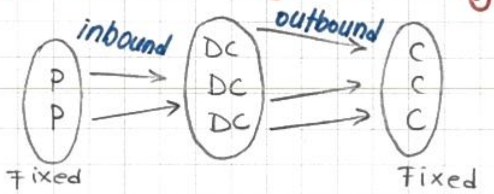
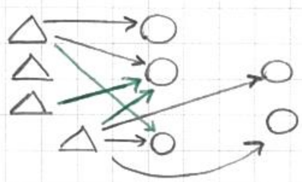
28.09.2016 - Çarşamba

Single Echelon (Layer) Single Commodity Location Models (SESC)



- If only one part of the journey important and the other part of the transportation is negligible it is single echelon.



Single Allocation
Multi Allocation

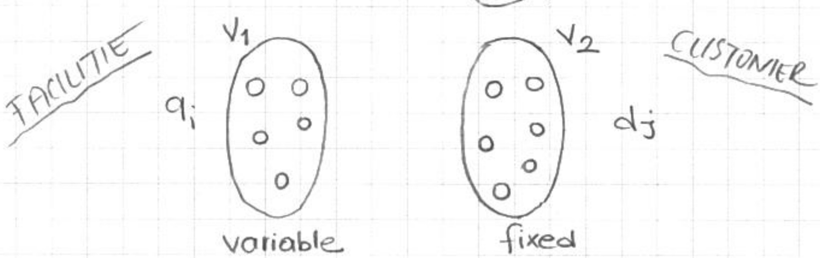
Assumptions:

- Homogenous facilities
- Single commodity
- Single echelon (either inbound or outbound traffic is negligible)
- Transportation cost is linear or piecewise linear and concave.
- Facility operating cost is linear and concave (or constant)

General Model

Let V_1 : Potential facilities $\qquad$ q_i : capacity of potential facility i
$\quad\;\;$ V_2 : Customers $\qquad\qquad\quad$ d_j : demand of customer j

* Assume demand is divisible (Multi Allocation)



Variables:

S_{ij} = Amount of product sent from DC_i to customer j
u_i = Total operations at potential facility i.

Define:

$C_{ij}(S_{ij})$ = Cost of transporting S_{ij} units from i to j.
$F_i(u_i)$ = Cost of operating facility i at level u_i.

MODEL:

$$\min \sum_i \sum_j C_{ij}(S_{ij}) + \sum F_i(u_i)$$

s.t

$$\sum_j S_{ij} = u_i \qquad \forall i \in V_1$$

u_i (induced variable)

$$\sum_i S_{ij} = d_j \qquad \forall j \in V_2$$

$$u_i \leq q_i \qquad \forall i \in V_1$$

$$u_i \geq 0, \quad S_{ij} \geq 0 \qquad \forall i \in V_1, \ \forall j \in V_2$$

→ Where to locate facilities? At i where $u_i > 0$

Special Cases:

1. $C_{ij}(\cdot)$ linear function
   $F_i(\cdot)$ fixed + linear

2. $C_{ij}(\cdot)$ linear function
   $F_i(\cdot)$ piecewise linear and concave

SESC Special Cases:

"$S_{ij} = d_j \cdot X_{ij}$"

(1.) Let $C_{ij}(S_{ij}) = \bar{C}_{ij} \cdot S_{ij} \qquad \forall i \in V_1, \ \forall j \in V_2$

↳ unit transportation cost over link (ij)

$$F_i(u_i) = \begin{cases} f_i + g_i \cdot u_i & \text{if } u_i > 0, \ i \in V_1 \\ 0 & \text{if } u_i = 0 \end{cases}$$

- f_i : fixed cost
- g_i : marginal cost

To represent $u_i > 0$, define $y_i \in \{0,1\}$ as location variable.

We may define X_{ij} : Fraction of demand d_j satisfy from i.
(bec. demand is divisible)

$$F_i(u_i) = f_i \cdot y_i + g_i \cdot u_i$$

→ usually negligible

$$\min \sum_i \sum_j \bar{c}_{ij} \cdot d_j \cdot x_{ij} + \sum_i f_i \cdot y_i + \sum_i g_i \left(\sum_j d_j \cdot x_{ij}\right)$$

s.t

$$\sum_i x_{ij} = 1. \qquad \forall j \in V_2$$

$$\sum_j d_j \cdot x_{ij} \leq q_i \cdot y_i \qquad \forall i \in V_1$$

$$0 \leq x_{ij} \leq 1 \qquad \forall i \in V_1, \ \forall j \in V_2$$

$$y_i \in \{0,1\} \quad \forall i \in V_1$$

$\Rightarrow$ If g_i is negligible

$$\min \sum_i \sum_j \underbrace{\bar{c}_{ij} \cdot d_j}_{c_{ij}} \cdot x_{ij} + \sum_i f_i \cdot y_i .$$

st.

$$\sum_i x_{ij} = 1 \qquad \forall j \in V_2$$

$$\sum_j d_j \cdot x_{ij} \leq q_i \cdot y_i \qquad \forall i \in V_1$$

$$0 \leq x_{ij} \leq 1 \qquad \forall i \in V_1, \ \forall j \in V_2$$

$$y_i \in \{0,1\} \quad \forall i \in V_1$$

"Capacitated Plant Location Problem" (CPL)

Variations:

If $q_i = \infty \quad \forall i$

$$\min \sum \sum c_{ij} \cdot x_{ij} + \sum_i f_i \cdot y_i$$

st.

$$\sum_i x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \leq y_i \qquad \forall i \in V_1$$

$$0 \leq x_{ij} \leq 1 \qquad \forall i \in V_1, \ \forall j \in V_2$$

$$y_i \in \{0,1\} \quad \forall i \in V_1$$

"Single Plant Location Problem" (SPL)

3 Eylül, Pazartesi

Variations:

$q_i = \infty$ and $f_i = f$ $\forall i \in V_1$ and want to open p facilities

Recall CPL

$$\min \sum_i \sum_j c_{ij} \cdot x_{ij} + \sum_i f_i \cdot y_i$$

$$\text{s.t.} \quad \sum_i x_{ij} = 1 \quad \forall j \in V_2$$

$$\sum_j d_j \cdot x_{ij} \leq q_i \cdot y_i \quad \forall i \in V_1$$

$$0 \leq x_{ij} \leq 1, \quad y_i \in \{0,1\} \quad \forall i,j$$

$q_i = \infty$, $f_i = f$
want to open p facilities

$$\min \sum_i \sum_j c_{ij} \cdot x_{ij} + \sum_i f \cdot y_i$$

$f \cdot p$

$$\text{s.t} \quad \sum_i x_{ij} = 1 \quad \forall j \in V_2$$

$$x_{ij} \leq y_i \quad \forall i,j$$

$\sum_i f \cdot y_i = f \cdot \sum_i y_i = f \cdot p$

$$\sum_{i \in V_1} y_i = p$$

$$0 \leq x_{ij} \leq 1 \quad \forall i,j$$

$$y_i \in \{0,1\}$$

$\equiv$

$$\min \sum_i \sum_j c_{ij} \cdot x_{ij}$$

$$\text{s.t.} \quad \sum_i x_{ij} = 1 \quad \forall j$$

$$x_{ij} \leq y_i \quad \forall i,j$$

(*) in addition to SPL: $\sum_{i \in V_1} y_i = p$

$$0 \leq x_{ij} \leq 1$$

$$y_i \in \{0,1\}$$

p-median problem

How to solve?

→ If possible; enumeration.

Say $|V_1| = n$, evaluate $\binom{n}{p}$ locations.

→ Integer programming techniques
- Use MIP solvers like CPLEX, Gurobi ?
- Lagrangean Relaxation

→ Heuristics
- Greedy
- Exchange (Teitz & Bond 1968)
- Neighbourhood search
- Genetic, Tabus

Heuristic

An algorithm that can find good solution to a decision problem but will not guarantee finding the optimal solution.

Heuristics

Construction

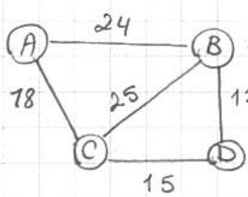
Neighborhood search { Improvement (like Simpl
Meta-heuristics

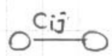
is defined based on the problem.

* Greedy Heuristic

- Applicable to decision problems of selecting a subset of things (for p-median facilities to open) that will optimize some objective.
- Starts with evaluating each site individually and selecting the one facility that yields the best objective function value.
  (The 1 facility with which $\sum\sum c_{ij}^{*} \cdot x_{ij}^{*}$ for j^{*} is minimum)
  That facility site is open.
- Evaluate the remaining alternatives and select the one which gives the best improvement in the objective function.
- Continue until open facilities is p.

Ex:





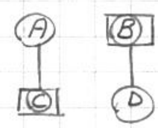
Say $V_1 = V_2$

1-median is at C

$A \longrightarrow 0+24+18+33 = 75$
$B \longrightarrow 24+0+25+12 = 61$
$C \longrightarrow 18+25+0+15 = 58$ *
$D \longrightarrow 33+12+15+0 = 60$

2-median

C & A	24+15 = 39	
✓ C & B	18+12 = 30	} same. Say locate C&B
✓ C & D	12+18 = 30	

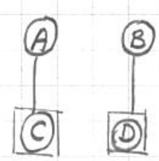


* Improvement Heuristics

- Begin with a good (or at least feasible) solution and seek to improve it. For p-median the most widely used is (still) $\boxed{\text{Teitz \& Bard}}$ 1968.
- Basic idea is to move a facility to an unused site. Each unused site is tri and when a move produces a better objective function value, then the relocation is accepted and we have a new improved solution.
- The process is repeated on the new solution
- The procedure stops one no improvement.

The improvement heuristics stucks at local optimum solution.

Ex: Apply T&B to previous exp. with Greedy stopped with 30 at C&D



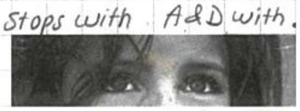
→ Try moving C to A ?

Cost is $0+12+15 = 27 < 30$ better
New Solution is A&D.

* Still not sure tha it's optimal
stuck at local opt

→ Try moving
$A \to B \Rightarrow B\&D = 35$ ✗
$A \to C \Rightarrow C\&D = 30$ ✗
$D \to B \Rightarrow A\&B = 30$ ✗
$D \to C \Rightarrow A\&C = 39$ ✗

Teitz & Bard Stops with A&D with.



Try the same B&C

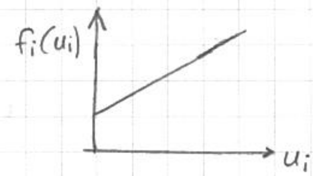
Special Cases recall:

(1) $C_{ij}(\cdot)$ linear function
$F_i(\cdot)$ fixed + $\underbrace{\text{linear}}_{\text{negligible}}$ → CPL / SPL / p-median

⇒ When not negligible?

If facility operating costs also depend on activity level.
→ transforms to previous models.

$$F_i(\cdot) = \begin{cases} f_i + g_i \cdot u_i & \text{if } u_i > 0 \\ 0 & \text{o.w} \end{cases}$$



How to convert?
Lets do for CPL

Recall CPL

$$\min \sum_i \sum_j C_{ij} \cdot x_{ij} + \sum_i f_i \cdot y_i$$

st.

$$\sum_i x_{ij} = 1 \quad \forall j$$

$$\sum_j d_j \cdot x_{ij} \leq q_i \cdot y_i \quad \forall i$$

$$x_{ij} \geq 0 \quad y_i \in \{0,1\}$$

Recall → $u_i = \sum_j d_j \cdot x_{ij}$

So $f_i(u_i) = f_i \cdot y_i + g_i \cdot \left(\sum_j d_j \cdot x_{ij}\right)$

$$\min \sum_i \sum_j C_{ij} \cdot x_{ij} + \sum_i f_i \cdot y_i + \sum_i g_i \cdot \left(\sum_j d_j \cdot x_{ij}\right)$$

$$\equiv \min \sum_i \sum_j C_{ij} \cdot x_{ij} + \sum_i f_i \cdot y_i + \sum_i \sum_j g_i \cdot d_j \cdot x_{ij}$$

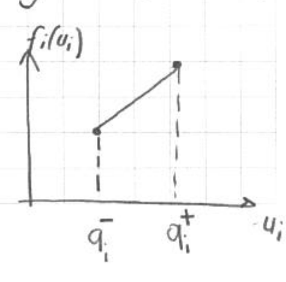
$$\equiv \min \sum_i \sum_j \underbrace{(C_{ij} + g_i \cdot d_j)}_{t_{ij}} x_{ij} + \sum_i f_i \cdot y_i \equiv \min \sum_i \sum_j t_{ij} \cdot x_{ij} + \sum_i f_i \cdot y_i$$

Variation

If $f_i(u_i)$ is like previous plus

- A facility can only be economically feasible for activity levels between q_i^- and q_i^+

i.e $f_i(u_i) = \begin{cases} f_i + g_i \cdot u_i & \text{if } q_i^- \leq u_i \leq q_i^+ \\ 0 & \text{ow} \end{cases}$



Model

$$\min \sum_i \sum_j t_{ij} \cdot x_{ij} + \sum_i f_i \cdot y_i$$

st.

$$\sum_i x_{ij} = 1 \quad \forall j \in V_2$$

$$q_i^- \cdot y_i \leq \sum_j d_j \cdot x_{ij} \leq q_i^+ \cdot y_i \quad \forall i$$

$$1 \geq x_{ij} \geq 0 , \quad y_i \in \{0,1\} \quad \forall i,j$$

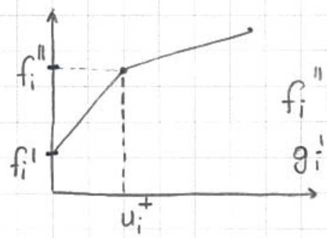
Special Cases ② Facility operating cost is piecewise and concave

- Facility operating costs exhibits economies of scale
- For different activity levels different configurations of fixed and variab costs are preferable

$\Rightarrow$ $F_i(u_i)$ is piecewise linear and concave.

Simplest case (2 pieces)

$$F_i(u_i) = \begin{cases} f_i'' + g_i'' \cdot u_i & \text{if} \quad u_i \geq u_i^+ \\ f_i' + g_i' \cdot u_i & \text{if} \quad 0 < u_i \leq u_i^+ \\ 0 & \text{if} \quad u_i = 0 \end{cases}$$



$f_i'' < f_i'$

$g_i' > g_i''$

* at u_i^+ $\quad f_i' + g_i' \cdot u_i^+ = f_i'' + g_i''$

➡ For each $i \in V_1$ define i' and i'' (dublicate each alternative facility)
Also include $y_i' + y_i'' \leq 1$. Due to the cost function, correct y_i will be 1

$\Rightarrow$ The first case with doubled locations. (Special case 1'in modeli)

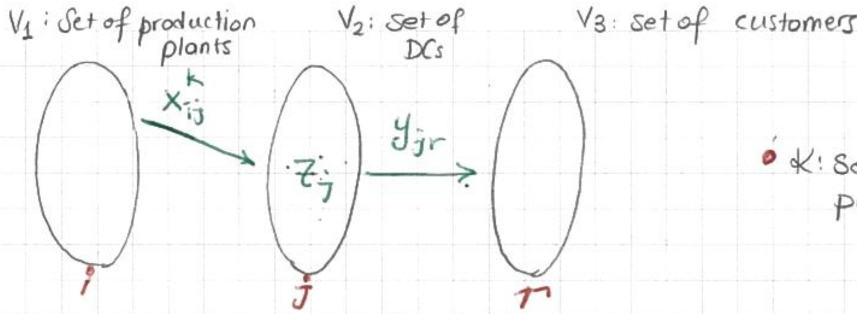
* Remark = If demand is not divisible; define x_{ij} as binary.

Two Echelon Multi Commodity Location Models (TEMC)

Assumptions:

- Homogenous facilities
- Both inbound and outbound is relevant.
- Nonhomogenous material flow
- Transportation costs are linear
- Facility operating costs consist of fixed and variable costs.
- Demand is not divisible.

