Technische Universität Ilmenau 28. November 2023

Vorlesung

Netzalgorithmen

Prof. Dr.-Ing Günter Schäfer

NeXTeX

Vorlesungsmitschrift von
Adrian Schollmeyer

Inhaltsverzeichnis

1 Einführung

2 Modeling Network Design Problems

2.1 Modeling Design Problems in Link-Path Formulation

Preliminary notation for undirected demands:

- $x - y$ ...Link between node x and y
- $\hat{h}_{xy} = b$ ...Demand d between nodes x and y (i. e. Demand b bandwidth between nodes x and y)
- $\hat{x}_{v_1 v_2} = f$ ...Flow on the path between v_1 and v_2
- $\hat{x}_{v_1 v_2 v_3} = f$ ...Flow on the path v_1, v_2, v_3
- $\hat{c}_{v_1 v_2}$ ...Capacity of the link between v_1 and v_2

This generally yields equations for satisfying the demands and inequalities for limiting the bandwidth to the link capacities, e. g.

$$\hat{x}_{12} + \hat{x}_{132} = 5 \tag{2.1}$$
$$\hat{x}_{13} + \hat{x}_{123} = 7 \tag{2.2}$$
$$\vdots \tag{2.3}$$
$$\hat{x}_{132} + \hat{x}_{13} + \hat{x}_{213} \leq 10 \tag{2.4}$$
$$\hat{x}_{132} + \hat{x}_{123} + \hat{x}_{23} \leq 15 \tag{2.5}$$

As such a system usually has multiple solutions, an **objective function** is defined, which can then be optimized, e. g.:

$$F = \hat{x}_{12} + 2\hat{x}_{132} + \hat{x}_{13} + 2\hat{x}_{123} + \hat{x}_{23} + 2\hat{x}_{213} \tag{2.6}$$

In this example, flow routed over two links are weighted with factor two, so direct links are preferred.

Such an optimization is called a **multi-commodity flow problem**. The *optimal solution* to this problem is marked with an asterisk, e. g.:

$$\hat{x}^*_{12} = 5 \qquad \hat{x}^*_{13} = 7 \qquad \hat{x}^*_{23} = 8 \tag{2.7}$$

Important observations:

- Changing the objective function usually affects the optimal solution to a problem.
- Formulation a good objective function for the particular network is important for obtaining meaningful solutions.

2.2 Modeling Design Problems in Node-Link Formulation

In this section, links and demands are assumed to be directed. Undirected links $x - y$ are replaced with two directed links $x \to y$ and $y \to x$.

Notation:

- $v_1 \to v_2$ ...Directed link from node v_1 to node v_2
- $\langle v_1 : v_2 \rangle$ ...Demand from node v_1 to node v_2
- $\tilde{x}_{a,d}$ ...Flow over arc a for demand d (e. g. $\tilde{x}_{v_1 v_2, v_1 v_2}$ flow over arc $v_1 \to v_2$ for demand $\langle v_1 : v_2 \rangle$)

Backflows (e. g. $\tilde{x}_{21,12}$) are also possible, but make practically no sense, so they are set to 0.

2.3 Link-Demand-Path-Identifier-Based Notation

Demands and links are assigned label indexes. Thus, tables for mapping indexes to actual demands and links are required.

Notation:

- h_i ...Demand with index i
- c_e ...(Known) capacity of link e
- y_e ...*Unknown* capacity of link e
- P_i ...Number of candidate paths for demand i
- P_{ij} ...jth candidate path for demand i
- $(v_1, e_1, v_2, e_2, \ldots, e_n, v_{n+1})$ ...n-hop path
- $v_1 \to v_2 \to \ldots \to v_{n+1}$ ...node representation of a path (directed, use $-$ instead of $\to$ for undirected)
- $\{e_1, e_2, \ldots, e_n\}$ ...link representation of undirected paths
- $(e_1, e_2, \ldots, e_n)$ ...link representation of directed paths
- v ...Node
- e ...Link
- d ...Demand
- p ...Path
- V, E, D, P ...total numbers of the aforementioned items

- ξ_e ...Cost of alink e

Example equations:

$$x_{11} = 15 \tag{2.8}$$
$$x_{21} + x_{22} = 20 \tag{2.9}$$
$$x_{31} + x_{32} = 10 \tag{2.10}$$

