Design and Operation of Traffic and Telecommunication Networks

Lectures on Location Problems
by Prof. Dr. Ralf Borndörfer at FU Berlin

LaTeX by Hoan Do and Felix Thoma*
Illustrations by Prof. Dr. Ralf Borndörfer

February 10, 2016

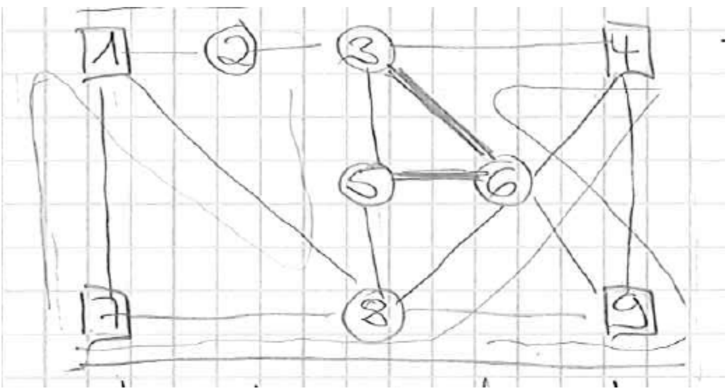
*Please report errors to thoma@campus.tu-berlin.de!

Contents

1 Introduction

1.1 Introduction

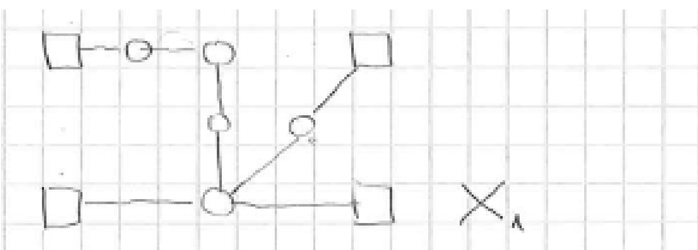
1.1.1 Example



$T = \{1, 4, 7, 9\}$ terminal nodes,
$S = \{2, 3, 5, 6, 8\}$ Steiner nodes
$V = S \dot{\cup} T$ nodes
$E = \{12, 17, 18, \ldots, 89\}$ edges
$G = (V, E)$ undirected graph

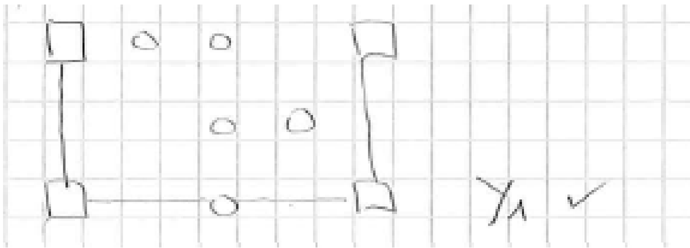
$d_{19} = d_{79} = 1$ demand
$u_e = 1$ capacities
$P = \{1234, 789, 7183, 784, 43698\}$ predefined paths
$c_e = \begin{cases} 1 & , e \neq 36, 56 \\ 2 & , e = 36, 56. \end{cases}$ costs,
$c_p = 1$ predefined path costs

a) **Minimum Spanning Tree Problem (MST):**
Connect V at minimum cost.



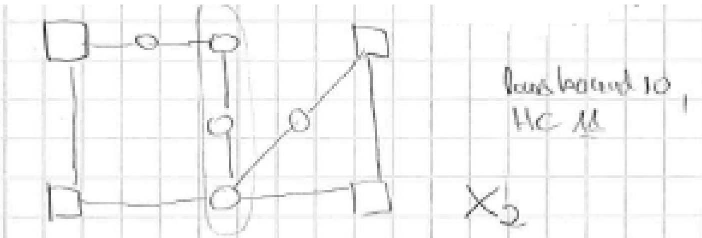
X_1 tree with minimum cost $C(X_1) = 9$

b) **Steiner Tree Problem (STP):**
Connect T at minimum cost.



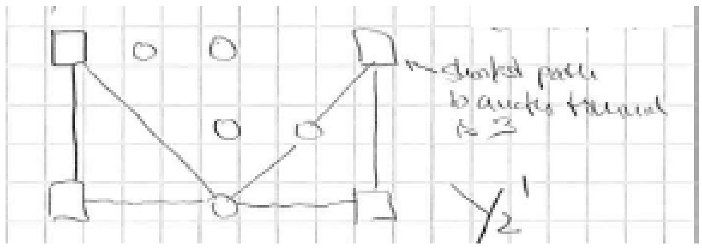
Y_1 Steiner tree with minimum cost $C(Y_1) = 4$

c) **2-Edge Connected Graph Problem (2EC(V)):**
Connect V at minimum cost such that one edge can fail.



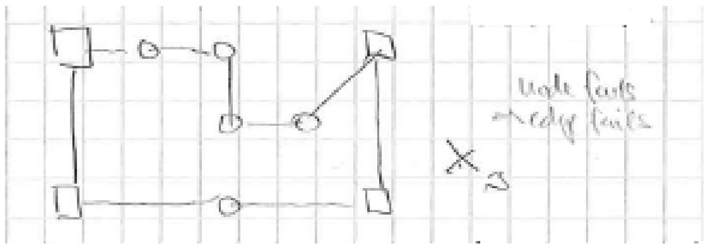
X_2 2-edge connected graph with minimum cost $c(X_2) = 11$

d) **2-Edge Connected Subgraph Problem (2EC(T)):**
Connect T at minimum cost such that one edge can fail.



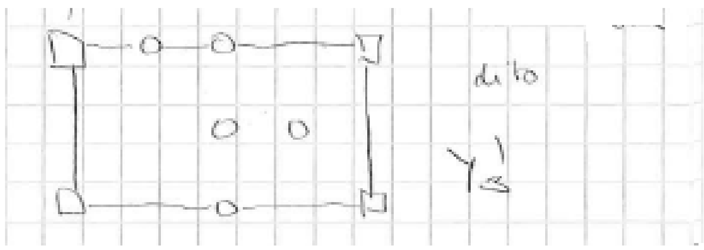
Y_2' 2-edge connected subgraph with minimum cost $c(Y_2') = 7$

e) **2-Node Connected Graph Problem (2NC(V)):**
Connect V at minimum cost such that one node can fail.



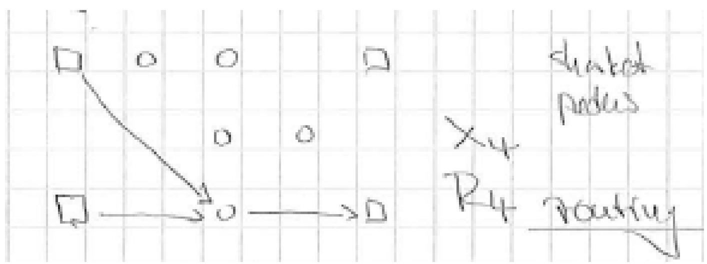
X_3 2-node connected graph with minimum cost $c(X_3) = 11$

f) **2-Node Connected Subgraph Problem (2NC(T)):**
Connect T at minimum cost such that one node can fail.



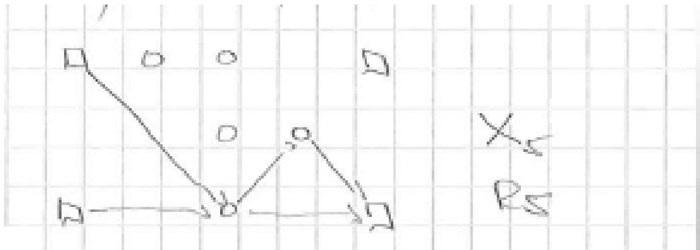
Y_3 2-node connected subgraph with minimum cost $c(Y_3') = 11$

g) **Uncapacitated Network Design Problem (UNDP):**
Route demand on shortest paths at minimum cost.



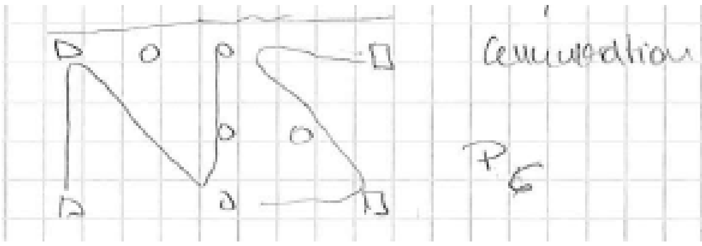
optimal solution $c(X_4) = 3, c(R_4) = 2 + 2 = 4$

h) **Capacitated Network Design Problem (UNDP)**:
Route demand on shortest paths at minimum cost w.r.t. edge capacities.



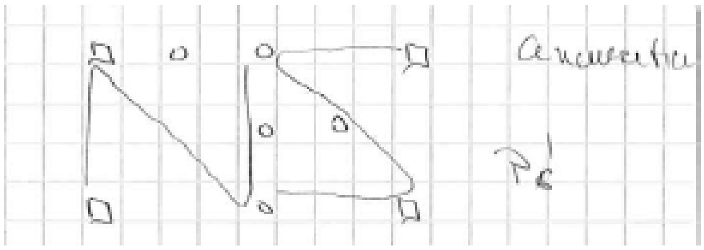
optimal solution $c(X_5) = 5, c(R_5) = 5$

i) **Spanning Set Problem (SSP):**
Connect V at minimum cost by paths.



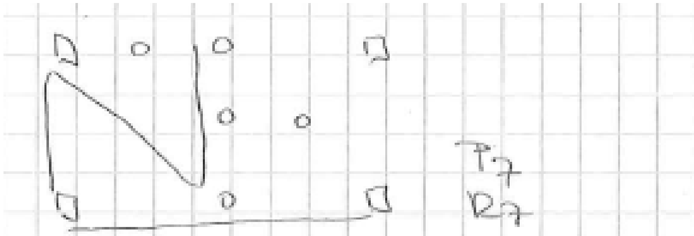
P_6 path system with optimal cost $c(P_6) = 3$

j) **Steiner Connectivity Problem (SCP):**
Connect T at minimum cost by paths.



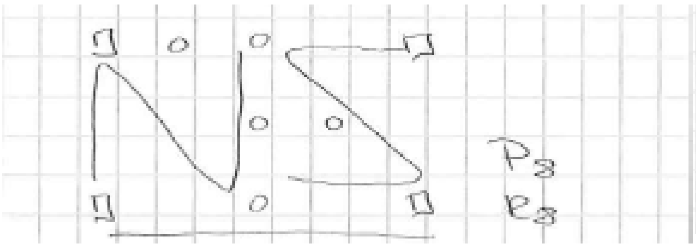
P'_6 path system with optimal cost $c(P'_6) = 2$

k) **Uncapacitated Line Planning Problem (ULPP):**
Route demand on shortest path at minimum cost by paths.



optimal solution $c(P_7) = 2, c(R_7) = 4$

l) **Capacitated Line Planning Problem (CLPP):**
Route demand at minimum cost by paths w.r.t. path capacities.



optimal solution $c(P_8) = 3, c(R_8) = 4$

1.1.2 Definition (Network Design Problem)
A network design problems involves a (supply) network of potential nodes, edges, ... with associated costs, travel times, capacities, ... that can be installed such that a transportation demand between constraint nodes can be routed according to some routing scheme subject to connectivity, robustness, ... requirements at minimum total construction and routing costs.

1.1.3 Definition (Graph Theory Notation)

a) V finite set of nodes
$E \subseteq \{\{\{u, v\} : u, v \in V\}\}$ set of (undirected) edges.
$A \subseteq \{\{(u, v) : u, v \in V\}\}$ set of (directed) arcs.
$w \in \{\{(v, v) : v \in V\}\}$ loop.
$G = (V, E)$ undirected graph.
$D = (V, A)$ directed graph.
$H = (U, F), W \subseteq V, F \subseteq E$ subgraph of G
$G[W] = (W, \{xy \in \begin{cases} E \\ A \end{cases} : x, y \in W\})$ subgraph of G induced by $W \subseteq V$

b) $\delta(W) := \{\{u,v\} \in E : u \in W, v \notin W\}$ cut induced by $W \subseteq V$
$\delta^+(W) := \{(u,v) \in A : u \in W, v \notin W\}$ outcut induced by $W \subseteq V$
$\delta^-(W) := \{(u,v) \in A : u \notin W, v \in W\}$ incut induced by $W \subseteq V$
$\delta(v) := \delta(\{v\})$
$\deg(v) := |\delta(v)|$ degree.

c) $(v_0, v_0v_1, v_1, \ldots, v_{k-1}, v_{k-1}v_k, v_k)$ v_0v_k-path
if $v_iv_{i+1} \in E$ and $v_i \neq v_j$ for all $i \neq j$ except possibly $v_0 = v_k$.
Cycle: v_0v_0-path, i.e. v_0v_k-path with $v_0 = v_k$, with at least 3 nodes
Hamiltonian cycle: cycle with $k = |V|$ nodes
s, t connected :⇔ There exists a st-path.
G connected :⇔ There exists a st-path for all $s \neq t$.
$G[W]$ component :⇔ W maximal (by inclusion) such that $s, t \in W$ connected.
$e \in E$ bridge :⇔ $G \backslash e : (V, E \backslash \{e\})$ has more components than G.
st k-node connected :⇔ $\exists\, k$ st-paths with disjoint inner vertices.
G k-node connected :⇔ $\forall s, t \in V :$ st k-node connected.
st k-edge connected :⇔ $\exists\, k$ st-paths with disjoint edges.
G k-edge connected :⇔ $\forall s, t \in V :$ st k-edge connected.

d) G tree :⇔ G is connected and contains no cycle.
G forest :⇔ G contains no cycle.
D arborescence with root v :⇔ $\forall t \neq v$: ∃! vt-path
D branching :⇔ D disjoint union of arborescences
G bipartite :⇔ $V = X \cup Y, E \subseteq \{xy : x \in X, y \in Y\}$.
G complete :⇔ $E = \{xy : x, y \in V, x \neq y\} \Leftrightarrow \forall x, y \in V, x \neq y :\ xy \in E$.
$F \subseteq E$ st-cut :⇔ $\exists W \subseteq V : s \in W, t \notin W, F = \delta(W)$.
$W \subseteq V$ st-node cut :⇔ $s, t \notin W$ are not connected in $G[V \backslash W]$.

e) $c \in \mathbb{R}^E$ edge cost.
$c(F) = \sum_{e \in F} c_e$
$\min\limits_{\text{p st-path}} c(p)$ shortest st-path problem
$\min\limits_{\text{T spanning tree}} c(T)$ minimum spanning tree problem
$u \in \mathbb{Q}^E_{\geq 0}$ edge capacities
$X \in \mathbb{Q}^E_{\geq 0}$ st-flow :⇔ $\forall\, v \neq s, t :\ X(\delta^+(v)) = X(\delta^-(v))$.
$V(X) := X(\delta^+(s)) - X(\delta^-(s))$ value of st-flow X
X feasible for u :⇔ $X \leq u$
$\max\limits_{\text{X feasible flow}} V(X)$ maximum flow problem
$\min\limits_{\text{X feasible flow of value V}} V(X)$ minimum cost flow problem
$(d_{st}) \in \mathbb{Q}^{V \times V}_{\geq 0}$ demand matrix (undirected symmetric case)
$X \in \mathbb{Q}^{V \times V \times E}_{\geq 0}$ multi-commodity flow :⇔ $\forall\, s, t :\ X^{st}$ st-flow
X feasible for u, d :⇔ $\forall\, s, t \in V :\ V(X^{st}) = d_{st}, \forall\, e \in E :\ \sum_{st} X^{st}_e \leq u_e$
$\min\limits_{\text{X feasible multi-commodity flow}} c^T x$ multi-commodity flow problem.

1.1.4 Definition (Linear Programming Terminology)
$\min_{Ax \leq b} c^T x$ linear program (LP)
$c \in \mathbb{R}^n$ objective function
$A \in \mathbb{R}^{m \times n}$ constraint matrix
$b \in \mathbb{R}^m$ right hand side
$l \leq x \leq u, l \in \mathbb{R}^n, u \in \mathbb{R}^n$ lower and upper variable bounds
$P(A,b) := \{x \in \mathbb{R}^n : Ax \leq b\}$ polyhedron
$P(A,b)$ polytope $:\Leftrightarrow P(A,b)$ bounded
$x \in P(A,b)$ vertex $:\Leftrightarrow \nexists y, z \in P, \lambda \in]0,1[: x = \lambda y + (1 - \lambda z) y$
$\min_{Ax=b, x \geq 0} c^T x$ LP in standard form
$P^{=}(A,b) := \{x \in \mathbb{R}^n : Ax = b\}$
$\max_{y^T A \leq c^T} y^T b$, dual of $\min_{Ax=b, x \geq 0} c^T x$
$P(A,b)$ integer $:\Leftrightarrow$ All vertices of $P(A,b)$ are integer.

1.1.5 Theorem (Linear Programming)

a) $P(A,b)$ nonempty and bounded $\Leftrightarrow \exists V \in \mathbb{R}^{n \times k} : P(A,b) = \text{conv}(V)$.

b) $P(A,b)$ nonempty and bounded $\Leftrightarrow \min_{x \in P(A,b)} c^T x$ has an optimal vertex solution.

c) $P(A,b)$ nonempty and bounded $\Leftrightarrow \min_{Ax=b, x \geq 0} c^T x = \max_{y^T A \leq c^T} y^T b$.

d) $P^{=}(A,b)$ nonempty with A having full row rank, v vertex of $P^{=}(A,b)$
$\Rightarrow$ There exists a basis $B \subseteq \{1, \ldots, n\}$ such that $A_B = (a_{ij})_{i=1,\ldots,m, j \in B}$ regular,
$v_B = (v_j)_{j \in B} = A_B^{-1} b$ and $v_N = 0$.

1.1.6 Definition (Integer Programming Terminology)
$\min_{Ax+Dy=b; x,y \geq 0; x \in \mathbb{Z}} c^T x + d^T y$ mixed integer program (MIP)
$\min_{Ax+Dy=b; x,y \geq 0} c^T x + d^T y$ LP relaxation of this MIP

1.1.7 Observation (LP Relaxation)
$$\min_{Ax+Dy=b; x,y \geq 0; x \in \mathbb{Z}} c^T x + d^T y \leq \min_{Ax+Dy=b; x,y \geq 0} c^T x + d^T y$$

1.1.8 Definition (Duality Gap, Total Unimodularity)
a) $\min_{Ax+Dy=b; x,y \geq 0; x \in \mathbb{Z}} c^T x + d^T y - \min_{Ax+Dy=b; x,y \geq 0} c^T x + d^T y$ duality gap.
b) $A \in \mathbb{R}^{m \times n}$ totally unimodular $:\Leftrightarrow \det A' \in \{-1, 0, 1\}$ for all square submatrices of A.

1.1.9 Corollary
If (A,D) is totally unimodular,
the LP-relaxation of $\min_{Ax+Dy=b; x,y \geq 0; x \in \mathbb{Z}^n} c^T x + d^T y$ is integer.

1.1.10 Definition (Complexity Theory Notation)

a) $I \in \Pi \subseteq \{0,1\}^*$ instance (e.g. G)
$P : \{0,1\}^* \supseteq \Pi \to \{\underbrace{0}_{=\text{true}}, \underbrace{1}_{=\text{false}}\}$ decision problem (e.g. "Is G connected?")
$A : \Pi \to \{0,1\}, I \mapsto A(I)$ Algorithm.
Algorithm A solves decision problem P $:\Leftrightarrow \forall I \in \Pi(P) : A(I) = P(I)$

b) $\langle I \rangle :=$ length/size of bit string I.
$\langle z \rangle := \lceil \log_2 |z| + 1 \rceil + 1 \quad$ for $z \in \mathbb{Z}$
$\langle \frac{p}{q} \rangle := \langle p \rangle + \langle q \rangle$ for $\frac{p}{q} \in \mathbb{Q}$.
$\langle x \rangle := \sum \langle x_i \rangle$ for $x \in \mathbb{Q}^n$
$\langle A \rangle := \sum \langle a_{ij} \rangle$ for $A \in \mathbb{Q}^{m \times n}$
$\langle G \rangle := \sum\limits_{uv \in E} \langle u \rangle + \langle v \rangle$ for a graph G which is stored as edge list
$\langle G \rangle := \left\langle \left(\delta_{ve} = \begin{cases} 1 & v \in e \\ 0 & v \notin e \end{cases} \right)_{V \times E} \right\rangle$ if G is stored as node-edge incidence matrix
$\langle G \rangle := \left\langle \left(\delta_{uv} = \begin{cases} 1 & uv \in e \\ 0 & uv \notin e \end{cases} \right)_{V \times V} \right\rangle$ if G is stored as node-node incidence matrix.

c) $f_A : \mathbb{N} \to \mathbb{N}, n \mapsto$ maximum number of elementary operations (addition, subtraction, multiplication, division, comparison, assignment) that A needs on a random access machine (RAM) to process and instance of problem P of size n
(runtime function)
$s_A : \mathbb{N} \to \mathbb{N}, n \mapsto$ maximum size of a number that A computes while processing a instance of problem P of size n
A is polynomial :⇔
There exist polynomials $p, q : f_A(n) \leq p(n), s_A(n) \leq q(n)$ for all $n \in \mathbb{N}$

d) Algorithm A solves decision problem P non-deterministically
$:\Leftrightarrow \forall I \in P^{-1}(0) : \exists J \in \{0,1\}$: A can prove $P(I) = 0$ using J.
A is nondeterministically polynomial $:\Leftrightarrow A(\cdot, J(\cdot))$ is polynomial.

e) $A : P \to P', I \to A(I) : P(I) = P'(A(I))$ transformation from problem P to P'
A polynomial transformation :⇔ A transformation which is polynomial
P set of problems that can be solved by a polynomial algorithm
NP set of problems that can be solved by a non-deterministic polynomial algorithm
NPC set of problems to which every NP problem can be polynomially transformed
$P : \pi \in \{0,1\}^* \to \mathbb{Q}$ optimization problem
$P' : (\pi, L) \mapsto P(I) \leq L$ decision problem associated with P
NPH set of optimization problems with an associated decision problem in NPC

f) Let $f, g : \mathbb{N} \to \mathbb{N}$ be functions.
$g \in O(f) :\Leftrightarrow \exists c \in \mathbb{N} : \forall n \in \mathbb{N} : g(n) \leq cf(n)$ (g is of the order of f).

1.1.11 Definition (Satisfiability Problem)
Instance: Literals (boolean variables and their negations) $x_1, \dots, x_n$.
Clauses (disjunction of literals) $C_1(x), \dots, C_n(x)$.
Question: Is there a satisfying truth assignment $x \mapsto \{0,1\}^m$ such that $\bigwedge_{i=1}^{m} C_i(x) = 0$?