05. 10. 2016
Çarşamba



- K: Set of different products

For each time unit;

- d_r^k : The quantity of product k required by customer r, $r \in U_3$, $k \in K$
- p_i^k : max quantity of product k, that plant i can supply (manuf.) $i \in U_1$, $k \in K$

✓ For each Potential DC$_i$, there is a min and max activity level q_i^- and q_i^+, $i \in V_2$

- f_j : fixed cost of opening a DC at j, $j \in V_2$
- g_j : marginal cost of DC$_j$.
- C_{ij}^k : Unit transportation cost of commodity k over link (i,j), $k \in K$, $(i,j) \in V_1 \times V_2 \cup V_2 \times V_3$

Let
- $z_j = \begin{cases} 1 & \text{if DC}_j \text{ is located at } j, \; j \in V_2 \\ 0 & \text{ow.} \end{cases}$

* Since demd. is not divs. it's binary. ow, fraction again

- $y_{jr} = \begin{cases} 1 & \text{if customer } r \text{ is served by DC at } j \\ 0 & \text{ow} \end{cases}$ $r \in V_3$, $j \in V_2$
- x_{ij}^k = Amount of commodity k supplied from plant i to DC$_j$, $i \in V_1$, $j \in V_2$

MODEL
(1)

$$\min \sum_k \sum_i \sum_j C_{ij}^k \cdot x_{ij}^k + \sum_k \sum_r \sum_j C_{jr} \cdot d_r^k \cdot y_{jr} + \sum_j z_j \cdot f_j + \sum_j g_j \left(\sum_r \sum_k d_r^k \cdot y_{jr} \right)$$

s.t

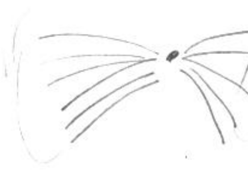
$$\sum_j y_{jr} = 1 \qquad \forall r \in V_3$$ * (Single Allocation)

$$z_j \cdot q_j^- \leq \sum_{r \in V_3} \sum_{k \in K} d_r^k \cdot y_{jr} \leq z_j \cdot q_j^+ \qquad \forall j \in V_2$$ * (Capacities of DCs)

(8)

$$\sum_j x_{ij}^k \leq p_i^k \qquad \forall k \;\; \forall i$$ * (Supply) -from plant-

$x_{ij}^k \le M z_j$, y_{jr} sağlatır.



$$\sum_i x_{ij}^k = \sum_{r \in V_3} d_r^k \cdot y_{jr} \qquad \forall j \in V_2,\ \forall k \in K \qquad \text{(balance)}$$

$$z_j \in \{0,1\} \quad \forall j \in V_3$$

$$y_{jr} \in \{0,1\} \quad \forall j \in V_2,\ r \in V_3$$

$$x_{ij}^k \geq 0 \quad \forall i \in V_1,\ j \in V_2,\ k \in K$$

• Remark: $x_{ij}^k \leq M \cdot z_j$ is valid but not necessary

(2) Another MIP

z_j & y_{jr} are the same

S_{ijr}^k = Quantity of item k transported from plant i to customer r via DC j, $k \in K$, $i \in V_1$, $j \in V_2$, $r \in V_3$

For simplicity

$$c_{ijr}^k = (C_{ij}^k + c_{jr}^k)$$

$$\min \sum_i \sum_j \sum_r \sum_k c_{ijr}^k \cdot S_{ijr}^k + \sum_j f_j z_j + \sum_j g_j \cdot \left(\sum_r \sum_k d_r^k \cdot y_{jr}\right)$$

s.t

$$\sum_j y_{jr} = 1 \qquad \forall r$$

$$q_j^- \cdot z_j \leq \sum_r \sum_k d_r^k \cdot y_{jr} \leq q_j^+ \cdot z_j \qquad \forall j \in V_2$$

Even if we define S_{ijr}, we have to include y_{jr}

$$\sum_j \sum_r S_{ijr}^k \leq p_i^k \qquad \forall i,\ \forall k$$

Remark: We still need y_{jr} variable in order to guarantee single allocation requirements of customer. Since demand is not divisible.

$$\sum_i S_{ijr}^k = d_r^k\, y_{jr} \qquad \forall j,\ \forall r,\ \forall k \implies$$ balance constraint i a little diffe

$$z_j \in \{0,1\} \quad \forall j$$

$$y_{jr} \in \{0,1\} \quad \forall j, r$$

$$S_{ijr}^k \geq 0 \quad \forall i \in V_1,\ j \in V_2,\ r \in V_3,\ k \in K$$

TEMC Model Variations

- Not all commodities can be produced at all plants. (p_i^k Assign as zero)
- Not all assignments for DCs to plants are feasible. (y_{jr} Assign related zero)
- Not all assignments of customers to DCs are feasible.
- Some potential DC's must be opened.
- Some demand must be served by a particular DC

Facility Location in the Public Sector

Public and private sector applications are different.

Ex. • For private sector = profit maximization, capture of market share from competit.
• For public sector = equality of service, efficiency (difficult to measure)

Ex. for public sector = fire stations, emergency station, dump sites, ...

* A major objective besides cost efficiency is to provide an adequate service to all potential clients.

→ Consider specific models;
- Center Models = the service time of the most disadvanteged client has to be minim
- Cover Models

1 → Covering Models:

In some location models, each client has to be reaced within a prespecified time from the closest facility. Eg. Ambulances should reach each call within 9 minutes. These kinds of apriori set value are called as "covering distance". A demand point is set to be covered by a facility if the facility is within the coverage distance from the demand point.

Formulations:

Let V_2 : set of clients
V_1 : set of potential sites that facilities can be built at.

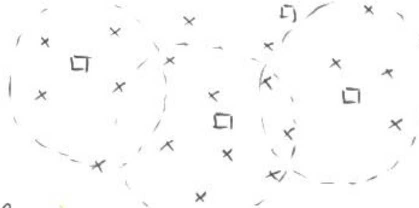
t_{ij} : travel distance betwen i & j.
T : threshold, covering distance

f_i : cost of building a facility at site i.

- Find the least cost facilities such that every client is covered.

Let $y_i = \begin{cases} 1, & \text{if a facility is opened at } i \in V_1 \\ 0, & \text{ow.} \end{cases}$

i host. j client



$t_{ij} \cdot y_j \leq$

* **Set Covering:** Each (min cost la) customer has to be covered.

$$\boxed{N_j = \{ i : t_{ij} \leq T \}}$$

$$\min \sum_i f_i \cdot y_i$$

s.t

$$\sum_{i \in N_j} y_i \geq 1 \qquad \forall j \in V_2$$

$(OR \;\; i : t_{ij} \leq T)$

$$\left(OR \;\; \sum_{i \in V_1} a_{ij} \cdot y_i \geq 1 \qquad \forall j \in V_2 \right) \longleftarrow a_{ij} = \begin{cases} 1 & \text{if } t_{ij} \leq T \\ 0 & \text{ow} \end{cases}$$

Different Variations:

→ **Maximal Covering Problem:** Say we have a budget and we want to maximiz the # of people covered.

Additional Parameters =
- h_j : # of people living in location j, $j \in V_2$
- C : total budget
- y_i : the same above
- $x_j : \begin{cases} 1 & \text{if customer node } j \text{ is covered} \\ 0 & \text{ow} \end{cases}$

$$\max \sum_j h_j \cdot x_j$$

st.

$$\sum_i f_i \cdot y_i \leq C$$

$$\sum_{i \in N_j} y_i \geq x_j$$

$$a_{ij} \cdot y_i \geq x_j$$

$$y_i, x_j \in \{0,1\} \quad i \in V_1, j \in V_2$$

→ **Penalty cost for not covering:** Say we have a penalty cost for uncover customers

p_j: penalty cost for not covering demand at j

$$z_j = \begin{cases} 1, & \text{if customer } j \text{ is not covered.} \\ 0, & \end{cases}$$

Remark: If p_j are very high, mo will set all $z_j = 0$, it tu to a set covering prob. ✓

$$\min \sum_i f_i \cdot y_i + \sum_j p_j \cdot z_j$$

st.

$$* \;\; z_j + \sum_{i \in N_j} y_i \geq 1 \qquad \forall j \in V_2$$

$$y_i, z_j \in \{0,1\}$$

10.10.2016
Pazartesi

→ Equal Fixed Costs:

If fixed costs are equal for each potential site, alternative solutions with an equal number of facilities can be compared
→ based on the total travel time.

ie. $f_i = f$ $\forall i$ We may want to minimize the total travel distance additionally.

Assume set covering requirements (each customer has to be covered)

y_i is the same

$x_{ij} = \begin{cases} 1 & \text{if customer } j \text{ is served by facility } i. \\ 0 & \text{ow.} \end{cases}$

parameters f, t_{ij}, a_{ij} (or N_j) ⊛ $A = [a_{ij}]$ $a_{ij} = \begin{cases} 1 & \text{if } t_{ij} \leq T \\ 0 & \text{ow} \end{cases}$

$$\min \sum_i f \cdot y_i + \sum_i \sum_j t_{ij} \cdot x_{ij}$$

s.t.

$$x_{ij} \leq y_i \qquad \forall i, j$$

$$\sum_i a_{ij} \cdot x_{ij} \geq 1 \qquad \forall j$$

$$x_{ij}, y_i \in \{0,1\} \qquad \forall i, j$$

* Solution Technique

Let use take the one with penalty cost (Transforms to set covering when p_j is very large)

$$\min \sum_i f_i \cdot y_i + \sum_j p_j \cdot z_j$$

st.

$$\sum_i a_{ij} \cdot y_i + z_j \geq 1 \qquad \forall j$$

$$y_i, z_j \in \{0,1\}$$

→ LP relaxation:

$$\min \sum_i f_i \cdot y_i + \sum_j p_j \cdot z_j$$

st.

$$(w_j) \quad \sum_i a_{ij} \cdot y_i + z_j \geq 1 \quad \forall j$$

$$y_i, z_j \geq 0$$

$$\max \sum_j w_j$$

st.

$$\sum_j a_{ij} \cdot w_j \leq f_i \quad \forall i \in V_1$$

$$0 \leq w_j \leq p_j \quad \forall j \in V_2$$

min cx
st $Ax \geq b$
$x \geq 0$

max wb
$wA \leq c$
$w \geq 0$

Find w^* such that

$$w_j^* \leq p_j, \ \sum_j w_j^* \text{ is maximized and } \sum_j a_{ij} \cdot w_j^* \leq f_i \quad \forall i$$

We will construct w^* recursively.

Greedy Approach

Given $w_1, w_2, \ldots, w_{k-1}$ such that

$$\sum_{j=1}^{k-1} a_{ij} \cdot w_j \leq f_i \quad \forall i \; ; \qquad w_j \leq P_j \quad \forall j = 1, \ldots, k-1$$

Find w_k

$$w_k = \min \left\{ P_k, \; \min_{\substack{i \in V_1 \\ \text{if } a_{ik}=1}} \left(f_i - \sum_{j=1}^{k-1} a_{ij} \cdot w_j \right) \right\}$$

For $k=1$

$$w_1 = \min \left\{ P_1, \; \min_{\substack{i \in V_1 \\ a_{i1}=1}} f_i \right\}$$

For $k=2$

$$w_2 = \min \left\{ P_2, \; \min_{\substack{i \in V_1 \\ a_{i2}=1}} (f_i - a_{i1} \cdot w_1) \right\}$$

$$\vdots$$

Each $w_j \leq P_j$

$\forall i \quad \sum_j a_{ij} w_j \leq P_j$. by construction

- Is the greedy approach optimal for the problem? Only for certain $A = [a_{ij}]$ matri

→ How to utilize w^*?
→ How to construct y^* and z^* from w^*?

Complementary Slackness

$\min \; cx$
st.
$Ax \geq b$
$x \geq 0$

$\max \; wb$
$wA \leq c$
$w \geq 0$

- CS conditions

$$w \cdot (Ax - b) = 0$$
$$(wA - c) \cdot x = 0$$

$$w_j \cdot \left(\sum_i a_{ij} \cdot y_i + z_j - 1 \right) = 0 \quad \forall j$$

$$\left(\sum_j a_{ij} w_j - f_i \right) \cdot y_i = 0 \quad \forall i$$

$$(w_j - P_j) \cdot z_j = 0 \quad \forall j$$

- Given w, utilize CS conditions to find Y &

Ex:

Say 6 possible locations with $f = [2\ 4\ 4\ 4\ 3\ 8]$.
7 customers $\quad p_j = 2 \quad \forall j \in [1 \dots 7]$

Let

$$
\begin{array}{c|ccccccc|c}
\text{Locat.}\backslash\text{Custm.} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & f_i \\
\hline
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 \\
2 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & \cancel{4}\ \cancel{2}\ 0 \\
3 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & \cancel{4}\ \cancel{2}\ 0 \\
4 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \cancel{4}\ \cancel{2}\ 1 \\
5 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \cancel{3}\ \cancel{1}\ 0 \\
6 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 8\ \cancel{6}\ \cancel{4}\ \cancel{2}\ 1 \\
\hline
w & 2 & 2 & 2 & 0 & 0 & 2 & 1 &
\end{array}
$$

Facility

$(f_i - \sum_j a_{ij} \cdot w_j)$

Solve $\quad \max \sum_j w_j$

st. $\quad \sum_{j=1}^{7} a_{ij} \cdot w_j \leq f_i \quad \forall i$

$0 \leq w_j \leq 2$

$w_5 \leq f_1 = 2$

$w_2 + w_3 + w_4 + w_5 \leq f_2 = 4$

$w_2 + w_3 + w_4 + w_5 \leq f_3 = 4$

$w_6 + w_7 \leq f_4 = 4$

$w_1 + w_7 \leq f_5 = 3$

$w_2 + w_3 + w_4 + w_5 + w_6 + w_7 \leq f_6 = 8$

- $\big(\underbrace{\sum_j a_{ij} \cdot w_j - f_i}_{\neq 0}\big) \cdot y_i = 0$

  $\Rightarrow y_1 = 0$
  $y_4 = 0$
  $y_6 = 0$

- $(w_j - p_j) \cdot z_j = 0$

  $\neq 0 \Rightarrow z_4 = 0$
  $z_5 = 0$
  $z_7 = 0$

$Y = [\ 0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0\ ]$
$Z = [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ ]$

- $\overset{\neq 0}{w_j} \cdot \left(\sum_i a_{ij} \cdot y_i + z_j - 1 \right) = 0$

- For $j = 1, 2, 3, 6, 7$ ($w_j \neq 0$ olonlar)

  $\sum_i a_{ij} \cdot Y_i + z_j - 1 = 0$

  $j=1 \quad y_5 + z_1 = 1 \Rightarrow y_5 = 1$ & $z_1 = 0$

  $y_2 + y_3 + y_6 + z_2 = 1 \longrightarrow y_2 + y_3 + z_2 = 1$

  $y_2 + y_3 + y_6 + z_3 = 1 \longrightarrow y_2 + y_3 + z_3 = 1$

  $y_4 + y_6 + z_6 = 1 \Longrightarrow y_4 = 0,\ y_6 = 0,\ z_6 = 1$

  $y_4 + y_5 + y_6 + z_7 = 1 \Longrightarrow y_5 + \underset{0}{z_7} = 1 \Rightarrow y_5 =$

if $y_2 = 1 \Rightarrow y_3 = 0,\ z_2 = z_3 = 0$ & $\text{cost} = 7 + 2 = 9$

if $y_3 = 1 \Rightarrow y_2 = 0,\ z_2 = z_3 = 0$ & $\text{cost} = 9$

They're feasible, we ca[n] choose both

if $y_2 = y_3 = 0$ $z_2 = 1,\ z_3 = 1$ $\text{cost} = \overset{\text{opening 5}}{3} + \overset{4+3+2}{3.2} = 9 \Longrightarrow$ Not feasible X

actually 4 & 5 are not covered, but here it looks as if they're covered. z_4 & $z_5 = 0$

Only 5 open

(Since its greedy, it is not optim[al])

② Center Problem:

p facilities have to be located such that the max travel time (distance etc) o[f] a client to the closest facility is minimized.