Demands:

$$\sum_{p=1}^{P_d} x_{dp} = h_d \qquad d = 1, \ldots, D \tag{2.11}$$

Short form, when iterating over all candidate paths:

$$\sum_{p} x_{dp} = h_d \qquad d = 1, \ldots, D \tag{2.12}$$

The vector of all flows (path flow variables) is called the **flow allocation vector** or short **flow vector**:

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D) \tag{2.13}$$
$$= (x_{11}, x_{12}, \ldots, x_{1P_1}, \ldots, x_{D1}, x_{D2}, \ldots, x_{DP_D}) \tag{2.14}$$
$$= (x_{dp} : d = 1, 2, \ldots, D; p = 1, 2, \ldots, P_d) \tag{2.15}$$

Important: vectors are represented with bold letters $\mathbf{x}$ and scalar values are represented with normal letters x.

The relationship between links and paths is written down with the **link-path incidence relation** δ_{edp}:

$$\delta_{edp} = \begin{cases} 1 & \text{if link } e \text{ belongs to path } p \text{ for demand } d \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

δ_{edp} can be written as a table:

e	$P_{11} = \{2,4\}$	$P_{21} = \{5\}$	$P_{22} = \{3,4\}$
1	0	0	0
2	1	0	0
3	0	0	1
4	1	0	1
5	0	1	0

The **load** in link e can be written as:

$$\underline{y}_e = \underline{y}_e(\mathbf{x}) = \sum_{d=1}^{D} \sum_{p=1}^{P_d} \delta_{edp} x_{dp} \tag{2.17}$$

Important: The actual link loads $\underline{y}_e$ are determined by the path flow variables x_{dp} of a solution and are not the same as the link capacity variables y_e.

Cost of a path P_{dp}:

$$\zeta_{dp} = \sum_e \delta_{edp} \xi_e, \qquad d = 1, 2, \ldots, D \quad p = 1, 2, \ldots, P_d \tag{2.18}$$

Shortest-Path Allocation Rule for Dimensioning Problems For each demand, allocate its entire demand to its shortest path with respect to link costs and candidate paths. If there is more than one shortest path for a given demand, then the demand volume can be arbitrarily split amon the shortest paths.

2.4 Shortest-Path Routing

For shorted-path routing, demands will only be routed on their shortest paths. The path length is determined by adding up link costs w_e according to some weight system $\mathbf{w} = (w_1, w_2, \ldots, w_E)$.

Single Shortest Path Allocation Problem For given link capacities $\mathbf{c}$ and demand volumes $\mathbf{h}$, find a link weight system $\mathbf{w}$ such that the resulting shortest paths are unique and the resulting flow allocation vector is feasible.

Zusammenfassung fortführen

3 General Optimization Methdos for Network Design

3.1 Extreme Points and Basic Solutions

3.1.1 Phase 2

$$S(x) \text{ linear unabhängig, wenn } \sum_{i=1}^{k} \alpha_i \vec{x}_i = 0 \implies \forall i \in \{1, \dots, k\} : \alpha_i = \vec{o} \tag{3.1}$$

3.2 The Simplex-Algorithm

Note: Solving the system of equations means transforming the target vector into the basis of the input vector.

If all d_i are non-negative, the basic solution ist the optimal solution as the $d_i y_i$ are always subtracted and y_i is always non-negative, thus positive d_i can only decrease the result and not increase it.

$\vec{z}_k$ is non-negative for all three cases as $t_{k,j}$ is non-positive and $\alpha > 0$. For items that are not j and not part of the base, the result is 0 and thus it can be removed from the sum.

If we find that all d_j are negative, the objective function is unbounded from above.

In case 3, ε is always positive due to the way it is defined and the constrains on the values in this case. As ε is the minimum of the given set, it can not be < 0.

Thus, the resulting vector $\vec{z}$ can be in the feasible set.

Basically, we have walked from one corner to the next and restarted the algorithm. In each calculation, we display $\vec{b}$ in another basis. Thus, we need to watch out for cycling between multiple results. This is what Bland's rule is used for.

3.2.1 Phase 1

For distinction, a new variable vector $\vec{y}$ is introduced. This is later transformed by extending $\vec{x}$. All y_i are weighted 1.

This auxiliary problem can be solved by setting $\vec{x} = \vec{o}$.