1.1.12 Theorem (Cook [1971])
$SAT \in NPC$.

1.1.12 Definition (Subset Sum)
Instance: $a_1, \dots, a_n \in \mathbb{N}$.
Question: Is there $I \subseteq [n]$ such that $a(I) = \sum_{i \in I} a_i = \sum_{i=1}^{n} \frac{a_i}{2} = \frac{a[n]}{2}$

1.1.13 Theorem (Karp [1971])
a) Hamiltonian cycle is NP-complete.
b) Clique $\geq k$ is NP-complete.
c) Stable set $\geq k$ is NP-complete.
d) Subset sum is NP-complete.
e) Finding a shortest Hamiltonian cycle is NP-hard.
f) Integer programming is NP-hard.

1.1.14 Example
a) Hamiltonian cycle is NP-complete.
b) Directed Hamiltonian cycle is NP-complete.

Proof. Lecture. □

1.1.15 Algorithm (ggT(a,b))
Input: $a, b \in \mathbb{N}, b \geq a$
Output: ggT(a,b)
Data structure: $c \in \mathbb{N}$

1. $c \leftarrow b$
2. $b \leftarrow a$
3. $a \leftarrow c \mod b$
4. If $a \neq 0$: Goto 1.
5. Output b

1.1.16 Theorem
$f_{ggT}(n) \in O(n^2)$

Proof. Every line only contains elementary operations and is thus in $O(1)$. b is at least halved in every iteration. Hence, the number of loops in step 4 is $k \leq \log_2 b \leq \langle b \rangle \leq n$ with respect to the initial value of b.
$\Rightarrow f_{ggT}(n) \leq n^2 \in O(n^2)$. □

2 Location Problems

2.1 Introduction

2.1.1 Motivation
Given a set V of points (in $\mathbb{R}^n$ or in a network N),
find a number of medians (in $\mathbb{R}^n$ or N, respectively),
i.e. new points that minimize the distance (e.g. $l_1, l_2, l_2^2, l_\infty$) to V.

a) Median point in the plane (Fermat [17th century]):
Given a triangle, find a median (point in the plane)
that minimizes the sum of the distances to the triangle vertices.

b) Location-allocation problem (Weber [20th century]):
As Fermat, but with $n \geq 3$ points, $p \geq 1$ facilities (medians) and
distance weights w_F to account for costumer demands.

c) Absolute median problem (Hakimi [1960s]):
V $\widehat{=}$ vertices of a graph,
medians $\widehat{=}$ points on vertices and edges,
dist $\widehat{=}$ distance in graph

2.1.2 Definition (Classification Scheme for Location Problems)
(Hamacher & Nickel [1998])
Input: new locations / domain / specifics / distances / objective, given points

- new locations: p=1 (one), n (many)
- domain: N (network), $\mathbb{R}^2$ (planar), $\mathbb{R}^n$ ($n \in \mathbb{N}$)
- specifics: $R = \ldots$ (restricted positions), $B = \ldots$ (barrier)
- distances: e.g. sp (shortest path), $l_1, l_2, l_2^2, l_\infty$
- objective: $\sum$ (of distances), max (of distances)
- given points: finite $V \subseteq$ domain (implicit)

Output: $P = \underset{P \subseteq \text{ domain w.r.t. specifics}}{\operatorname{argmin}} \text{ objective}$

2.1.3 Example (Schöbel & Schmidt [2009])
a) Desert well: $1/\mathbb{R}^2/\cdot/l_2/\sum$
b) Manhattan fire brigade: $1/\mathbb{R}^2/R = \text{buildings}/l_1/\max$
c) Warehouses: $n/N/\cdot/sp/\sum$

2.2 Medians in the Plane

2.2.1 Definition (Median of numbers)
Let $\{x_1, \dots, x_m\} \subseteq \mathbb{R}$ be a set of $m \in \mathbb{N}$ numbers.

$$\text{Then } \operatorname{med}\{x_i\}_{i=1}^m := \begin{cases} \{\operatorname{argmin}_{\lfloor \frac{m}{2} \rfloor}\{x_i\}\} & m \text{ odd} \\ [\operatorname{argmin}_{\frac{m}{2}}\{x_i\}, \operatorname{argmin}_{\frac{m}{2}+1}\{x_i\}] & m \text{ even} \end{cases}$$

2.2.2 Example (Manhattan distances (Schöbel & Schmidt [2009]))
Let $V = \{(a_i, b_i)\}_{i=1}^6 = \{(0,6), (4,5), (10,5), (5,3), (9,2), (2,0)\}$. Then
$\min A/\mathbb{R}^2/\cdot/l_1/\sum$
$= \min \sum_{i=1}^6 (|x - a_i| + |y - b_i|)$
$= \min \sum_{i=1}^6 |x - a_i| + \min \sum_{i=1}^6 |y - b_i|$
$= [\min_3\{a_i\}, \min_4\{a_i\}] \times [\min_3\{b_i\}, \min_4\{b_i\}]$
$= \operatorname{med}\{a_i\} \times \operatorname{med}\{b_i\}$
$= [4, 5] \times [3, 5]$

2.2.3 Theorem (Explicit solution formula for $1/\mathbb{R}^2/\cdot/l_1/\sum$)
Let $V = \{(a_i, b_i)\}_{i=1}^m$. Then $\underset{(x,y)\in\mathbb{R}^2}{\operatorname{argmin}}\ 1/\mathbb{R}^2/\cdot/l_1/\sum = \operatorname{med}\{a_i\} \times \operatorname{med}\{b_i\}$.

Proof. Similar to Example 2.2.2. □

2.2.4 Example (l_2^2)
$V = \{(a_i, b_i)\}_{i=1}^m, 1/\mathbb{R}^2/\cdot/l_2^2/\sum w_i$

$$\min \sum_{i=1}^m w_i((x - a_i)^2 + (y - b_i)^2) = \min \sum_{i=1}^m w_i(x - a_i)^2 + \min \sum_{i=1}^m w_i(y - b_i)^2 \Rightarrow$$

$$\frac{d}{dx} \sum_{i=1}^m w_i(x - a_i)^2 = 2 \sum_{i=1}^m w_i(x - a_i) = 0 \Rightarrow x = \frac{\sum_{i=1}^m w_i a_i}{\sum_{i=1}^m w_i}$$

$$\frac{d}{dx} \sum_{i=1}^m w_i(y - b_i)^2 = 2 \sum_{i=1}^m w_i(y - b_i) = 0 \Rightarrow x = \frac{\sum_{i=1}^m w_i b_i}{\sum_{i=1}^m w_i}$$

2.2.5 Theorem (Explicit solution formula for $1/\mathbb{R}^2/\cdot/l_2^2/\sum_{i=1}^m w_i$)

Let $V = \{(a_i, b_i)\}_{i=1}^m$, $w_i \geq 0$ and $\sum_{i=1}^m w_i > 0$.

Then $\underset{(x,y)\in\mathbb{R}^2}{\operatorname{argmin}}\ 1/\mathbb{R}^2/\cdot/l_2^2/\sum w_i = \frac{1}{\sum_{i=1}^m w_i}(\sum_{i=1}^m w_i a_i, \sum_{i=1}^m w_i b_i)$.

Proof. The point x from Example 2.2.4 minimizes the objective because it is convex. □

2.2.5 Example (l_2)
$V = \{(a_i, b_i)\}_{i=1}^m =: \{v_i\}_{i=1}^m, 1/\mathbb{R}^2/\cdot/l_2/\sum$

$$\min_{p=(x,y)\in\mathbb{R}^2} f(p) = \min_{p=(x,y)\in\mathbb{R}^2} \sum_{i=1}^m \sqrt{(x - a_i)^2 + (y - b_i)^2} = \min_{p=(x,y)\in\mathbb{R}^2} \sum_{i=1}^m \|p - v_i\|_2$$

- f is convex (as a sum of convex functions)
- f is differentiable at $p \notin V$ and
  $\nabla f(p) = (\frac{\delta f}{\delta x}, \frac{\delta f}{\delta y})(p) = (\sum_{i=1}^{m} \frac{x-a_i}{\|p-v_i\|_2}, \sum_{i=1}^{m} \frac{y-b_i}{\|p-v_i\|_2}) = \sum_{i=1}^{m} \frac{p-v_i}{\|p-v_i\|_2}$
- $\nabla f(p) = \sum_{i=1}^{m} \frac{p-v_i}{\|p-v_i\|_2} = 0$
  $\Rightarrow p \underbrace{\sum_{i=1}^{m} \frac{1}{\|p-v_i\|_2}}_{=:\lambda(p)} = \sum_{i=1}^{m} \frac{p}{\|p-v_i\|_2} = \sum_{i=1}^{m} \frac{v_i}{\|p-v_i\|_2}$
  $\Rightarrow p = \frac{1}{\lambda(p)} \sum_{i=1}^{m} \frac{v_i}{\|p-v_i\|_2} =: T(p)$
- $p_1, p_2 := T(p_1), p_3 := T(p_2), \ldots \to p^* = \operatorname{argmin}_{p \in \mathbb{R}^2} \sum_{i=1}^{m} \|p - v_i\|_2$

2.2.6 Theorem (Weiszfeld [1937])
Let $V = \{v_i\}_{i=1}^m$ be a set of points with $\dim V > 1$ (i.e. no line contains all points). Then there exists a unique point $p^* := \operatorname{med}_{\|\cdot\|_2}\{v_i\}_{i=1}^m := \operatorname{argmin}_{p\in\mathbb{R}^2} \sum_{i=1}^{m} \|p - v_i\|_2$.
p^* is characterized as follows:

a) $p^* \notin V \Leftrightarrow \sum_{i=1}^{m} \frac{p^*-v_i}{\|p^*-v_i\|_2} = 0$

b) $p^* = v_k \Leftrightarrow \| \sum_{i\in\{1,\ldots,m\}\setminus k} \frac{p^*-v_i}{\|p^*-v_i\|_2} \|_2 \leq 1$

c) Each sequence $(p_j)_{j\in\mathbb{N}}$ with $p_j \notin V$ and $p_{j+1} = T(p_j)$ for all $i \in \mathbb{N}$ converges to p^*.

2.2.7 Lemma
Let $p(\lambda) = u + \lambda w, \lambda \in \mathbb{R}$ be a line in $\mathbb{R}^2$ and $V \nsubseteq p(\mathbb{R})$.
Then $g : \mathbb{R} \to \mathbb{R}, \lambda \mapsto w^T \sum_{i=1}^{m} \frac{v_i - p(\lambda)}{\|v_i - p(\lambda)\|_2}$ is strictly monotonously decreasing.

Proof. $w^T \frac{v_i-p(\lambda)}{\|v_i-p(\lambda)\|_2} = \|w\|_2 \cos\sphericalangle(w, v_i - p(\lambda))$ decreases
and decreases strictly for those v_i that are not on the line. □

2.2.8 Lemma
There is at most one point satisfying 2.2.6 a).

Proof. Suppose $p_1, p_2 \in \mathbb{R}^2$ with $p_1 \neq p_2$ do and consider $p(\lambda) = p_1 + \lambda(p_2 - p_1)$:
$\sum_{i=1}^{m} \frac{p_1-v_i}{\|p_2-v_i\|_2} = 0 = \sum_{i=1}^{m} \frac{p_2-v_i}{\|p_2-v_i\|_2}$

$\Rightarrow (p_2 - p_1)^T \sum_{i=1}^{m} \frac{\overbrace{p_1}^{=p_1+0(p_2-p_1)} - v_i}{\|p_1-v_i\|_2} = (p_2 - p_1)^T \sum_{i=1}^{m} \frac{\overbrace{p_2}^{=p_1+1(p_2-p_1)} - v_i}{\|p_2-v_i\|_2}$ ↯

□

2.2.9 Lemma
If p^* satisfies 2.2.6 a), there is no $v_i \in V$ satisfying 2.2.6 b).

Proof. W.l.o.g. suppose v_1 does. Consider $p(\lambda) = v_1 + \lambda \frac{p^*-v_1}{\|p^*-v_1\|_2}$ with $\{v_1, ..., v_n\} \nsubseteq p(\mathbb{R})$.
Then $\lambda \mapsto \frac{(p^*-v_1)^T}{\|p^*-v_1\|_2}(\sum_{i=2}^{m} \frac{v_i-p(\lambda)}{\|v_i-p(\lambda)\|_2})$ is strictly monotonously decreasing.

$$\Rightarrow \underbrace{\frac{(p^*-v_1)^T}{\|p^*-v_1\|_2}}_{\|\cdot\|_2=1} \underbrace{(\sum_{i=2}^{m} \frac{v_i-v_1}{\|v_i-v_1\|_2})}_{\|\cdot\|\leq 1 \text{ by b)}} > \underbrace{\frac{(p^*-v_1)^T}{\|p^*-v_1\|_2}((\sum_{i=2}^{m} \frac{v_i-p^*}{\|v_i-p^*\|_2}) + \frac{v_1-p^*}{\|v_1-p^*\|_2})}_{=0 \text{ by a)}} \underbrace{-(p^*-v_1)^T \frac{v_1-p^*}{\|v_1-p^*\|_2^2}}_{=+1} \lightning$$

□

2.2.10 Lemma
At most one $v_i \in V$ satisfies 2.2.6 b).

Proof. W.l.o.g. suppose v_1 and v_2 do and consider $p(\lambda) = v_1 + \lambda \frac{v_2-v_1}{\|v_2-v_1\|}$.
Then $\{v_3, \ldots, v_m\} \nsubseteq p(\mathbb{R})$ and $\frac{(v_2-v_1)^T}{\|v_2-v_1\|_2}(\sum_{i=3}^{m} \frac{v_i-v_1}{\|v_i-v_1\|_2}) > \frac{(v_2-v_1)^T}{\|v_2-v_1\|_2}(\sum_{i=3}^{m} \frac{v_i-v_2}{\|v_i-v_2\|_2})$

$$\Rightarrow 1 \overset{\text{b)}}{\geq} \|\sum_{i=2}^{m} \frac{v_i-v_1}{\|v_i-v_1\|_2}\|_2 \geq \frac{(v_2-v_1)^T}{\|v_2-v_1\|_2}(\sum_{i=2}^{m} \frac{v_i-v_1}{\|v_i-v_1\|_2}) = \underbrace{\frac{\|v_2-v_1\|_2^2}{\|v_2-v_1\|_2^2}}_{=+1} + \underbrace{\frac{(v_2-v_1)^T}{\|v_2-v_1\|_2}(\sum_{i=3}^{m} \frac{v_i-v_1}{\|v_i-v_1\|_2})}_{\leq 0}$$

$$\Rightarrow \frac{(v_2-v_1)^T}{\|v_2-v_1\|_2}(\sum_{i=3}^{m} \frac{v_i-v_2}{\|v_i-v_2\|_2}) < 0 \Rightarrow \underbrace{\frac{(v_2-v_1)^T}{\|v_2-v_1\|_2} \frac{v_1-v_2}{\|v_1-v_2\|_2}}_{=-1} + \frac{(v_2-v_1)^T}{\|v_2-v_1\|_2}(\sum_{i=3}^{m} \frac{v_i-v_2}{\|v_i-v_2\|_2}) < 0$$

$$\Rightarrow \|\sum_{i\in\{1,\ldots,m\}\setminus\{2\}} \frac{v_i-v_2}{\|v_i-v_2\|_2}\|_2 > 1 \lightning.$$

□

2.2.11 Lemma
$\sum_{i=1}^{m} \|p_{j+1} - v_i\|_2 \leq \sum_{i=1}^{m} \|p_j - v_i\|_2$ for $j = 1, .., m$ with equality if and only if $p_{j+1} = p_j$.

Proof. Let $w_i := \frac{1}{\|p_j-v_i\|_2}$ for $i = 1, \ldots, m$.

$$\overset{\text{Theorem 2.2.5}}{\Rightarrow} \operatorname*{argmin}_{p\in\mathbb{R}^2} \sum_{i=1}^{m} w_i \|p - v_i\|_2^2 = \frac{1}{\sum_{i=1}^{m} w_i} \sum_{i=1}^{m} w_i v_i = \frac{1}{\lambda(p_j)} \sum_{i=1}^{m} \frac{1}{\|p_j-v_i\|_2} v_i = T(p_j) = p_{j+1}$$

$$\Rightarrow \sum_{i=1}^{m} \|p_j - v_i\|_2 = \sum_{i=1}^{m} \underbrace{w_i}_{=\frac{1}{\|p_j-v_i\|_2}} \|p_j - v_i\|_2^2$$

$$\geq \sum_{i=1}^{m} \frac{1}{\|p_j-v_i\|_2} \|p_{j+1} - v_i\|_2^2 = \sum_{i=1}^{m} \frac{1}{\|p_j-v_i\|_2}(\|p_j - v_i\|_2 + (\|p_{j+1} - v_i\|_2 - \|p_j - v_i\|_2))^2$$

$$= \sum_{i=1}^{m} \frac{1}{\|p_j-v_i\|_2}(\|p_j-v_i\|_2^2 + 2\|p_j-v_i\|_2(\|p_{j+1}-v_i\|_2 - \|p_j-v_i\|_2) + (\|p_{j+1}-v_i\|_2 - \|p_j-v_i\|_2)^2)$$

$$= \sum_{i=1}^{m} \|p_j - v_i\|_2 + 2\sum_{i=1}^{m} \|p_{j+1} - v_i\|_2 - 2\sum_{i=1}^{m} \|p_j - v_i\|_2 + \sum_{i=1}^{m} \frac{(\|p_{j+1}-v_i\|_2 - \|p_j-v_i\|_2)^2}{\|p_j-v_i\|_2}$$

$$\Rightarrow 2\sum_{i=1}^{m}\|p_j - v_i\|_2 \geq 2\sum_{i=1}^{m}\|p_{j+1} - v_i\|_2 + \underbrace{\frac{\sum_{i=1}^{m}(\|p_{j+1} - v_i\|_2 - \|p_j - v_i\|_2)^2}{\|p_j - v_i\|_2}}_{>0} \qquad \square$$

2.2.12 Lemma
(p_i) is bounded and hence has an accumulation point.

Proof. For $i > 2$ the p_i as centers of gravity are located in the convex closure of V. □

2.2.13 Lemma
$T(p) = p$ for any accumulation point $p \notin V$ of (p_j).

Proof. Suppose $T(p) = p' \neq p$, then p' is also an accumulation point of (p_j),
as T is continuous, i.e. images of points near p are near p'.
Lemma 2.2.11 yields $\sum_{i=1}^{m}\|p - v_i\|_2 > \sum_{i=1}^{m}\|p' - v_i\|_2$,
but as $\sum_{i=1}^{m}\|p_j - v_i\|_2$ is monotonous and bounded and thus converges for $j \to \infty$,
the function values of all accumulation points must be equal:
$\sum_{i=1}^{m}\|p - v_i\|_2 = T(p) = T(p') = \sum_{i=1}^{m}\|p' - v_i\|_2$. ↯ □

2.2.14 Lemma
Any accumulation point $p \notin V$ of (p_j) satisfies 2.2.6 a).

Proof. Lemma 2.2.13 states $p = T(p)$.
Hence $p = \frac{\sum_{i=1}^{m}\frac{v_i}{\|p-v_i\|_2}}{\sum_{i=1}^{m}\frac{1}{\|p-v_i\|_2}} \Leftrightarrow \frac{\sum_{i=1}^{m}\frac{p-v_i}{\|p-v_i\|_2}}{\sum_{i=1}^{m}\frac{1}{\|p-v_i\|_2}} = 0 \Rightarrow \sum_{i=1}^{m}\frac{p-v_i}{\|p-v_i\|_2} = 0$ □

2.2.14 Corollary
(p_j) admits at most one accumulation point $p \notin V$.

Proof. Lemma 2.2.8 states that 2.2.6 a) is satisfied by at most one point. □

2.2.15 Lemma
If v_i is an accumulation point of (p_j), it is the only condensation point.

Proof. Without loss of generality let $v_i = v_1$.
Assume there are other accumulation points. They are only finitely many,
because they can only be $v_2, \ldots, v_m$ or the unique point p satisfying 2.2.6 a) if it exists.
Choose $\varepsilon > 0$ such that $U_\varepsilon(v_1) \cap \{v_2, \ldots, v_m, p\} = \emptyset$.
Consider $(j_k)_{k\in\mathbb{N}}$ such that $p_{j_k} \in U_\varepsilon(v_1), p_{j_k+1} \notin U_\varepsilon(v_1)$.
$\Rightarrow \|p_{j_k} - v_1\|_2 \to 0 \Rightarrow \|p_{j_k} - v_l\|_2 \to \|v_1 - v_l\|_2$ for $l = 2, \ldots, m$. **(1)**
$\Rightarrow \frac{\|p_{j_{k+1}} - v_1\|_2}{\|p_{j_k} - v_1\|_2} > \frac{\varepsilon}{\|p_{j_k} - v_1\|_2} \overset{k\to\infty}{\to} \infty$

Without loss of generality let $v_1 = 0$. Then

$$\frac{\|p_{j_{k+1}}-v_1\|_2}{\|p_{j_k}-v_1\|_2} = \frac{\|T(p_{j_k})\|_2}{\|p_{j_k}\|_2} = \frac{\|\sum_{i=2}^{m} \frac{v_i}{\|p_{j_k}-v_i\|_2}\|_2}{\underbrace{\sum_{i=1}^{m} \frac{1}{\|p_{j_k}-v_i\|_2}}_{=\frac{1}{\|p_{j_k}\|_2}+\sum_{i=2}^{m}\frac{1}{\|p_{j_k}-v_i\|_2}}} \cdot \frac{1}{\|p_{j_k}\|_2} = \frac{\|\sum_{i=2}^{m} \frac{v_i}{\|p_{j_k}-v_i\|_2}\|_2}{1+\sum_{i=2}^{m}\frac{\|p_{j_k}\|_2}{\|p_{j_k}-v_i\|_2}}.$$

$\|p_{j_k}\|_2 \overset{k\to\infty}{\to} 0$ and $\|p_{j_k} - v_i\|_2 \to \|v_1 - v_i\|_2 \overset{k\to\infty}{\to} \|v_i\|_2$ yields

$$\frac{\|p_{j_{k+1}}-v_1\|_2}{\|p_{j_k}-v_1\|_2} = \frac{\|\sum_{i=2}^{m} \frac{v_i}{\|p_{j_k}-v_i\|_2}\|_2}{1+\sum_{i=2}^{m}\frac{\|p_{j_k}\|_2}{\|p_{j_k}-v_i\|_2}} \overset{k\to\infty}{\to} \|\sum_{i=2}^{m}\frac{v_i}{\|v_i\|_2}\|_2 =: \nu < \infty \ \textbf{(2)} \lightning$$

□

2.2.16 Lemma
If v_i is an accumulation point of (p_j), it satisfies 2.2.6 b).

Proof. Without loss of generality let $v_i = v_1 = 0$.
Lemma 2.2.15 states that v_1 is the only accumulation point of (p_j)
Hence $p_j \overset{j\to\infty}{\to} v_1 \Rightarrow \|p_j - v_1\|_2 \overset{j\to\infty}{\to} 0$
(1) implies $\|p_j - v_l\|_2 \overset{j\to\infty}{\to} \|v_1 - v_l\|_2$ for $l = 2, \dots, m$
(2) implies $\frac{\|p_{j+1}-v_1\|_2}{\|p_j-v_1\|_2} \overset{j\to\infty}{\to} \underbrace{\|\sum_{i=2}^{m}\frac{v_i}{\|v_i\|_2}\|_2}_{=\|\sum_{i=2}^{m}\frac{v_i-v_1}{\|v_i-v_1\|_2}\|_2} = \nu < 1.$

□

2.2.17 Lemma
Suppose $p_j \notin V$ for $j \in \mathbb{N}$.
Then (p_j) converges to a point p^* satisfying 2.2.6 a) or 2.2.6 b).
Moreover, p^* does not depend on p_1.

