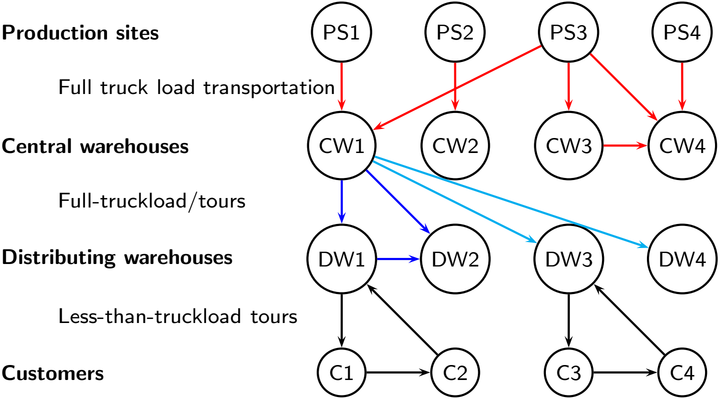
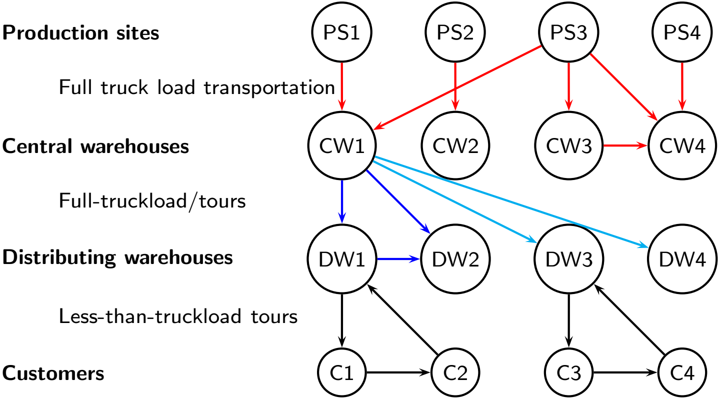
Transportation Logistics

Part II: Location problems and the design of transportation networks


Production sites
Full truck load transportation
Central warehouses
Full-truckload/tours
Distributing warehouses
Less-than-truckload tours
Customers
PS1
PS2
PS3
PS4
CW1
CW2
CW3
CW4
DW1
DW2
DW3
DW4
C1
C2
C3
C4



How many warehouses shall be built?
Which warehouses shall be built?
How should the transportation network be designed?

Median problems

b_i...weight of node i
d_{ij}...distance between nodes i and j

Median problems

b_i...weight of node i
d_{ij}...distance between nodes i and j

Undirected graph

Median: the node with the shortest weighted distance to all other nodes
$\min_{i \in V} \sigma(i)$ and $\sigma(i) = \sum_{j \in V} d_{ij} b_j$.

Median problems

b_i...weight of node i
d_{ij}...distance between nodes i and j

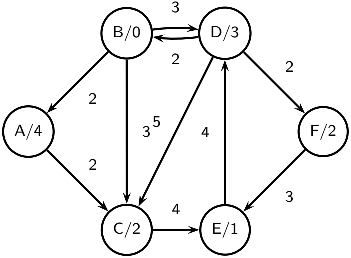
Undirected graph

Median: the node with the shortest weighted distance to all other nodes
$\min_{i\in V}\sigma(i)$ and $\sigma(i)=\sum_{j\in V} d_{ij}b_j$.

Directed graph

Out-median: the node with the shortest weighted distance **to** all other nodes. $\min_{i\in V}\sigma_{out}(i)$ and $\sigma_{out}(i)=\sum_{j\in V} d_{ij}b_j$.
In-median: the node with the shortest weighted distance **from** all other nodes. $\min_{i\in V}\sigma_{in}(i)$ and $\sigma_{in}(i)=\sum_{j\in V} d_{\mathbf{ji}}b_j$.
Median node i for which $\sigma_{out}(i)+\sigma_{in}(i)$ is minimal.