Note that the auxiliary problem is to minimize, not maximize the result. As all y_i need to remain positive, the algorithm will approach $\forall i : y_i = 0$ eventually.

3.2.2 Simplex Tableaus

Page 35 last line: $\vec{\vec{b}}$ is actually $\vec{b}$...

Example 1:

$$\max x_1 + x_2 \tag{3.2}$$

such that:

$$-x_1 + x_2 \leq 1 \tag{3.3}$$
$$x_1 \leq 3 \tag{3.4}$$
$$x_2 \leq 2 \tag{3.5}$$
$$x_1, x_2 \geq 0 \tag{3.6}$$

This needs to be transformed from canonical form into the standard form. For this, additional variables $x_3, x_4, \ldots$ need to be introduced:

$$-x_1 + x_2 + x_3 = 1 \tag{3.7}$$
$$x_1 + x_4 = 3 \tag{3.8}$$
$$x_2 + x_5 = 2 \tag{3.9}$$
$$\vec{x} \geq \vec{o} \tag{3.10}$$

A basic solution can be obtained directly:

$$x_1 = 0 \tag{3.11}$$
$$x_2 = 0 \tag{3.12}$$
$$x_3 = 1 \tag{3.13}$$
$$x_4 = 3 \tag{3.14}$$
$$x_5 = 2 \tag{3.15}$$

From this, the current maximum value is $z = 0$.

We now work with the simplex tableau.

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & b \\ -1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array} \tag{3.16}$$

The last row represents the objective function and the last cell if the last row is z.

This matrix can now be transformed using Gauss-Jordan elimination. During elimination, we need to ensure that $\vec{b}$ remains positive!

We apply Blant's rule and find the first column where the value in the last row is non-zero. Then, we try to set this cell to 0 using the Gauss-Jordan transformation. Additionally, we transform the matrix to get a unity matrix in the first 3 columns. We continue until we have a unity matrix in the first 3 columns (or can transform the system of equations into such a form by changing the order of rows).

Every time a value in the last row is non-zero (except for $-z$, of course), the result can be improved further!

$$\left|\begin{array}{cccccc} -1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right| \begin{array}{l} \longleftarrow\!\rceil^{+} \\ \longrightarrow\!\rfloor_{1} \ \ \rceil^{-1} \\ \\ \longleftarrow\!\!\!\longrightarrow\!\rfloor_{+} \end{array} \tag{3.17}$$

$$\Longrightarrow \left|\begin{array}{cccccc} 0 & 1 & 1 & 1 & 0 & 4 \\ 1 & 0 & 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & -1 & 0 & -3 \end{array}\right| \begin{array}{l} \longleftarrow\!\rceil^{+} \\ \\ \longrightarrow\!\rfloor_{-1} \ \ \rceil^{-1} \\ \longleftarrow\!\!\!\longrightarrow\!\rfloor_{+} \end{array} \tag{3.18}$$

Row 1 can not be subtracted from row 3 as it would result in a negative b_3. We can, however, subtract row 3 from row 1. We obtain a matrix where all coefficients in the last row are negative. Thus, we have obtained an optimal solution and $-z$ is in the last cell of the matrix.

$$\Longrightarrow \left|\begin{array}{cccccc} 0 & 0 & 1 & 1 & -1 & 2 \\ 1 & 0 & 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & -1 & -1 & -5 \end{array}\right| \tag{3.19}$$

Thus, we obtain:

$$\vec{x} = \begin{pmatrix} 3 \\ 2 \\ 2 \\ 0 \\ 0 \end{pmatrix} \tag{3.20}$$

This solution indeed satisfies all constraints. After the Gauss-Jordan transformation, we have obtained the values for $t_{k,j}$. From the structure of $\hat{A}$ we see that

$$\left(\hat{A}_N\right)_{k,j} = t_{k,j} \tag{3.21}$$

for all $k \in B$ and $j \in N$. What happens with the $t_{k,j}$ is basically solving three systems of equations at once.