Proof. (p_j) has an accumulation point by Lemma 2.2.12.
If $p^* \in V$, it is unique by Lemma 2.2.16 and satisfies 2.2.6 b) by Lemma 2.2.17.
If $p^* \notin V$, it is unique by Lemma 2.2.15 and satisfies 2.2.6 a) by Lemma 2.2.14.
Let $p'_j \neq p$ and suppose $\lim_{j\to\infty} p'_j = p' \neq p^*$. We have one of the following cases:

a) $\sum_{i=1}^{m}\frac{p^*-v_i}{\|p^*-v_i\|_2} = 0, \sum_{i=1}^{m}\frac{p'-v_i}{\|p'-v_i\|_2} = 0 \overset{\text{Lem. 2.2.8}}{\Rightarrow} \lightning$

b) $\sum_{i=1}^{m}\frac{p^*-v_i}{\|p^*-v_i\|_2} = 0, \|\sum_{i=1}^{m}\frac{p'-v_i}{\|p'-v_i\|_2}\|_2 \leq 1 \overset{\text{Lem. 2.2.9}}{\Rightarrow} \lightning$

c) $\|\sum_{i=1}^{m}\frac{p^*-v_i}{\|p^*-v_i\|_2}\|_2 \leq 1, \sum_{i=1}^{m}\frac{p'-v_i}{\|p'-v_i\|_2} = 0 \overset{\text{Lem. 2.2.9}}{\Rightarrow} \lightning$

d) $\|\sum_{i=1}^{m}\frac{p^*-v_i}{\|p^*-v_i\|_2}\|_2 \leq 1, \|\sum_{i=1}^{m}\frac{p'-v_i}{\|p'-v_i\|_2}\|_2 \leq 1 \overset{\text{Lem. 2.2.10}}{\Rightarrow} \lightning$

□

2.2.18 Proposition

If $p_j \notin V$ for all $j \in \mathbb{N}$, then $p^* := \lim\limits_{j\to\infty} p_j = \operatorname{argmin} \sum\limits_{i=1}^{m} \|p - v_i\|_2$.

Proof. a) For $x \neq p^*$ and $x \notin V$ let $x_1 = x, x_2 = T(x_1), \ldots$

$$\sum_{i=1}^{m} \| \underbrace{x}_{=x_1} - v_i\|_2 \overset{\text{Lemma 2.2.11}}{>} \sum_{i=1}^{m} \|x_2 - v_i\|_2 > \ldots > \lim_{j\to\infty} \sum_{i=1}^{m} \|x_j - v_i\|_2 = \sum_{i=1}^{m} \|p^* - v_i\|_2$$

b) For $x \neq p^*$ and $x = v_i$ (w.l.o.g. $x = v_1$) suppose $\sum\limits_{i=1}^{m} \| \underbrace{x}_{=v_1} - v_i\|_2 \leq \sum\limits_{i=1}^{m} \|p^* - v_i\|_2$.

Let c be the median of v_1 and p^* and consider the triangles v_1, p^*, v_i for $i = 2, \ldots, m$.
$\|v_i - c\|_2 \leq \frac{\|v_i - v_1\|_2 + \|v_i - p^*\|_2}{2}$ with equality if and only if $v_i \in \operatorname{lin}\{v_1, p^*\}$

$$\Rightarrow \sum_{i=2}^{m} \|c - v_i\|_2 \leq \frac{\sum\limits_{i=2}^{m} \|v_1 - v_i\|_2 + \sum\limits_{i=2}^{m} \|p^* - v_i\|_2}{2} \leq \sum_{i=2}^{m} \|p^* - v_i\|_2.$$

$c \notin V$ contradicts the inequality in a) $\lightning$. For $c \in V$ consider the median of c and p' and repeat this case distinction this leads to $c^* \notin V$ due to the finiteness of V. □

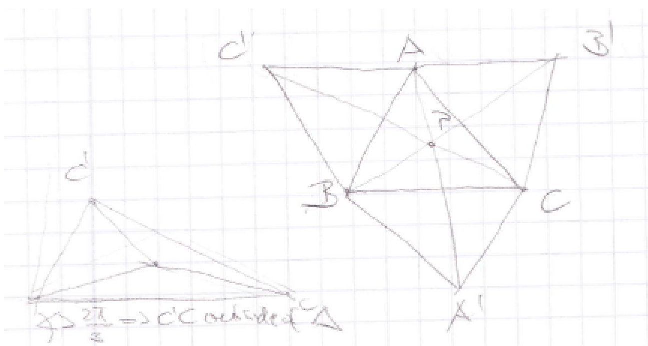
2.2.19 Remark

a) $p_j \in V$ can be repaired, e.g. by choosing p_1 appropriately.
b) There are solution formulas for $m = 3$ and $m = 4$.
c) $m = 5$ cannot be solved by a formula involving only elementary algebraic operations $(+/-, \cdot/:, \sqrt{\cdot})$.

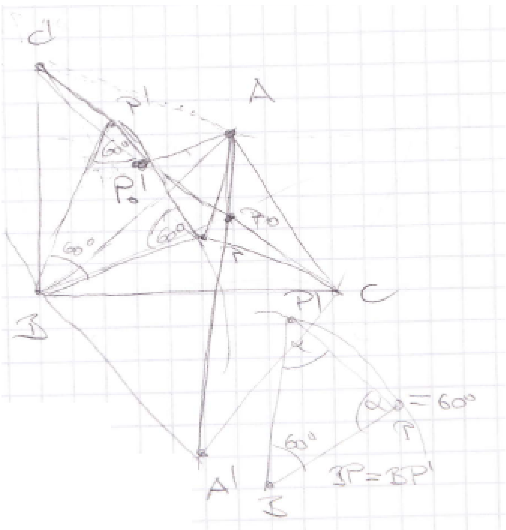
2.2.20 Theorem (Fermat Point or Torricelli Point, Napoleon's Theorem)
Let ABC be a triangle with angles $< \frac{2}{3}\pi$
and C', A', B' the outside vertices of equilateral triangles over AB, BC, AC.
Then $A'A, B'B, C'C$ are concurrent (intersect) in

$$P = \operatorname*{argmin}_{\{A,B,C\}} 1/\mathbb{R}^2/\cdot/l_2/\sum = \operatorname{med} \triangle ABC$$

P is called Fermat or Toricelli or 1st isogonic point.

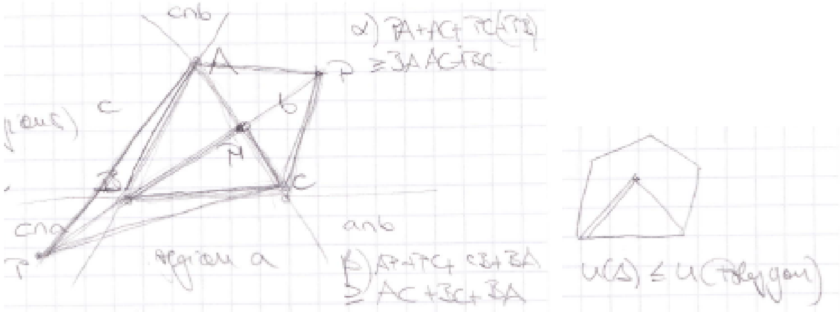


Proof (Hofmann[1929]).
a)



Select an arbitrary $P \in \triangle ABC$.
Rotate $\triangle BPA$ $60°$ around B into $BP'C'$.
Then $PA = P'C' = C'P'$ and $\triangle BPP'$ is equilateral $\Rightarrow PB = P'P$
This yields $PA + PB + PC = C'P' + P'P + PC \geq C'C$
C' as the image of A does not depend on P. BAC' is equilateral.
Construct another equilateral $\triangle BA'C$ and let $P_0 = CC' \cap AA'$.
$\triangle C'CB$ is a $60°$ rotation of $\triangle A'AB$ around B.
$\Rightarrow$ The $60°$ rotation P_0' of P_0 around B lies on CC'.
$\Rightarrow PA + PB + PC = C'P_0' + P_0'P_0 + P_0C = C'C$ is minimum and for all $P \neq P_0$ it holds that $P_0 \notin CC'$ or $P_0' \notin CC'$ and thus $PA+PB+PC = C'P'+P'P+PC > C'C$, hence $P_0 = \underset{P \in \triangle}{\operatorname{argmin}} PA + PB + PC$ is the unique minimum. Permutation of the nodes yields that P_0 is located not only on AA' and CC' but also on BB'.

b)



Select an arbitrary $P \notin \triangle ABC$.
(i) P in two regions (w.l.o.g. $P \in c \cap a$) $\Rightarrow PA + PB + PC > BA + BB + BC$.
(ii) P in one region (w.l.o.g. $P \in b$). Let $P' = AC \cap PB$.
$\Rightarrow PA + PB + PC > P'A + P'B + P'C$ □

2.2.21 Definition (k-th Minimum and Median of Numbers)
Let $a_i \in \mathbb{R}$ for $i = 1, \ldots, m$, without loss of generality $a_1 \leq \ldots \leq a_m$.
a) $\min_{i=1\ k}^{m} a_i := \min_k \{\{a_i\}\}_{i=1}^m = a_k$
b) $\operatorname{med}_{i=1}^{m}{}_k a_i := \operatorname{med}_k \{\{a_i\}\}_{i=1}^m = \operatorname*{argmin}_{p \in \mathbb{R}} \sum_{i=1}^{m} a_i - p$

2.2.22 Proposition

Let $a_1, \ldots, a_m \in \mathbb{R}$ with $a_1 \leq \ldots \leq a_m$. Then $\operatorname{med}\{\{a_i\}\} = \begin{cases} \{a_{\lceil \frac{m}{2} \rceil}\} & \text{m odd} \\ [a_{\frac{m}{2}}, a_{\frac{m}{2}} + 1] & \text{m even} \end{cases}$

Proof. Exercise. □

2.2.23 Example $(1/\mathbb{R}^1/\cdot/l_1/\sum_{w_i})$
$\{\{a_i\}\}_{i=1}^{25} = \{\{2, 5, 8, 18, 21, 15, 15, 15, 15, 15, 15, 19, 19, 19, 21, 19, 20, 23, 25, 28, 5, 13, 25, 28, 31\}\}$
Sort $\{\{a_i\}\}$ as $(2, 5, 5, 8, 13, 15, 15, 15, 15, 15, 15, 18, 19, 19, 19, 19, 20, 21, 21, 23, 25, 25, 28, 28, 31)$
$\Rightarrow \operatorname{med}\{\{a_i\}\}_{i=1}^2 5 = \min_{13}\{\{a_i\}\}_{i=1}^2 5 = 19$
Running time is $O(m \log m)$ dominated by the sorting. Is there a better algorithm?

2.2.24 Algorithm (Select $(k, a_1 \ldots, a_m)$)
Input: $a \in \mathbb{R}^m, k \in \{1, \ldots, m\}$.
Output: $\min_k \{\{a_i\}\}_{i=1}^m$.
Data structures: $(P_j) \in \mathbb{R}^{\frac{m}{5}}, p \in \mathbb{R}$ (ignoring integrability issues)

1. Calculate medians of groups of five elements: // run time in $O(\frac{m}{5})$
   For $j = 1, \ldots, \lceil \frac{m}{5} \rceil$: $p_j \leftarrow \text{Select}(3, a_{(j-1)5}, \ldots, a_{(j-1)5+4})$
2. Calculate the median of medians:
   $p \leftarrow \text{Select}(\lceil \frac{m}{10} \rceil, p_1, \ldots, p_{\lceil \frac{m}{5} \rceil})$.
3. Split $\{\{a_i\}\}$: //run time in $O(m)$:
   $L \leftarrow \{\{a_i : a_i < p\}\}$
   $E \leftarrow \{\{a_i : a_i = p\}\}$
   $G \leftarrow \{\{a_i : a_i > p\}\}$
4. Recursive call:
   - If $|L| \geq k$: $p \leftarrow \text{Select}(k, L)$ //run time: see Thm. 2.2.25
   - Else if $|L| + |E| < k$: $p \leftarrow \text{Select}(k - |L| - |E|, G)$ //run time: see Thm. 2.2.25
5. Return p //run time in $O(m)$

2.2.25 Theorem (Blum, Floyd, Praff, Rivest, Tarjan [1972])
Let $T(m,k) := \sup_{\langle k,a_1,\ldots,a_m\rangle} f_{\text{Select}}(\langle k,a_1,\ldots,a_m\rangle)$ and $T(m) = \max_{k=1}^{m} T(m,k)$.
Then $T(m) \in O(m)$,
i.e. the k-th minimum of m numbers can be computed with Select in linear time.

Proof.

a) W.l.o.g let $p_1 \leq \cdots \leq p_{m/5}$. Then

$$|L| + |E| \geq |L \cup E| \geq \Big| \bigcup_{i=1}^{\lceil \frac{m}{5}\cdot\frac{1}{2}\rceil} \underbrace{\bigcup_{j=1}^{3} \{\min_j\{a_{(i-1)5},\ldots,a_{(i-1)5+4}\}\}}_{\text{3 out of 5 numbers are } \leq \text{ due to median}} \Big| \geq \lceil \tfrac{m}{10} \rceil 3 \geq \tfrac{3}{10}m$$

$\Rightarrow |G| = m - |L| - |E| \leq \frac{7}{10}m$
Analogously $|G| + |E| \geq |G \cup E| \geq \frac{3}{10}m \Rightarrow |L| = m - |G| - |E| \leq \frac{7}{10}m$.

b) $T(m) = cm + T(\frac{m}{5}) + T(\frac{7m}{10})$
$= cm + T(\frac{2}{10}m) + T(\frac{7}{10}m)$

$$\overset{\text{stack of bricks method}}{\leq} \quad cm \textstyle\sum_{i=0}^{\infty} (\frac{9}{10})^i = \frac{1}{1-\frac{9}{10}} cm = 10cm \in O(m).$$

c) In general (i.e. in particular for $5 \nmid m$):

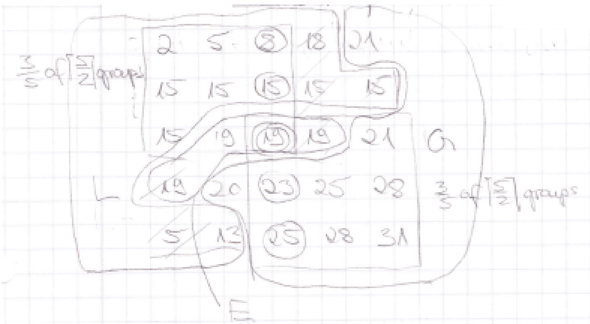
(i) $|L| + |E| \geq |L \cap E| \geq (\lceil \frac{m}{5} \rceil - 1) \cdot \frac{1}{2} \cdot 3 \geq \frac{3}{10}m - \frac{3}{2} \geq \frac{3}{10}m(1-\epsilon)$
$\Leftrightarrow m \geq \frac{5}{\epsilon}$. Analogously $|G| + |E| \geq |G \cap E| \geq \frac{3}{10}(1-\epsilon)$.
$\Rightarrow |L|, |G| \leq \frac{10}{10}m - \frac{3}{10}(1-\epsilon) = \frac{7+3\epsilon}{10}m$.

(ii) $\lceil \frac{m}{5} \rceil \leq \frac{m}{5} + 1 \leq \frac{m}{5}(1+\epsilon) \Leftrightarrow 1 \leq \frac{m\epsilon}{5} \Leftrightarrow \frac{5}{\epsilon} \leq m$.

(iii) $T(m) = cm + T((1+\epsilon)\frac{m}{5}) + T(\frac{7+3\epsilon}{10}m)$
$= cm + T(\frac{2+2\epsilon}{10}m) + T(\frac{7+3\epsilon}{10}m)$
$= cm \cdot \sum_{i=0}^{\infty} (\frac{2+2\epsilon+7+3\epsilon}{10}) = \sum_{i=0}^{\infty} \delta$ where $\delta = \frac{2+2\epsilon+7+3\epsilon}{10} = \frac{9+5\epsilon}{10}$ with $\delta < 1 \Leftrightarrow \epsilon < \frac{1}{5}$
$= cm\frac{1}{1-\delta} \in O(m)$ for $m \geq \frac{5}{\epsilon} > \frac{5}{\frac{1}{5}} = 25$.
$\Rightarrow$ Solve problem for $m \leq 25$ (by sorting) in constant time.

□

2.2.25 Example (Numbers from Example 2.2.23)



2.2.26 Theorem
Let $T : \mathbb{N}_0 \to \mathbb{N}_0$ satisfy the recursion $T(m) = c_0 m + \sum_{i=1}^{k} T(\lceil c_i m \rceil)$ with $c_i \geq 0$ for $i = 0, \ldots, k$ and $\sum_{i=1}^{k} c_i < 1$. Then $T(m) \in O(m)$.

Proof. Stack of bricks method and rounding. □

2.2.27 Algorithm (Quickselect $(k, a_1, \ldots, a_m)$)
Input: $a \in \mathbb{R}^m, k \in \{1, \ldots, m\}$
Output: $\min_k \{\{a_i\}\}_{i=1}^m$
Data structures: $p \in \mathbb{R}^m, l$ (global)

1. Randomly select $i \in \{1, \ldots, m\}$ and set $p \leftarrow a_i$ // run time in $O(1)$
2. Relabel such that $a_1, \ldots, a_{i-1} < p = a_i = \ldots = a_j = p < a_{j+1}, \ldots, a_m$
   // run time in $O(m-1)$
3. Recursive call:
   - If $i = k$: $p \leftarrow a_i$
   - If $i > k$: $p \leftarrow$ Quickselect$(k, a_1, \ldots, a_{i-1})$
   - If $i < k$: $p \leftarrow$ Quickselect$(k - i, a_{i+1}, \ldots, a_m)$
4. Return p // run time in $O(1)$

2.2.28 Theorem
Let $\tilde{T}(m,k) = E[f_{\text{Quickselect}}(\langle k, a_1, \ldots, a_m \rangle)]$ and $\tilde{T}(m) = \max_{k=1}^m \tilde{T}(m,k)$.
Then $\tilde{T}(m) \in O(m)$, i.e. Quickselect runs in expected linear time.

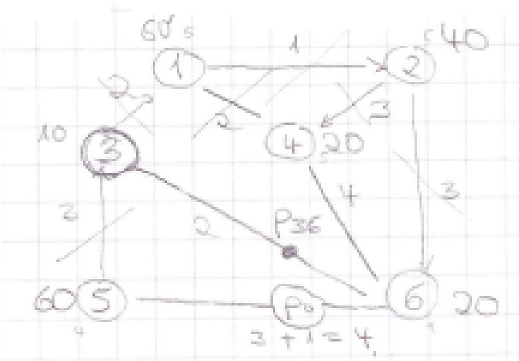
Proof. Let $c < m$ be the constant run time for steps 1 and 4 in one function call and $\tilde{T}(i) \leq 4i$ for $i < m$. Then the total run time fulfills
$\tilde{T}(m) = c + (m-1) + \frac{1}{m} \sum_{i=0}^{m-1} \tilde{T}(i) \leq 2(m-1) + \frac{4}{m} \frac{(m-1)m}{2} = 4(m-1) \leq 4m \in O(m)$ □

2.3 Medians in Networks

This chapter considers the warehouse location problem on networks $(1/N/\cdot/sp/\sum)$.

2.3.1 Example (Schöbel & Schmidt [2009])
Warehouse: $\underbrace{1}_{\in V\cup A\cup E}/N/\cdot/sp/\sum$
with shortest path distance defined for vertices as the shortest path on the network and for edge/arc points by $sp(\lambda u+(1-\lambda)v,w) := \min\{(1-\lambda)c_{uv}+sp(u,w), \lambda c_{uv}+sp(v,w)\}$
$N=(V,E\cup A)$ network, E undirected edges, A directed arcs
$c_e, c_a \in \mathbb{R}_+$ edge weights.

$$\text{Let } (c_{uv})_{u\in V, v\in V} = \begin{pmatrix} 0 & 1 & \infty & 2 & \infty & \infty \\ \infty & 0 & \infty & 3 & \infty & 3 \\ 2 & \infty & 0 & \infty & 2 & 2 \\ 2 & \infty & \infty & 0 & \infty & 4 \\ \infty & \infty & 2 & \infty & 0 & 4 \\ \infty & \infty & 2 & 4 & 4 & 0 \end{pmatrix} \text{ with } c_{uv}=\infty \Leftrightarrow uv\notin E \wedge \{u,v\}\notin A.$$



i) Let $p=4$. Then $\sum_{v\in V} sp(p,v) = 2+3+6+0+8+4 = 23$.
ii) Let p be located on edge 56 with distances $c_{5p_0}=3=c_{p_05}$ and $c_{6p_0}=1=c_{p_06}$):
Then $\sum_{v\in V} sp(p,v) = 5+6+3+5+3+1 = 23$.

2.3.2 Theorem (Node solution (Hakimi [1964]))
There is $v\in V$ such that $v\in \operatorname{argmin} 1/N/\cdot/sp/\sum$,
i.e. the 1-median problem on a network has an optimal node solution.