Eg. Fire stations : reach everyone as quick as possible.

Let V_1 = set of alternative locations
V_2 : set of customers

t_{ij} = time/distance between $i \in V_1$, $j \in V_2$

Define $y_i = \begin{cases} 1 & \text{if facility is located at } i \in V_1 \\ 0 & \text{ow} \end{cases}$

$x_{ij} = \begin{cases} 1 & \text{if client } j \text{ receives service from facility } i \\ 0 & \text{ow} \end{cases}$

$$\min \overbrace{\left(\max_{i,j} t_{ij} \cdot x_{ij}\right)}^{z} \qquad \Longrightarrow \qquad \min z$$

st
$$x_{ij} \le y_i \quad \forall j, \forall i$$
$$\sum_i x_{ij} = 1 \quad \forall j \in V_2$$
$$\sum_i y_i = p.$$
$$x_{ij}, y_i \in \{0,1\}$$

st.
$$* \ z \geq t_{ij} \cdot x_{ij} \quad \forall i, j$$
$$- \ x_{ij} \le y_i \quad \forall j, \forall i$$
$$\sum_i x_{ij} = 1$$
$$- \ \sum_i y_i = p$$
$$x_{ij}, y_i \in \{0,1\}$$

* Vertex restricted p-center probl.

• Why we need x_{ij}? instead of $z \geq t_{ij} \cdot y_i$

(*) Requires each customer to reach each facility

- Vertex restricted p-center problem: We have a set of alternative locations.
- Absolute p-center problem: Can locate facilities anywhere on the plane.
  Continuous location, Off from logistics network
  → If we can locate anywhere along the links, may make se[nse]

12 Ekim, Çarşamba

How to solve?

- If $p = 1$, Vertex restricted:

→ Among V_1 select the facility for which the longest distance to any customer in V_2 is mini

t_{ij} = Time of travel between i & j

$$\begin{array}{c|c} \text{Loc.} \backslash \text{Cust.} & 1 \; \cdots \; j \; \cdots \; m \\ \begin{array}{c} 1 \\ \vdots \\ i \\ \vdots \\ n \end{array} & \left[\begin{array}{ccc} \; & \; & \; \\ & & \\ & & \\ & & \\ & & \end{array} \right] \end{array}$$

$\gamma_i = \max_j t_{ij}$

Choose y^* such that $y^* = \min_i \gamma_i$

optimum 1-center location

- If we allow locations on edges

- Let $G = (V, E)$ V: set of vertices (customers, intersection points, ...)
  E: set of edges (physical roads)

Facilities can be located on any point along the edges (including the end points)

Let t_{ij} : Time to traverse link (i, j)

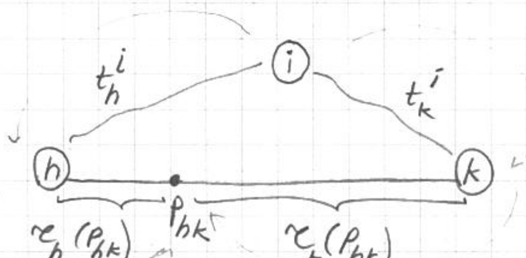
t_i^j : Shortest travel time between i & j (from i to j)

- By definition $t_i^j \leq t_{ij}$, if $(i,j) \notin E \Rightarrow t_{ij} = \infty$?

$\bar{T} = [t_i^j]$ (symmetric)

Let P_{hk} denote any point along link (h,k)

Let $\tau_h(P_{hk})$: travel time between $h \in V$ and P_{hk} on (h,k)



By definition

- $\tau_k(P_{hk}) = t_{hk} - \tau_h(P_{hk})$

1-Center algorithm (Hakimi 1969)

Step 1.: For each edge $(h,k) \in E$ and for each vertex $i \in V$, determine the travel time from i to a point P_{hk} along (h,k)

- $T_i(P_{hk}) = \min\{ t_i^h + \tau_h(P_{hk}),\ t_i^k + \tau_k(P_{hk}) \}$

Step 2: For each edge $(h,k) \in E$, determine the local center

P_{hk}^{*} : the point on (h,k) which minimizes the travel time from the most disadvantaged vertex.

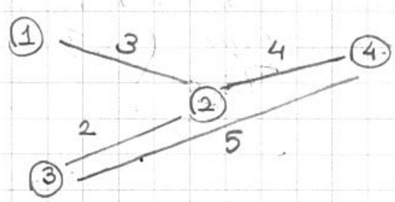
$$P_{hk}^{*} : \arg\min \max_{i \in V} T^{i}(P_{hk})$$

Step 3: 1-center location is $y^{*} = \min_{(h,k) \in E} P_{hk}^{*}$

For each edge (Step 1 & 2)
- calculate the distances to each node.
- find the worst node(s).
- find the location which minimizes the worst.

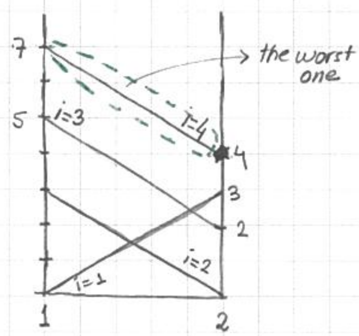
Select the best location among the edges. (Step 3)

Example =



$V = \{1,2,3,4\}$
$E = \{(1,2),(2,3),(2,4),(3,4)\}$

Edge (1,2)

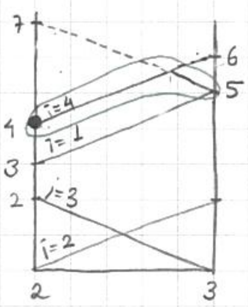


$i=3$ from 1 : $5 \to 8$
2 : $2 \to 5$

$i=4$ from 1 : $7 \to 10$
2 : $4 \to 7$ *

For edge (1,2) worst node is 4.
To minimize the distance to the worst locate at node 2.
Cost is 4.

Edge (2,3)

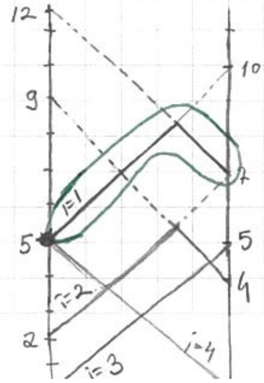


$i=1$ 2 : $3 \to 5$
3 : $5 \to 7$

$i=4$ 2 : $4 \to 6$
3 : $5 \to 7$

Worst node is 4. to minimize the distance to 4 locate at 2, cost is 4.

Edge (3,4)

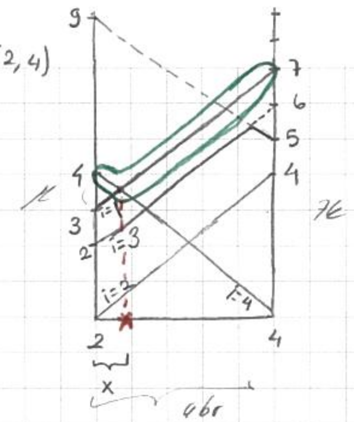


$i=1$: 3 : $5 \to 10$
4 : $7 \to 12$

$i=2$ 3 : $2 \to 7$
4 : $4 \to 9$

- Upper envelope is determined by node locate at node 3. Cost is 5.

Edge (2,4)



1: 2: 3 → 7
   4: 7 → 11 ×

3: 2: 2 → 6
   4: 5 → 9

3k = 4
k = 0.5

Upper envelope is determine by 2 nodes. Node 1 & 4
For the upper envelope best is at a point where

(*) the distance to 4 = distance to 1

$3 + x = 4 - x$

$x = 0.5$

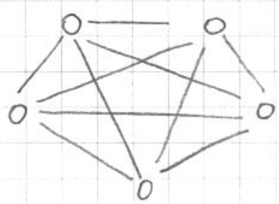
cost is 3.5

→ Optimum 1-center location is 0.5 from node 2.
Because it gives min. cost

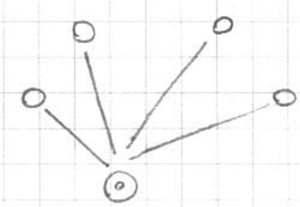
The Hub Location Problem

Hub location is a special location problem. Locating hubs in many to many distribution systems. Demand is not from a facility. It is between origin destination pair. Hubs act as switching, connection, consolidation centers.

Say you have 5 nodes.



$\binom{5}{2}$ connections. $\Longrightarrow$



Main Application areas: Airline
Cargo
Telecommunication

Why Hubbing?

→ Hubbing allows indirect connection between city pairs
- which cannot generate enough volume of traffic.
- which are too far away.

for direct connections.

→ Shorter non-stop segment
→ Smaller maintenence, processing, guardience costs due to centeralization.
→ Positive effect in marketing due to more frequent flights, better service e

→ Hubbing allows
consolidate & disseminate flows at certain locations called hubs.

→ Flows from the same origin with different destinations are consolidated on the route to the hub node and are combined with the flows with different origi but same destinations.

→ Replace direct flows with the indirect ones.

→ The hub location problem ←

Given: n demand centers
Known cross traffic

Problem: where to locate hub nodes?
how to allocate demand centers?

Allocation: Two variants: single
multi

- In single allocation all the incoming and outgoing traffic of every demand center is routed through a single hub.
- In multi allocation each demand center can receive/send flow through more than one hub.

* Performance measures:
- Cost: Minimize the total travel cost / travel cost + set up cost
- Time: Minimize the longest journey time (esp. for cargo delivery)
  the # of hubs by limiting the journey times.

* Economies of scale:

p-Hub median problem

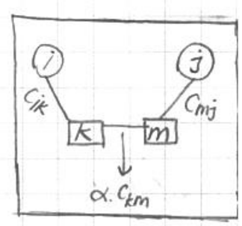
min total transportation cost

Let
- w_{ij} : flow between i & j
- c_{ij} : unit transportation cost between i & j

Let
- $x_{ij} = \begin{cases} 1 & \text{if } i \text{ is served by hub } j \\ 0 & \text{ow} \end{cases}$

Observe,

$$x_{jj} = \begin{cases} 1 & \text{if node } j \text{ is a hub} \\ 0 & \text{ow} \end{cases}$$



- Single Allocation:

economics of scale

$$\min \sum_i \sum_j w_{ij} \Big(\sum_k \sum_m (c_{ik} + \alpha . c_{km} + c_{mj}) \cdot \underbrace{x_{ik} \cdot x_{jm}}_{\text{quadratic}} \Big)$$

Nonlinear!

s.t.

$$\sum_k x_{ik} = 1 \quad \forall i$$

$$x_{ik} \leq x_{kk} \quad \forall i, k$$

$$\sum_k x_{kk} = p$$

$$x_{ik} \in \{0, 1\}$$

17.10.2016, Pazartesi

Linearize

Let $X_{ijkm} = \begin{cases} 1, & \text{if } x_{ik}=1,\ x_{jm}=1 \\ 0, & \text{ow} \end{cases}$ $\qquad X_{ijkm} = x_{ik} \cdot x_{jm}$

Claim:

$$\left.\begin{array}{l} \bullet\ x_{ijkm} \geq x_{ik} + x_{jm} - 1 \\ \bullet\ x_{ijkm} \leq \dfrac{x_{ik}+x_{jm}}{2} \end{array}\right\} \text{Correctly linearizes } x_{ijkm} = x_{ik} \cdot x_{jm}$$

$\rightarrow (x_{ijkm} \leq x_{ik}\ \&\ x_{ijkm} \leq x_{jm})$

Proof:

x_{ik}	x_{jm}	$x_{ijkm} \leq \frac{x_{ik}+x_{jm}}{2}$	$x_{ijkm} \geq x_{ik}+x_{jm}-1$		x_{ijk}
1	1	$x_{ijkm} \leq 1$	active $\{ x_{ijkm} \geq 1$	$\Rightarrow$	1
1	0	$x_{ijkm} \leq 0.5$ active	$x_{ijkm} \geq 0$	$\Rightarrow$	0
0	1	$x_{ijkm} \leq 0.5$ active	$x_{ijkm} \geq 0$	$\Rightarrow$	0
0	0	$x_{ijkm} \leq 0$ active	$x_{ijkm} \geq -1$	$\Rightarrow$	0

QED ✓

Linear Model

$$\min \sum_i \sum_j \sum_k \sum_m w_{ij} \cdot (C_{ik} + \alpha \cdot C_{km} + C_{mj}) \cdot x_{ijkm}$$

s.t

$$x_{ik} \leq x_{kk} \qquad \forall\, i,k$$

$$\sum_k x_{ik} = 1 \qquad \forall\, i$$

$$\sum_k x_{kk} = p$$

$$x_{ijkm} \leq \frac{x_{ik}+x_{jm}}{2} \qquad \forall\, i,j,k,m$$

$$x_{ijkm} \geq x_{ik} + x_{jm} - 1$$

$$x_{ik},\ x_{ijkm} \in \{0,1\}$$

Another IP

x_{ik} is the same

$x_{ijkm} = \begin{cases} 1 & \text{if } i \to k \to m \to j \\ 0 & \text{ow} \end{cases}$

$$\min \sum_i \sum_j \sum_k \sum_m w_{ij} \, (c_{ik} + \alpha . c_{km} + c_{mj}) . x_{ijkm}$$

s.t

$$\sum_k x_{ik} = 1 \quad \forall i$$

$$x_{ik} \leq x_{kk} \quad \forall ik$$

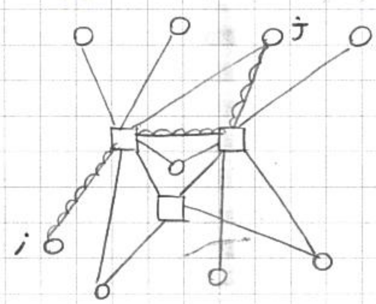
$$\sum_k x_{kk} = p$$

$$\sum_m x_{ijkm} = x_{ik} \quad \forall ikj$$

$$\sum_k x_{ijkm} = x_{jm} \quad \forall jmi$$

$$x_{ik}, x_{ijkm} \in \{0, 1\} \quad \forall ijkm$$

- Multiallocation



Define

- $y_k = \begin{cases} 1 & \text{if } k \text{ is a hub} \\ 0 & \text{ow} \end{cases}$
- No need for x_{ik}
- x_{ijkm} valid.

$$\min \sum_i \sum_j \sum_k \sum_m x_{ijkm} . w_{ij} . (c_{ik} + \alpha . c_{km} + c_{mj})$$

s.t

$$\sum_k y_k = p$$

$$\sum_m x_{ijkm} \leq y_k \quad \forall ijk$$

$$\sum_k x_{ijkm} \leq y_m \quad \forall ijm$$

$$\sum_k \sum_m x_{ijkm} = 1 \quad \forall ij$$

$$x_{ijkm}, y_k \in \{0,1\} \quad \forall ijkm$$

x_{ikkj} } 1 hub
kk cost : zero

p-Hub center problem

Find a set of hubs such that the maximum "cost" between any o-d pair is minimized.

Single allocation version

X_{ik}, X_{ijkm} same

min z
st.

$z \geq w_{ij} \cdot (C_{ik} + \alpha \cdot C_{km} + C_{mj}) \cdot X_{ijkm} \qquad \forall\, i\, j\, k\, m$

$\sum_k X_{ik} = 1 \qquad \forall i$

$X_{ik} \leq X_{kk} \qquad \forall ik$

$\sum_k X_{kk} = p$

$\sum_m X_{ijkm} = X_{ik} \qquad \forall\, ikj$

$\sum_k X_{ijkm} = X_{jm} \qquad \forall\, jmi$

X_{ik}, $X_{ijkm} \in \{0, 1\}$

Exercise $\rightarrow$ Multiallocation one!

