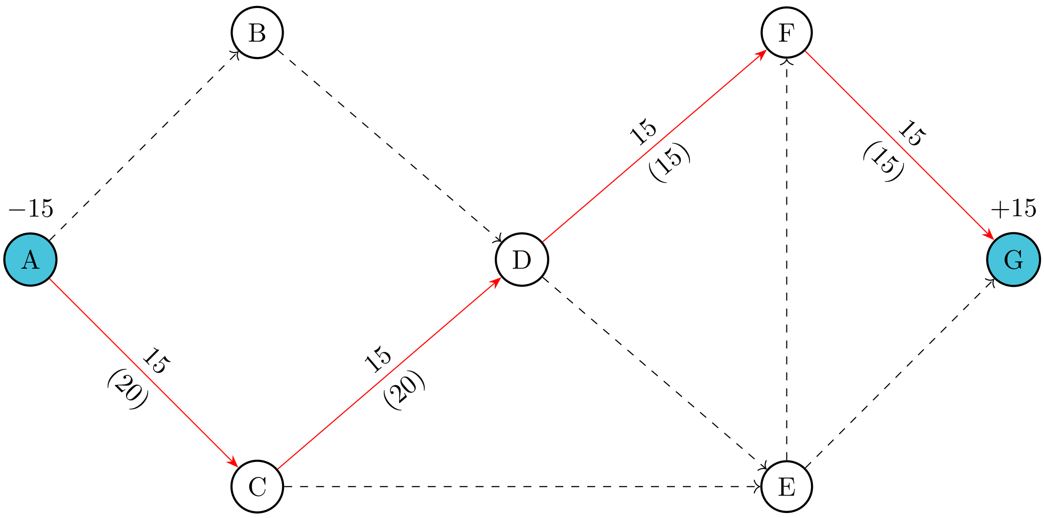

Figure 4: Optimal path obtained in the modified scenario using model 3