Course:	Telecommunication Network Design
Teacher:	Piotr Chołda piotr.cholda@agh.edu.pl
Studies:	Electronics and Telecommunications
Speciality:	Networks and Services
Semester:	2nd sem. MSc stud., Fall

Draft of the lecture

Piotr Chołda

November 22, 2017

1 Problems of Resource Allocation and Basic Dimensioning

1. Network design is based on the so-called multi-commodity flow problems.
2. Dimensioning problems, resource/flow allocation problems (typically combined with network flows routing), capacitated vs. uncapacitated network.
3. The simplest dimensioning problem — Uncapacitated Flow Allocation [and Dimensioning] Problem (UFAP) — linear programming optimization problem/task:
   - Indices:
     - ⋆ $e = 1, 2, \ldots, E$ arcs;
     - ⋆ $v = 1, 2, \ldots, V$ nodes;
     - ⋆ $d = 1, 2, \ldots, D$ demands, a demand from v_1 to v_2 is not the same as a demand from v_2 to v_1.
   - Constants:
     - ⋆ h_d volume of demand d;
     - ⋆ ξ_e unit/marginal cost of link e (cost of usage, leasing, installation), e.g., $\xi_4 = 2 \times 10^{-6}$ $/b/s;
     - ⋆ s_d source node of demand d;
     - ⋆ t_d sink/destination node of demand d;
     - ⋆ a_{ev} $= 1$ if arc e starts in node v; 0, otherwise;
     - ⋆ b_{ev} $= 1$ if arc e finishes in node v; 0, otherwise.
   - Variables:
     - ⋆ $x_{ed} \geq 0$ continuous flow realizing/satisfying demand d on arc e ($x_{ed} \in \mathbb{R}$);
     - ⋆ y_e continuous capacity to be used on arc e.
   - Goal function: $\min \sum_e \xi_e y_e$ (minimization of the capacity usage/leasing/installation cost).
   - Constraints:
     - ⋆ $\sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} = \begin{cases} h_d & \text{if } v = s_d \\ 0 & \text{if } v \neq s_d, v \neq t_d \\ -h_d & \text{if } v = t_d \end{cases}$

       for all NODEs: $d = 1, 2, \ldots, D$, $v = 1, 2, \ldots, V$ (flow conservation constraints, additionally they are demand constraints);

  - ⋆ $\sum_d x_{ed} = y_e$ for all LINKs (arcs): $e = 1, 2, \ldots, E$ (capacity constraints);
  - ⋆ **all** variables are **non-negative and continuous**.
- Constraints denotation $\sum_d x_{ed} = y_e \quad e = 1, 2, \ldots, E$ typically appears as $\forall_{e \in \{1,\ldots,E\}} \sum_d x_{ed} = y_e$ (in various books or papers), thus it represents E various constraints.
- Instead of $\sum_e a_{ev} x_{ed}$ we could also write $\sum_{i \in N: e=(v,i) \in A} x_{ed}$, where N is the set of nodes ($N = \{1, 2, \ldots, V\}$), and A is the set of arcs ($A = \{1, 2, \ldots, E\}$). In fact, we will prefer this approach while implementing problems in CPLEX.

4. Capacitated Flow Allocation Problem (CFAP):
   - Indices: (as before).
   - Constants:
     - ⋆ h_d (as before);
     - ⋆ s_d (as before);
     - ⋆ t_d (as before);
     - ⋆ a_{ev} (as before);
     - ⋆ b_{ev} (as before);
     - ⋆ ξ_e marginal cost of using of (leased/installed before) unit of capacity in arc e;
     - ⋆ c_e capacity installed in arc e.
   - Variables:
     - ⋆ x_{ed} (as before),
     - ⋆ y_e (as before).
   - Goal function: min $\mathbf{F}(\mathbf{y}) = \sum_e \xi_e y_e$ (in this case, we may be interested in finding feasible flows only, that is in finding only the feasible solutions of the optimization problem — then we can put anything in the goal function).
   - Constraints:
     - ⋆ $\sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} = \begin{cases} h_d & \text{if } v = s_d \\ 0 & \text{if } v \neq s_d, v \neq t_d \\ -h_d & \text{if } v = t_d \end{cases}$
       $d = 1, 2, \ldots, D,\ v = 1, 2, \ldots, V;$
     - ⋆ $\sum_d x_{ed} = y_e \qquad e = 1, 2, \ldots, E$
     - ⋆ $y_e \leq c_e \qquad e = 1, 2, \ldots, E;$
       (variables y_e are auxiliary, it is easier to present the goal function with them)
     - ⋆ all variables are non-negative and continuous.