Hub Covering

An o-d pair $(i\,j)$ is said to be covered by hubs k&m if the "cost" from i to j via hubs k&m does not exceed the predefined bound γ (γ_{ij} if you want to differentiate between o-d pairs)

If $C_{ik} + \alpha \cdot C_{km} + C_{mj} \leq \gamma_{ij}$ hubs k and m cover the o-d pair i&j.

Single Allocation

X_{ik}, X_{ijkm} the same

$X_{ik} = \begin{cases} 1 & \text{if } i \rightarrow k \\ 0 & \text{ow} \end{cases} \qquad X_{ijkm} = \begin{cases} 1 & \text{if } i \rightarrow k \;\; j \rightarrow m \\ 0 & \text{ow} \end{cases}$

Define parameter; $V_{ijkm} = \begin{cases} 1 & \text{if } c_{ik} + \alpha \cdot c_{km} + c_{mj} \leq \gamma_{ij} \\ 0 & \text{ow} \end{cases}$

$$\min \sum_k x_{kk}$$

st.

$$\sum_k x_{ik} = 1 \quad \forall i$$

$$x_{ik} \leq x_{kk} \quad \forall ik$$

$$\sum_k \sum_m V_{ijkm} \cdot x_{ijkm} \geq 1 \quad \forall i, j$$

$$\sum_k x_{ijkm} = x_{jm} \quad \forall ijm$$

$$\sum_m x_{ijkm} = x_{ik} \quad \forall ijk$$

$$x_{ik}, x_{ijkm} \in \{0,1\} \quad \forall ijkm$$

Exercise → Multiallocation one!

Freight Transportation

Transportation accounts for roughly a third of total logistics costs.

Players;

- shippers : originate the demand
- Carriers : supply the transportation services
- governments : provide transport infrastructure and regulate the indu

Long-Haul: Relatively long distances.

Short-Haul: Usually between pickup and delivery points in the same area (sh duration, usually finishes within a worksh

Long-Haul Transportation

- Privately operated transportation : from restricted number of origins to sever destinat
  e.g. Shell 9 → 575 (few to many transportation system)

Public carriers : (many-to-many)

- customized
  A vehichle serves a single request
- Consolidation based
  Routes, pick up and deliver for more than one customer.

Common Decision Problems

* Strategic Level: Definition of the operating strategy.
Design of the physical network
Mode choice
Acquisition of resources

* Tactical Level: Allocation of resources (vehicles, crew, ...)
Increase/decrease in capacity to cope with fluctuations in den

* Operational Level: Adjusting vehicle and crew schedules for last minute chang

* For privately operated transportation systems are relatively simple.
Determine the mix of owned and leased vehicles over the year.
The objective is usually minimize the cost while maintaining a service level.

* Consolidation based transportation system;
- Strategic level: Carrier should decide on what kind of commodities to tran and which o-d pair to serve. Number of terminals and locations should be determined.
- Tactical level: Determining the frequency and number of intermediate sh

* Customized transportation systems;
- Strategic level: Most suitable fleet.
- Tactical level: Pricing
- Operational level: Dynamic reallocation.

In Public Transport (both customized & Consolidation based case) objective is usually maximize profit.

Traffic Assignment

Determining the best "least cost" routing of goods over a transportation network. The demand allocation decisions in locational problems are traffi assignment problems.

Classification

* Static
- Decisions are not affected by time.
All parameters are known.

* Dynamic
- Parameters change by time.

19.10.2016, Çarşamba

Solving the Traffic assignment problem

Let $G=(V,A)$
- V: set of vertices (terminals, w/hs, plants)
  Origin, destination, transhipment points
- A: physical links
- K: set of commodities
- For each link, there is a cost.

For each commodity $k \in K$
Let
- $O(k)$: set of origins
- $D(k)$: set of destinations
- $T(k)$: set of transhipment cost

- O_i^k for $i \in O(k)$: supply of comd. k at node i
- d_i^k for $i \in D(k)$: demand of comd k at node i
- u_{ij}^k: maximum flow of comd k over $(i,j) \in A$
- u_{ij}: total capacity of arc $(i,j) \in A$
- $C_{ij}^k(a)$: cost function of traversing arc (i,j) with a unit k.

Let x_{ij}^k Amount of commodity k over link (i,j)

$$\min \sum_k \sum_i \sum_{j \in A} C_{ij}^k (x_{ij}^k)$$

st.
$$\underbrace{\sum_{j:(i,j)\in A} x_{ij}^k}_{(out)} - \underbrace{\sum_{j:(j,i)\in A} x_{ij}^k}_{(in)} = \begin{cases} O_i^k & \text{if } i \in O(k) \\ -d_i^k & \text{if } i \in D(k) \\ 0 & \text{ow} \end{cases} \qquad \forall i \in V,\ k \in K$$

$$x_{ij}^k \leq u_{ij}^k \qquad \forall k,\ \forall (i,j) \in A$$

$$\sum_k x_{ij}^k \leq u_{ij} \qquad \forall (i,j) \in A \qquad \text{(Aggregated Capt.)}$$

$$x_{ij}^k \geq 0 \qquad \forall k,\ \forall (i,j) \in A$$

Remark: $\sum_i O_i^k = \sum_i d_i^k$ is required, ow. problem is infeasible.

parameters

Special Variations:

→ Linear Cost, Single Commodity

K is single $(|K| = 1)$

$$C_{ij}^{k}(X_{ij}^{k}) = c_{ij}^{k} \cdot X_{ij}^{k}$$

*** P**

$$\min \sum_i \sum_j c_{ij} \cdot x_{ij}$$

$$\text{st.} \quad \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)} x_{ij} = \begin{cases} O_i & \text{if } i \in O \\ -d_i & \text{if } i \in D \\ 0 & \text{ow} \end{cases} \quad (\forall i) \quad \pi_i$$

• Network Flow Problem (Network simplex)

1 more constraints needed

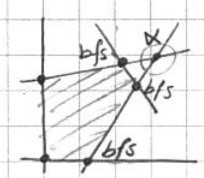
$$0 \leq x_{ij} \leq U_{ij} \quad \forall (i,j) \in A \rightarrow \lambda_{ij}$$

min için ters, o yüzden dual

$\begin{cases} \min cx \\ \text{st } Ax = b \quad A \text{ has full rank} \\ x \geq 0 \end{cases}$

$B.x_B + N.x_N = b$, $x_N = 0$

$x_B = B^{-1} \cdot b$



(dual)

$\max wb$

st.

$wA \leq c$

w free

$w = c_B \cdot B^{-1}$

$c_B \cdot B^{-1} A - c \leq 0$

→ node-arc incidence matrix

Property:

A basic solution have $|V| - 1$ basic variables (cardinality)

Each basic solution corresponds to a tree.

To find a basic solution construct a tree and find the required flo

If feasible, ($0 \leq$ all flows $\leq U$) then bfs is found.

Check the optimality

Is the tree constructed optimal? <u>Use duality.</u>

*** D**

$$\max \sum_i O_i \pi_i - \sum_i d_i \pi_i + \sum_{ij} \lambda_{ij} \cdot U_{ij}$$

st.

$$\pi_i - \pi_j + \lambda_{ij} \leq c_{ij}$$

$$\pi_i \text{ free}$$

$$\lambda_{ij} \leq 0$$

Utilize CS conditions to find the corresponding dual solution of the feasible tree on hand (x_{ij})

$$\pi_i \cdot \underbrace{(\text{Flow balance})}_{=0} = 0 \Rightarrow \text{Not useful.}$$

$$\lambda_{ij} \, (x_{ij} - U_{ij}) = 0$$

$$(\pi_i - \pi_j + \lambda_{ij} - c_{ij}) \cdot x_{ij} = 0$$

We have a feasible X. Find π and and check their feasibility.

Given x_{ij}

Use

$$\lambda_{ij}(x_{ij} - u_{ij}) = 0$$

$$(\pi_i - \pi_j + \lambda_{ij} - c_{ij}) \cdot x_{ij} = 0$$

- if $x_{ij} < u_{ij} \Rightarrow \lambda_{ij} = 0$
- if $x_{ij} > 0 \Rightarrow \pi_i - \pi_j + \lambda_{ij} = c_{ij}$

if $0 < x_{ij} < u_{ij} \Rightarrow \pi_i - \pi_j = c_{ij}$

For tree arcs $0 < x < u \Rightarrow \boxed{\pi_i - \pi_j = c_{ij}}$

For nonbasic arcs $x_{ij} = 0$, We check dual feasibility

$$\pi_i - \pi_j + \lambda_{ij} \leq c_{ij}$$

$$x_{ij} \leq 0$$

For nonbasics if $x_{ij} = 0 < u_{ij} \Rightarrow \lambda_{ij} = 0$

if $x_{ij} = u_{ij}$ $\quad \pi_i - \pi_j + \lambda_{ij} = c_{ij}$

$$\lambda_{ij} = -(\pi_i - \pi_j - c_{ij}) \leq 0$$

$$\boxed{\pi_i - \pi_j - c_{ij} \geq 0}$$

Summary:

Construct a feasible tree

For tree arcs, find π_i s using $\pi_i - \pi_j = c_{ij}$

For co-tree arcs

if $x_{ij} = 0$ check if $\pi_i - \pi_j \leq c_{ij}$

if $x_{ij} = u_{ij}$ check if $\pi_i - \pi_j \geq c_{ij}$

If all satisfied OPT.

else find a co-tree edge (h,k) violating optimality conditions.

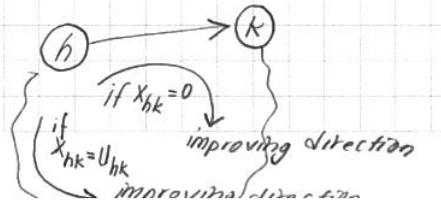
Arc (h,k) will become basic $\Rightarrow$ entering variable.

Leaving variable?

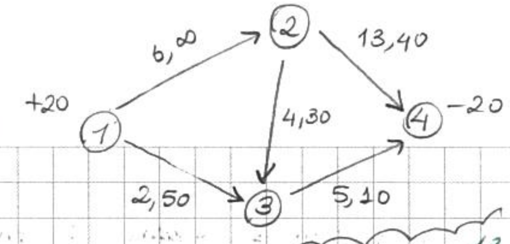
If $x_{hk} = 0 \Rightarrow$ we'll increase flow over (h,k)

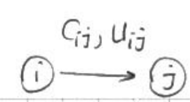
$x_{hk} = u_{hk} \Rightarrow$ we'll decrease flow over (h,k)

Arc (h,k) together with tree arcs will constitute a unique cycle. "Ratio test" will be performed for the arcs in that cycle based on their directions. If the arcs are in the improving direction the flow over them will increase, if they are in the reverse direction their flows will decrease. Minimum wi[ll] determine the leaving arc. The leaving arc can also be (h,k) if it change[s] its bound earlier than the rest of the [arcs] in the cycle hitting arc of the bound[s]



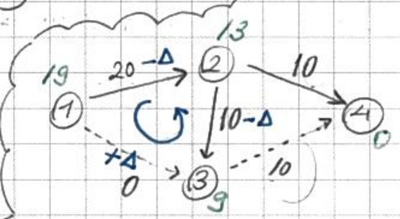
Example:





Initial tree arcs : (1,2)
(2,4)
(2,3)

Co-tree arc (3,4) is at its upper bound.



$\pi_i - \pi_j = c_{ij}$

$\pi_1 - \pi_2 = 6$ let $\pi_4 =$

$\pi_2 - \pi_4 = 13$

$\pi_2 - \pi_3 = 4$

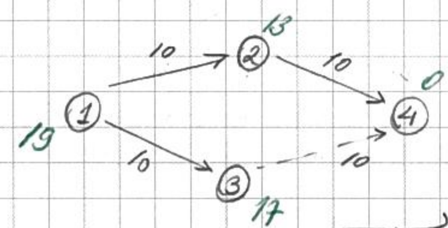
$\pi_i - \pi_j \leq c_{ij}$

Co-tree arcs : (1,3) $\pi_1 - \pi_3 \leq c_{13} \Rightarrow 19 - 9 = 10 \overset{\alpha}{\leq} 2 \Rightarrow$ enter (1,3)

at upper (3,4) $\pi_3 - \pi_4 \geq 5$ Violating.

$\Delta = \min\{50, 10, 20\} = 10$ (2,3) becomes non basic

New tree



$\pi_1 - \pi_2 = 6$ let $\pi_4 = 0$

$\pi_2 - \pi_4 = 13$

$\pi_1 - \pi_3 = 2$

(2,3) $\pi_2 - \pi_3 \leq c_{23}$ ✓ Thus it is optimum.

(3,4) $\pi_3 - \pi_4 \geq c_{34}$

24.10.2016, Pazartesi

Finding an initial feasible tree

- Add a dummy node with 0 demand and supply
- Add artificial arcs from each $i \in V$ to dummy node if $i \in O$ arc $i \rightarrow$ dummy

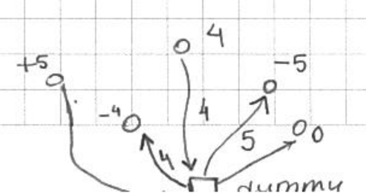
  if $i \in D$ arc $i \leftarrow$ dummy

  assign $c_{ij} = 0 \quad \forall (i,j) \in A$

  $c_{ij} = 1$ for artificial arcs

  The capacity of artificial arcs $\rightarrow \infty$
- Apply network simplex with the starting bfs as the artificial arcs.
  (If there are nodes $i \in V$ where $i \notin O$ and $i \notin D$, i is transhipment node, assign an artificial arc between this node and dummy with 0 flow over it, the direction of this arc is not important.)

Once all artificial arcs are out of the tree we have a feasible tree with original arcs. Put back to original costs and continue with network simplex.
If we stop with some artificial arcs having positive flow, then the original flow is infeasible.



Fixed Charge Network Design Models

In traffic assignment no fixed cost for arcs.
Including the fixed costs $\longrightarrow$ Fixed charge Network design models.

- which arcs to be employed
- how to transport the commodities

Some parameters and variables o_i^k, d_i^k, c_{ij}^k

x_{ij}^k

Additional parameters:

f_{ij} : fixed cost of using arc (i,j)

y_{ij} : $\begin{cases} 1 & \text{if arc } (i,j) \text{ is used} \\ 0 & \text{ow} \end{cases}$

$$\min\left(\sum_k \sum_i \sum_j c_{ij}^k \cdot x_{ij}^k + \sum_{(i,j)\in A} f_{ij} \cdot y_{ij} \right)$$

s.t.

$$\sum_{j:(i,j)\in A} x_{ij}^k - \sum_{j:(j,i)\in A} x_{ij}^k = \begin{cases} o_i^k & \text{if } i \in O(k) \\ -d_i^k & \text{if } i \in D(k) \\ 0 & \text{ow} \end{cases} \qquad \forall i \in V,\ k \in K$$

$$x_{ij}^k \leqslant u_{ij}^k \qquad \forall ij \quad \forall k$$

(açmazsan kullanamazsın) $$\sum_k x_{ij}^k \leqslant u_{ij} \cdot y_{ij} \qquad \forall ij$$

$$x_{ij}^k \geqslant 0, \quad y_{ij}^k \in \{0,1\} \qquad \forall k \in K,\ \forall ij \in A$$

The problem is very hard. Lagrangean relaxation is helpful (Out of scope of this

Lower bounds & Heuristics

- Weak continuous Relaxation

$\Rightarrow$ Relax the integrality requirements.

$$\min\left(\sum_k \sum_{(i,j)\in A} c_{ij}^k \cdot x_{ij}^k + \sum_{(i,j)\in A} f_{ij} \cdot y_{ij} \right)$$

s.t.

$$\sum_{j:(i,j)\in A} x_{ij}^k - \sum_{j:(j,i)\in A} x_{ij}^k = \begin{cases} o_i^k & \text{if } i \in O(k) \\ -d_i^k & \text{if } i \in D(k) \\ 0 & \text{ow} \end{cases}$$

$$x_{ij}^k \leqslant u_{ij}^k$$

$$\sum x_{ij}^k \leqslant u_{ij} \cdot y_{ij}$$

$$x_{ij}^k \geqslant 0 \ ; \quad \underbrace{0 \leqslant y_{ij}^k \leqslant 1}_{*} \qquad \forall i,j \ \forall k$$

* Since $f_{ij} > 0$ and relatively large
if $y_{ij} = 1$ utilize the link fully $\sum_k x_{ij}^k = u_{ij} \cdot y_{ij}$; $y_{ij} = \frac{\sum_k x_{ij}^k}{u_{ij}}$

- $\min \sum_k \sum_{(i,j)\in A} \left(c_{ij}^k + \frac{f_{ij}}{u_{ij}}\right) \cdot x_{ij}^k$

st ______ flow balance

$$x_{ij}^k \leq u_{ij}^k$$

$$\sum_k x_{ij}^k \leq u_{ij}$$

$$x_{ij}^k \geq 0$$

Minimum cost flow with $|K|$ commodities. Easier to solve.
The optimal value of this problem is a lower bound for the original problem.
Let us denote the objective with LB_w^*.