Proof. Consider $p=\lambda v_1+(1-\lambda)v_2, \lambda\in]0,1[$ for:

a) $v_1v_2\in A: \sum\limits_{v\in V} sp(p,v) > \sum\limits_{v\in V} sp(v_2,v)$.

b) $v_1v_2 \in E$:

$$\begin{aligned}
\sum_{v\in V} sp(p,v) &= \sum_{v\in V} \min\{\underbrace{(1-\lambda)c_{v_1v_2} + sp(v_1,v)}_{\text{then let } v\in V_1}, \underbrace{\lambda c_{v_1v_2} + sp(v_2,v)}_{\text{then let } v\in V_2}\} \\
&= \sum_{v\in V_1} [(1-\lambda)c_{v_1v_2} + sp(v_1,v)] + \sum_{v\in V_2} [\lambda c_{v_1v_2} + sp(v_2,v)] \\
&= c_{v_1v_2}\big[(1-\lambda)|V_1| + \lambda|V_2|\big] + \sum_{v\in V_1} sp(v_1,v) + \sum_{v\in V_2} sp(v_2,v) \\
&\overset{\text{w.l.o.g. } |V_1|\le|V_2|}{\geq} c_{v_1v_2}[1\cdot|V_1| + 0|V_2|] + \sum_{v\in V_1} sp(v_1,v) + \sum_{v\in V_2} sp(v_2,v) \\
&\overset{p\to v_2}{=} \sum_{v\in V_1} \underbrace{[c_{v_1v_2} + sp(v_1,v)]}_{\geq sp(v_2,v)} + \sum_{v\in V_2} sp(v_2,v) \geq \sum_{v\in V} sp(v_2,v)
\end{aligned}$$

□

2.3.3 Algorithm (for $1/N/\cdot/sp/\sum$)
Input: $N = (V, E\cup A), c_{uv} \in \mathbb{R}_+, uv \in E\cup A$.
Output: $p^* = \text{argmin}\, 1/N/\cdot/sp/\sum$.

1. Compute the shortest path matrix $(sp(u,v))_{u\in V, v\in V}$,
   e.g. by using the Floyd-Warshall algorithm.
2. Compute the row sums $sp^T\mathbb{1}$.
3. Output the index of the row sum minimum $p = \text{argmin}\, sp^T\mathbb{1}$.

2.3.3 Example (Median of the Network from Example 2.3.1)

$$sp^T\mathbb{1} = \begin{pmatrix} 0 & 1 & 6 & 2 & 8 & 4 \\ 5 & 0 & 5 & 3 & 7 & 3 \\ 2 & 3 & 0 & 4 & 6 & 2 \\ 2 & 3 & 6 & 0 & 8 & 4 \\ 5 & 6 & 3 & 7 & 0 & 4 \\ 4 & 5 & 2 & 4 & 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 21 \\ 23 \\ 17 \\ 23 \\ 25 \\ 19 \end{pmatrix}$$

$\Rightarrow p^* = \text{argmin}\, sp^T\mathbb{1} = 3, \min sp^T\mathbb{1} = 17$

2.3.4 Example (Weighted Median of the Network from Example 2.3.1)
Consider $1/N/\cdot/sp/\sum w_i$ with volumes $w_i \in \mathbb{R}$.
Let $w_i := (60, 40, 10, 20, 60, 20)$.

$$sp^T w = \begin{pmatrix} 0 & 1 & 6 & 2 & 8 & 4 \\ 5 & 0 & 5 & 3 & 7 & 3 \\ 2 & 3 & 0 & 4 & 6 & 2 \\ 2 & 3 & 6 & 0 & 8 & 4 \\ 5 & 6 & 3 & 7 & 0 & 4 \\ 4 & 5 & 2 & 4 & 4 & 0 \end{pmatrix} \begin{pmatrix} 60 \\ 40 \\ 10 \\ 20 \\ 60 \\ 20 \end{pmatrix} = \begin{pmatrix} 700 \\ 890 \\ 720 \\ 860 \\ 790 \\ 780 \end{pmatrix}$$

$\Rightarrow p^* = \text{argmin}\, sp^T w = 1, \min sp^T w = 700$

2.4 Centers in Networks

This chapter considers the fire brigade location problem on networks $(1/N/\cdot/sp/\max)$.

2.4.1 Example (Edge Solution)
Let $G = \{\{v_1, v_2\}, \{v_1v_2\}\}$ and p^* be the center of the edge v_1v_2.
Then $\operatorname{argmin} 1/N/\cdot/sp/\max = p^*$.

2.4.2 Observation
Center problems in networks do not necessarily have optimal node solutions.

2.4.3 Algorithm (for $1/N|V/\cdot/sp/\max$)
Input: $N = (V, E \cup A), c_{uv} \in \mathbb{R}_+$ for $u, v \in E \cup A$.
Output: $p^* = \operatorname{argmin} 1/N|V/\cdot/sp/\max$ ($N|V$: only node solutions allowed)

1. Compute the shortest path matrix $(sp(u, v))_{u\in V, v\in V}$.
2. Compute the row maxima $\max_{v\in V} sp(u, v)$.
3. Output the index of the minimum row maximum $p^* = \operatorname*{argmin}_{u\in V} \max_{v\in V} sp(u, v)$

2.4.3 Example (Network from Example 2.3.1)

$$sp = \begin{pmatrix} 0 & 1 & 6 & 2 & 8 & 4 \\ 5 & 0 & 5 & 3 & 7 & 3 \\ 2 & 3 & 0 & 4 & 6 & 2 \\ 2 & 3 & 6 & 0 & 8 & 4 \\ 5 & 6 & 3 & 7 & 0 & 4 \\ 4 & 5 & 2 & 4 & 4 & 0 \end{pmatrix} \Rightarrow \max_{v\in V} sp(u, v) = \begin{pmatrix} 8 \\ 7 \\ 6 \\ 8 \\ 7 \\ 5 \end{pmatrix}$$

$$\Rightarrow p^* = \operatorname*{argmin}_{u\in V} \max_{v\in V} sp(u, v) = 6, \min_{u\in V} \max_{v\in V} sp(u, v) = 5$$

2.4.4 Lemma (Preprocessing)
Let $p = \lambda v_1 + (1 - \lambda)v_2, \lambda \in]0, 1[, v_1v_2 \in E \cup A$. Then $p \notin \operatorname{argmin} 1/N/\cdot/sp/\max$ if

a) $v_1v_2 \in A$ and $c_{v_1v_2} > 0$.

b) $v_1v_2 \in E$ and $\min\{sp(v_1, \bar{v}), sp(v_2, \bar{v})\} > \max_{v\in V} sp(\bar{p}, v)$
for some $\bar{v} \in V$ and $\bar{p} \in \operatorname{argmin} 1/N|V/\cdot/sp/\max \subseteq V$.

Proof.
a) $\max_{v\in V} sp(p, v) > \max_{v\in V} sp(v_2, v)$.
b) $\max_{v\in V} sp(p, v) \geq sp(p, \bar{v}) \geq \min\{sp(v_1, \bar{v}), sp(v_2, \bar{v})\} > \max_{v\in V} sp(\bar{p}, v)$ □

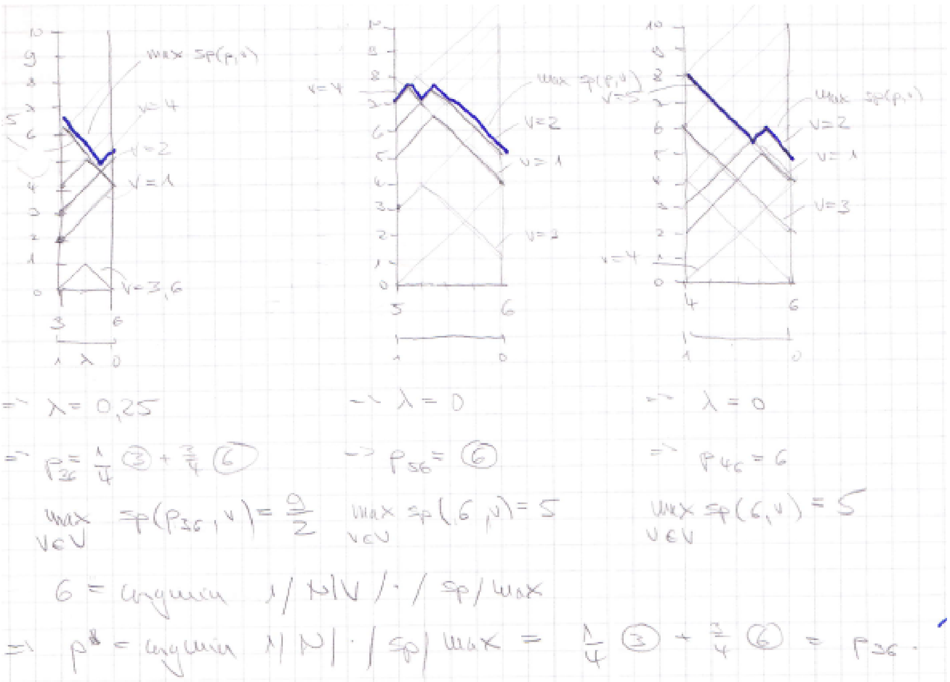
2.4.4 Example (Network from Example 2.3.1)
Let $p \in 14, 14 \in E$ and $\bar{p} := 6$.
Then $sp(p, 5) \geq \min\{sp(1, 5), sp(4, 5)\} = \min\{8, 8\} = 8 > 5 = \max_{v\in V} sp(6, v)$
implies $p \notin \operatorname{argmin} 1/N/\cdot/sp/\max$.

2.4.5 Observation (Undirected Edges)
Let $p = \lambda v_1 + (1-\lambda)v_2, \lambda \in]0,1[, v_1v_2 \in E$.

a) $sp(p,v) = \min\{(1-\lambda)c_{v_1v_2} + sp(v_1,v), \lambda c_{v_1v_2} + sp(v_2,v)\}$
is for each fixed $v \in V$ a piecewise affine, continuous function in λ with ≤ 2 pieces.

b) $\max\limits_{v \in V} sp(p,v)$ is a piecewise affine, continuous function in λ with $\leq 2|V|$ pieces.

2.4.5 Example (Network from Example 2.3.1)



$\Rightarrow \text{argmin}\, 1/N|V/\cdot/sp/\max = 6 \Rightarrow p^* = \text{argmin}\, 1/N/\cdot/sp/\max = \frac{1}{4}v_3 + \frac{3}{4}v_6$.

2.7 Centers in the Plane

2.7.1 Proposition (Smallest Enclosing Circle)
Let $V = \{v_i\}_{i=1}^m \subseteq \mathbb{R}^2$.

a) There is a unique circle C of smallest diameter enclosing V.

b) $|C \cap V| \geq 2$

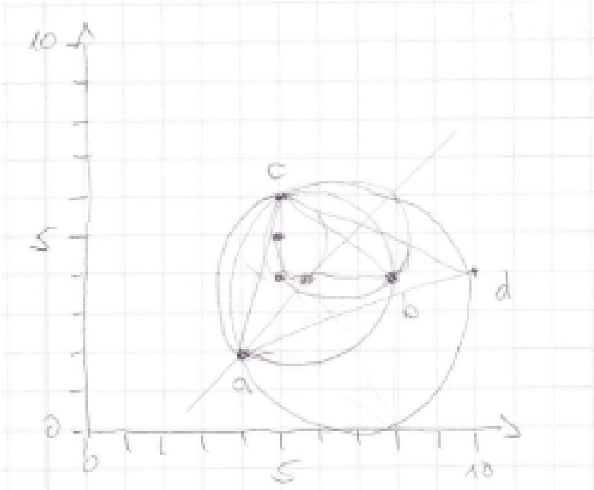
c) $|\underbrace{C \cap V}_{=\{v_1,v_2\}}| = 2 \Rightarrow \text{diam } C = \text{ dist } (v_1, v_2) = \text{diam } C \cap V$.

Proof. Exercise. □

2.7.2 Algorithm (Elzinga & Hearn [1971])
Input: $v_1, \ldots, v_m \in \mathbb{R}^2$.
Output: $p^* \in \text{argmin}\, 1/\mathbb{R}^2/\cdot/l_2/\max$.
Data structures: $a, b, c, d, p^* \in \mathbb{R}^2$.

1. Choose two points $a, b \in V$. Set $V \leftarrow V \backslash \{a, b\}$.
2. Set $C \leftarrow$ circle with diameter ab and center $p^* = \frac{1}{2}a + \frac{1}{2}b$.
   - If $\text{conv}\, C \supseteq V$: Goto 5.
   - Else: Choose c from $V \backslash \text{conv}\, C$. Set $V \leftarrow V \backslash \{c\}$.
3. If $\triangle abc$ has a right or obtuse angle:
   - Relabel a, b and c so that c is at right or obtuse angle.
   - Drop c.
   - Goto 2.
4. Set $C \leftarrow$ smallest enclosing circle of $\triangle abc$.
   Set $p^* \leftarrow$ center of C.
   - If $\text{conv}\, C \supseteq V$: Goto 5.
   - Else:
     Choose d from $V \backslash \text{conv}\, C$. Set $V \leftarrow V \backslash \{d\}$.
     Relabel a, b, c so that $\|a - d\|_2 \geq \|b - d\|_2$ and $\|a - d\|_2 \geq \|c - d\|_2$.
     Construct a line L_{ap^*} through a and p^*.
     Relabel b and c so that b and d are on the same side of L_{ap^*}.
     Set $b \leftarrow d$. Goto 3.
5. Output p^*.

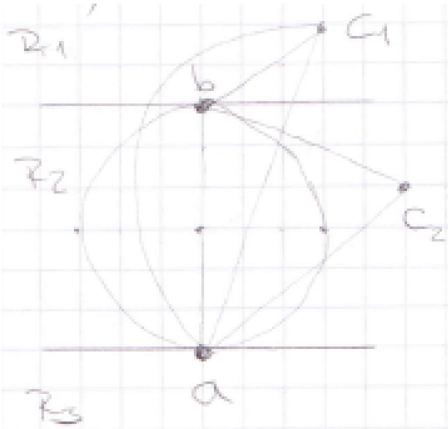
2.7.3 Example
$V = \{\underbrace{(5,4)}_{=a}, \underbrace{(5,6)}_{=b}, (5,5), (8,4), (4,2), (10,4)\}$



2.7.4 Theorem
Algorithm 2.7.2 is correct.

Proof. The algorithm produces circles C defined by 2 points in step 2 and by 3 points in step 4. We show that C has strictly increasing diameter. Then the algorithm terminates finitely with $C \supseteq \operatorname{conv} V$, as there is only a finite number of 2-point and 3-point circles.

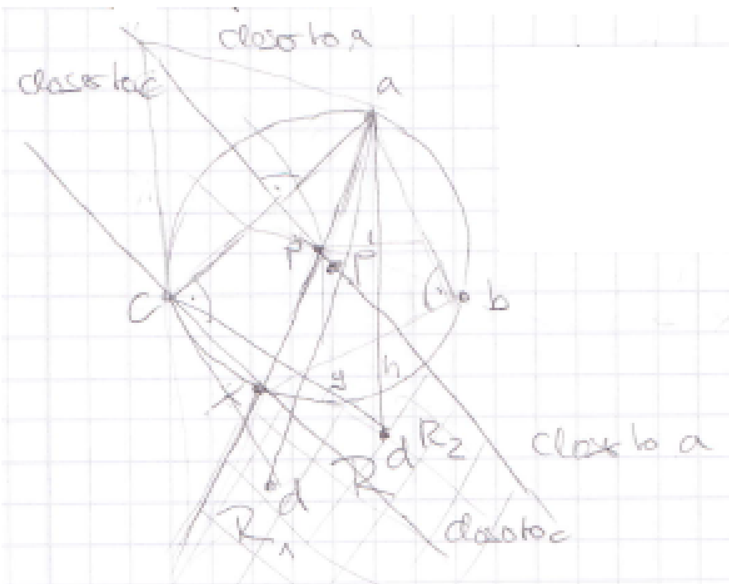
a) Let C be defined by 2 points a and b. Consider the three regions R_1, R_2, R_3 defined by a line through a and a line through b both orthogonal to ab.



(i) $\max \angle \triangle abc \geq \frac{\pi}{2} \Leftrightarrow c \in R_1 \cup R_3$, w.l.o.g. $c \in R_1$.
The next circle is defined by ac and ac > ab holds.

(ii) $\max \angle \triangle abc < \frac{\pi}{2}$.
The next circle is defined by $\triangle abc$, is distinct from the current circle and by Proposition 2.7.1 has a larger diameter.

b) Let C be defined by 3 points forming an acute triangle abc.
Let x be the opposite of a on C.



$\Rightarrow b \in \overset{\frown}{ax}, c \in \overset{\frown}{xa}$ ($\triangle abx$ is right)
$\Rightarrow d \in R$ with R defined as the intersection of
the c-side of the perpendicular ac bisector and the b-side of $L_{ax} = L_{ap^*}$
Split R by L_{cx} into regions R_1 and R_2.

(i) $d \in R_1$:
Then the triangle acx is right.
Hence the triangle acd is obtuse and $\|a - d\|_2 > \|a - x\|_2$.
The next circle has a larger diameter and comes within case a).

(ii) $d \in R_2$:
Then the triangle acd is acute at all of the three vertices a, c and d,
whose corresponding vertices are denoted α, γ and δ:
a: $ad > cd \Rightarrow (ad)^2 + (ac)^2 > (cd)^2 \overset{\text{cosine rule}}{=} (ad)^2 + (ac)^2 - 2(ad)(ac)\cos\alpha$
$\Rightarrow \cos\alpha > 0 \Leftrightarrow \alpha < \frac{\pi}{2}$
c: This follows directly from the location of d in R_2.
d: Let g and h be the intersections of cd and ad with C.
$\gamma = \frac{\overset{\frown}{ag}}{2}, \alpha = \frac{\overset{\frown}{hc}}{2} \Rightarrow \delta = \pi - \frac{\overset{\frown}{ag} + \overset{\frown}{hc}}{2} = \pi - \frac{\overset{\frown}{ac} + \overset{\frown}{gh}}{2} < \frac{\pi}{2}$
Let C' be the cycle through the triangle acd.
Then its center p' is situated on the perpendicular ac-bisector.
$pd > pc, p'd = p'c \Rightarrow p'd < pd, \underbrace{p'c}_{\text{radius of C'}} > \underbrace{pc}_{\text{radius of C}}.$

□

2.7.5 Algorithm (Smallest Enclosing Circle)
Input: $v_1, \ldots, v_m \in \mathbb{R}^2$
Output: $C(v_1, \ldots, v_m)$ // smallest circle enclosing $v_1, \ldots, v_m$
Data structures: $i \in \mathbb{N}$, circles C_i for $i = 1, \ldots, m-2$.

1. If $m = 1$: Output $C(v_1) = \{v_1\}$. Stop.
2. Set $i \leftarrow 2$. Choose $v_1, v_2 \in V$ randomly. Set $C_1 \leftarrow C(v_1, v_2)$.
3. If $i = m$: Output C_m. Stop.
4. Set $i \leftarrow i + 1$. Choose $v_i \in V \backslash \{v_1, \ldots, v_{i-1}\}$ randomly.
   - If $v_i \in \operatorname{conv} C_{i-1}$: Set $C_i \leftarrow C_{i-1}$.
   - Else: Set $C_i \leftarrow C^1(v_1, \ldots, v_i)$
     // smallest enclosing circle of $v_1, \ldots, v_{i-1}$ with v_i on the boundary.

   Goto 3.

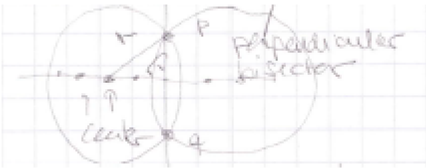
Function $C^1(v_1, \ldots, v_i)$
Input: $v_1, \ldots, v_i \in \mathbb{R}^2$
Output: $C^1(v_1, \ldots, v_i)$ // smallest circle enclosing $v_1, \ldots, v_{i-1}$ with v_i on the boundary
Data structures: $j \in \mathbb{N}$, circles C^1_j for $j = 1, \ldots, i-1$

1. If $i = 1$: Output $C^1(v_1) = \{v_1\}$. Stop.
2. Set $j \leftarrow 1$. Choose $v_1 \in V \backslash \{v_i\}$ randomly. Set $C^1_j \leftarrow C^1(v_1, v_2)$.
3. If $j = i - 1$: Output C^1_{i-1}. Stop.
4. Set $j \leftarrow j + 1$. Choose $v_j \in V \backslash \{v_1, \ldots, v_{j-1}, v_i\}$ randomly.
   - If $v_j \in \operatorname{conv} C^1_{j-1}$: Set $C^1_j \leftarrow C^1_{j-1}$.
   - Else: Set $C^1_j \leftarrow C^2(v_1, \ldots, v_j, v_i)$
     // smallest enclosing circle of $v_1, \ldots, v_{j-1}$ with v_j and v_i on the boundary.

   Goto 3.

Function $C^2(v_1, \ldots, v_j, v_i)$
Input: $v_1, \ldots, v_j, v_i \in \mathbb{R}^2$ // $m = j + 1$
Output: $C^2(v_1, \ldots, v_j, v_i) = C(v_1, \ldots, v_j, v_i)$
Data structures: circles C^2_L, C^2_R, C^2

1. Subdivide $\{v_1, \ldots, v_{j-1}\}$ into $L \dot\cup R$ along the perpendicular $v_j v_i$-bisector.
2. Set $C^2_L \leftarrow \max_{v \in L} C(v, v_j, v_i)$, $C^2_R \leftarrow \max_{v \in R} C(v, v_j, v_i)$ and $C^2 \leftarrow \max\{C^2_L, C^2_R\}$.
3. Output C^2.



2.7.6 Observation
a) The center of a circle through two points p and q lies on the perpendicular pq-bisector (either left or right).
b) Let $v_1, \dots, v_m, p, q \in \mathbb{R}^2$.
Then $C^2(v_1, \dots, v_m, p, q) = \max\{\underbrace{\max_{v_i \text{ left of } L_{pq}} C(v_i, p, q)}_{\text{leftmost centered circle}}, \underbrace{\max_{v_i \text{ right of } L_{pq}} C(v_i, p, q)}_{\text{leftmost centered circle}}\}$.