Generic form of the matrix:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \vdots & a_{1n} & b_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} & b_m \\ c_1 & c_2 & c_3 & \dots & c_n & z \end{pmatrix} \tag{3.22}$$

Example 2:

$$\max x_2 \tag{3.23}$$

such that

$$x_1 - x_2 \leq 1 \tag{3.24}$$

$$-x_1 + x_2 \leq 2 \tag{3.25}$$

$$x_1, x_2 \geq 0 \tag{3.26}$$

First of all, we write this in matrix form. We also introduce slack variables:

$$\begin{vmatrix} 1 & -1 & 1 & 0 & 1 \\ -1 & 1 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 & 0 \end{vmatrix} \begin{array}{l} \rceil\, 1 \quad \rceil -1 \\ \leftarrow\!\rfloor + \\ \leftarrow\!\!-\!\!-\!\!\rfloor + \end{array} \tag{3.27}$$

$$\Longrightarrow \begin{vmatrix} 1 & -\mathbf{1} & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 3 \\ 0 & \mathbf{1} & -1 & 0 & -1 \end{vmatrix} \tag{3.28}$$

We can see in the second column that c_2 is positive and all other values negative or 0. This means that the solution is unbounded! Thus, the solution is infinitely high.

Example 3 contains errors!

3.3 Duality – Motivation

Consider the LOP:

$$\max f(\vec{x}) = 2x_1 + 3x_2 \tag{3.29}$$

such that

$$4x_1 + 3x_1 \leq 12 \qquad \text{(I)} \tag{3.30}$$

$$2x_1 + x_2 \leq 3 \qquad \text{(II)} \tag{3.31}$$

$$3x_1 + 2x_2 \leq 4 \qquad \text{(III)} \tag{3.32}$$

$$x_1, x_2 \geq 0 \tag{3.33}$$

While this system of equation has more constraints than variables, we get more variables than constraints as soon as we add slack variables.

As $x_1, x_2 \geq 0$, we have:

$$2x_1 + 3x_1 \leq 4x_1 + 8x_2 \leq 12 \tag{3.34}$$

Thus, 12 is an upper bound for $f(\vec{x})$. If we divide the first equality by 2, we obtain an even better bound

$$2x_1 + 3x_2 \leq 2x_1 + 4x_2 \leq 6. \tag{3.35}$$

Even better, if we calculate $\frac{1}{3} \cdot (\mathrm{I}) \cdot (\mathrm{II})$.

$$2x_1 + 3x_2 \leq \frac{1}{3}(4x_1 + 8x_2 + 2x_1 + x_2) \tag{3.36}$$

$$\leq \frac{1}{3}(12 + 3) \tag{3.37}$$

$$= 5 \tag{3.38}$$

Hence, $f(\vec{x})$ can not get larger than 5. How good can an upper bound get this way? So, we are trying to derive an inequality of the form

$$d_1 x_1 + d_2 x_2 \leq h \tag{3.39}$$

with $d_1 \geq 2$, $d_2 \geq 3$ and h as small as possible. So $\forall x_1, x_2 \geq 0$ we have

$$2x_1 + 3x_2 \leq d_1 x_1 + d_2 x_2 \leq h. \tag{3.40}$$

Combining the 3 inequalities of the original LOP with some non-negative coefficients y_1, y_2, y_3 (so that direction of inequalities is not reversed), we obtain

$$\underbrace{(4y_1 + 2y_2 + 3y_3)}_{d_1} x_1 + \underbrace{(8y_1 + y_2 + 2y_3)}_{d_2} x_2 \leq \underbrace{12y_1 + 3y_2 + 4y_3}_{h} \tag{3.41}$$

How to choose y_i? We need to ensure that $d_1 \geq 2$ and $d_2 \geq 3$ and we want to have h as small as possible under these constraints. So we have a new LOP:

$$\min 12y_1 + 3y_2 + 4y_3 \tag{3.42}$$

such that

$$4y_1 + 2y_2 + 3y_3 \geq 2 \tag{3.43}$$

$$8y_1 + y_2 + 2y_3 \geq 3 \tag{3.44}$$

$$y_1, y_2, y_3 \geq 0 \tag{3.45}$$

This is called the **dual LOP** to the original LOP.