26.10.2016, Garsombo

- Strong LP relaxation.

Add
$x_{ij}^k \leq u_{ij}^k \cdot y_{ij}$ instead of $x_{ij}^k \leq u_{ij}^k$

$$\min \sum_k \sum_{(i,j)\in A} c_{ij}^k \cdot x_{ij}^k + \sum_{(i,j)} f_{ij} \cdot y_{ij}$$

st.

$$\sum_j x_{ij}^k - \sum_j x_{ji}^k = \begin{cases} O_i^k & i \in O(k) \\ -d_i^k & i \in D(k) \\ 0 & ow \end{cases} \quad \forall i, \forall k$$

$\longrightarrow$ $x_{ij}^k \leq u_{ij}^k \cdot y_{ij}$ add here, too

$$\sum x_{ij}^k \leq u_{ij} \cdot y_{ij}$$

$$y_{ij} \in \{0,1\} \quad , \quad x_{ij}^k \geq 0 \quad \forall k, \forall ij$$

No special structure.
- Solve as LP.
Let LB_s^* be the optimal objective function of the LP relaxation. $LB_s^* \geq LB_w^*$

Add & Drop Heuristic

If y_{ij}'s are known, then we will have multi commodity flow problem.
The main idea is to fix a set of arcs, say $A' \subset A$;
If Linear Cost Multi Commodity Network Flow Problem (LCMCNFP) over A' (LCMCNFP(A')) is feasible, a solution is found, else change A'.

Drop Heuristic starts with $A' = A$ and deletes one arc at a time if possi
Add Heuristic starts with $A' = \emptyset$ and adds one arc at a time.
until feasibility
until infeasibility

→ Drop Heuristic

Step 0: Set $h=0$ $A^h = A$
Let $x_{ij}^{k,h}$ be the optimal solution, if exists, of LCMCNFP (A^h). If infea
original problem is also infeasible.

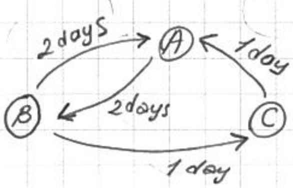
Step 1: For each arc $(i,j) \in A^h$
Let $A'^h = A^h \setminus \{(i,j)\}$ and solve LCMCNFP(A'^h)
If all the min cost problems are infeasible (you cannot drop anymore, the set of arcs A^h and the solution $x_{ij}^{k,h}$ is the best solution.
Otherwise, let (v,w) be the arc whose removal from A^h allows the least c
solution. Cost is the cost of LCMCNFP(A^h) + $\sum_{(i,j)\in A^h} f_{ij}$

Step 2: Set $A^{h+1} = A^h \setminus \{(v,w)\}$
$h = h+1$, go to step 1.

Dynamic Models

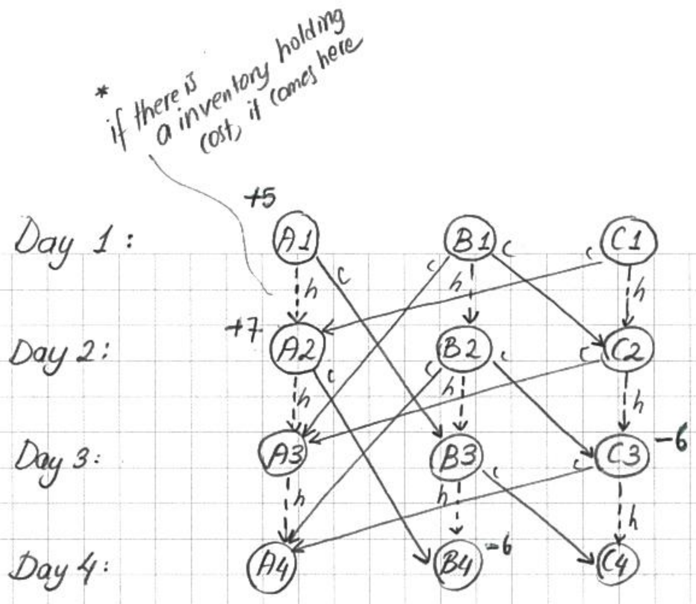
→ In dynamic models, the parameters will change by time.
→ In dynamic models, we usually replicate the physical network for each time period and add temporal links.
→ The temporal links between two replicas of the same node (between period t and t+1) may represent the time, cost, ... for loading/unloading.

Ex:



A supplies at day 1 5 units
2 7 units
C requires at day 3 6 units
B requires at day 4 6 units

Say staying over requires h \$/unit
travelling requires c \$/unit



Shipment Consolidation & Dispatching

- Producers may need to choose the best way of delivering timely over a planning horizon T (days, ...)

→ Decisions: * Mode of transportation
* Consolidation
* Schedule : start times, intermediate steps, the routing, ...

A General Formulation

Let $k \in K$: represents a specific order (shipment)

i_k : destination

w_k : weight

r_k : release date (the day at which the order is ready to be dispatched)

d_k : deadline (the day the order must be delivered)

The company may transport via "one-way" truck trips or by using a common carrier.

- For "renting" one-way truck trips.

Let R: set of all routes from the manufacturer to all $i_k \in K$

For each $r \in R$

S_r = set of stops at router r in a given sequence

i.e. a set of destinations that can be served via router r (a shipment can be transported via r, if $i_k \in S$)

f_r = fixed cost

q_r = capacity of vehicle operating over route r.

π_{kr} = # of days needed to deliver shipment k via route r.

- For common carrier:

g_k = Cost of transporting shipment k by common carrier.

π'_k = # of days needed to deliver shipment k by common carrier.

Variables:

$k \in K$ • $z_k = \begin{cases} 1, & \text{if shipment } k \text{ is transported via common carrier} \\ 0, & \text{ow} \end{cases}$

• $y_{rt} = \begin{cases} 1 & \text{if route } r \text{ is started at day } t \\ 0 & \text{ow} \end{cases}$ $\forall r \in R, t \in T$

• $x_{krt} = \begin{cases} 1 & \text{if shipment } k \text{ is assigned to route } r \text{ starting from time } t \\ 0 & \end{cases}$

$$\text{Min} \sum_k g_k \cdot z_k + \sum_r \sum_t f_r \cdot y_{rt}$$

S.t.

- $$z_k + \sum_{r:\, i_k \in S_r} \sum_{t:\, r_k \leq t \leq d_k - \pi_{kr}} x_{krt} = 1 \quad \forall k \in K$$

A shipment can only be assinged to a route if time is okay. if $t \geq r_k$, $t + \pi_{kr} \leq d_k$ and also if $i_k \in S_r$

- $$\sum_{\substack{k:\, i_k \in S_r, \\ r_k \leq t \leq d_k - \pi_{kr}}} x_{krt} \cdot w_k \leq y_{rt} \cdot q_r \quad \forall r, \forall t$$

- $z_k, y_{rt}, x_{krt} \in \{0,1\}$

(Do we need? $\sum_t y_{rt} = 1 \;\; \forall r$ → No ⇒ if needed can assign a different vehicle starting at a different date

- You can also write the model for $\forall r\ \forall t$ and then set the inappropriate X variables to zero.

$x_{krt} = 0$ if $i_k \notin S_r$ OR $x_{krt} = 0$ if $t < r_k$

OR

$x_{krt} = 0$ if $t > d_k - \pi_{kr}$

2 Kasım, Çarşamba

Vehicle Allocation Problems

* Faced by carrier. Main decisions:
  - which load to accept/reject
  - ceposition empty vehicles

→ General Formulation over time expanded network
- Single type vehicle
- Full trucks (no consolidation, pick up or delivery on the way)

Let $1, \dots, T$ denote time periods

Let
- d_{ijt} : # of loads available at time t to be moved from i to j
- π_{ij} : travel time between i & j
- p_{ij} : profit (revenue − cost) derived from moving a load from i to j.
- c_{ij} : cost of moving an empty vehicle from i to j.
- m_{it} : # of vehicles enter the system at node i, at time t.
  ($m_{depo,0}$ = fleet size)

Variables:

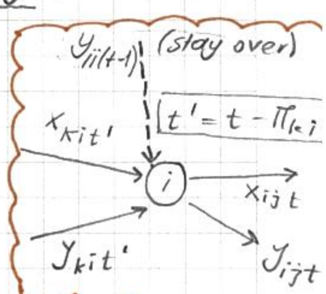
- x_{ijt} : the number of vehicles start moving fully from i to j, at time
- y_{ijt} : the number of vehicles start moving empty from i to j, at time

$$\text{Max} \quad \sum_i \sum_j \sum_t p_{ij} \cdot x_{ijt} - \sum_i \sum_j \sum_t c_{ij} \cdot y_{ijt}$$

s.t.

$$x_{ijt} \leq d_{ijt} \quad \forall i, j, t$$

$$\sum_j x_{ijt} + \sum_j y_{ijt} - \sum_{\substack{k: k \neq i \\ t > \pi_{ki}}} \left(x_{ki(t-\pi_{ki})} + y_{ki(t-\pi_{ki})} \right) - y_{ii(t-1)} = m_{it} \quad \forall i, t$$



→ $d_{ijt} - x_{ijt}$ represents the rejected loads.

→ $x_{ijt} > 0$. represents the accepted loads.

Dynamic Driver Assignment Problem

Arises in tracking where full load trips are assigned to drivers.

Let $D = \{1, \ldots, n\}$ set of drivers waiting to be assigned

$L = \{1, \ldots, m\}$ current set of full load trips to be performed

Let c_{ij} $\left.\begin{matrix} i \in D \\ j \in L \end{matrix}\right\}$ "cost" of assigning driver i to load j. (can be considered the cost of moving empty from the current location of i to location of j)

$$x_{ij} = \begin{cases} 1, & \text{if driver } i \text{ is assigned to load } j. \\ 0, & \text{ow} \end{cases}$$

$$\min \sum_i \sum_j c_{ij} \cdot x_{ij}$$

s.t

$$\sum_i x_{ij} = 1 \qquad \forall j \in L$$

$$\sum_j x_{ij} = 1 \qquad \forall i \in D$$

(Hungarian path)

$$x_{ij} \in \{0,1\}$$

= Assignment Problem and the Hungarian Method =

$$\min \sum_i^{n} \sum_j^{n} c_{ij} \cdot x_{ij}$$

→ cardinality is n

s.t.

$$\sum_i x_{ij} = 1 \qquad \forall j$$

$$\sum_j x_{ij} = 1 \qquad \forall i$$

$$x_{ij} \in \{0,1\}$$

In an optimal solution n x_{ij}'s will be 1, rest are 0.

Special case of traffic assignment (All $s_i = d_j = 1$)

Basis has $n+m-1$ positives

Since we have $n{:}n$ assignments the solution is highly degenerate.

⇒ Hungarian Method

Hungarian Method:

Step 1:
- Find the min. element in each row of the $n \times n$ cost matrix
- Construct a new cost matrix by subtracting the row min. from each cost.
- For this new cost matrix, find the mincost in each column. Construct a new cost matrix by subtracting from each cost the column min.

$\Rightarrow$ leads to a new cost matrix, say C'

Step 2: Cover all zeros in C' with min number of horizontal or vertical lines.
If n lines are required, stop. Optimum is achieved.
O.W. go to step 3.

Step 3: Find the smallest uncovered entry in C' (call it k). Subtract k from each uncovered entry of C' and add k to each twice covered entry.
Return to step 2 with the new C'.

$\rightarrow$ Recovering the optimum solution.
Assign $x_{ij} = 1$ to cells with 0 in the found C'.
Be careful to assign one x_{ij} to each row & column.

Example: $n = 4$

$$C = \begin{bmatrix} 14 & 5 & 8 & 7 \\ 2 & 12 & 6 & 5 \\ 7 & 8 & 3 & 9 \\ 2 & 4 & 6 & 10 \end{bmatrix} \quad \begin{array}{c} \text{row min} \\ 5 \\ 2 \\ 3 \\ 2 \end{array} \longrightarrow \begin{array}{c} \begin{bmatrix} 9 & 0 & 3 & 2 \\ 0 & 10 & 4 & 3 \\ 4 & 5 & 0 & 6 \\ 0 & 2 & 4 & 8 \end{bmatrix} \\ \text{col. min} \quad 0 \;\; 0 \;\; 0 \;\; 2 \end{array} \longrightarrow \begin{bmatrix} 9 & 0 & 3 & 0 \\ 0 & 10 & 4 & 1 \\ 4 & 5 & 0 & 4 \\ 0 & 2 & 4 & 6 \end{bmatrix} = C'$$

min # of lines = $3 < 4$

$k = 1$

$$C' = \begin{bmatrix} 10 & 0 & 4 & 0 \\ 0 & 9 & 4 & 0 \\ 4 & 4 & 0 & 3 \\ 0 & 1 & 4 & 5 \end{bmatrix}$$

# of lines = 4, Stop, optimum

	cost
$1 \rightarrow 2$	5
$2 \rightarrow 4$	5
$3 \rightarrow 3$	3
$4 \rightarrow 1$	+ 2
	15

- Justification of Optimality of Hungarian.

Observe that if a constant is added or subtracted from each cost in a row or column of an assignment problem, the optimal solution is uncha

Proof:

Say we add k to each entry of row s

$$\text{new } z = \text{old } z + k.(\underbrace{x_{s_1} + x_{s_2} + x_{s_3} + \cdots + x_{s_n}}_{1}) \Rightarrow \text{new } z = \text{old } z +$$

Similar arg. holds for column

So step 1 of Hungarian creates another assignment problem having the same optimal solution as the original one.

Step 3 is equivalent to adding k to each cost that lies in a cove[red] row and subtracting k from each cost that lies in an uncovered column (or vise versa).

⟹ The Hungarian Method creates a sequence of assignment problem[s] that all have the same optimal solution as the org. one.

Now consider an assignment problem where all costs are ≥ 0. If a solution with $z=0$ is found, that it is optimum. This is when Hungarian stops.

14 Kasım, Pazartesi

Planning and managing Short Haul Freight transportation

Short haul ⟶ pick up & delivery of goods in a "small" area using a fleet of trucks.

Usually;
- → Vehicles have single depot
- → Tours performed in a single shift.

Decision:

- Strategic : depot, wh locations
- Tactical : fleet sizing
- Operational : Build vehicles tours to satisfy user requests (V[RP])

Vehicle Routing Problem (VRP)

⊛ Determining the routes to be used by a fleet of vehicles to serve a set of users.

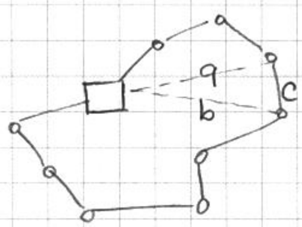
- Operational Constraints:

- The # of vehicles, m can be a variable (with an upper bound const[raint]) or a parameter.
- Vehicle capacity
- The duration of any vehicle should not exceed a work shift duration.