2.7.6 Proposition
Let $p_1, \dots, p_i \in \mathbb{R}^2$ with $p_i \notin \operatorname{conv} C(p_1, \dots, p_{i-1})$. Then

a) $p_i \in \delta C(p_1, \dots, p_i)$

b) $\forall P' \subseteq \{p_1, \dots, p_{i-1}\} : p_i \in \delta C(P', p_i)$

c) $p_j \notin \operatorname{conv} C^1(p_1, \dots, p_{j-1}, p_i) = \operatorname{conv} C(p_1, \dots, p_{j-1}, p_i) \Rightarrow p_j, p_i \in \delta C(p_1, \dots, p_j, p_i)$

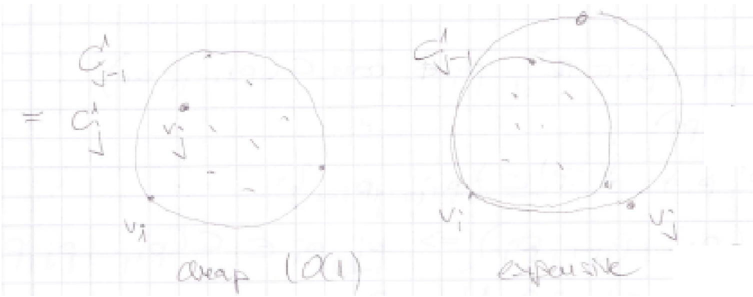
Proof.
a) clear
b) clear
c) follows from a) and b). □

2.7.7 Theorem
Let $T^2(m), T^1(m), T(m)$ be the expected run time
of computing $C^2(v_1, \dots, v_m), C^1(v_1, \dots, v_m), C(v_1, \dots, v_m)$, respectively.
Then $T(m) = T^1(m) = T^2(m) = O(m)$.

Proof.

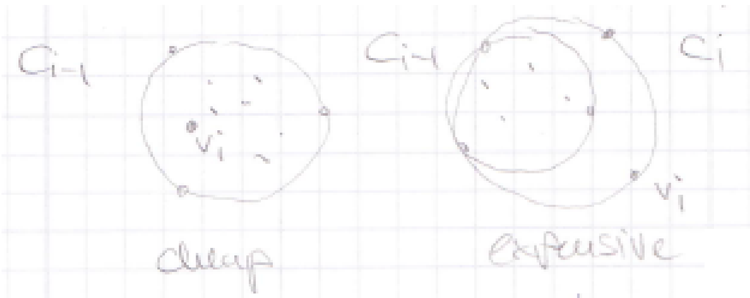
a) $T^2(m) = O(m)$ because $j - 1 = m - 2$ circles are constructed
and among these the circle with maximum diameter is selected and returned.

b) For $T^1(m)$ consider the treatment of v_j for $j = 1, \dots, i-1$.
Constructing C_j^1 out of C_{j-1}^1



is cheap (in $O(1)$) if $v_j \in C_{j-1}^1$ and expensive (in $O(j)$) otherwise.
$P[C_j^1 \text{ expensive}] = P[\forall\, V' \subseteq \{v_1, \dots, v_j\}, |V'| = j-1 : C(V') \neq C_j^1]$
$\leq \max\{\frac{1}{j}, \frac{2}{j}\} = \frac{2}{j}$.
The expected time for iteration j is $T_j^1 \in \frac{j-2}{j} O(1) + \frac{2}{j} O(j) = O(1)$.
$\Rightarrow T^1(m) = \sum_{j=1}^{m} T_j^1 \in O(m)$.

c) For $T(m)$ consider the treatment of v_i for $i = 1, \dots, m$.
Constructing C_i out of C_{i-1}



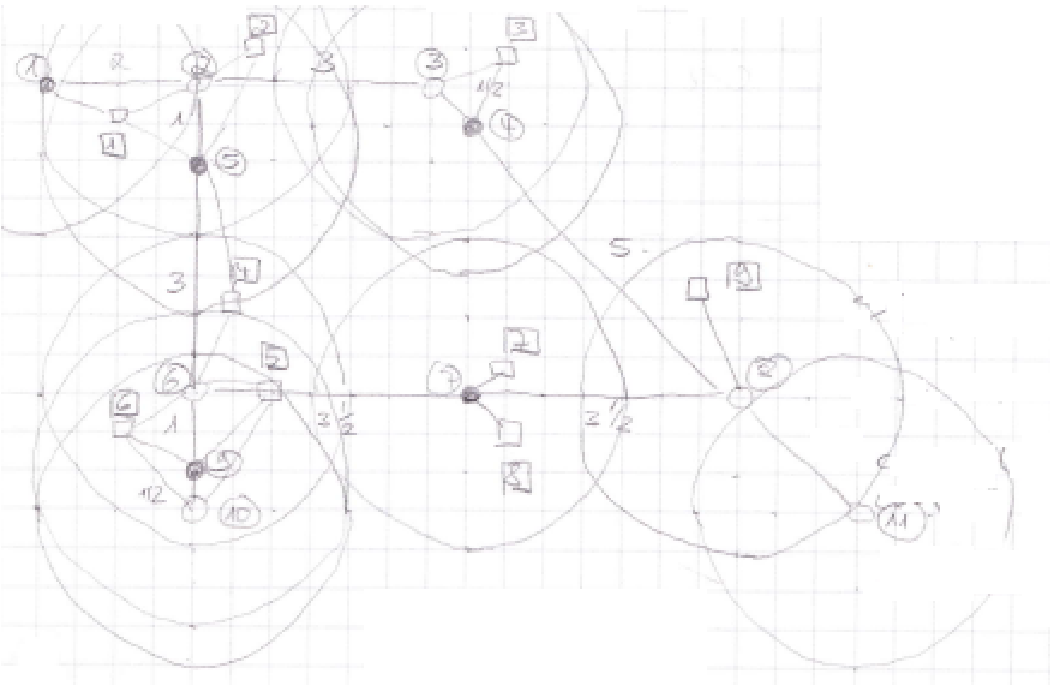
is cheap (in $O(1)$) if $v_i \in C_{i-1}$ and expensive (in $O(i)$) otherwise.
$P[C_i \text{ expensive}] = P[\forall\, V' \subseteq \{v_1, \dots, v_i\}, |V'| = i-1 : C(V') \neq C_i]$
$\leq \max\{\frac{2}{i}, \frac{3}{i}\} = \frac{3}{i}$.
The expected time for iteration i is $T_i \in \frac{i-3}{i} O(1) + \frac{3}{i} O(i) = O(1)$.
$\Rightarrow T(m) = \sum_{i=1}^{m} T_i \in O(m)$.

$\square$

2.8 Stop Location Problems in Networks

2.8.1 Definition (Bus Stop Location Problem (BSLP) (Schöbel [2007]))
$S \subseteq \mathbb{R}^2$ breakpoints, $T \subseteq \mathbb{R}^2$ existing stops
$N := (S \cup T, E)$ planar bus network
$V \subseteq \mathbb{R}^2$ demand points
$r \in \mathbb{R}_+$ covering radius
$\text{cov}_r(U) := \{v \in V : \|v, U\| \leq r\}$ cover of $U \subseteq \mathbb{R}^2$, $\text{cov}_r(u) = \text{cov}_r(\{u\})$ cover of $u \in \mathbb{R}^2$
$A_r^{\text{cov}}(U) := (\chi_{\text{cov}_r(u)}(v))_{v \in V, u \in U}$ covering matrix associated with $U \subseteq \mathbb{R}^2$

2.8.1 Example (Bus Stop Location Problem (BSLP) (Schöbel [2007]))
Let $S := \{2, 3, 6, 8, 10, 11\}$, $T := \{1, 4, 5, 7, 9\}, r := 2, V := \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.



Then $A_r^{\text{cov}}(U) = (\chi_{\text{cov}_r(u)}(v))_{v \in V, u \in S \cup T} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$.

Minimum cardinality $p = |U| = 5$ is attained at $U = \{3, 5, 6, 7, 8\}$.

2.8.2 Definition (Discrete Bus Stop Location Problems)

a) $(DSL):\ p = |U|/S/\mathrm{cov}_r(U) = V/l_2/p$ planning from scratch

b) $(DSL_1):\ p = |U|/T/\mathrm{cov}_r(U) = V/l_2/p$ closing stops

c) $(DSL_2):\ p = |U|/S/\mathrm{cov}_r(U \cup T) = V/l_2/p$ opening stops

d) $(DSL_3):\ p = |U|/S \cup T/\mathrm{cov}_r(U) = V/l_2/p$ closing and opening stops.

2.8.3 Observation
$(DSL_1), (DSL_2), (DSL_3)$ are special cases of (DSL).

Proof. Exercise. □

2.8.4 Definition (Continuous Bus Stop Location Problem)
$(CSL):\ p = |U|/N/\mathrm{cov}_r(U) = V/l_2/p$ planning from scratch

2.8.5 Remark (Alternative Objective Functions)
One can also consider:

a) $p = |U|/\{V, N\}/\,\mathrm{cov}_V(u)/l_2/\sum_{u \in U} w_u$ construction costs

b) $p = |U|/\{V, N\}/\,\mathrm{cov}_V(u)/sp/\sum_{ij \in V \times V} w_{ij}$ travel time (more difficult)

2.8.6 Proposition (Set Covering Model for the DSL)
$(DSL):\ p = |U|/\{V, N\}/\,\mathrm{cov}_r(U) = V/l_2/p$
can be formulated as a set covering problem as follows:

$$\begin{array}{lllll} (DSL) & \min & \mathbb{1}^T x & & \\ & (i) & A_r^{\mathrm{cov}}(s)x & \geq \mathbb{1} & \\ & (ii) & x & \in \{0,1\}^S & \end{array} \quad = \quad \begin{array}{lllll} (DSL) & \min & \sum_{u \in S} x_u & & \\ & (i) & \sum_{u \in S\,:\, v \in \mathrm{cov}_r(u)} x_u & \geq 1 & \forall v \in V \\ & (ii) & x_u & \in \{0,1\} & \forall u \in S \end{array}$$

2.8.7 Corollary (Complexity of (DSL))
(DSL) is NP-hard (and even PX-hard).

Proof. The Set Covering Problem is NP-hard (Exercise). □

2.8.8 Definition (Set Covering Problem)
Let $A \in \{0,1\}^{m \times n}$ and $c \in \mathbb{R}^n_{>0}$. Then a set covering problem is given by

$$\begin{array}{llll} (DSL) & \min & c^T x & \\ & (i) & Ax & \geq \mathbb{1} \\ & (ii) & x & \in \{0,1\}^n \end{array}$$

2.8.9 Algorithm (Greedy Algorithm for the Set Covering Problem)
Input: $c \in \mathbb{R}^n_{>0}$, $A \in \{0,1\}^{m \times n}$ with $Ax \geq \mathbb{1}$
Output: cover J_*
Data structures: $A_j \subseteq \{1, ..., m\}$ for $j = 1, \ldots, n$

0. Set $J_* \leftarrow \emptyset$ and $A_j \leftarrow \operatorname{supp}(A_{\cdot j})$ for $j = 1, \dots, n$.

1.
   - If $A_j = \emptyset$ for $j = 1, \dots, n$: Output J_*. Stop.
   - Else: Set $k \leftarrow \operatorname{argmax} \frac{|A_j|}{c_j}$.

2. Set $J_* \leftarrow J_* \cup \{k\}$ and $A_j \leftarrow A_j \backslash A_k$ for $j = 1, \dots, n$. Goto 1.

2.8.10 Remark

$\operatorname{argmax} \frac{|A_j|}{c_j} = \operatorname{argmin}_{|A_j|>0} \frac{c_j}{|A_j|}$

i.e. Algorithm 2.8.9 adds in each iteration a column k to the cover-to-be that minimizes the costs per yet uncovered rows.

2.8.11 Example

a) $\min \sum\limits_{j=1}^{7} x_j$ with respect to $\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_7 \end{pmatrix} \geq \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

Iteration 0: Set $J_* \leftarrow \emptyset$.
Iteration 1: Choose $k \in \{1, 2, 3, 5\}$, e.g. $k \leftarrow 1$, and set $J_* \leftarrow \{1\}$
Iteration 2: Choose $k \in \{3, 7\}$, e.g. $k \leftarrow 3$, and set $J_* \leftarrow \{1, 3\}$
Iteration 3: Output J_*.

$c(J_{\text{opt}}) = 2$, because there is no column that covers every row, but the two columns $\{1, 3\}$ do.

b) $A_j = \{j\}$ and $c_j = \frac{1}{j}$ for $j \in \{1, \dots, m\}$, $A_{m+1} = \{1, \dots, m\}$ and $c_{m+1} =: \alpha > 1$,

i.e. $c = \begin{pmatrix} 1 & \frac{1}{2} & \dots & \frac{1}{m} & \alpha \end{pmatrix}$ and $A = \begin{pmatrix} 1 & & & 1 \\ & \ddots & & \vdots \\ & & 1 & 1 \end{pmatrix}$

Iteration 0: Set $J_* \leftarrow \emptyset$.
Iteration 1: Set $k \leftarrow \operatorname{argmax}\{\frac{1}{\frac{1}{1}}, \frac{1}{\frac{1}{2}}, \dots, \frac{1}{\frac{1}{m}}, \frac{m}{\alpha}\} = m$ and $J_* \leftarrow \{m\}$
Iteration 2: Set $k \leftarrow \operatorname{argmax}\{1, \dots, m-1, \frac{m-1}{\alpha}\} = m - 1$ and $J_* = \{m - 1, m\}$
...
Iteration m: $k \leftarrow \operatorname{argmax}\{1, \frac{1}{\alpha}\} = 1, J_* \leftarrow \{1, \dots, m\}$
Iteration m+1: Output $J_* = \{1, \dots, m\}$.

$$c(J_*) = \underbrace{\sum_{i=1}^{m} \frac{1}{i}}_{=:H(m)} \leq H(m)\alpha = H(m)c(\{m + 1\}) = H(m)c(J_{\text{opt}})$$

2.8.12 Theorem (Harmonic Approximation Ratio for the Set Covering Problem)
Algorithm 2.8.9 produces a solution J_* such that
$c(J_*) \leq \sum\limits_{j \in J_{\text{opt}}} H(|A_j|)c_j \leq H(\max\limits_{j \in \{1,\dots,m\}} |A_j|)c(J_{\text{opt}}) \leq H(m)c(J_{\text{opt}})$

where $H(m) := \sum\limits_{i=1}^{m} \frac{1}{i}$ is the m-th harmonic number.

Proof. Consider the following chain of inequalities:

$$\begin{array}{lll} \min & \sum\limits_{j=1}^{n} H(\overbrace{|A_j|}^{=\sum_{i=1}^{m} a_{ij}})c_j x_j & \\ & \sum\limits_{j=1}^{n} a_{ij}x_j \geq 1 & \forall i \in \{1,...,m\} \\ & x_j \in \{0,1\} & \forall j \in \{1,...,n\} \end{array}$$

$$\overset{\text{LP relaxation}}{\geq} \begin{array}{lll} \min & \sum\limits_{j=1}^{n} H(\sum\limits_{i=1}^{m} a_{ij})c_j x_j & \\ & \sum\limits_{j=1}^{n} a_{ij}x_j \geq 1 & \forall i \in \{1,...,m\} \\ & x_j \geq 0 & \forall j \in \{1,...,n\} \end{array}$$

$$\overset{\text{LP duality}}{=} \begin{array}{lll} \max & \sum\limits_{i=1}^{n} y_i & \\ & \sum\limits_{i=1}^{n} a_{ij}y_i \overset{(1)}{\leq} H(\sum\limits_{i=1}^{m} a_{ij})c_j & \forall j \in \{1,...,n\} \\ & y_i \overset{(2)}{\geq} 0 & \forall i \in \{1,...,m\} \end{array}$$

$$\overset{(3)}{\geq} c(J_*) \geq \begin{array}{ll} \min & c^T x \\ & Ax \geq \mathbb{1} \\ & x \in \{0,1\}^n \end{array}$$

If (3) holds, the claim follows by setting $x = \chi_{J_{\text{opt}}}$.

Denote

a) $A_j^r :=$ set A_j at the beginning of iteration r

b) $w_j^r := |A_j^r|$

c) $J_*^r :=$ set J_* at the beginning of iteration r
(without loss of generality assume $J_*^r = \{1, \dots, r\}$)

d) $y_i := \frac{c_r}{w_r^r}$ for $i \in A_r^r$ (can be considered as the price to cover row i)

e) $t :=$ number of iterations in which J_* grows.

Then
(i) $\frac{w_r^r}{c_r} \geq \frac{w_j^r}{c_j}$ for $r = 1, ..., t$ and $j = 1, ..., n$
(ii) $c(J_*) = \sum_{i=1}^{t} c_j$
We claim that y as defined in d) satisfies (1), (2), (3):
(1): clear
(2): $\sum_{i=1}^{m} y_i = \sum_{r=1}^{t} (\sum_{i \in A_r^r} y_i) = \sum_{r=1}^{t} w_r^r \frac{c_r}{w_r^r} = \sum_{r=1}^{t} c_r = c(J_*)$
(3): $\sum_{i=1}^{m} a_{ij} y_i = \sum_{r=1}^{t} (\sum_{i \in A_j \cap A_r^r = A_j^r \setminus A_j^{r+1}} y_i) = \sum_{r=1}^{t} (w_j^r - w_j^{r+1}) y_i$
With $s := \max_{r=1}^{t} \{w_j^r \,:\, w_j^r > 0\}$ we obtain for $j = 1, ..., n$:
$\sum_{i=1}^{m} a_{ij} y_i = \sum_{r=1}^{s} (w_j^r - w_j^{r+1}) \frac{c_j}{w_r^r} \leq c_j \sum_{r=1}^{s} \frac{w_j^r - w_j^{r+1}}{w_j^r} \leq c_j \sum_{r=1}^{s} (H(w_j^r) - H(w_j^{r+1})) \leq c_j H(w_j^1)$

□

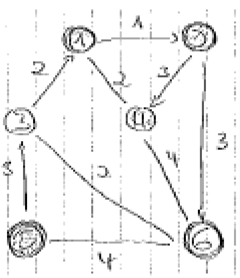
2.8.13 Example (Instance from Example 2.8.11 a))
$2 = c(\{1, 3\}) = c(J_*) \leq H(3) c(J_{\text{opt}}) = (1 + \frac{1}{2} + \frac{1}{3}) c(J_{\text{opt}}) = \frac{11}{6} c(J_{\text{opt}}) \Rightarrow c(J_{\text{opt}}) \geq 2$

3 The Uncapacitated Facility Location Problem

3.1 The General Uncapacitated Facility Location Problem

3.1.1 Definition (Uncapacitated Facility Location Problem (UFL))
$I \subseteq V$ potential facilities
$J \subseteq V$ customers (clients)
$A \subseteq I \times J$, $G = (V, E \cup A)$ network
$f_i \in \mathbb{R}_{\geq 0}, i \in I$ setup costs
$d_{ij} \in \mathbb{R}_{\geq 0}, ij \in A$ transportation costs $i \to j$
Uncapacitated Facility Location Problem (UFL): $\min\limits_{I^* \subseteq I} \sum\limits_{i \in I^*} f_i + \sum\limits_{j \in J} \min\limits_{i \in I^* : ij \in A} d_{ij}$
$G = (I \cup J, A)$ associated bipartite graph

3.1.1 Example (Schöbel & Schmidt [2009])



$I = \{1, 2, 5, 6\}$
$J = \{1, 2, 4, 5\}$
$A = I \times J$
$f = (f_i)_{i \in I} = (5, 5, 5, 5)$

$$d = (d_{ij})_{i \in I, j \in J} = \begin{pmatrix} 0 & 1 & 8 & 4 \\ 5 & 0 & 7 & 3 \\ 2 & 3 & 8 & 4 \\ 6 & 3 & 0 & 4 \end{pmatrix}$$

3.1.2 Proposition (IP Formulation for the UFL, Balinski [1965])
$y_i \in \{0, 1\}, i \in I$ facility setup variables
$x_{ij} \in \{0, 1\}, ij \in A$ client assignment variables

$$\begin{array}{llrrll} (UFL) & \min & \sum\limits_{i \in I} f_i y_i & + \sum\limits_{ij \in A} d_{ij} x_{ij} & & \\ & (i) & & \sum\limits_{i \in I} x_{ij} & \geq 1 & \forall j \in J \\ & (ii) & y_i & -x_{ij} & \geq 0 & \forall ij \in A \\ & (iii) & y_i & & \in \mathbb{Z}_+ & \forall i \in I \\ & (iv) & & x_{ij} & \in \mathbb{Z}_+ & \forall ij \in A \end{array}$$

a) (UFL) has an optimal 0/1-solution.

b) $(UFL) \Leftrightarrow (UFL)(i), (ii), (iii), x_{ij} \geq 0 \; \forall ij \in A$
i.e. the client assignment variables are automatically integral.

Proof. Exercise. □

3.1.3 Proposition (Set Covering Model for the UFL)
$J(i) := \{j \in J \;:\; ij \in A\}$
$\mathcal{J} := \{(i, J') \;:\; i \in I, \emptyset \subsetneq J' \subseteq J(i)\}$
$c_{(i,J')} = f_i + \sum_{j \in J'} d_{ij}$

$$\begin{array}{llll} (SCP) & \min & \sum_{(i,J') \in \mathcal{J}} c_{(i,J')} z_{(i,J')} & \\ & (i) & \sum_{(i,J') \in \mathcal{J} \,:\, j \in J'} z_{(i,J')} \quad \geq 1 & \forall j \in J \\ & (ii) & z_{(i,J')} \quad \in \{0,1\} & \forall (i,J') \in \mathcal{J} \end{array}$$

There is an one-to-one correspondence
between optimal solutions of (SCP) and (UFL).

Proof. Exercise. □

3.1.4 Corollary
a) The greedy algorithm for UFL is $H(\max_i |J(i)|)$-approximate.
b) UFL is APX-hard.

Proof. a) follows from Theorem 2.6.8.
b) SCP is APX-hard. Given $(SCP) \min c^T x, Mx \geq 1, x \in \{0,1\}^n$
with $M \in \{0,1\}^{m \times n}$ and $c \in \{0,1\}^n$,
construct in polynomial time an UFL with
$I = \{1, \dots, n\}, J = \{1, \dots, m\}, A = \{ij \in I \times J \;:\; m_{ij} = 1\}$ and $f_i = c_i$ for $i = 1, \dots, n$.
Then $(UFL) \Leftrightarrow (SCP)$ and the equivalence is approximation-preserving. □

3.1.5 Observation (Relation to the p-median problem)
Let $f_i = 0$ for all $i \in \{1, ..., n\}$. Then

a) $(UFL) \Leftrightarrow |I|/I/\cdot/d_{ij}/\sum$

b) $(UFL), (v) \sum_{i \in I} y_i \leq p \Leftrightarrow p/I/\cdot/d_{ij}/\sum$

3.2 The Metric Uncapacitated Facility Location Problem

3.2.1 Definition (Metric Uncapacitated Facility Location Problem (MUFL))
A Metric Uncapacitated Facility Location Problem (MUFL)
is an Uncapacitated Facility Location Problem (UFL) with

a) $I, J \subseteq \mathbb{R}^k, k \in \mathbb{N}$ (I and J are embedded in $\mathbb{R}^k$)

b) $A = I \times J$ (all assignments are possible)

c) $d_{ij} = \|i, j\|$ (distances with respect to some norm)

3.2.2 Algorithm (LP Rounding)
Input: $(MUFL)\ I, J \in \mathbb{R}^k, f_i \in \mathbb{R}_{\geq 0}\ \forall i \in I, d_{ij} = \|i, j\| \in \mathbb{R}\ \forall ij \in I \times J$
Output: (x, y) feasible for $(MUFL)$.