5. Various types of formulations of flow allocation problems with use of linear programming LP: node-link formulation (N-L), aggregated node-link formulation (A/N-L), link-path formulation (L-P). Before, N-L was used:
   - Indices: d, e, $v\ldots$

- Variables: $x_{ed}, \ldots$
- Constraints:
  - ⋆ for all NODEs: $v = 1, 2, \ldots, V, \ldots$;
  - ⋆ for all LINKs: $e = 1, 2, \ldots, E$;
  - ⋆ ...

6. CFAP problem in the aggregated formulation A/N-L:

- Indices:
  - ⋆ $e = 1, 2, \ldots, E$ (as before);
  - ⋆ $v, w = 1, 2, \ldots, V$ nodes.
- Constants:
  - ⋆ h_{vw} volume of demand from source node v to destination node w;
  - ⋆ H_v $= \sum_{w \neq v} h_{vw}$ total volume of demands sourced in v;
  - ⋆ ξ_e (as before);
  - ⋆ a_{ev} (as before);
  - ⋆ b_{ev} (as before);
  - ⋆ c_e (as before).
- Variables:
  - ⋆ $x_{ev} \geq 0$ flow on arc e, which summarizes all flows satisfying demands sourced in v.
- Goal function: min $\sum_e \sum_v \xi_e x_{ev}$.
- Constraints:
  - ⋆ $\sum_e a_{ev} x_{ev} = H_v$
    AGGREGATED for all NODEs: $v = 1, 2, \ldots, V$;
  - ⋆ $\sum_e a_{ew} x_{ev} - \sum_e b_{ew} x_{ev} = -h_{vw}$
    AGGREGATED for all NODEs: $v, w = 1, 2, \ldots, V$, $v \neq w$;
  - ⋆ $\sum_v x_{ev} \leq c_e$ for all LINKs: $e = 1, 2, \ldots, E$.

7. CFAP problem with use of link-path (L-P, also known as arc-flow) formulation:

- Indices:
  - ⋆ $e = 1, 2, \ldots, E$ edges/links (here, not necessarily arcs);
  - ⋆ $d = 1, 2, \ldots, D$ demands;
  - ⋆ $p = 1, 2, \ldots, P_d$ candidate paths for flows satisfying demand d, a candidate path with a defined number p for a selected demand d is denoted as $\mathcal{P}_{dp}$ (e.g., $\mathcal{P}_{101,72}$ is related to the 72. candidate path for demand 101).
- Constants:
  - ⋆ δ_{edp} $= 1$ if link e is located in candidate path p satisfying demand d; 0, otherwise ($\delta_{edp} = 1 \Leftrightarrow e \in \mathcal{P}_{dp}$);
  - ⋆ h_d (as before);
  - ⋆ ξ_e (as before);
  - ⋆ c_e (as before).

- Variables:
  - ⋆ x_{dp} size of a flow satisfying demand d using path p;
  - ⋆ y_e (as before).
- Goal function: min $\mathbf{F}(\mathbf{y}) = \sum_e \xi_e y_e$.
- Constraints:
  - ⋆ $\sum_d \sum_p \delta_{edp} x_{dp} = y_e$ for all LINKs $e = 1, 2, \ldots, E$;
  - ⋆ $y_e \leq c_e$ for each of LINKs $e = 1, 2, \ldots, E$;
  - ⋆ $\sum_p x_{dp} = h_d$ $d = 1, 2, \ldots, D$ constraints describing demand realization with use of various PATHs (demand constraints);
  - ⋆ all variables are non-negative continuous.