- Customers may require to be served within time windows
- Some customers may require special vehicles.
- We may have precedence relations between customers.
- A customer may be served by a single or many vehicles.

(*) In the absence of operational constraints
$\Rightarrow$ Visit each node from a single depot.

If the objective is related with link costs (travel costs) satisfying $\Delta\neq$ triangular ineq
the we have a TSP.



$\Delta\neq \; c < a + b.$

Travelling Salesman Problem (TSP)

Let $G = (V, A, E)$ — set of vertices, set of arcs (one-way streets), set of edges (two-way)

Vertex 0 is depot.

$U \subseteq V$ set of customers (need to be ser…

$\Rightarrow$ Construct an auxilary directed complete graph.

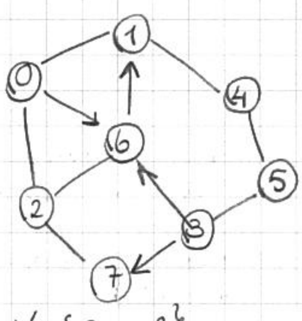
$G'(V', A')$ where $V' = U \cup \{0\}$

- $A' = \{(i,j) : i \in V', \; j \in V', \; j \neq i\}$

c_{ij} = cost of the least cost path from i to j

They satisfy $\Delta\neq$ $\quad c_{ij} \leq c_{ik} + c_{kj} \quad (i,j) \in A' \quad k \in V' \quad k \neq i,j$

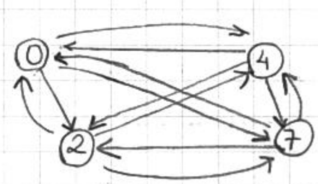
G



$V = \{0, \dots, 7\}$

$U = \{2, 4, 7\}$

G' $\quad V' = \{0, 2, 4, 7\}$



Compute c_{ij} for all (i,j) using
costs and the structure of …

TSP tour: visit each vertex in V' exactly once and form a cy
$\longrightarrow$ Hamiltonian tour in G'.

if $c_{ij} = c_{ji} \longrightarrow$ symmetric TSP
ow. asymmetric TSP (more general)

Formulation of ATSP

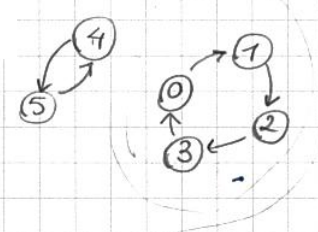
Let $x_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ is in the tour} \\ 0 & \text{ow} \end{cases}$ $\quad i \rightarrow j \quad$ j is visited right after

$$\min \sum_{(i,j)\in A'} c_{ij} \cdot x_{ij}$$

$$\sum_i x_{ij} = 1 \qquad \forall j \in V'$$

$$\sum_i x_{ji} = 1 \qquad \forall j \in V'$$

+subtour breaking constraints $\longrightarrow$



$$x_{ij} \in \{0,1\}$$

Subtour breaking constraints

→ Set Version

- $\sum_{i\in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \qquad S \subset V' \qquad |S| \geq 2$

OR

- $\sum_{i\in S} \sum_{j \notin S} x_{ij} \geq 1 \qquad S \subset V', \ |S| \geq 2$

Need to include subtour breaking constraints for every $S \subset V'$ which is exponentially many.

Methodology:

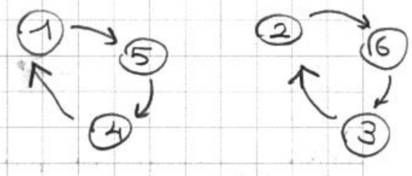
Solve the relaxed problem (= original problem without subtour eliminatio constraints = AP (solve with Hungarian)) and obtain a lower bound
2 possibilities
→ 1. Unique tour (no subtours) which is feasible to the original original problem - Hungarian Solution is optimal.
→ 2. There are subtours. → Add appropriate constraints & reoptimize.

Ex: Say $V' = \{1, \dots, 6\}$

$$\min \sum_{i=1}^{6} \sum_{j=1}^{6} C_{ij} \cdot x_{ij}$$

$$\text{st.} \quad x_{11} + x_{12} + \dots + x_{16} = 1$$
$$x_{21} + x_{22} + \dots + x_{26} = 1$$
$$\vdots$$
$$x_{61} + x_{62} + \dots + x_{66} = 1$$
$$x_{ij} \in \{0, 1\}$$

Say we optimize and get



To avoide these subtours we need to solve

$$\min \sum_i \sum_j C_{ij} \cdot x_{ij}$$

$$\sum_i x_{ij} = 1 \quad \forall j.$$
$$\sum_i x_{ji} = 1 \quad \forall j.$$
$$x_{15} + x_{14} + x_{45} + x_{41} + x_{51} + x_{54} \leq 2.$$
$$x_{26} + x_{63} + x_{32} + x_{23} + x_{36} + x_{62} \leq 2$$
$$x_{ij} \in \{0, 1\}$$

- After reoptimizing new subtours may arise and we may additionall include new subtour elimination constraints.

⇒ Main idea: Solve the assignment problem with $C_{ii} = +\infty$ (instead of 0)

In the optimal solution of assignment problem we may have subtours.
- If no subtour, AP solves ATSP optimally.
- Else either branch & bound for optimal solution or heuristics to g a tou
  ⇒ Patching Heuristics

√ Patching Heuristic

Step 0: Let $C = \{C_1, \dots, C_p\}$ be the p subtours in the AP optimal solution
If $p=1$, stop. AP optimum is feasible and so optimum for ATSP.

Step 1: Identifying the two subtours say C_h and C_k with the longest # of vertices.

Step 2: Merge C_h and C_k in such a way that the cost increase is kept minimum. Remove one arc from C_h and one from C_k. and ad two arcs between C_k and C_h.
So that we have a single tour from C_h & C_k

Step 3: Update C and let $p = p-1$. If $p = 1$, stop. A feasible solution is found.
O.w. go to step 1.

Ex: $C_1 = \{(1,4),(4,3),(3,9),(9,1)\}$ $C_2 = \{(20,22),(22,20)\}$, $C_3 = \{(18,0),(0,18)\}$

Step 1: Pick C_1 & C_2 (C_3 can also be selected).

Step 2: Say remove (9,1)
(18,0)

$1 \rightarrow 4 \rightarrow 3 \rightarrow 9$
$0 \rightarrow 18$

$\Delta = -4.1 - 3.9 + 3.8 + 7.4 = +$

or say remove (18,0)
(3,9)

$9 \rightarrow 1 \rightarrow 4 \rightarrow 3$
$0 \rightarrow 18$

$\Delta = -3.9 - 3.2 + 4.2 + 3.3 = \underline{\underline{+0.4}}$ is the best

* You need to check all possibilities.

Step 3: $p = 2$

Say remove (18,9) and (20,22), add (18,22) (20,9) $\Rightarrow$ $\Delta = 0.3$ is the best

Fig 7.8 is optimal.

✓ Nearest Neighbor is quick and is 10-15% away from optimum.

At each iteration insert the nearest neighbor of the last inserted node

Ex:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 4 & 1 & 3 & 6 & 9 \\ 8 & 0 & 5 & 7 & 3 & 1 \\ 7 & 6 & 0 & 2 & 4 & 4 \\ 6 & 2 & 1 & 0 & 6 & 1 \\ 9 & 8 & 2 & 5 & 0 & 4 \\ 4 & 1 & 3 & 3 & 5 & 0 \end{bmatrix} \end{array}$$

→ We tried to solve it via Hungarian to get a lower bound ($c_{ii} = M$)

$1 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 2 \rightarrow 5 \rightarrow 1$ Cost 17.

√ **Cheapest Insertion Method.**

- Start with a particular tour.
- Gradually insert other vertices (unvisited) between two vertices of the current tour
- At each iteration the cheapest insertion is performed.

Ex: (Same)   * Aslında $1 \to 3 \to 1$ ile başlanıp devam edilir.

Say we have $1 \to 3 \to 4 \to 1$, Unvisited vertices $\{2,5,6\}$

			cost difference
Try inserting 2 between	1&3	$1 \to 2 \to 3 \to 4 \to 1$	$+4+5-1=8$
	3&4	$1 \to 3 \to 2 \to 4 \to 1$	$+6+7-2=11$
	4&1	$1 \to 3 \to 4 \to 2 \to 1$	$+2+8-6=4$
Try inserting 5 between	1&3	$1 \to 5 \to 3 \to 4 \to 1$	$6+2-1=7$
	3&4	$1 \to 3 \to 5 \to 4 \to 1$	$4+5-2=7$
	4&1	$1 \to 3 \to 4 \to 5 \to 1$	$6+9-6=9$
Try inserting 6 between	1&3	$1 \to 6 \to 3 \to 4 \to 1$	$9+3-1=11$
	3&4	$1 \to 3 \to 6 \to 4 \to 1$	$4+3-2=5$
	4&1	$1 \to 3 \to 4 \to 6 \to 1$	$1+4-6=-1$

$1 \to 3 \to 4 \to 6 \to 1$ cost is 8.   unvisited $\{2,5\}$

Turn back to step 1.

* Nearest neighbour, patching, and cheapest insertion are constructive heuristics.

- Improvement heuristics?
  - r-opt heuristics.
    - Remove r edges from the tour
    - Reconnect the tour. If an improvement is realized, a new tour has been constructed

Most common values for r are 2 & 3.

2-opt usually brings you within 5% of the optimum.

Ex: (Same) $1 \to 5 \to 3 \to 4 \to 6 \to 2 \to 1$ Cost is 20.

Apply 2-opt. remove $1 \to 5$ & $6 \to 2$

$5 \to 3 \to 4 \to 6$

$2 \to 1$

Connect the tour with 5-2, 6-1

- $1 \to 6 \to 4 \to 3 \to 5 \to 2 \to 1$ cost is 33
- $1 \to 2 \to 5 \to 3 \to 4 \to 6 \to 1$ cost is 16.

(*) Everything we did in ATSP is applicable, since symmetric TSP is a special case of ATSP.

21.10.2016
Pazartesi

Symmetric TSP

$C_{ij} = C_{ji}$, $G' = (V', E')$

Use $i<j$ convention - (i)—(j)

- $x_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ is in the tour } (i-j) \\ 0 & \text{ow} \end{cases}$

$$\min \sum_{i<j}\sum c_{ij} \cdot x_{ij}$$

st.

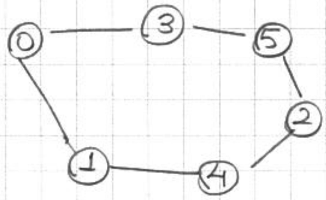
$$\sum_{i<k} x_{ik} + \sum_{k<i} x_{ki} = 2 \qquad \forall k \in V'$$

I. $$\sum_{(i,j)\in S} x_{ij} \leq |S|-1 \qquad \forall S \subset V', \ |S| \geq 3$$

OR

II. $$\sum_{\substack{i\in S:\, j\notin S \\ i<j}} x_{ij} + \sum_{\substack{i\in S:\, j\notin S \\ j<i}} x_{ji} \geq 2 \qquad \forall S \subset V', \ 2 \leq |S| \leq [V'/2]$$

$$x_{ij} \in \{0,1\}$$

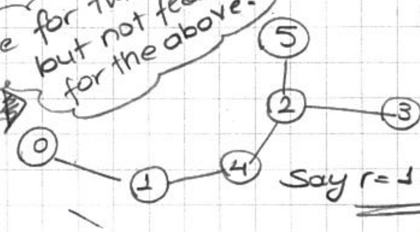


$x_{03} = x_{35} = x_{25} = x_{24} = x_{14} = 1$

A Lower Bound

Select an arbitrary $r \in V'$

feasible for this one but not feasible for the above.



Say $r=1$.

Consider

$$\min \sum_{i<j}\sum c_{ij} \cdot x_{ij}$$

st.

$$\sum_{i<r,\, i\in V'} x_{ir} + \sum_{r<i,\, i\in V'} x_{ri} = 2 \qquad \text{(instead of } \forall j \text{ we write only for } r.)$$

$$\sum_{\substack{i\in S:\, j\notin S \\ i<j}} x_{ij} + \sum_{\substack{i\in S:\, j\notin S \\ j<i}} x_{ji} \geq 1 \qquad S \subset V', \ 1 \leq |S| \leq [V/2]$$

Minimum cost spanning r-tree problem

Output is a connected Subgraph where node r has degree 2.

→ Why it is a lower bound for the symmetric TSP?

Enough to show

Let $\bar{x}$ be an optimal solution to spanning r-tree problem.

→ Is it feasible to STSP? (Not necessarily)

If $\bar{x}$ solves STSP → $\bar{x}$ is also feasible to spanning r-tree probl

Any feasible solution to STSP is also feasible to spanning r-tree prob

Spanning tree can output a better $\bar{x}$ (not feasible to STSP)

⟹ It is a lower boun

How to solve min cost spanning r-tree problem?

Step 1: Determine mincost spanning tree $V' \setminus r$

Step 2: Insert node r with two least cost arcs

Step 1 requires min cost spanning tree problem.

Kruskal's Algorithm (Optimal) ⟶ min cost spanning tree.

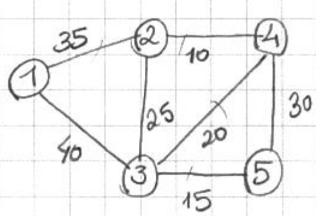
Step 0: Sort all the arcs in nondecreasing order of cost. Let SORT be th set of sorted arcs. Let LIST = ∅ {This set will have the arcs of spann tree at the end

Step 1: Let k be the first arc in SORT (Smallest arc)
Consider adding k to LIST.

- If arcs in LIST with k forms a cycle, disregard k (don't include LIST), delete k fom SORT
- If no cycle is formed, add k to LIST

Step 2: Repeat step 1, Until $|LIST| = n-1$ (where $n = |V' \setminus r|$)

EX:



Find the cheapest spanning tree.

SORT = {(2,4), (3,5), (3,4), (2,3), (4,5), (1,2), (1,3)}

LIST = ∅

SORT = {(2,4), (3,5), (3,4), (1,2)} & cost = 80

① SORT = {(3,5), ...
LIST = {(2,4)}

② SORT = {(3,4), ...
LIST = {(2,4), (3,5)}

⋮

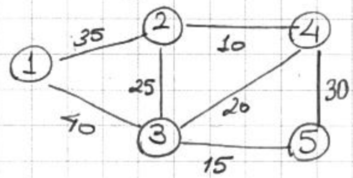
23.11.2016, Çarşamba

Kruskal's Algorithm is for minimum spanning tree

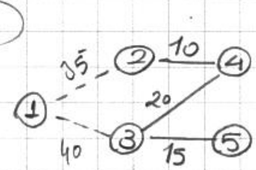
We require minimum spanning r-tree.
For each $r \in V'$
- Apply Kruskal to $V' \setminus r$
- Add r with two least cost arcs

In order to get a lower bound for STSP, we select the best among the r's

Ex:

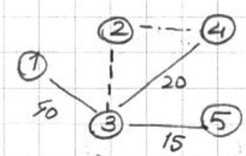


(r=1)



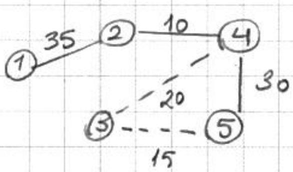
cost: 120

(r=2)



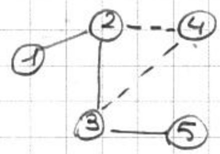
cost: 110

(r=3)



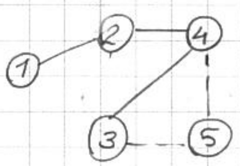
cost: 110

(r=4)



cost: 105

(r=5)



cost: 110

(*) Lower Bound is 120

Christofides Heuristic

(It is prooved that this is at most 50% off from the optimum)

Step 1: Compute the min spanning tree for V' using Kruskal.
Let E'_1 be the edges in the tree.