0. LP Solving: $(\overline{x}, \overline{y}) \leftarrow \text{argmin } (MUFL)_{LP}$ where

$$\begin{array}{llllll} (MUFL)_{LP} & \min & \sum\limits_{i \in I} f_i y_i & + \sum\limits_{ij \in A} d_{ij} x_{ij} & & \\ & (i) & & \sum\limits_{i \in I} x_{ij} & \geq 1 & \forall j \in J \\ & (ii) & y_i & -x_{ij} & \geq 0 & \forall ij \in I \times J \\ & (iii) & y_i & & \geq 0 & \forall i \in I \\ & (iv) & & x_{ij} & \geq 0 & \forall ij \in I \times J \end{array}$$

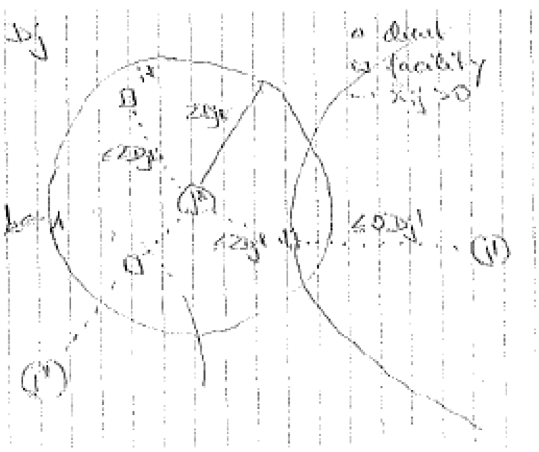
1. Filtering:

$D_j \leftarrow \sum\limits_{i \in I} d_{ij} \overline{x}_{ij}\ \forall j \in J$ // total fractional distance

$N_j \leftarrow \{i \in I\ :\ \underbrace{\overline{x}_{ij} > 0}_{\text{fractionally assigned}} \wedge \underbrace{d_{ij} \leq 2D_j}_{\text{near}}\}\ \forall j \in J$ // neighborhood

$x'_{ij} \leftarrow \begin{cases} 0 & i \notin N_j \\ 2\overline{x}_{ij} & \text{else} \end{cases} \quad \forall ij \in I \times J$

$y'_i \leftarrow \max\limits_{j \in J} x'_{ij} \forall i \in I$



2. Rounding:

   2a. $J^* \leftarrow \emptyset, (x^0, y^0) \leftarrow (x', y'), k \leftarrow 1$

   2b. $j^* \leftarrow \operatorname{argmin}_{j \in J \setminus J^*} D_j$

   $i^* \leftarrow \operatorname{argmin}_{i \in N_{j^*}} f_i$

   $EN_{j^*} \leftarrow \{j \in J \setminus J^* \ : \ N_{j*} \cap N_j \neq \emptyset\} \ni J^*$//extended neighborhood of j^*

$$y_i^k \leftarrow \begin{cases} 1 & i = i^* \\ 0 & i \in N_{j^*} \setminus \{i^*\} \text{ // select } i^*\text{, drop } N_{j^*} \setminus \{i^*\} \\ y_i^{k-1} & \text{else} \end{cases}$$

$$x_{ij}^k \leftarrow \begin{cases} 1 & i = i^*, j \in EN_{j^*} \\ 0 & i \in N_{j^*} \setminus \{i^*\}, j \in J \\ 0 & i \notin N_{j^*}, j \in EN_{j^*} \\ x_{ij}^{k-1} & \text{else} \end{cases} \text{ // network for clients in } EN_{j^*} \text{ to } i^*$$

   $J^* \leftarrow J^* \cup EN_{j^*}$

   $k \leftarrow k + 1$

   If $J^* \neq J$: Goto 2b.

   2c. Output (x^k, y^k).

3.2.3 Claim (w.r.t. Step 1)
$\forall j \in J : \sum_{i \in I} \overline{x}_{ij} \geq \frac{1}{2}$, in particular $N_j \neq \emptyset$.

3.2.4 Observation (w.r.t. Step 1)
(x', y') is feasible for $(MUFL)_{LP}$.

3.2.5 Observation (w.r.t. Step 1)
(x', y') satisfies

a) $x'_{ij} > 0 \Rightarrow d_{ij} \leq 2D_j$

b) $(x', y') \leq 2(\overline{x}, \overline{y})$

3.2.6 Claim (w.r.t. Step 2)

a) $j' \in EN_{j^*} \Rightarrow d_{i^*j'} \leq 2 \cdot 2D_{j^*} + 2D_{j'} \overset{D_{j^*} \leq D_{j'}}{\leq} 6D_{j'}$

b) $f_{i^*} \leq \underbrace{\sum_{i \in N_{j^*}} \underbrace{y'_i}_{\geq x'_{ij^*}}}_{\geq 1} f_i$

c) $$\underbrace{\sum_{i \in I} f_i y''_i}_{\overset{b)}{\leq} \sum_{i \in I} f_i y'_i \overset{3.2.5b)}{\leq} 2 \sum_{i \in I} f_i \overline{y}_i} + \underbrace{\sum_{ij \in I \times J} d_{ij} x''_{ij}}_{\overset{a)}{\leq} 6 \sum_{j \in J} D_j = 6 \sum_{j \in J} \sum_{i \in I} \overline{x}_{ij} d_{ij} \leq 6 \sum_{ij \in I \times J} \overline{x}_{ij} d_{ij}} \leq 6\left(\sum_{i \in I} f_i \overline{y}_i + \sum_{ij \in I \times J} d_{ij} \overline{x}_{ij}\right)$$

3.2.7 Theorem (Shmoys, Tardos & Aardal [1997])
Algorithm 3.2.3 is 6-approximate.

3.2.8 Lemma (Generalization of Claim 3.2.4)
For $\beta > 0$ let $N_j(\beta) := \{i \in I \,:\, \overline{x}_{ij} > 0 \wedge d_{ij} \leq \beta D_j\}$.
Then $\sum\limits_{i \in N_j(\beta)} \geq 1 - \frac{1}{\beta}$.

Proof. Let $F_j(\beta) := \{i \in I \,:\, \overline{x}_{ij} > 0, d_{ij} > \beta D_j\}$
and $s_j := \sum\limits_{i \in I} \overline{x}_{ij} \geq 1$.
Suppose $\sum\limits_{i \in F_j(\beta)} \overline{x}_{ij} > \frac{s_j}{\beta}$. Then
$D_j = \sum\limits_{i \in I} d_{ij}\overline{x}_{ij} = \sum\limits_{i \in F_j(\beta)} d_{ij}\overline{x}_{ij} + \sum\limits_{i \in N_j(\beta)} d_{ij}\overline{x}_{ij}$
$\geq \sum\limits_{i \in F_j(\beta)} d_{ij}\overline{x}_{ij} \geq \beta D_j \sum\limits_{i \in F_j(\beta)} \overline{x}_{ij} > \beta D_j \frac{s_j}{\beta} \geq D_j \lightning$
$\Rightarrow s_j = \underbrace{\sum\limits_{i \in F_j(\beta)} \overline{x}_{ij}}_{\leq \frac{s}{\beta}} + \sum\limits_{i \in N_j(\beta)} \overline{x}_{ij}$
$\Rightarrow \sum\limits_{i \in N_j(\beta)} \overline{x}_{ij} \geq s_j - \frac{s_j}{\beta} = s_j(1 - \frac{1}{\beta}) \geq 1 - \frac{1}{\beta}$ □

3.2.9 Definition (Dual of $(MUFL)_{LP}$)

$$
\begin{array}{llcclll}
(MUFL)_{LP} & \min & \sum\limits_{i \in I} f_i y_i & + \sum\limits_{ij \in I \times J} d_{ij} x_{ij} & & & \\
& (i) & & \sum\limits_{i \in I} x_{ij} & \geq 1 & \forall j \in J & \to \alpha_j \\
& (ii) & y_i & -x_{ij} & \geq 0 & \forall ij \in I \times J & \to \beta_{ij} \\
& (iii) & y_i & & \geq 0 & \forall i \in I & \\
& (iv) & & x_{ij} & \geq 0 & \forall ij \in I \times J & \\
(MUFL)_{DP} & \max & \sum\limits_{j \in J} \alpha_j & & & & \\
& (i) & & \sum\limits_{j \in J} \beta_{ij} & \leq f_i & \forall i \in i & \leftarrow y_i \\
& (ii) & \alpha_j & -\beta_{ij} & \leq d_{ij} & \forall ij \in I \times J & \leftarrow x_{ij} \\
& (iii) & \alpha_j & & \geq 0 & \forall j \in J & \\
& (iv) & & \beta_{ij} & \geq 0 & \forall ij \in I \times J &
\end{array}
$$

Interpretation of the dual program $(MUFL)_{DP}$:
α_j: total cost paid by j to get assigned to an open facility
β_{ij}: cost paid by j to open facility i
By complementary slackness optimal solutions satisfy:
$y_i > 0 \Rightarrow \sum\limits_{j \in J} \beta_{ij} = f_i$: To open a facility i, f_i needs to be paid by the clients.
$x_{ij} > 0 \Rightarrow \alpha_j - \beta_{ij} = d_{ij} \Rightarrow \alpha_j = d_{ij} + \beta_{ij}$:
If j is connected to i, j pays the assignment cost and its share for opening i.
$\alpha_j > 0 \Rightarrow \sum\limits_{i \in I} x_{ij} = 1$ (not interesting)
$\beta_{ij} > 0 \Rightarrow x_{ij} - y_i = 0 \Rightarrow x_{ij} = y_i$: Client j only pays its share of f_i if it uses facility i.

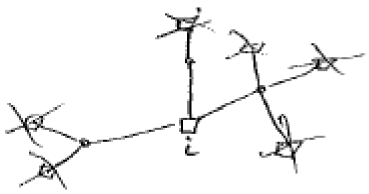
3.2.10 Algorithm (Primal-dual approximation algorithm for $(MUFL)$ (Jain & Vazirani [2001]))
Input: $(MUFL)$ $I, J \in \mathbb{R}^k, f_i \in \mathbb{R}_{\geq 0}\ \forall i \in I, d_{ij} = \|i,j\| \in \mathbb{R}\ \forall ij \in I \times J$
Output: (x,y) feasible for $(MUFL)$
Definitions used in the algorithm:
- $T := \{i \in I\ :\ \sum_{j \in J} \beta_{ij} = f_i\}$ temporarily opened facilities
- j neighbors i $:\Leftrightarrow \alpha_j \geq d_{ij}$
- j contributes to i $:\Leftrightarrow \beta_{ij} > 0$

1. Initial Construction:

   1a. Set $x, y, \alpha, \beta \leftarrow 0$.

   1b. While there exists a client not neighboring a facility in T:
   - Raise α_j uniformly for all such j.
   - If j becomes a neighbor of some $i \notin T$,
     increase β_{ij} at the same rate to maintain dual feasibility.
   - If j becomes a neighbor of some $i \in T$, freeze α_j.

2. Cleaning:

   While there exists a client contributing to at least two facilities in $T(\beta)$:

   - Pick such a facility i.
   - Set $T \leftarrow T \backslash \{h \in T \backslash \{i\}\ :\ \exists j \in J : \beta_{ij} > 0 \wedge \beta_{hj} > 0\}$,
     i.e. close all other facilities contributed by at least one client
     which also contributes i.

3. Final Assignment:
   For all $i \in T$: Set $y_i = 1$.
   For all $j \in J$:

   - If j contributes to an $i \in T$ (unique after Step 2): Set $x_{ij} = 1$.
   - Else if j does not contribute to an $i \in T$ but neighbors some $i \in T$:
     Set $x_{ij} = 1$ for one such i.
   - Else (j has no neighbors in T):
     Let i be the facility in T closest to j. Set $x_{ij} = 1$.

   Output (x,y).

3.2.11 Lemma
If at the end of Step 2 there is a client j with no neighbors in T,
there exists $i \in T$ such that $d_{ij} \leq 3\alpha_j$.

Proof.

Claim 1: Given such j, there exist facilities h and i and a client k such that

- we stopped increasing α_j when j neighbored some $h \in T$.
- h was removed from T in Phase 2
  because some client was contributing both to h
  and to some i that remained in T.

Claim 2: $d_{ij} \leq 3\alpha_j$

Proof by triangle inequality:

- $d_{ij} \leq \alpha_j$ because h and j are neighbors.
- k contributes to h. $\Rightarrow \beta_{hk} > 0 \Rightarrow \alpha_k > d_{hk}$
- k contributes to i. $\Rightarrow \beta_{ik} > 0 \Rightarrow \alpha_k > d_{ik}$

When α_j stops growing, h is in T. k contributed to h
$\Rightarrow$ k already neighbors h.
$\Rightarrow \alpha_k$ does not grow after α_j stops. $\Rightarrow \alpha_j \geq \alpha_k$
$\Rightarrow \alpha_j \geq \alpha_k \geq d_{hk}, d_{ik} \Rightarrow d_{ij} \leq d_{hj} + d_{hk} + d_{ik} \leq 3\alpha_j$ □

3.2.12 Theorem
Algorithm 3.2.10 is 3-approximative.

Proof. For every facility i let $A(i) := \{\text{neighbors j of } i \ : \ x_{ij} = 1\}$
$\Rightarrow A(i)$ and $A(i')$ are disjoint for two distinct facilities i and i'.
Let $Z := J \setminus \bigcup_{i \in T} A(i)$ the set of clients with no neighbors in T.

$$\sum_{i \in T} \sum_{j \in Z} d_{ij} x_{ij} = \sum_{j \in Z} \sum_{i \in T} d_{ij} x_{ij} \overset{x_{ij}=1 \Leftrightarrow \text{ i closest facility to j}}{\leq} \sum_{j \in Z} \min_{i \in T} d_{ij} \overset{\text{Lemma 3.2.9}}{\leq} \sum_{j \in Z} 3\alpha_j$$

$$\Rightarrow \sum_{i \in T} f_i + \sum_{j \in J} d_{ij} x_{ij} = \sum_{i \in T} \sum_{j \in A(i)} \beta_{ij} + \sum_{j \in J} d_{ij} x_{ij} \leq \sum_{i \in T} \sum_{j \in A(i)} \alpha_j + \sum_{j \in Z} 3\alpha_j \leq 3 \sum_{j \in J} \alpha_j$$

$\leq 3OPT_{DP} \leq 3OPT_{LP}$ □

3.3 Local Search for the Metric p-Median Problem

3.3.1 Definition (Neighborhood, Swap for the Metric p-Median Problem)
Given $V = J \in \mathbb{R}^k$, consider the metric p-median problem $p/I/\cdot/\|\cdot\|/\sum$.

a) $\mathcal{S} := \{S \subseteq I \ : \ |S| = p\}$ (set of feasible solutions for the p-median problem)

b) $c(S) := \sum_{j \in J} \|S, j\| = \sum_{j \in J} \min_{i \in S} \|i, j\|$ (cost of a solution)

c) $\Gamma(S) := \{S' \in \mathcal{S} \ : \ |S' \triangle S| = 2\}$ (neighborhood of a solution)

d) $i \to i' \ : \ \{S \in \mathcal{S} \ : \ i \in S, i' \notin S\} \to \mathcal{S}, S \mapsto S \backslash \{i\} \cup \{i'\}$ (swap)

e) $S \in \mathcal{S}$ local optimum $:\Leftrightarrow \forall S' \in \Gamma(S) : c(S) \leq c(S')$

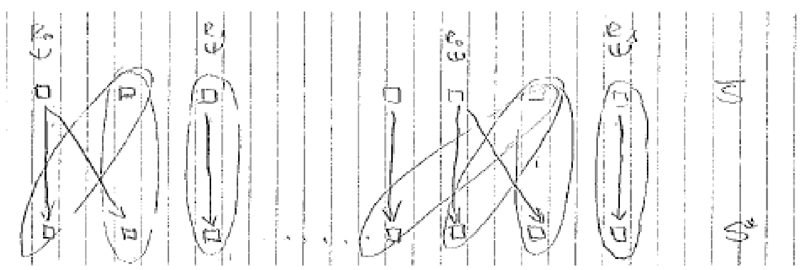
3.3.2 Remark
$p/I/\cdot/\|\cdot\|/\sum \leftrightarrow (MUFL) \sum_{i \in I} y_i = p$ with $f = 0$ constant

3.3.3 Algorithm (Local Search for the Metric p-Median Problem)
Input: $p/I/\cdot/\|\cdot\|/\sum, V = J, S \in \mathcal{S}$
Output: $S \in \mathcal{S}$ local optimum
While $S' \in \Gamma(S)$ with $c(S') < c(S)$ exists: Set $S \leftarrow S'$.
Output S.

3.3.4 Remark (Arya, Gary, Mulagala, Pandit [2001])
Let $\overline{S}$ be a local optimum and S^* be a global optimum of the metric p-median problem.
Then the following holds:
a) $c(\overline{S}) \leq 3c(S^*)$
b) For all $S \in \mathcal{S}$ exists $S' \in \Gamma(S)$ with decrease per swap $c(S) - c(S') \geq \frac{c(S) - 3c(S^*)}{n^2}$.
c) This yields a $3 + O(1)$-approximate polynomial time local search algorithm.

3.3.5 Theorem (5-Approximation Ratio for Local Search)
Let S^* be a global optimum of the metric p-median problem.
Then Algorithm 3.3.3 yields a local optimum $\overline{S}$ such that $c(\overline{S}) \leq 5c(S^*)$.

Proof. Define a map $\eta : S^* \to \overline{S}, i^* \to \underset{i \in \overline{S}}{\operatorname{argmin}} \|i, i^*\|$
that maps each facility in S^* to the closest facility in $\overline{S}$.



For $k = 0, 1$ let $R_k := \{i \in \overline{S} \;:\; \eta(i^*) = i \text{ for exactly k facilities } i^* \in S^*\}$.
Construct a set P of k swaps, one for each $i^* \in S^*$, as follows:

a) $i \in R_1 \Rightarrow i \rightarrow \eta^{-1}(i) \in P$

b) $|R_0| + \underbrace{|\overline{S}\backslash(R_0 \cup R_1)|}_{\leq \frac{1}{2}|S^*\backslash\eta^{-1}(R_1)|} = |S^*\backslash\eta^{-1}(R_1)|$
$\Rightarrow 2|R_0| \geq |S^*\backslash\eta^{-1}(R_1)|$
$i \in R_0 \Rightarrow i \rightarrow i^* \in P$ for at most 2 arbitrarily chosen $i^* \in S^*\backslash\eta^{-1}(R_1)$

Remark:

a) $i \in R_1$ is close to $\eta^{-1}(i)$ and all other $i^* \in S^*$ are far away
$\Rightarrow i \rightarrow \eta^{-1}(i)$ can be handled by assigning all of i's clients to $\eta^{-1}(i)$.

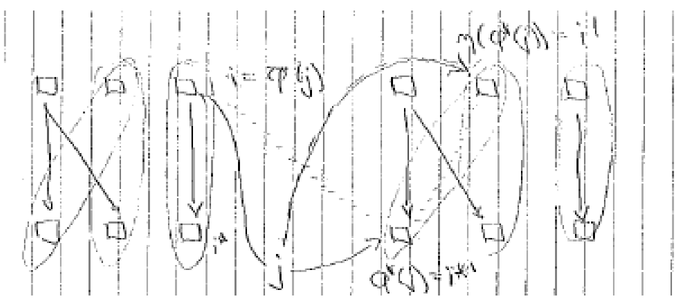
b) $i \in \overline{S}\backslash(R_0 \cup R_1)$ is close to a several facilities in $\eta^{-1}(i)$.
$\Rightarrow i \rightarrow i^* \in \eta^{-1}(i)$ and assigning all of i's clients to i^* can cost much.
We consider only swaps $(i \rightarrow i^*) \in P$ with $i \in R_0 \cup R_1$.

Claim 1: For $i \in \overline{S}, i^*, i^{*\prime} \in S^*$ and $(i \rightarrow i^*) \in P$ it holds $\eta(i^{*\prime}) \neq i$.

Let
$\overline{\varphi} : J \rightarrow \overline{S}, j \mapsto \text{argmin}_{s \in \overline{S}} \|\overline{S}, j\|$,
$\varphi^* : J \rightarrow \overline{S}, j \mapsto \text{argmin}_{s \in \overline{S}} \|\overline{S}^*, j\|$
be functions mapping j to the closest facility in $\overline{S}$ and S^*, respectively, and
$\overline{N}(i) = \overline{\varphi}^{-1}(i), i \in \overline{S}$
$N^*(i) = \varphi^{*-1}(i), i \in S^*$
be the sets of clients assigned to $i \in \overline{S}$ and $i \in S^*$, respectively.