8. Candidate paths:
   - In fact, we have $p(d) = 1, 2, \ldots, P_d$, thus precisely we should write $x_{d,p(d)}$, e.g., $x_{101,72(101)}$ (flow satisfying demand no. 101 at its candidate path no. 72.[1])
   - Each path satisfying demand d number p is simply a subset of links: $\mathcal{P}_{dp} \subseteq \{1, 2, \ldots, E\}$.

9. '$D+E$ property' for a solution of the CFAP problem (the property is clear for the L-P formulation). We should remember that in the formulation given above we use the redundant number of constraints — simply, we can present the goal function as $\mathbf{F}(\mathbf{y}) = \sum_e \sum_d \xi_e \delta_{edp} x_{dp}$, and the LINK constraint can be simply given as $\sum_d \sum_p \delta_{edp} x_{dp} \leq c_e$ (we do not need variables y_e).

10. UFAP given with the L-P formulation:
    - Indices and constants: (as before).
    - Variables:
      - ⋆ x_{dp} (as before);
    - Goal functionmin $\mathbf{F}(\mathbf{y}) = \sum_e \xi_e y_e$.
    - Constraints:
      - ⋆ $\sum_p x_{dp} = h_d$ $d = 1, 2, \ldots, D$;
      - ⋆ $\sum_d \sum_p \delta_{edp} x_{dp} = y_e$ $e = 1, 2, \ldots, E$.

    How to find the optimal solution without using the simplex method:
    - $\sum_p x_{dp} = h_d$ $d = 1, 2, \ldots, D$;
    - $\sum_d \sum_p \delta_{edp} x_{dp} = y_e$ $e = 1, 2, \ldots, E$;
    - Goal function: min $\mathbf{F}(\mathbf{y}) = \sum_e \xi_e y_e = \sum_e \xi_e \sum_d \sum_p \delta_{edp} x_{dp} =$
      $= \sum_d \sum_p \left(\sum_e \xi_e \delta_{edp}\right) x_{dp} = \sum_d \sum_p \kappa_{dp} x_{dp}$;
    - $\kappa_{dp} = \sum_e \xi_e \delta_{edp}$: cost of path $\mathcal{P}_{dp}$;

[1] But in the compact description given during the lecture we do not do like that (i.e., we skip to index indices). However, this precise description should be used while implementing various problems in CPLEX. In the general description, we skip commas between indices, too.

- The optimal solution is based on usage of the cheapest paths for each demand separately: $\mathbf{F}(\mathbf{y})|_{\min} = \sum_d \zeta_d h_d$;
- ζ_d: cost of the cheapest/shortest path (with respect to weights ξ_e) satisfying demand d;
- we can use any shortest-path algorithm — it is possible only due to the fact that in case of UFAP we can **decompose** the optimization problem to D independent problems.

11. Differences between various formulations:

- Number of variables and constraints is given in the table:

Formulation	**Number of variables** x	**Number of constraints N+L or L+P**
N-L	$\sim \frac{k\times V\times V\times(V-1)}{2} = \mathcal{O}(V^3)$	$\sim V\times V\times(V-1) + \frac{k\times V}{2} = \mathcal{O}(V^3)$
A/N-L	$\sim \frac{k\times V\times V}{2} = \mathcal{O}(V^2)$	$\sim V + V\times(V-1) + \frac{k\times V}{2} = \mathcal{O}(V^2)$
L-P	$\sim P\times V\times(V-1) = \mathcal{O}(V^2)$	$\sim \frac{k\times V}{2} + V\times(V-1) = \mathcal{O}(V^2)$

V: number of nodes, k: average node degree (when digraph is treated as a graph).