We now show that *every* feasible solution we can find for the dual LOP is an upper bound for the original LOP and a feasible solution to the original LOP is a lower bound for the dual LOP. The original problem is referred to as the **primal problem**.

$$\max \vec{c}^\top \vec{x} \text{ s. t. } A\vec{x} \leq \vec{b}, \vec{x} \geq \vec{o}(P) \tag{3.46}$$

$$\min \vec{b}^\top \vec{y} \text{ s. t. } A^\top \vec{y} \geq \vec{c}, \vec{y} \geq \vec{o}(D) \tag{3.47}$$

We can see that $\vec{y}$ in the dual problem is multiplied with the *transposed* matrix. Thus, if the primal problems has 2 variables and 3 constraints, the dual problem has 3 variables, but only 2 constraints.

We later show that the optimal solution for both problems is the same and any feasible solution is a bound for the optimal solution. If the primal optimization problem is unbounded, the dual problem has no solution. Vice versa, if the primal problem has no solution, the dual problem is unbounded.

3.4 Duality

Note that we now calculate $y^\top A$ in the slide. Let's recall:

Let A be an $m \times n$ matrix, $\vec{x}$ be an $m \times 1$ vector, $\vec{b}$ be an $m \times 1$ vector.

$$A \cdot \vec{x} = \vec{b} \tag{3.48}$$

$$\iff (A \cdot x)^\top = \vec{b}^\top \tag{3.49}$$

$$\iff x^\top \cdot A^\top = \vec{b}^\top \tag{3.50}$$

Thus, we can remove the parentheses for the transposition by applying the transposition for all inner parts of the parentheses.

We can multiply:

- a row vector and a matrix: $(1 \times m) \cdot (m, n) = (1 \times n)$
- a matrix and a column vector: $(m \times n) \cdot (n \times 1) = (m \times 1)$

3.5 Duality (7)

$(\hat{P})$ expands upon (P) by adding slack variables:

$$\max \vec{c}^\top \vec{x} \tag{3.51}$$

s. t.

$$A\vec{x} + \vec{z} = \vec{b} \tag{3.52}$$

$$\iff \tag{3.53}$$

$$\max \left(\vec{x}^\top, \vec{o}^\top\right) \begin{pmatrix} \vec{x} \\ \vec{z} \end{pmatrix} \tag{3.54}$$

s. t.

$$(A|I)\underbrace{\begin{pmatrix}\vec{x}\\ \vec{z}\end{pmatrix}}_{\vec{\hat{x}}} = \vec{b} \tag{3.55}$$

Note that $\hat{h}_B$ and $\hat{h}_N$ have nothing to do with h. $\vec{\hat{y}}$ has nothing to do with the $\vec{y}$ from the duality explanation, but it is rather the $\vec{y}$ that was the solution of the LOP as in the simplex algorithm.

3.6 Duality (8)

$$\hat{A}_B^{-1}\hat{A}_B\vec{\hat{x}}_B = \hat{A}_B^{-1}\hat{A}_B\vec{\hat{y}}_B + \hat{A}_B^{-1}\hat{A}_n\vec{\hat{y}}_N \tag{3.56}$$
$$\iff \vec{\hat{x}}_B = \vec{y}_B + \hat{A}_B^{-1}\hat{A}_N\vec{\hat{y}}_N \tag{3.57}$$
$$\iff \vec{\hat{y}}_B = \vec{\hat{x}}_B - \hat{A}_B^{-1}\hat{A}_N\vec{\hat{y}}_N \tag{3.58}$$

As $\vec{\hat{x}}$ is a basic solution, $\vec{\hat{x}}_N = \vec{o}$, so there is no $\vec{\hat{x}}_N$ in the equations.

3.7 Duality (10)

$$\vec{u}^\top A \geq \vec{c}^\top \tag{3.59}$$
$$\vec{u} \geq \vec{o} \tag{3.60}$$
$$\vec{u}^\top := \vec{c}^\top \hat{A}_B^{-1} \tag{3.61}$$
$$\vec{\hat{c}}^\top \hat{A}_B^{-1} A \geq \vec{c}^\top \tag{3.62}$$
$$\vec{\hat{c}}^\top \hat{A}_B^{-1} \geq \vec{o}^\top \tag{3.63}$$

$$A\vec{x} = \vec{b} \tag{3.64}$$
$$\iff \hat{A}_B\vec{\hat{x}}_B = \vec{\hat{b}} \tag{3.65}$$
$$\iff \hat{A}_B^{-1}\hat{A}_B\vec{\hat{x}}_B = \hat{A}_B^{-1}\vec{\hat{b}} \tag{3.66}$$

Stichwortverzeichnis