Claim 2: For each swap $i \rightarrow i^* \in P$ it holds

$$0 \overset{\overline{S}\text{ local optimum}}{\leq} c(\overline{S}\backslash\{i\} \cup \{i^*\}) - c(\overline{S}) \leq \sum_{j \in N^*(i^*)} (\|S^*, j\| - \|\overline{S}, j\|) + \sum_{j \in N(i)} 2\|S^*, j\|$$



Consider the following assignment $\varphi' : J \to \overline{S}\backslash\{i\} \cup \{i^*\}$:

$$\varphi'(j) := \begin{cases} i^* & j \in N^*(i) \\ i' := \eta(\varphi^*(j)) & j \in \overline{N}(i)\backslash N^*(i^*) \\ \overline{\varphi}(j) & j \in J\backslash(\overline{N}(i) \cap N^*(i^*)) \end{cases}$$

From claim 1 follows $i \neq i'$, so φ' is well defined. Further

$$\begin{aligned}
&c(\overline{S}\backslash\{i\} \cup \{i^*\}) - c(\overline{S}) \\
&= \sum_{j\in J}(\|\overline{S}\backslash\{i\} \cup \{i^*\}, j\|) - \|\overline{S}, j\| \\
&= \sum_{j\in N^*(i^*)} (\|i^*, j\| - \|\overline{S}, j\|) + \sum_{j\in\overline{N}(i)\backslash N^*(i^*)} (\|i', j\| - \|i, j\|) \\
&\leq \sum_{j\in N^*(i^*)} (\|i^*, j\| - \|\overline{S}, j\|) + \sum_{j\in\overline{N}(i)\backslash N^*(i^*)} (\|i', j\| - \|i, j\|) \\
&\leq \sum_{j\in N^*(i^*)} (\underbrace{\|i^*, j\|}_{=\|S^*,j\|} - \|\overline{S}, j\|) + \sum_{j\in\overline{N}(i)\backslash N^*(i^*)} (\underbrace{\|i^{*\prime}, j\| + \underbrace{\|i^{*\prime}, i'\|}_{\leq\|i^{*\prime},i\|} - \|i, j\|}_{\leq\|i^{*\prime},j\|}) \\
&\leq \sum_{j\in N^*(i^*)} (\|S^*, j\| - \|\overline{S}, j\|) + \sum_{j\in\overline{N}(i)\backslash N^*(i^*)} 2\|S^*, j\| \\
&\leq \sum_{j\in N^*(i^*)} (\|S^*, j\| - \|\overline{S}, j\|) + \sum_{j\in\overline{N}(i)} 2\|S^*, j\| \text{ (Claim 2)}
\end{aligned}$$

In the summation of the inequality of Claim 2 over all $(i \to i^*) \in P$ every $i \in S$ appears twice and every $i^* \in S^*$ appears once:

$$0 \leq \sum_{i^*\in S^*} \sum_{j\in N^*(i^*)} (\|S^*, j\| - \|\overline{S}, j\|) + 2 \sum_{i\in S} \sum_{j\in\overline{N}(i)} 2\|S^*, j\| = 5c(S^*) - c(\overline{S})$$

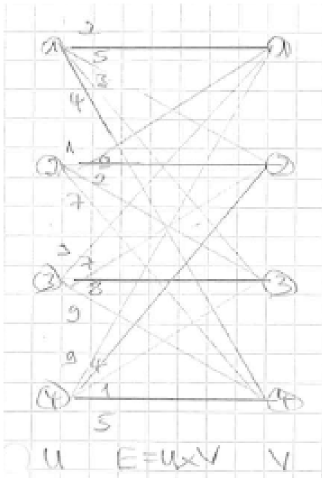
$\Rightarrow c(\overline{S}) \leq 5c(S^*)$. □

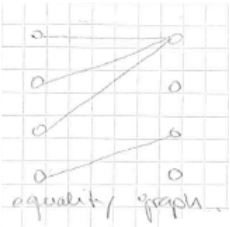
3.4 The Assignment Problem

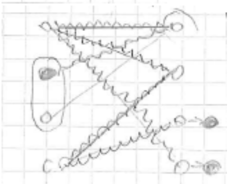
3.4.1 Example (Assignment Problem or Perfect Bipartite Matching Problem)
$G = (U \dot\cup V, E)$ complete bipartite graph with $|U| = |V|$ (equal sized shores)
$M \subseteq E : |M \cap \delta(w)| = 1 \; \forall w \in U \cup V$ perfect matching
$(c_{ij})_{i \in U, j \in V}$ cost matrix with $c_{ij} \geq 0$ costs
$u_i := \min\limits_{j \in V} c_{ij}$ row minima
$(c'_{ij})_{i \in U, j \in V} := (c_{ij} - u_i)_{i \in U, j \in V}$ reduced cost matrix with respect to u
$\overline{E} := \{e \in E \;:\; c'_e = 0\}$, $\overline{G} := (U \cup V, \overline{E})$ equality graph
$v_j := \min\limits_{i \in U} c'_{ij} = \min\limits_{i \in U}(c_{ij} - u_i)$ column minima
$(\overline{c}_{ij})_{i \in U, j \in V} := (c_{ij} - u_i - v_j)_{i \in U, j \in V}$ reduced cost matrix with respect to u and v
$\min c(M) = \sum\limits_{ij \in M} c_{ij}$ minimum cost perfect matching

Let $(c_{ij}) = \begin{pmatrix} 2 & 5 & 3 & 4 \\ 1 & 9 & 2 & 7 \\ 3 & 7 & 8 & 9 \\ 9 & 4 & 1 & 5 \end{pmatrix}, u_i = \begin{pmatrix} 2 \\ 1 \\ 3 \\ 1 \end{pmatrix}$

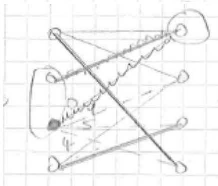
$(c'_{ij}) = (c_{ij} - u_i) = \begin{pmatrix} 0 & 3 & 1 & 2 \\ 0 & 7 & 1 & 6 \\ 0 & 4 & 5 & 6 \\ 8 & 3 & 0 & 4 \end{pmatrix}, v_i = \begin{pmatrix} 0 & 3 & 0 & 2 \end{pmatrix}$



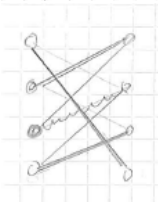




$(\overline{c}_{ij}) = (c'_{ij} - v_j) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 4 & 1 & 4 \\ 0 & 1 & 5 & 4 \\ 8 & 0 & 0 & 2 \end{pmatrix}, u_i = \begin{pmatrix} 2 \\ 1 \\ 3 \\ 1 \end{pmatrix}, v_i = \begin{pmatrix} 0 & 3 & 0 & 2 \end{pmatrix}$



$\overset{\text{dual update}}{\rightarrow} \overline{c}_{ij} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 4 & 1 & 4 \\ 0 & 1 & 5 & 4 \\ 8 & 0 & 0 & 2 \end{pmatrix}, u_i = \begin{pmatrix} 2 \\ 2 \\ 4 \\ 1 \end{pmatrix}, v_i = \begin{pmatrix} -1 & 3 & 0 & 2 \end{pmatrix}$



$\overset{\text{primal update}}{\rightarrow} \overline{c}_{ij} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 3 & 0 & 3 \\ 0 & 0 & 4 & 3 \\ 9 & 0 & 0 & 2 \end{pmatrix}, u_i = \begin{pmatrix} 2 \\ 2 \\ 4 \\ 1 \end{pmatrix}, v_i = \begin{pmatrix} -1 & 3 & 0 & 2 \end{pmatrix}$

3.4.2 Definition (Assignment Problem or Perfect Bipartite Matching Problem)
Input: $G = (U \dot{\cup} V, E)$ with $|U| = |V|, E = U \times V, c_{ij} \in \mathbb{R}\ \forall ij \in E$
Output: $M \subseteq E$ perfect matching of minimum cost $c(M) = \sum\limits_{ij \in M} c_{ij}$

3.4.3 Definition (Assignment Terminology)
Let $G = (U \dot{\cup} V, E)$ with $|U| = |V|$, $E = U \times V$ and $c_{ij} \in \mathbb{R}$ for all $ij \in E$.

a) Let $F \subseteq E$:
$U(F) := \{i \in U\ :\ \exists\, j \in V\ :\ ij \in F\}$
$V(F) := \{i \in V\ :\ \exists\, j \in U\ :\ ij \in F\}$

b) Let $u \in \mathbb{R}^U, v \in \mathbb{R}^V, ij \in E$:
$\overline{c}_{ij} := \overline{c}_{ij}(u, v) = c_{ij} - u_i - v_j$ reduced cost with respect to u and v
$\overline{E} := \{ij \in E : \overline{c}_{ij} = 0\}, \overline{G} := (U \cup V, \overline{E})$ equality graph

c) Let $u \in \mathbb{R}^U, v \in \mathbb{R}^V, M \subseteq \overline{E}$ matching, $P \subseteq \overline{E}$ path, $u_0 \in U(P) \backslash U(M)$
P alternating u_0v_k-path $:\Leftrightarrow P = \overbrace{u_0v_1}^{\notin M}, \overbrace{v_1u_2}^{\in M}, \overbrace{u_2v_3}^{\notin M}, \ldots, \overbrace{u_{k-1}v_k}^{\notin M}$
P augmenting u_0v_k-path $:\Leftrightarrow P = u_0 \ldots v_k$ alternating with $v_k \in V \backslash V(M)$

d) Let $u \in \mathbb{R}^U, v \in \mathbb{R}^V, M \subseteq \overline{E}$ matching, $T \subseteq E$ tree, $r \in U(T) \backslash U(M)$
T alternating tree $:\Leftrightarrow$ all paths in T are alternating

e) $u \in \mathbb{R}^U, v \in \mathbb{R}^V$ dual feasible $:\Leftrightarrow \overline{c}_{ij} = c_{ij} - u_i - v_j \geq 0 \Leftrightarrow u_i + v_j \leq c_{ij}$

3.4.4 Proposition (IP Formulation of the Assignment Problem)
The assignment problem can be formulated as an IP as follows:

$$
\begin{array}{llll}
(AP) & \min & \sum\limits_{ij \in E} c_{ij} x_{ij} & \\
& & \sum\limits_{j \in V} x_{ij} = 1 & \forall i \in U \\
& & \sum\limits_{i \in U} x_{ij} = 1 & \forall j \in V \\
& & x_{ij} \in \{0,1\} & \forall ij \in E
\end{array}
$$

$$
\begin{array}{lll}
\Leftrightarrow: & \min & c^T x \\
& & A(G)x = \mathbb{1} \\
& & x \in \{0,1\}^E
\end{array}
$$

3.4.5 Proposition (Total Unimodularity of $A(G)$)
A(G) is totally unimodular.

Proof. Exercise. □

3.4.6 Corollary (LP Formulation of the Assignment Problem)
The LP relaxation of the assignment problem has 0/1 extreme solutions.
Hence, the assignment problem can be formulated as the following LP and its dual:

$$
\begin{array}{llllll}
(LP) & \min & \sum\limits_{ij\in E} c_{ij}x_{ij} & & & \\
& & \sum\limits_{j\in V} x_{ij} & = 1 & \forall i \in U & \to u_i \\
& & \sum\limits_{i\in U} x_{ij} & = 1 & \forall j \in V & \to v_j \\
& & x_{ij} & \geq 0 & \forall ij \in E &
\end{array}
$$

$$
\begin{array}{llllll}
\Leftrightarrow (DP) & \max & \sum\limits_{i} u_i & + \sum\limits_{i} v_j & & \\
& & u_i & +v_j & \leq c_{ij} & \forall ij \in E
\end{array}
$$

3.4.7 Corollary (Dual Solution of the Assignment Problem)
Let $u \in \mathbb{R}^U, v \in \mathbb{R}^V$ be dual feasible.
Then $\sum\limits_{i\in U} u_i + \sum\limits_{j\in V} v_j \leq c(M)$ for all matchings M.

Proof. $c(M) = \sum\limits_{ij\in M} c_{ij} = \sum\limits_{i\in U} c_{ij(i)} \geq \sum\limits_{i\in U} u_i + v_{j(i)} = \sum\limits_{i\in U} u_i + \sum\limits_{j\in V} v_j.$ □

3.4.8 Proposition (Alternating Paths)
Let $M \subseteq \overline{E}$ be a matching. Then the following are equivalent:
M is maximal with respect to inclusion.
⇔ There is no augmenting path in $\overline{G}$.
⇔ There is no augmenting path in a maximal forest of alternating trees in $\overline{G}$.

Proof. Exercise. □

3.4.9 Algorithm (Hungarian Method (Kuhn [1955] & Munkres [1957]))
Input: $G = (U\dot\cup V, E)$ with $|U| = |V|$, $E = U \times V, c_{ij} \in \mathbb{R}\ \forall ij \in E$
Output: $M \in \text{argmin}(LP), (u, v) \in \text{argmax}(DP) \Rightarrow c(M) = \sum\limits_{i\in U} u_i + \sum\limits_{j\in V} v_j$

Data structures: $\overline{G}, \overline{c}$

1. Set $u \leftarrow (\min\limits_{j\in V} c_{ij})_{i\in U}$ and $v \leftarrow (\min\limits_{i\in U}(c_{ij} - u_i))_{j\in V}$.
2. Choose a maximum matching M in $\overline{G}$.
3. If $|M| = |U|$: Output M and (u, v). Stop.
4. Choose an unmatched node $r \in U(T)$.
   Calculate the maximal alternating r-tree T in $\overline{G}$.
   - 4a. If there is an augmenting rj-path P in T:
     Primal update: Set $M \leftarrow M\backslash(P \cap M) \cup (P\backslash M)$. Goto 3.
   - 4b. Dual update:
     Calculate $\varepsilon \leftarrow \min\limits_{ij\in U(T)\times(V\backslash V(T))} c_{ij} - u_i - v_j$.
     Set $u_i \leftarrow u_i + \varepsilon$ for all $i \in U(T)$.
     Set $v_j \leftarrow v_j - \varepsilon$ for all $j \in V(T)$. Goto 4.

3.4.10 Theorem (Runtime of the Hungarian Method (Kuhn [1960]))
Algorithm 3.4.9 is correct.

Proof. We obtain the correctness from the following claims:

a) **Claim:** Without loss of generality c_{ij} is integer for each edge $ij \in E$.

b) **Claim:** M grows by exactly one edge in step 4a.
Proof. The definition of an augmenting path P yields $|P \cap M| = |P \cap M| + 1$.

c) **Claim:** $|U(T)| > |V(T)|$ in step 4b.
Proof. Every alternating path in T is non-augmenting and therefore terminates at a node from $U(T)$. Because T consists of alternating paths starting in $U(T)$, for each node in $V(T)$ there is a matching edge in T to a node in $U(T)\backslash\{r\}$. On the other hand, each node in $U(T)\backslash\{r\}$ is matched to exactly one node in $V(T)$. This yields a bijection $U(T)\backslash\{r\} \leftrightarrow V(T)$ and $r \in U(T)$ implies $|U(T)| = |V(T)| + 1$.

d) **Claim:** $\varepsilon \geq 1$
Proof. Let T be a maximal alternating tree T in $\overline{E}$, $i \in U(T)$ and $j \in V\backslash V(T)$.
Then $ij \notin M$ because of $|M \cap \delta(i)| \leq 1$
and $ij \notin \overline{E}$, because otherwise $ij \in \overline{E}\backslash M$ could be added to T.
Hence $U(T) \times (V\backslash V(T)) \subseteq E\backslash\overline{E}$ and $\varepsilon = \min\limits_{ij \in U(T)\times(V\backslash V(T))} c_{ij} \geq \min\limits_{ij \in E\backslash\overline{E}} c_{ij} > 0$.

e) **Claim:** (u, v) is dual feasible in each step of the algorithm.
Proof. By definition of u and v, $(c_{ij} - u_i) - v_j \geq (c_{ij} - \min\limits_{j \in V} c_{ij}) - \min\limits_{i \in U}(c_{ij} - \min\limits_{j \in V} c_{ij}) \geq 0$ holds for each pair $ij \in U \times V$ immediately after initialization.
Primal update does not change u and v. If (u, v) is dual feasible, then (u', v') obtained after a dual update step remains feasible for all edges of each type: - $ij \in U(T) \times V(T) \Rightarrow c_{ij} - u'_i - v'_j = c_{ij} - (u_i + \varepsilon) - (v_j - \varepsilon) = c_{ij} - u_i - v_j \geq 0$ - $ij \in U(T)\times(V\backslash V(T)) \Rightarrow c_{ij} - u'_i - v'_j = c_{ij} - (u_i + \varepsilon) - v_j = \overline{c}_{ij} - \min\limits_{ij \in U(T)\times(V\backslash V(T))} c_{ij} - u_i - v_j \geq 0$ - $ij \in (U\backslash U(T)) \times V(T) \Rightarrow c_{ij} - u'_i - v'_j = c_{ij} - u_i - (v_j - \varepsilon) \geq c_{ij} - u_i - v_j \geq 0$ - $ij \in (U\backslash U(T)) \times (V\backslash V(T)) \Rightarrow c_{ij} - u'_i - v'_j = c_{ij} - u_i - v_j \geq 0$

f) **Claim:** If Algorithm 3.4.9 terminates in step 3, M is optimal.
Proof. $\overline{c}(M) = 0 \Leftrightarrow \sum\limits_{ij \in M} \overline{c}_{ij} = \sum\limits_{ij \in M} c_{ij} - u_i - v_j = 0$
$\Leftrightarrow \sum\limits_{ij \in M} c_{ij} = \sum\limits_{i \in U} u_i + \sum\limits_{j \in V} v_j \leq c(M')$ for an arbitrary perfect matching M'.

g) **Claim:** Algorithm 3.4.9 terminates finitely.
Proof. This follows from b) and d).

□

3.4.11 Proposition (Runtime of the Hungarian Method)
Algorithm 3.4.9 runs in $O(|V|^3)$ time.

Proof. Exercise. □

3.5 Lagrangian Relaxation

3.5.1 Remark (Compound Optimization Problem)
Consider (P) $\min c^T x, \quad \underbrace{Dx = d,}_{\text{complicated } (C)} \quad \underbrace{Ax = b, x \geq 0}_{\text{tractable } (T)}.$
How to reduce (P) $(C), (T)$ to (P) (T)? / How to get rid of (C)?

3.5.2 Definition (Lagrangian Relaxation)
Let $D \in \mathbb{R}^{m \times n}$, $\underbrace{X \subseteq \mathbb{R}^n}_{\text{e.g. } Ax \leq b, x \in \{0,1\}^n}$ closed and $(P) \min c^T x, Dx = d, x \in X$. Then

a) $(L_{Dx=d})$ $\max\limits_{\lambda \in \mathbb{R}^m} \min\limits_{x \in X} (c^T x - \lambda^T (Dx - d))$
is the Lagrangian relaxation of (P) with respect to $Dx = d$.

b) $f : \mathbb{R}^m \to \mathbb{R}, \lambda \mapsto f(\lambda) := \min\limits_{x \in X} (c^T x - \lambda^T (Dx - d))$
is the Lagrangian function of (P) with respect to $Dx = d$.

c) $f(\lambda) := \min\limits_{x \in X} (c^T x - \lambda^T (Dx - d))$ is the subproblem of $(L_{Dx=d})$.

3.5.3 Theorem (Elementary Properties of the Lagrangian Relaxation (Geoffrion))
Let $\nu(P) := \min c^T x, Dx = d, x \in X$ with $\nu(P) \in \mathbb{R} \cup \{+\infty\}$.

a) $\max f(\lambda) \leq \nu(P)$

b) Let $X = \{Ax = b, x \geq 0\}$ with $X \cap \{Dx = d\} \neq \emptyset$. Then $\max f(\lambda) = \nu(P)$.
In particular, feasible LPs and their Lagrange relaxations
have the same optimal objective values.

c) Let X be finite ($X = \{x_1, \ldots, x_k\} \neq \emptyset$) or a polytope ($X = \operatorname{conv}\{x_1, \ldots, x_k\} \neq \emptyset$).
Then f is
(i) concave,
(ii) piecewise affine,
(iii) bounded from above.

Proof. a) $f(\lambda) = \min\limits_{x \in X} (c^T x - \lambda^T (Dx - d)) \leq \min\limits_{\substack{x \in X \\ Dx = d}} (c^T x \underbrace{- \lambda^T (Dx - d)}_{=0}) = \nu(P)$

b) (P)
$$\begin{array}{llll} \min & c^T x & & \\ & Ax & = b & \to \mu \\ & Dx & = d & \to \lambda \\ & x & \geq 0 & \end{array}$$

$$\stackrel{\text{duality}}{=} \begin{array}{lllll} \max & \mu^T b & + \lambda^T d & & \\ & \mu^T A & + \lambda^T D & \leq c^T & \leftarrow x \end{array}$$

$$= (\max_{\lambda\in\mathbb{R}^m} \lambda^T d) + \begin{pmatrix} \max & \mu^T b & & \\ & \mu^T A & \le c^T - \lambda^T D & \to x \end{pmatrix}$$

$$\overset{\text{duality}}{=} (\max_{\lambda\in\mathbb{R}^m} \lambda^T d) + \begin{pmatrix} \min & (c^T - \lambda^T D)x & & \\ & Ax & = b & \leftarrow \mu \\ & x & \ge 0 & \end{pmatrix}$$

$$= \max_{\lambda\in\mathbb{R}^m} \left(\begin{pmatrix} \min & (c^T - \lambda^T D)x & \\ & Ax & = b \\ & x & \ge 0 \end{pmatrix} + \lambda^T d \right)$$

$$= \max_{\lambda\in\mathbb{R}^m} \min_{x\in X} (c^T x - \lambda^T (Dx - d)) = \max_{\lambda\in\mathbb{R}^m} f(\lambda)$$

c) $f(\lambda) = \min_{x\in X}(c^T x - \lambda^T(Dx-d)) = \min_{i=1}^{k} \underbrace{(c^T x_i - \lambda^T(Dx_i - d))}_{\text{affine in } \lambda} \le \nu(P) \underbrace{< \infty}_{X\cap\{Dx=d\}\neq\emptyset}$

□

3.5.4 Definition (Subgradient, Subdifferential)
Let $f : \mathbb{R}^m \to \mathbb{R}$ be concave and $\lambda_0 \in \mathbb{R}^m$.

a) Subgradient of f at λ_0: $u \in \mathbb{R}^m$ such that $\forall\, \lambda \in \mathbb{R}^m : f(\lambda) \le f(\lambda_0) + u^T(\lambda - \lambda_0)$.

b) Subdifferential of f at λ_0: $\delta f(\lambda_0) = \{u \in \mathbb{R}^m \ : \ u \text{ subgradient of } f \text{ at } \lambda_0\}$

3.5.5 Proposition (Differentiable Case)
Let $f : \mathbb{R}^m \to \mathbb{R}$ be concave and differentiable at $\lambda_0 \in \mathbb{R}^m$. Then $\delta f(\lambda_0) = \{f'(\lambda_0)\}$.

Proof. Exercise. □

3.5.6 Proposition (Polyhedral Case)
Let X be finite ($X = \{x_1, \ldots, x_k\} \neq \emptyset$) or a polytope ($X = \operatorname{conv}\{x_1, \ldots, x_k\} \neq \emptyset$
and f the Lagrangian function of $\min c^T x, Dx = d, x \in X$ with respect to $Dx = d$.
Then $\delta f(\lambda_0) = \operatorname{conv}\{-(Dx_i - d) \ : \ x_i \in \operatorname{argmin} f(\lambda_0)\}$ for all $\lambda_0 \in \mathbb{R}^m$.