- The N-L formulation has to use arcs, while the L-P formulation can use links only (if demands use bi-directional connections and the demand volumes are symmetrical).
- The optimal solution of a problem given in the L-P formulation directly presents the values of flows (it is useful from the management and administrative viewpoint), while in the case of N-L formulations, the flow values are given indirectly and the solution should be processed to find the exact configuration of the optimal flows (in fact, it is necessary to solve a max-flow problem for each demand, where the capacity of link e is equal to the optimal value of x_{ed}).
- In the L-P formulation, we do not have to use all the possible candidate paths for demands (in the N-L formulation all of them are present indirectly), e.g., we can limit the length of the flows (it is important for optical networks).
- A 'path' used by the L-P formulations can represent any set of links (a tree, a cycle, or even a non-connected set of links) — it depends on the optimization problem which is dealt with.
- Looking for candate paths in L-P is not necessarily a trivial task, separated from solving the flow/resource allocation problem.
- Problems given in the L-P formulation are easier to decompose.

1.1 Exercises

1. A weighted digraph is given, the weights represent the capacities of links represented as arcs (see Figure 1). Find the volume of the maximum flow between s and t, and then formulate a linear programming LP problem to find this volume (indices, constants, variables, a goal function, and constraints should be adjusted to the given graph).

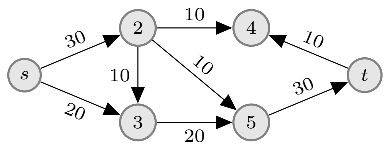

Figure 1: Weighted digraph related to exercise 1.

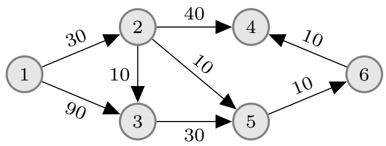

Figure 2: Weighted digraph related to exercise 2.

2. A weighted digraph (representing a network) is given, the weights represent the unit costs of capacity usage in links (see Figure 2). Formulate a linear programming LP problem to find the minimum cost flow allocation assuming that there are two demands in this network: the first one between nodes 1 and 6, and the second one between nodes 2 and 4. The traffic volumes to be carried for each of the demands equal 10. Apply the node-link N-L formulation; indices, constants, variables, a goal function, and constraints should be adjusted to the given graph. What is the optimal solution of this problem?

3. A weighted digraph (representing a network) is given, the weights put in brackets represent as follows: the first value — a unit cost of capacity usage in a link, the second value — the capacity available in a link (see Figure 3). Formulate a linear programming LP problem to find the minimum cost flow allocation assuming that there are two demands in this network: the first one between nodes 1 and 6, and the second one between nodes 2 and 4. The traffic volumes to be carried for each of the demands equal 10. Apply the link-path L-P formulation; indices, constants, variables, a goal function, and constraints should be adjusted to the given graph. What is the optimal solution of this problem?

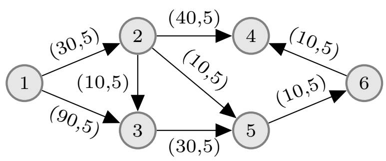

Figure 3: Weighted digraph related to exercise 3.

1.2 Reading

1.2.1 Contents of the lecture

Problems described in this lecture are generally dealt with in the following positions:

- Deepankar Medhi and Karthikeyan Ramasamy. *Network Routing. Algorithms, Protocols, and Architectures.* Morgan Kaufmann Publishers—Elsevier, San Francisco, CA, 2007: chapter 4.
- Michał Pióro and Deepankar Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Publishers—Elsevier, San Francisco, CA, 2004: chapter 4.1, 4.4-4.6, 5.1.1, 5.1.3, 5.1.4.

1.2.2 Auxiliary references

- Terje Jensen. Network Planning—Introductory Issues. *Telektronikk*, 99(3/4):9–46, 2003: network design from the viewpoint of the management plane.
- Michał Pióro and Deepankar Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Publishers—Elsevier, San Francisco, CA, 2004: basic network design problems.
- Poompat Saengudomlert. *Optimization for Communications and Networks.* CRC Press/Science Publishers, Boca Raton, FL, 2012: overview of optimization problems for telecommunication and computer networks.