Step 2: Let V'_D be the vertices of odd degree in the tree.
(degree of vertex = # of arcs incident to it.)

Compute the least cost perfect matching among V'_D
Let E'_D be the edges in the matching and
let

$$E'_H = E'_1 \cup E'_D \quad . \quad \text{Let } H = (V', E'_H)$$

Step 3: If there is a vertex $j \in V'$ of degree greater than 2 in H elimin[ate]
2 edges from E'_H and odd are edge as eliminate
(j,k) (j,h) and add (h,k) (or (k,h)).

or (k,j) & (h,j)

Repeat step3 until all vertices in V' has degree

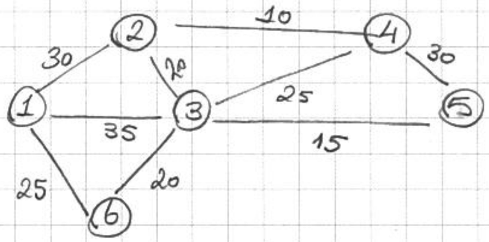
2 in H. E'_H is the resulting Hamiltonian cycle or STSP tour.

Matching: Given a graph $G=(V,E)$ a matching $M \subseteq E$ is a set of disjoint edges such that at most one edge of M is incid... to any node $v \in V$.

A matching is perfect if for each $v \in V$, there is an arc in M inciden... to it. For a graph with $|V| = 2k$, a perfect matching has k arcs.

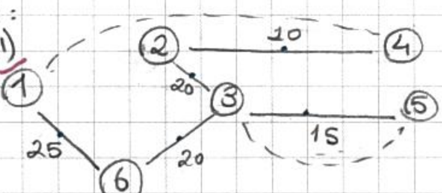
How to solve the min. weight perfect matching of step? By enumeration (in this course)

Ex:



	1	2	3	4	5	6
1	–	30	35	40	50	25
2		–	20	10	35	40
3			–	25	15	20
4				–	30	45
5					–	35
6						–

Step 1: (Kruskal)



Step 2: $V'_D = \{1,3,4,5\}$ vertices w/ odd degrees.

Matchings
- 1-3, 4-5 ⟶ 35+30=65
- 1-4, 3-5 ⟶ 40+15=55 * (1,4) (3,5)
- 1-5, 3-4 ⟶ 50+25=75

Step 3:

Vertices with degree > 2 ⟹ node 3

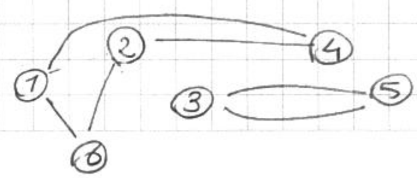
eliminate

2 3 3 5	add 2 5	⟶ $\Delta = -20-15+35-0$	by chan... $\Delta = 0$ ol... he...
2 3 3 6	add 2 6	⟶ $\Delta = -20-20+40=0$	
3 5 3 6	add 5 6	⟶ $\Delta = -20-15+35=0$	

triangular inequality Δ

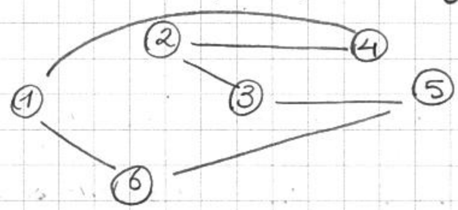
According to Δ, choose one arbitrarly
Normally due to $\Delta \neq$ we expect decrease in cost.
But be careful, the tour should remain connected

If option 2 is selected

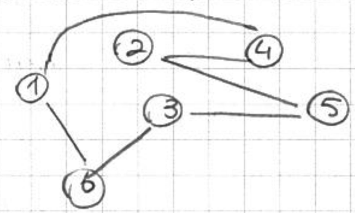


NOT Valid. Do not choose.

If 3 or 1 is adopted, you get a STSP tour with cost 145



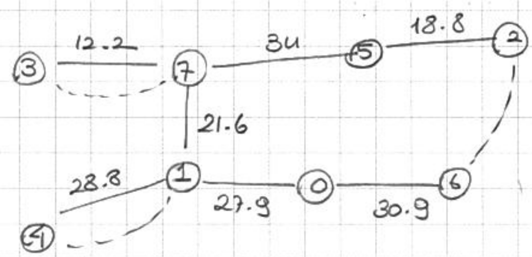
OR



Ex:

	0	1	2	3	4	5	6	7
0	–	27.9	54.6	42	56.5	37	20.9	34.1
1		–	67.2	25.6	28.8	48.4	57.4	21.6
2			–	60.5	95.8	18.8	60.4	52.1
3				–	39.4	43.1	7.2	12.2
4					–	77.2	84.5	44.4
5						–	51.6	34
6							–	59.3
7								–

Step 1:



Cost = 174.2

Step 2:

$V'_D = \{1, 2, 3, 4, 6, 7\}$ (You can also use Hungarian with $c_{ii} = M$)

34   17   26
12   67
16   27

31   47   26
24   67
46   27

37   14   26 = 101.4* best
24   16
46   12

23   14   67
47   16
46   17

36   14   27
47   12
24   17

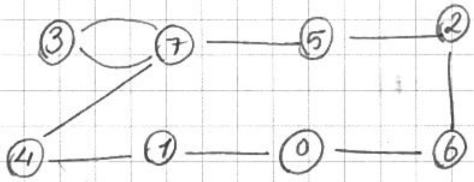
Step 3: Nodes with degree > 2 : $\{1,7\}$

Node 1 : (1,4) (1,4) (0,1) (1,7)

eliminate (1,4) (0,1) add (0,4) $\Rightarrow \Delta = -28.8 - 27.9 + 56.5 = -0.2$
(1,4) (1,7) " (4,7) $\Rightarrow \Delta = -28.8 - 21.6 + 44.4 = -6$
(0,1) (1,7) " (0,7) $\Rightarrow \Delta = -27.9 - 21.6 + 34.1 = \underline{-15.4}$

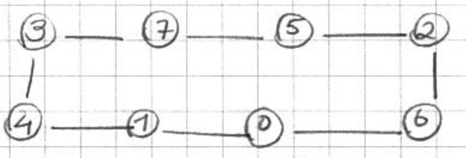
not connected
x Don't ch

Choose 2.



nodes with degree > 2 : 7
Arcs : (3,7) (4,7) (5,7)

eliminate (3,7) (4,7) add (3,4) $\Rightarrow \Delta = -17.2$ *
(3,7) (5,7) add (3,5) $\Rightarrow \Delta = -3.1$



Local Search Algorithms

- Starts with a feasible solution. Try to "improve" the solution. At each ste consider neighbor solutions.
- Stops at local optimum.

General Form.

Step 0: Find an initial feasible solution. Denote it by $x(0)$. Let $N(X^0)$ be its neighbourhood. Set $h=0$

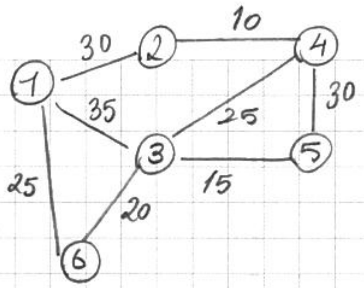
Step 1: Enumerate the feasible solutions belonging to $N(x^h)$. Select the best $x^{h+1} \in N(x^h)$

Step 2: If $cost(x^{h+1}) < cost(x^h)$ (for min.) set $h=h+1$ and go to step 1
ow. x^h is the best solution can be found.

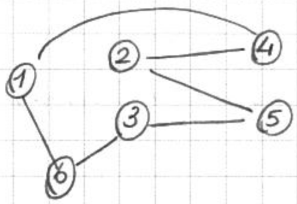
Main Issues: * How to find x_0
* How to define neighbourhood.

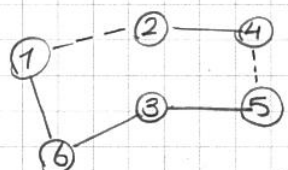
- For STSP a common neighbourhood definition is substituting k edges of the tour with k "other" edges (k-neighbourhood)

Ex:



After Christofides → Apply 2 neighbourhood:
Say delete (1,4) and (2,5)





$\Delta = -40-35+20+20$
$= -5$
improves

New STSP tour with 140.

28 Kasım 2016
Pazartesi

Vehicle Routing Problem

Find a least cost route for vehicles such that;

0. Visit each customer exatly one.
1. Each customer has a demand p_i (to be collected or delivered) an each vehicle has a capacity.
2. Upperbound L on the length of each route (eg. at most 1000 km
   " " 8 hrs)
3. Time windows. Customer i must be served between time a_i and b_i (time window $[a_i; b_i]$)
4. Combination of collection and delivery.
5. Multiple depots.
6. Heterogenous vehicle fleet. The vehicles can be grouped due to different loading constraints.
7. Special vehicles.
8. Periodic VRP. Customers may have different periods to be serve
   eg. customer 1 : Sundays & Thursdays
   2 : Monday, wednesday, friday
   Each customer has a frequency to be meet and the problem is determining the daily routes so that each customer's service frequency is met.
9. Vehicles may not be available whole day.

✓ Focus on symmetric case only.

- If we include capacities only (1)
  - p_i: demand of customer i
  - Q: total capacity of vehicle → VRP

p_i demand, Q capacity

Let $M = \{1, \dots, m\}$ denote the set of vehicles.

Let $x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ traverses arc } (i,j) \quad i<j \\ 0 & \text{ow} \end{cases}$

$$\min \sum_k \sum_{i<j} c_{ij} \cdot x_{ijk}$$

st

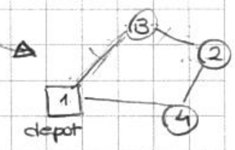
$$\sum_k \sum_{i: i<j} x_{ijk} + \sum_k \sum_{i: j<i} x_{jik} = 2 \qquad \forall j \neq 1 \qquad (\longrightarrow j \longrightarrow)$$

$$\sum_k \sum_{(1,j)} x_{1jk} = 2m$$

$$\sum_{\substack{(i,j): i<j \\ j,i \in S}} x_{ijk} \leq |S| - 1 \qquad S \subset U \setminus \{1\}, \ \forall k$$

$$\sum\sum_{i<j} \quad \sum_{(i,j): i<j} x_{ijk} \cdot (p_i + p_j) \leq 2Q_k \qquad \forall k$$

$$x_{ijk} \in \{0,1\} \qquad i<j, \ \forall k$$



$p_1 + p_2 + p_3 = Q$

$x_{13} = x_{23} = x_{24} = x_1$

- If Q_k is different for some k, the formulation is still valid.

Another IP (Only valid if $Q_k = Q \ \forall k$)

$x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is traversed} \\ 0 & \text{ow} \end{cases}$

$$\min \sum\sum_{i<j} c_{ij} \cdot x_{ij}$$

st.

$$\sum_{i: i<j} x_{ij} + \sum_{i: j<i} x_{ji} = 2 \qquad \forall j \neq 1$$

$$\sum_j x_{1j} = 2m$$

(subtour elm. & capacity together)
$$\sum_{\substack{i<j \\ i \in S, j \notin S}} x_{ij} + \sum_{\substack{j<i \\ i \in S, j \notin S}} x_{ji} \geq 2\alpha(S) \qquad S \subset U \setminus \{1\}, \ |S| \geq 2$$

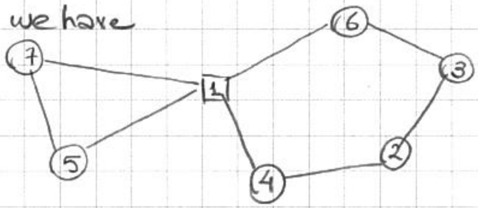
$$x_{ij} \in \{0, 1\}$$

$\alpha(S)$: min # of vehicles to serve set S $\quad \alpha(S) = \left\lceil \frac{\sum_{i \in S} p_i}{Q} \right\rceil$

Ex: for understanding the constraint

Say $N = \{1, \dots, 7\}$, 1 is depot, $p_2 = 10$, $p_3 = 5$, $p_4 = 5$, $p_5 = p_7 = 4$, $p_6 = 3$, $Q = 14$, $m = 3$

Say we have



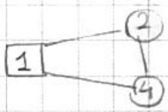
$\Rightarrow$ Infeasible due to Q

For $S = \{2, 3, 4, 6\}$

will break the tour

$$x_{27} + x_{37} + x_{47} + x_{67} + x_{25} + x_{35} + x_{45} + x_{12} + x_{13} + x_{14} + x_{16} + x_{65} \geq 4$$

$$\alpha(S) = \left\lceil \frac{10+5+5+3}{14} \right\rceil = \left\lceil \frac{23}{14} \right\rceil = 2$$



$S = \{2, 4\}$

$$x_{23} + x_{25} + x_{26} + x_{27} + x_{45} + x_{46} + x_{47} + x_{12} + x_{14} + x_{34} \geq$$

30 Kasım, Çarşamba

Subtour Elimination with Miller – Tucker – Zemlin (MTZ)

★ Cannot use $i<j$ convention, have to define all arcs

$x_{ij} = \begin{cases} 1 & \text{if } j \text{ is visited right after } i \quad i \to j \\ 0 & \text{ow} \end{cases}$

u_i = load of the vehicle right aft[er] node i

$$\min \sum_{i} \sum_{j, i \neq j} c_{ij} \cdot x_{ij}$$

st. $\sum_j x_{ij} = 1 \quad \forall i \neq 1$

$\sum_j x_{ji} = 1 \quad \forall i \neq 1$

$\sum_j x_{1j} = m \quad j = 2, \dots$

$\sum_j x_{j1} = m \quad j = 2, \dots$

(subtour & capacity) $u_i - u_j + Q \cdot x_{ij} \leq Q - p_j \quad \forall i, j = 2, \dots, n, \; i \neq j$

$p_i \leq u_i \leq Q$

$x_{ij} \in \{0, 1\}$

$(i) \xrightarrow{u_i} (j)$ u

$x_{ij} = 1 \to u_i - u_j + Q \leq Q - p_j$

$u_i + p_j \leq u_j$

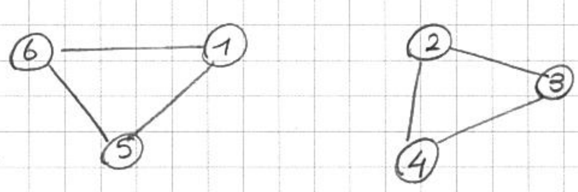
$x_{ij} = 0 \to u_i - u_j \leq Q - p_j$ is allw[ays] val[id]

$(Q - p_j) - \dots$

to obtain the max value for that u_i have to be max;

52

The case for $x_{ij}=1$ also acts as subtour elimination



$x_{23}=x_{34}=x_{42}=1$

$u_3 > u_2+p_3$

$u_4 \geqslant u_3+p_4 \geqslant u_2+p_3+p_4$

$u_2 \geqslant u_4+p_2 \geqslant u_2+p_3+p_4+p_2$

$u_2 \geqslant u_2+p_3+p_4+p_2$ ↯ contradiction

Distance Constrained VRP
(Including shift duration or total tour length bound)

For $i<j$

$$\min \sum_{i<j} c_{ij}\cdot x_{ij}$$

$$\text{s.t.} \quad \sum_{i:\, i<j} x_{ij} + \sum_{i:\, i<j} x_{ji} = 2 \qquad \forall j=1$$

$$\sum x_{1j} = 2m$$

$$\sum_{\substack{i<j \\ i\in S,\, j\notin S}} x_{ij} + \sum_{\substack{j<i \\ i\in S,\, j\notin S}} x_{ji} \geqslant 2\alpha(S) \qquad S\subset U\setminus\{1\},\ |S|\geqslant 2$$

$$\sum_{\substack{i<j \\ i\in S,\, j\notin S}} x_{ij} + \sum_{\substack{j<i \\ i\in S,\, j\notin S}} x_{ji} \geqslant 4 \qquad S\subset U\setminus\{1\},\ |S|\geqslant 2$$

$$t^*(S) \geqslant T$$

at least 2 vehicle ((in+out)*2)

$$x_{ij}\in\{0,1\}$$

where $t^*(S)$ is the value of minimum Hamiltonian (TSP) cycle span

S

MTZ version

Let t_i = time when we leave node i

$$\min \sum_i \sum_j c_{ij} \cdot x_{ij}$$

st.

$$\sum_j x_{ij} = 1 \qquad \forall i \neq 1$$

$$\sum_j x_{ji} = 1 \qquad \forall i \neq 1$$

$$\sum_j x_{1j} = \sum_j x_{j1} = m$$

$$u_i - u_j + Q \cdot x_{ij} \leq Q - p_j \qquad \forall i,j = 2,\dots,n$$

$$p_i \leq u_i \leq Q \qquad \forall i = 2,\dots,n$$

$$t_i - t_j + T \cdot x_{ij} \leq T - t_{ij} \cdot x_{ij} \qquad \forall i,j = 2,\dots,n$$

e to travel from i to j

$$0 \leq t_i \leq T - t_{i1} \qquad \forall i = 2,\dots n$$

$$x_{ij} \in \{0,1\}$$

t_i $\quad$ t_j

$$i \xrightarrow{t_{ij}} j$$

$$x_{ij} = 1 \implies t_i - t_j \leq -t_{ij} \qquad x_{ij} = 0 \implies t_i - t_j \leq T$$

$$t_j \geq t_i + t_{ij}$$

Exercise: • How would you modify if it takes S_j time to load/un[load] at node j.