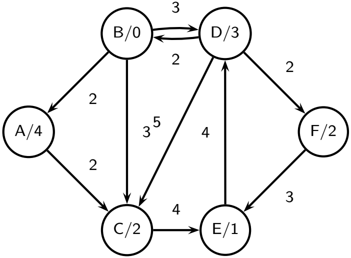
Median problems - Example



$$D = \begin{matrix} 0 & 12 & 2 & 10 & 6 & 12 \\ 2 & 0 & 3 & 3 & 7 & 5 \\ 12 & 10 & 0 & 8 & 4 & 10 \\ 4 & 2 & 5 & 0 & 5 & 2 \\ 8 & 6 & 9 & 4 & 0 & 6 \\ 11 & 9 & 12 & 7 & 3 & 0 \end{matrix} \qquad b = \begin{matrix} 4 \\ 0 \\ 2 \\ 3 \\ 1 \\ 2 \end{matrix}$$

Source: Domschke, Drexl (1990) Logistik: Standorte, Chapter 3.1.1.

Median problems - Example

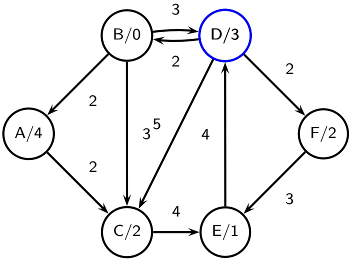


$$D = \begin{array}{cccccc} 0 & 12 & 2 & 10 & 6 & 12 \\ 2 & 0 & 3 & 3 & 7 & 5 \\ 12 & 10 & 0 & 8 & 4 & 10 \\ 4 & 2 & 5 & 0 & 5 & 2 \\ 8 & 6 & 9 & 4 & 0 & 6 \\ 11 & 9 & 12 & 7 & 3 & 0 \end{array} \qquad b = \begin{array}{c} 4 \\ 0 \\ 2 \\ 3 \\ 1 \\ 2 \end{array}$$

Which node is the out-median?

Source: Domschke, Drexl (1990) Logistik: Standorte, Chapter 3.1.1.

Median problems - Example



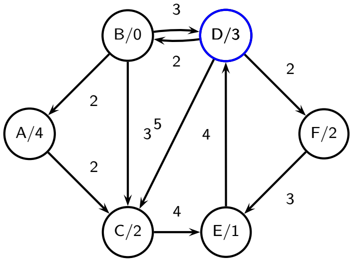
$$D = \begin{pmatrix} 0 & 12 & 2 & 10 & 6 & 12 \\ 2 & 0 & 3 & 3 & 7 & 5 \\ 12 & 10 & 0 & 8 & 4 & 10 \\ 4 & 2 & 5 & 0 & 5 & 2 \\ 8 & 6 & 9 & 4 & 0 & 6 \\ 11 & 9 & 12 & 7 & 3 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 3 \\ 1 \\ 2 \end{pmatrix}$$

Which node is the out-median?

$i = D$, $\sigma_{out}(D) = 35$

Source: Domschke, Drexl (1990) Logistik: Standorte, Chapter 3.1.1.

Median problems - Example



$$D = \begin{matrix} 0 & 12 & 2 & 10 & 6 & 12 \\ 2 & 0 & 3 & 3 & 7 & 5 \\ 12 & 10 & 0 & 8 & 4 & 10 \\ 4 & 2 & 5 & 0 & 5 & 2 \\ 8 & 6 & 9 & 4 & 0 & 6 \\ 11 & 9 & 12 & 7 & 3 & 0 \end{matrix} \qquad b = \begin{matrix} 4 \\ 0 \\ 2 \\ 3 \\ 1 \\ 2 \end{matrix}$$

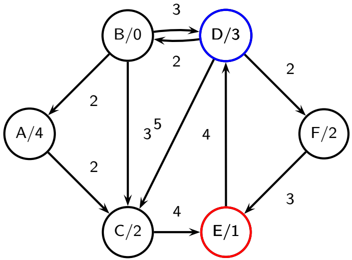
Which node is the out-median?

$i = D,\ \sigma_{out}(D) = 35$

Which node is the in-median?

Source: Domschke, Drexl (1990) Logistik: Standorte, Chapter 3.1.1.

Median problems - Example



$$D = \begin{array}{cccccc} 0 & 12 & 2 & 10 & 6 & 12 \\ 2 & 0 & 3 & 3 & 7 & 5 \\ 12 & 10 & 0 & 8 & 4 & 10 \\ 4 & 2 & 5 & 0 & 5 & 2 \\ 8 & 6 & 9 & 4 & 0 & 6 \\ 11 & 9 & 12 & 7 & 3 & 0 \end{array} \qquad b = \begin{array}{c} 4 \\ 0 \\ 2 \\ 3 \\ 1 \\ 2 \end{array}$$

Which node is the out-median?