Proof. Let $\lambda, \lambda_0 \in \mathbb{R}^m$, $x(\lambda_0) := \operatorname{argmin} f(\lambda_0)$ and $u_k := -(Dx_k - d)$ for $x_k \in x(\lambda_0)$.
$\supseteq$) Let $x_j \in x(\lambda_0)$, i.e. x_j minimizes $f(\lambda_0)$. Then
$f(\lambda_0) + u_j^T(\lambda - \lambda_0) = \min_{x\in X}(c^T x - \lambda_0^T(Dx - d)) - (Dx_j - d)^T(\lambda - \lambda_0)$
$= c^T x_j - \lambda_0^T(Dx_j - d) - (Dx_j - d)^T(\lambda - \lambda_0) = c^T x_j - \lambda^T(Dx_j - d)$
$\ge \min_{i=1}^{k} c^T x_i - \lambda^T(Dx_i - d) = f(\lambda)$
This also holds for any convex combinations of some u_k
because the corresponding convex combination of some x_k also minimizes $f(\lambda_0)$.
$\subseteq$) $\min_{x_i \notin x(\lambda_0)} c^T x_i - \lambda_0^T(Dx_i - d) - f(\lambda_0) > 0$
$\Rightarrow\ \exists\, \varepsilon > 0\ \forall\, \lambda \subseteq U_\varepsilon(\lambda_0) :\ x(\lambda) \subseteq x(\lambda_0)$.
Let $u \notin \operatorname{conv}\{u_i \ : \ x_i \in x(\lambda_0)\}$.
$\overset{\text{separating hyperplane theorem}}{\Rightarrow} \exists \pi \in \mathbb{R}^m : \forall x_i \in x(\lambda_0) : u^T\pi < u_i^T\pi$

$\Rightarrow f(\lambda) = f(\lambda_0 + \varepsilon\pi)$
$= \min\limits_{x_i \in x(\lambda_0)} (c^T x_i - (\lambda_0 + \varepsilon\pi)^T (Dx_i - d))$
$= \min\limits_{x_i \in x(\lambda_0)} (c^T x_i - \lambda_0^T (Dx_i - d) + \varepsilon\pi^T u_i)$
$= f(\lambda_0) + \min\limits_{x_i \in x(\lambda_0)} \varepsilon\pi^T u_i$
$> f(\lambda_0) + \varepsilon\pi^T u$
$= f(\lambda_0) + u^T(\lambda_0 + \varepsilon\pi - \lambda_0) = f(\lambda_0) + u^T(\lambda - \lambda_0)$ ↯ □

3.5.7 Algorithm (Subgradient Method)
Input: $f : \mathbb{R}^m \to \mathbb{R}$ concave, $\lambda_0 \in \mathbb{R}^m$ starting point, $(\alpha_k)_{k=1}^{\infty}$ with $\alpha_k \geq 0$ step lengths
Output: sequence $(\lambda_k)_{k=1}^{\infty}$ with $\lambda_k \in \mathbb{R}^m$

1. $k \leftarrow 0$
   $u_0 \leftarrow u \in \delta f(\lambda_0)$
2. $\lambda_{k+1} \leftarrow \lambda_k + \alpha_k u_k$
   $u_{k+1} \leftarrow u \in \delta f(\lambda_{k+1})$
3. Goto 2.

3.5.8 Theorem (Convergence of the Subgradient Method)
Let $f : \mathbb{R}^m \to \mathbb{R}$ be concave and

a) $f^* := \max f < \infty$

b) $\exists L \in \mathbb{R}\ \forall u \in \delta f : \|u\|_2 \leq L$

c) $\sum\limits_{k=1}^{\infty} \alpha_k^2 < \infty$ is convergent and $\sum\limits_{k=1}^{\infty} \alpha_k$ is divergent.

Then $\lim\limits_{k\to\infty} \max\limits_{j=0}^{k} f(\lambda_j) = f^*$.

Proof. Let $\lambda^* \in \operatorname{argmax} f$.
$\|\lambda_{k+1} - \lambda^*\|_2^2 = \|\lambda_k + \alpha_k u_k - \lambda^*\|_2^2 = \|\lambda_k - \lambda^* + \alpha_k u_k\|_2^2$
$= \|\lambda_k - \lambda^*\|_2^2 + 2\alpha_k \underbrace{u_k^T(\lambda_k - \lambda^*)}_{\leq f(\lambda_k) - f^*} + \alpha_k^2 \|u_k\|_2^2$

$\leq \|\lambda_0 - \lambda^*\|_2^2 - \sum\limits_{j=0}^{k} 2\alpha_k (f^* - f(\lambda_k)) + \sum\limits_{j=0}^{k} \alpha_j \underbrace{\|u_j\|_2^2}_{\leq L^2}$

$\Rightarrow \underbrace{\|\lambda_{k+1} - \lambda^*\|_2^2}_{\geq 0} + 2 \sum\limits_{j=0}^{k} \alpha_j \underbrace{(f^* - f(\lambda_j))}_{\geq f^* - \max_{j=0}^{k} f(\lambda_j)} \leq \|\lambda_0 - \lambda^*\|_2^2 + \sum\limits_{j=0}^{k} \alpha_j^2 L^2$

$\Rightarrow f^* - \max\limits_{j=0}^{k} f(\lambda_k) \leq (\underbrace{\|\lambda_0 - \lambda^*\|_2^2}_{=\text{const.}} + \underbrace{\sum_{j=0}^{k} \alpha_j^2 L^2}_{\leq \text{const.}}) \;/\; (\underbrace{2 \sum_{j=0}^{k} \alpha_j}_{\to\infty}) \underset{k\to\infty}{\to} 0$ □

3.5.9 Example (Capacitated Facility Location Problem (CFLP))

$$
\begin{array}{llllll}
(CFLP) & \min & \sum_{i\in I} f_i y_i & + \sum_{ij\in E} d_{ij}x_{ij} & & \\
(D) & & & \sum_{i\in I} x_{ij} & \geq 1 \quad \forall j\in J & \text{demand constraints} \\
(B) & & y_i & -x_{ij} & \geq 0 \quad \forall ij\in E & \text{variable upper bounds} \\
(C) & & s_i y_i & - \sum_{j\in J: ij\in E} w_j x_{ij} & \geq 0 \quad \forall i\in I & \text{capacity constraints} \\
(N_x) & 0 \leq & & x_{ij} & \leq 1 \quad \forall ij\in E & \text{non-negativity constraint for x} \\
(N_y) & 0 \leq & y_i & & \leq 1 \quad \forall i\in I & \text{non-negativity constraint for y} \\
(I) & & y_i & & \in \mathbb{Z} \quad \forall i\in I & \text{integrality constraint for y}
\end{array}
$$

The Lagrangian relaxation with respect to (C)

$$
\begin{array}{llllll}
(L_C) \max_{\lambda\in\mathbb{R}^{|I|}_{\geq 0}} & \min & \sum_{i\in I} f_i y_i & + \sum_{ij\in A} d_{ij}x_{ij} & - \sum_{i\in I} & \lambda_i(s_i y_i - \sum_{j\in J} w_j x_{ij}) \\
(D) & & & \sum_{i\in I} x_{ij} & \geq 1 \quad \forall j\in J & \text{demand constraints} \\
(B) & & y_i & -x_{ij} & \geq 0 \quad \forall ij\in A & \text{variable upper bounds} \\
(N_x) & 0 \leq & & x_{ij} & \leq 1 \quad \forall ij\in E & \text{non-negativity constraint for x} \\
(N_y) & 0 \leq & y_i & & \leq 1 \quad \forall i\in I & \text{non-negativity constraint for y} \\
(I) & & y_i & & \in \mathbb{Z} \quad \forall ij\in A & \text{integrality constraint for y}
\end{array}
$$

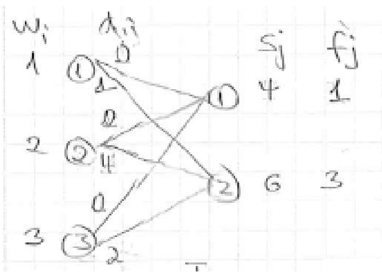
is equivalent to $(MUFL)$ with a modified objective.

(L_C) is feasible $\Leftrightarrow s_i y_i - \sum_{j\in J} w_j x_{ij} \geq 0 \Rightarrow \lambda_i = 0$.

(L_C) is infeasible $\Leftrightarrow s_i y_i - \sum_{j\in J} w_j x_{ij} < 0 \Rightarrow \lambda_i \to \infty$.

Let $I = \{1,2\}, J = \{1,2,3\}$,

$(s_i) = \begin{pmatrix} 4 \\ 6 \end{pmatrix}, (f_i) = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, (d_{ij}) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 4 & 2 \end{pmatrix}, (w_j) = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$



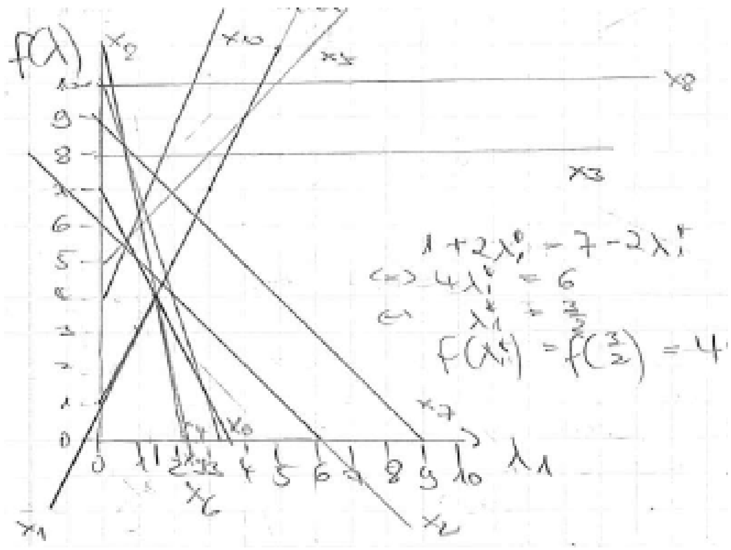
Optimal solution: $(x^*_{ij}) = \begin{pmatrix} 1 & 1 & \frac{1}{3} \\ 0 & 0 & \frac{2}{3} \end{pmatrix}$ with cost $\frac{16}{3}$

Optimal LP solution: $(x^*_{ij}) = \begin{pmatrix} \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}, (y^*_i) = \begin{pmatrix} 1 \\ \frac{2}{3} \end{pmatrix}$ with cost $\frac{9}{2}$.

Lagrangian function: $f(\lambda) = d^T x + f^T y - \lambda_1 C_1 - \lambda_2 C_2$
with $C_1 = s_1 y_1 - \sum_{j\in J} w_j x_{1j}$ and $C_2 = s_2 y_2 - \sum_{j\in J} w_j x_{2j}$

	$i=1$	$i=2$	$i=3$	$j=1$	$j=2$	d^Tx	$+f^Ty$	$-\lambda_1 C_1$	$-\lambda_2 C_2$	$=f(\lambda)$
x_1	1	1	1	1	0	0	+1	$-\lambda_1(-2)$	$-\lambda_2(0)$	$=1+2\lambda_1$
x_2	1	1	2	1	1	2	+4	$-\lambda_1(1)$	$-\lambda_2(3)$	$=6-\lambda_1-3\lambda_2$
x_3	1	2	1	1	1	4	+4	$-\lambda_1(0)$	$-\lambda_2(4)$	$=8-4\lambda_2$
x_4	1	2	2	1	1	6	+4	$-\lambda_1(3)$	$-\lambda_2(1)$	$=10-3\lambda_1-\lambda_2$
x_5	2	1	1	1	1	1	+4	$-\lambda_1(-1)$	$-\lambda_2(5)$	$=5+\lambda_1-5\lambda_2$
x_6	2	1	2	1	1	3	+4	$-\lambda_1(2)$	$-\lambda_2(2)$	$=7-2\lambda_1-2\lambda_2$
x_7	2	2	1	1	1	5	+4	$-\lambda_1(1)$	$-\lambda_2(3)$	$=9-\lambda_1-3\lambda_2$
x_8	2	2	2	0	1	7	+3	$-\lambda_1(0)$	$-\lambda_2(0)$	$=10$
x_9	2	2	2	1	1	7	+4	$-\lambda_1(4)$	$-\lambda_2(0)$	$=11-4\lambda_1$
x_{10}	1	1	1	1	1	0	+4	$-\lambda_1(-2)$	$-\lambda_2(6)$	$=1+2\lambda_1-6\lambda_2$

$\forall\, x_k : C_2 \leq 0 \Rightarrow \max\limits_{\lambda \in \mathbb{R}^2_{\geq 0}} f(\lambda)$ has $\lambda_2 = 0$



Considering the functions $f(\lambda_1) := f(\lambda_1, 0)$ yields $\lambda_1^* = \frac{3}{2}$ with $f(\lambda_1^*) = f(\frac{3}{2}) = 4$.

3.5.10 Definition (Capacitated Facility Location Problem (CFLP) - extended)
Let I, J be finite sets, $f \in \mathbb{R}^I, s \in \mathbb{R}^I_{>0}, w \in \mathbb{R}^J_{>0}, d_{ij} \in \mathbb{R}^{I \times J}$.

$(CFLP)$	$\min \sum\limits_{i \in I} f_i y_i + \sum\limits_{ij \in E} d_{ij} x_{ij} =: Z$		
(D)	$\sum\limits_{i \in I\,:\,ij \in E} x_{ij} \geq 1$	$\forall j \in J$	demand constraints
(B)	$y_i - x_{ij} \geq 0$	$\forall ij \in E$	variable upper bounds
(C)	$s_i y_i - \sum\limits_{j \in J : ij \in E} w_j x_{ij} \geq 0$	$\forall i \in I$	capacity constraints
(T)	$\sum\limits_{i \in I} s_i y_i - \sum\limits_{j \in J} w_j \geq 0$		total demand constraint
(N_x)	$0 \leq x_{ij} \leq 1$	$\forall ij \in E$	non-negativity constraint for x
(N_y)	$0 \leq y_i \leq 1$	$\forall i \in I$	non-negativity constraint for y
(I)	$y_i \in \mathbb{Z}$	$\forall ij \in A$	integrality constraint for y

For $R, L \subseteq S := \{D, B, C, N, I, T\}, R \cap L = \emptyset, \overline{R \cup L} = S \setminus (R \cup L)$
let $Z_L^R := Z_L(\overline{R \cup L})$, i.e. (Z_L^R) arises from (Z) by dropping the constraints from R and a Lagrangian relaxation with respect to the constraints in L.
$N_x \in R, L; \qquad N_y \in R, L$
have no computational gain.
$D \in R \qquad C \in R$
would remove essential properties.
The LP relaxation and its Lagrangian relaxations have the same optimal values, i.e. $I \in R \Rightarrow \forall X \in \{D, B, C, N, T\} : Z_X^R = Z^R$.
We will consider all other possible relaxations:
$B \in R, L, \overline{R \cup L}; \qquad T \in R, L, \overline{R \cup L}; \qquad D \in L, \overline{R \cup L}; \qquad C \in L, \overline{R \cup L}$
yield $3 \cdot 3 \cdot 2 \cdot 2 = 36$ relaxations.
$I \in R; \qquad B \in R, \overline{R \cup L}; \qquad T \in R, \overline{R \cup L}$
yield $2 \cdot 2 = 4$ relaxations.

3.5.11 Observation
Let $S_i := (A_i x \leq b_i), i = 1, \ldots, k$ be inequality systems,
$P(S_1 \ldots S_k) := \{A_1 x \leq b_1, \ldots, A_k x \leq b_k\}$ and
$\text{conv}(S_1 \ldots S_k I) := \text{conv}\{x \in \mathbb{Z}, A_1 x \leq b_1, \ldots, A_k x \leq b_k\}$. Then

a) $P(S_1 \ldots S_k) = \bigcap_{i=1}^{k} P(S_i)$

b) $\text{conv}(S_1 S_2 I) \subseteq P(S_1) \cap \text{conv}(S_2 I) \subseteq P(S_1 S_2)$

Proof. a) by definition. b) is also clear. □

3.5.12 Theorem (Lagrangian Relaxations of the CFLP (Sridharan et al. [1991]) The following equalities and inequalities between Lagrangian relaxations hold:

a) $Z^{BI} \leq Z^I \leq Z_C^T \leq Z_C \leq Z$

b) $Z^I \leq Z_D \leq Z_C$

c) $Z^{BI} \leq Z_C^B \leq Z_D$

d) $Z = Z_B = Z^B = Z_T = Z^T = Z_{TB} = Z_T^B = Z_B^T = Z^{TB}$

e) $Z_D = Z_{DC} = Z_{BD} = Z_{BC} = Z_{BDC} = Z_D^B$

f) $Z_C^B = Z_{DC}^B$

g) $Z_C^T = Z_{TC}$

h) $Z^I = Z^{TI} = Z_{TD} = Z_D^T = Z_{TDC} = Z_{DC}^T = Z_{BTC} = Z_{BC}^T$
$= Z_{BTDC} = Z_{BDC}^T = Z_{BTC} = Z_{BD}^T = Z_{TD}^B = Z_D^{BT}$

i) $Z^{BI} = Z^{TBI} = Z_{TC}^B = Z_C^{TB} = Z_{TDC}^B = Z_{DC}^{TB}$

Proof.

a) $Z^{BI} \leq Z^I \overset{\text{h)}}{=} Z^{TI} = Z_C^{TI} \leq Z_C^T \leq Z_C \leq Z$

b) $Z^I = Z_D^I \leq Z_D \overset{\text{e)}}{=} Z_{DC} \leq Z_C$

c) $Z^{BI} = Z_C^{BI} \leq Z_C^B \overset{\text{f)}}{=} Z_{DC}^B \leq Z_D$

d) **Claim:** $\text{conv}(NDCI) = \text{conv}(TBNDCI)$
Proof. $(C) \wedge (I) \Rightarrow (B)$

$$(D) \wedge (C) \Rightarrow (T) : \sum_{i \in I} s_i y_i \overset{\sum_{i \in I}(C)}{\geq} \sum_{ij \in E} w_j x_{ij} \geq \sum_{j \in J} (w_j \sum_{i \in I \,:\, ij \in E} x_{ij}) \overset{(D)}{\geq} \sum_{j \in J} w_j$$

e) (i) $Z_D = Z_{BDC} = Z_{DC} = Z_{BD} = Z_{BC}$:
The first equation $Z_D = Z_{BDC}$ implies the other three equations.
$Z_D = \min d^T x + f^T y, (x, y) \in P(D) \cap \text{conv}(BCINT)$
$Z_{BDC} = \min d^T x + f^T y, (x, y) \in \underbrace{P(DBC)}_{\overset{3.5.11\ \text{a)}}{=} P(D) \cap P(BC)} \cap \text{conv}(INT)$

Claim: $P(BC) \cap \text{conv}(INT) = \text{conv}(BCINT)$
Proof. ($\supseteq$) by Observation 3.5.11 b).
($\subseteq$): Let $(\overline{x}, \overline{y})$ be an extreme point of $P(BC) \cap \text{conv}(INT)$.
$(\overline{x}, \overline{y}) \in \{0, 1\}^{I \times J} \Rightarrow (\overline{x}, \overline{y}) \in \text{conv}(BCINT)$.
Otherwise $\overline{y} \in \underbrace{\text{conv}(INT)}_{\text{integer polytope}} \Rightarrow \overline{y} = \frac{1}{2} y_1 + \frac{1}{2} y_2; y_1, y_2 \in \text{conv}(INT), y_1 \neq y_2$.

Set $x_{ij}^k := \begin{cases} 0 & \overline{y}_i = 0 \\ \overline{x}_{ij} \frac{y_i^k}{\overline{y}_i} & \overline{y}_i \neq 0 \end{cases}$ for $k = 1, 2 \Rightarrow \overline{x} = \frac{1}{2} x_1 + \frac{1}{2} x_2$
$\Rightarrow (x_k, y_k) \in P(BC) \cap \text{conv}(INT)$ for $k = 1, 2 \Rightarrow (\overline{x}, \overline{y})$ not extreme ↯

(ii) $Z_D = Z_D^B$:
Claim: $\text{conv}(BCINT) \cap P(D) = \text{conv}(CINT) \cap P(D)$
Proof. $(C) \wedge (I) \wedge (N) \Rightarrow (B)$

f) **Claim:** $P(C) \cap \text{conv}(DINT) = P(CD) \cap \text{conv}(INT)$
Proof. The claim is equivalent to $\text{conv}(DINT) = P(D) \cap \text{conv}(INT)$
$(DINT)$ separates into constraints for x (DN) and y (INT).
The convexification only appears to y.

g) **Claim:** $P(C) \cap \text{conv}(DNBI) = P(TC) \cap \text{conv}(DNBI)$
Proof. The claim is equivalent to $\text{conv}(DNBI) = P(T) \cap \text{conv}(DNBI)$.
This follows from $(C), (D) \Rightarrow (T)$.

h) **Claim:** $Z^I = Z^{TI}$

Proof. $(C) \wedge (D) \Rightarrow (T)$

Claim: $Z_D^{BT} = Z_{TD}^B = Z_T^D$

Proof.

$$P(D) \cap \text{conv}(CNI) \underbrace{=}_{(D)\wedge(C)\Rightarrow(T)} P(TD) \cap \text{conv}(CNI) \underbrace{=}_{(C)\wedge(N)\wedge(I)\Rightarrow(B)} P(TD) \cap \text{conv}(CNIB)$$

Claim: $Z_{TD} = Z^I$

Proof. $P(TD) \cap \text{conv}(BCNI) = P(TDBCN) = P(TD) \cap P(BCN)$
$\Rightarrow \text{conv}(BCNI) = P(BCN)$.
As in e)(i), but without T, we obtain
$\text{conv}(BCNI) = P(BC) \cap \text{conv}(NI) = P(BC) \cap P(N) = P(BCN)$.

Claim: $Z_{BTC} = Z^I$

Proof. $P(BTC) \cap \text{conv}(DNI) = P(BTCDN)$
$\overset{\text{x,y separate}}{=} P(D) \cap \text{conv}(NI) = P(D) \cap P(N)$.

All other bounds are stronger than Z^I and weaker than Z_{TD} or Z_{BTC}.

i) **Claim:** $Z^{BI} = Z^{TBI}$

Proof. $(C) \wedge (D) \Rightarrow (T)$

Claim: $Z^{BI} = Z_{TC}^B$

Proof. $P(TCDN) = P(TC) \cap \text{conv}(DNI) = P(TC) \cap P(DN)$

The same argument yields the other equalities.

□