• How would you modify the MTZ for delivery cases.

Time Windowed VRP

Let e_i = earliest time at which service can finish at i.
ℓ_i = latest " " " " must " " i.

$$\min \sum_i \sum_j c_{ij} \cdot x_{ij}$$

st. $\sum_j x_{ij} = 1 \quad \forall i \neq 1, \qquad \sum_j x_{ji} = 1 \quad \forall i \neq 1$

$$\sum_j x_{1j} = \sum_j x_{j1} = m$$

$$u_i - u_j + Q\, x_{ij} \leq Q - p_j \qquad \forall i,j = 2,\dots,n$$

$$p_i \leq u_i \leq Q \qquad \forall i = 2,\dots,n$$

$$\longrightarrow \quad t_i - t_j + M \cdot x_{ij} \leq M - t_{ij} \qquad \forall i,j = 2,\dots,n$$

$$e_i \leq t_i \leq \ell_i$$

$$x_{ij} \in \{0,1\}$$

$M = ?$

$X_{ij} = 1 \Rightarrow \quad t_j \geq t_i + t_{ij} \qquad X_{ij} = 0 \Rightarrow \quad t_i - t_j \leq M - t_{ij}$

We know that $t_i - t_j \leq l_i - e_j$ is vali

$$t_{ij} - t_{ij} + t_i - t_j \leq l_i - e_j + t_{ij} - t_{ij}$$

$$t_i - t_j \leq (l_i - e_j + t_{ij}) - t_{ij}$$

for $M \geq l_i - e_j + t_{ij} \quad \forall$

the constraint is valid.

Path Formulation

Let K be the set of routes that satisfy capacity and length
C_K be the cost of route $K \in R$

Parameter:

$$a_{ik} : \begin{cases} 1, & \text{if node } i \text{ is in route } k \quad i \in N, k \in K \\ 0, & \text{ow} \end{cases}$$

Variable:

route?

$$y_K = \begin{cases} 1, & \text{if } \cancel{\text{node}} \; k \text{ selected} \\ 0, & \text{ow} \end{cases}$$

$$\min \sum_k C_K \cdot y_K$$

st.

$$\sum_k a_{ik} \cdot y_K = 1 \qquad \forall i$$

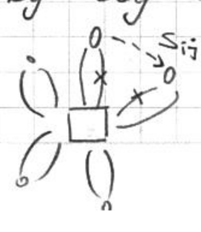
$$\sum_k y_K = m \rightarrow \text{\# of vehicle}$$

$$y_K \in \{0, 1\}$$

Heuristics for VRP

- **Constructive Heuristics:**

- Saving Heuristic: Initially assigns a dedicated route for each cus
Then merge a pair of routes (if capacity, T, ... is okay) according
savings.
Savings can be calculated by : say depot is at 0. $S_{ij} = C_{i0} + C_{0j}$ -

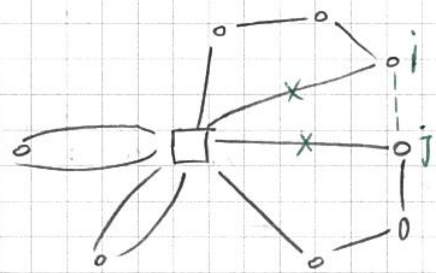


Step 0: Initialization

- Let C be the set of all routes $C_i = \{0, i, 0\}$ $i \in V' \setminus \{0\}$
- For each pair of vertices $i, j \in V' \setminus \{0\}$ compute s_{ij}.
- Let L be the list of savings sorted in a nondecreasing way

Step 1: Extract the largest savings s_{ij}.
If vertices i & j belong to two seperate routes in which i & j are directly linked to depot and if merging these two is feasible, then merge the routes.

Step 2: If $L = \emptyset$ stop, else go back to step 1



— Sweep Heuristic: (Gillett & Miller 1974)

You first determine a straight line (horizontal/vertical with depot) Then this line is rotated in a circle and whenever it touches a customer, this customer is questioned to be included in the tour. If feasible, included, else current tour is finalized and a new tour is started

- Two Phased Heuristics:

— Cluster first, route second: (Fisher & Jaikumar 1980)

2 main steps:
- Customers are partitioned into subsets, each subset will be served by one vehicle.
- Solve a TSP for each subset (capacity, T, ... will automaticall be satisfied due to clustering in 1)

Step 1 is usually a generalized assignment problem which is a lot more difficult than A.P.

For step 1, let d_{ik} : cost of assigning i to cluster k

$$x_{ik} = \begin{cases} 1 & \text{if } i \text{ is assigned to cluster } k \\ 0 & \text{ow} \end{cases}$$

$$\min \sum_{i,k} d_{ik} \cdot x_{ik}$$

st.

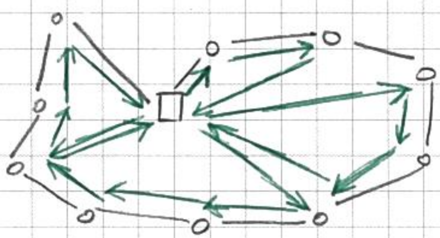
$$\sum_k x_{ik} = 1 \qquad \forall i \neq 1$$

$$\sum_i p_i \cdot x_{ik} \leq Q \qquad \forall k$$

$$x_{ik} \in \{0,1\}$$

– Route first, cluster second (Beosly, 1983)

First a giant TSP is created
Then the route is decomposed into feasible routes.



05.12.2016
Pazartesi

Petal Heuristic: (Hjouring, Ryan, Glover 1983; Renaud, Boctor, Laporte 1996)

- Generate many feasible routes (petals) using for example sweep or any other constructive heuristic.
- Make a selection (set partitioning problem)

parameters

Let c_k: cost of visiting all customers of petal k. (get this by solving a TSP for petal.

$a_{ik}: \begin{cases} 1, & \text{if customer } i \text{ is in petal } k \\ 0, & \text{ow} \end{cases}$

Variables:

$$X_k = \begin{cases} 1 & \text{if petal } k \text{ is chosen.} \\ 0 & \text{ow} \end{cases}$$

Let K: set of all petals.

larger K, better results but harder to solve (CPU tim
So, solution depends on K. diffe
that is why it is a heur

$$\min \sum_k c_k \cdot X_k$$

st.

$$\sum_k a_{ik} \cdot X_K = 1 \qquad \forall i$$

$$X_k \in \{0,1\} \qquad \forall k \in K$$

Criteria to assess the quality of heuristic:

1. Speed (5% of the planning horizon)
2. Flexibility
3. Accuracy
4. Simplicity

Heuristic	Speed	Flexibility	Accuracy	Simplicity
→ Savings	high	low	medium	high
→ Sweep	high	low	medium	high
→ Cluster first Route Second	high	low	good →	medium
→ Route first Cluster second	high	very low	very low	high
→ Petal	medium	high (can add new petals)	good →	medium

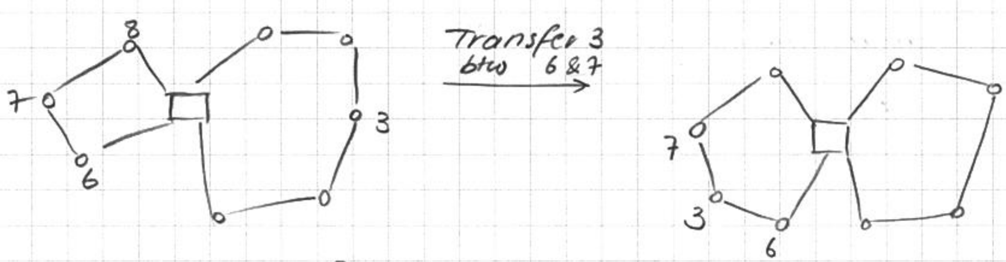
- **Improvement Heuristics:**

– Intro Route Movements (See TSP methods k-neighbourhood)
(within)

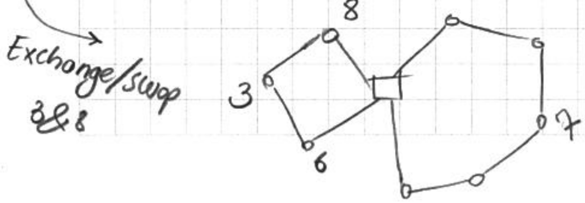
– Inter Route.

Transfer: Move customer i from route I and insert it in all positions of route II.

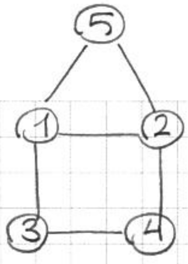
Swap: Exchange customer i from route I with custom j from route II.



Try all posible locations & choose the best one.



Ex:



depot is 1.

1 → 3 → 4 → 2 → 1 → 5 → 2 → 1

For undirected networks

- Necessary condition to have Eularian path
  - The graph must be connected
  - All vertex degrees must be even

End-pairing algoritm (1873) for undirected networks

Step 1: Draw a first cycle by visiting unvisited edges until no longer possible

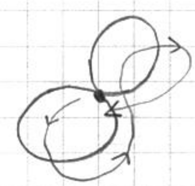
Step 2: Have all edges been visited?
If yes stop, if no go to step 3.

Step 3: Draw a second cycle in the same way.
Starting from an unvisited edge incident to the first cycle.

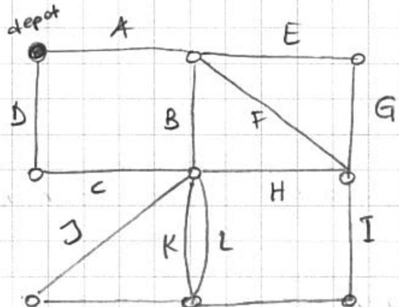
Step 4: Merge the two cycles and go to step 2.

Merging:

Start from the first cycle, move until you hit the second cycle. Then go in the second cycle and then finish the first cycle



Ex:



Connected ✓
Even degrees ✓
} ⇒ Endpairing works.

Cycle 1: A − B − C − D
Cycle 2: E − G − F

Merge: A − E − G − F − B − C − D

Cycle 3: H − L − N − I

Merge: A − E − G − H − L − N − I − F − B − C − D

Cycle 4: K − M − J

Merge: A − E − G − H − K − M − J − L − N − I − F − B − C − D

↳ Eularian cycle

(not strong connect

For directed Networks

Necessary & sufficient condition → Can reach anywhere from anywher

- Graph must be strongly connected.
- In degree = out degree for every node.

Then end-pairing applies.

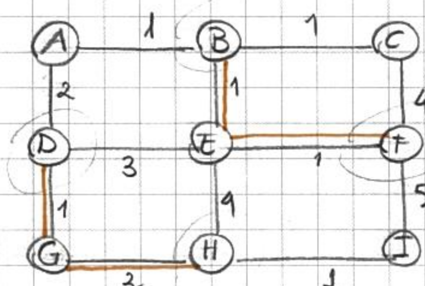
Chinese Postman Problem

Traverse all edges in the most economical way.

- For undirected case
  If not all vertices have even degree ⟹ Add chains to connect odd degree vertices. Do this in the most economical way.

Once all degrees are even, apply end-pairing.

Ex:



Find a chinese postman tour.

connected ✓
even degree ✗

B, D, F, H

B-D & F-H : 3+5
- B-F & D-H : 2+3
B-H & F-D : 5+4

Additional cost is 5. The graph which repeats B-E, E-F, D-G, G-H has an Eularian cycle which can be found by end-pairing.

→ How to match odd degree vertices?
- Complete enumeration
- Utilize hungarian (be careful! :))

- For directed case — strongly connected, in degree = out degree

Step1: If graph is Eularian, apply end-pairing, If no go to step 2
Step2: Identify the deficit (out degree > in degree) and suplus vertices (in degree > out degree).

Step3: Solve the following transportation problem

S: Set of Surplus vertices
D: " " deficit vertices
c_{ij}: length of shortest path from i to j.
s_i: surplus of node i
d_j: deficit of node j

x_{ij} = # of paths (chains added) from i to j

$$\min \sum_{i\in S}\sum_{j\in D} c_{ij}\cdot x_{ij}$$

st.

$$\sum_{j\in D} x_{ij} = s_i \qquad \forall i \in S$$

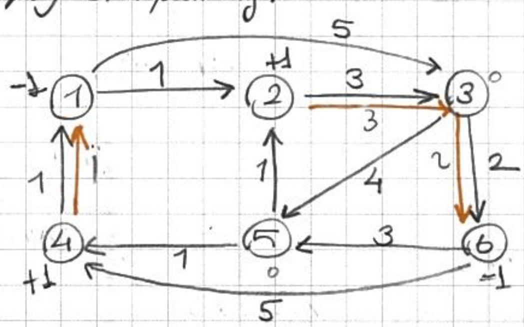
$$\sum_{j\in S} x_{ij} = d_j \qquad \forall j \in D$$

$$x_{ij} \geq 0 \ \& \text{ integer}$$

toplam + = toplam −

Then apply end pairing.

EX:



$S = \{2, 4\}$
$D = \{1, 6\}$

	1	6
2	$\boxed{9}$	$\boxed{5}$
4	$\boxed{1}$	$\boxed{7}$

2 → 6
4 → 1

Additional cost = 5+1 = 6

Total cost = 26+6 = 32

1−2−3−6−5−4−1−5−6−4−1

Routing Problems

The general routing problem (GRP) is a routing problem defined on a graph or network where a min. cost tour is to be found and where the route must include visiting certain required vertices and traversing certain required edges. (Orloff 1974, Eglese & Letchford 2009)

Let $G = (V, A)$

VR : required vertices to be visited
AR : " edges " " "

Basic Problem	VR	AR
GRP	$VR \subseteq V$	$AR \subseteq A$
VRP	$VR = V$	$AR = \emptyset$
ARP	$VR = \emptyset$	$AR = A$
Selective Versions		
General Routing Prob with profits	$VR = \emptyset$, Select $V' \subseteq V$	$AR = \emptyset$, Select $A' \subseteq A$
Routing with profits	$VR = \emptyset$, select $V' \subseteq V$	$AR = \emptyset$
Arc Routing with Profits (Rural postman problem) RPP	$VR = \emptyset$	$AR = \emptyset$, $A' \subseteq A$

Selective Problem	Objective	Constraint	Node (Common Name)	Arc (Common Name)
• Prize Collection	min cost	bound on the total prize	Prize Collecting TSP (1989)	" " RPP (19
• Profitable	max (profit − cost)		(1993)	(200
• Orienteering	max profit	bound on total length or time	OP (1957)	(2009)