$i = D,\ \sigma_{out}(D) = 35$

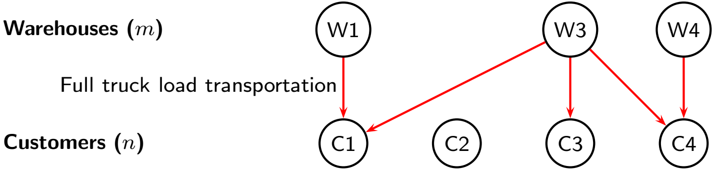
Which node is the in-median?

$i = E,\ \sigma_{in}(E) = 53$

Source: Domschke, Drexl (1990) Logistik: Standorte, Chapter 3.1.1.

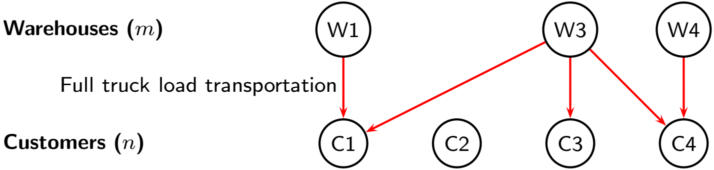
Uncapacitated warehouse location problems (WLP)

We will first consider its simplest version, that is, the single stage/single level problem:



Uncapacitated warehouse location problems (WLP)

We will first consider its simplest version, that is, the single stage/single level problem:



There are n customers, each with a given demand. The company aims at reducing their distribution costs. In order to achieve this goal, it plans to setup and operate distributing warehouses.
There are m potential locations available. If a warehouse is built at site i fixed of f_i EUR occur. The transportation costs for transporting the entire demand of customer j from warehouse i are given by c_{ij} EUR.

Uncapacitated warehouse location problems (WLP)

We want to minimize the total transportation and holding costs under the condition that the demands of all customers are satisfied.

- How many warehouses shall be built?
- At which locations shall they be built?

Note that ...

... in the **uncapacitated** case it is never necessary to supply a customer from more than one warehouse.

WLP: Two possible solutions

Solution 1: build all warehouses

i,j	1	2	3	4	5	6	7	f_i
1	1	2	10	9	6	7	3	5
2	2	9	0	7	3	6	10	7
3	7	6	1	5	3	10	5	5
4	6	5	10	2	6	3	6	6
5	6	4	6	3	7	2	6	5

fixed costs = 5+7+5+6+5=28
transportation costs =
1+2+0+2+3+2+3=13
total costs = 28+13=41

WLP: Two possible solutions

Solution 1: build all warehouses

i,j	1	2	3	4	5	6	7	f_i
1	1	2	10	9	6	7	3	5
2	2	9	0	7	3	6	10	7
3	7	6	1	5	3	10	5	5
4	6	5	10	2	6	3	6	6
5	6	4	6	3	7	2	6	5

fixed costs = 5+7+5+6+5=28
transportation costs =
1+2+0+2+3+2+3=13
total costs = 28+13=41

Solution 2: build only two: warehouse 1 and 3

i,j	1	2	3	4	5	6	7	f_i
1	1	2	10	9	6	7	3	5
3	7	6	1	5	3	10	5	5

fixed costs = 5+5=10
transportation costs =
1+2+1+5+3+7+3=22
total costs = 10+22=32

Source: Domschke, Drexl (1990) Logistik: Standorte, Chapter 3.3.1

WLP: MIP formulation

- in the case where the locations/warehouses are already selected:
  - the total costs can be calculated immediately
  - BUT: $2m - 1$ selection possibilities (for $m = 10 \rightarrow 1023$ possible solutions; m...number of potential warehouse locations)

WLP: MIP formulation

- in the case where the locations/warehouses are already selected:
  - the total costs can be calculated immediately
  - BUT: $2m - 1$ selection possibilities (for $m = 10 \rightarrow 1023$ possible solutions; m...number of potential warehouse locations)

Formulation in terms of a Mixed Integer Program (MIP)

$$y_i = \begin{cases} 1, \text{ if at location } i \text{ a warehouse is built,} \\ 0, \text{ otherwise.} \end{cases}$$

x_{ij} = share of customer j's demand that is covered by warehouse i

y_i...binary, x_{ij}...continuous, $i \in \{1, \dots, m\}$, $j \in \{1, \dots, n\}$

WLP: MIP formulation

- in the case where the locations/warehouses are already selected:
  - the total costs can be calculated immediately
  - BUT: $2m - 1$ selection possibilities (for $m = 10 \rightarrow 1023$ possible solutions; m...number of potential warehouse locations)

Formulation in terms of a Mixed Integer Program (MIP)

$$y_i = \begin{cases} 1, \text{ if at location } i \text{ a warehouse is built,} \\ 0, \text{ otherwise.} \end{cases}$$

x_{ij} = share of customer j's demand that is covered by warehouse i

y_i...binary, x_{ij}...continuous, $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$

How many decision variables do we have?

WLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{1}$$

$$x_{ij} \leq y_i \qquad \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \in \{1,\ldots,n\} \tag{3}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\ldots,m\} \tag{4}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{5}$$

WLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{1}$$

$$x_{ij} \leq y_i \qquad \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \in \{1,\ldots,n\} \tag{3}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\ldots,m\} \tag{4}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{5}$$

The objective function (1) minimizes total transportation and site costs.

WLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{1}$$

$$x_{ij} \leq y_i \qquad \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \in \{1,\ldots,n\} \tag{3}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\ldots,m\} \tag{4}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{5}$$

The objective function (1) minimizes total transportation and site costs. Constraints (2) ensure that a customer can only be served by a warehouse that is built.

WLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{1}$$

$$x_{ij} \leq y_i \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \in \{1,\dots,n\} \tag{3}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\dots,m\} \tag{4}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{5}$$

The objective function (1) minimizes total transportation and site costs.
Constraints (2) ensure that a customer can only be served by a warehouse that is built.

Constraints (3) ensure that the entire demand of customer j is delivered.

WLP: MIP formulation

Difficulty

m*n continuous variables and m binary variables $\rightarrow$ for larger instances, the computation of optimal solutions becomes time consuming.

Solution: employ heuristics

WLP: MIP formulation

Difficulty

m*n continuous variables and m binary variables $\rightarrow$ for larger instances, the computation of optimal solutions becomes time consuming.
Solution: employ heuristics

Classification of heuristics

- Construction heuristics (to obtain a feasible initial solution)
- Improvement heuristics (to improve on a given initial solution)

WLP: construction heuristic ADD

Notation

$I = \{1, \ldots, m\}$	set of all potential warehouse locations.
I_0	set of definitely forbidden locations.
I_0^{pot}	set of potentially forbidden locations.
I_1	set of definitely realized locations.
ω_i	savings in transportation costs if location i is realized in addition to the already selected.
Z	total costs (objective value).

WLP: construction heuristic ADD

- **Initialization**
  - determine which location should be realized if exactly one warehouse is built: for each warehouse location i calculate $c_i = \sum_j^m c_{ij}$; select location k with the smallest cost value $c_k + f_k$
  - set $I_1 = \{k\}$, $I_0^{pot} = I \setminus \{k\}$ and $Z = c_k + f_k$
  - calculate the savings in transportation costs $\omega_{ij} = \max\{c_{kj} - c_{ij}, 0\}$ for each i in I_0^{pot} and all customers j, and the row sum $\omega_i = \sum_{j=1}^n \omega_{ij}$

WLP: ADD - Initialization

c_{ij}	1	2	3	4	5	6	7	f_i	c_i	$f_i + c_i$
1	1	2	10	9	6	7	3	5	38	43
2	2	9	0	7	3	6	10	7	37	44
3	7	6	1	5	3	10	5	5	37	42
4	6	5	10	2	6	3	6	6	38	44
5	6	4	6	3	7	2	6	5	34	39

WLP: ADD - Initialization

c_{ij}	1	2	3	4	5	6	7	f_i	c_i	$f_i + c_i$
1	1	2	10	9	6	7	3	5	38	43
2	2	9	0	7	3	6	10	7	37	44
3	7	6	1	5	3	10	5	5	37	42
4	6	5	10	2	6	3	6	6	38	44
5	6	4	6	3	7	2	6	5	34	39

first location $k = 5$ with $Z := c_5 + f_5 = 39$, $I_1 = \{5\}$, $I_0^{pot} = \{1, 2, 3, 4\}$

WLP: ADD - Initialization

c_{ij}	1	2	3	4	5	6	7	f_i	c_i	$f_i + c_i$
1	1	2	10	9	6	7	3	5	38	43
2	2	9	0	7	3	6	10	7	37	44
3	7	6	1	5	3	10	5	5	37	42
4	6	5	10	2	6	3	6	6	38	44
5	6	4	6	3	7	2	6	5	34	39

first location $k = 5$ with $Z := c_5 + f_5 = 39$, $I_1 = \{5\}$, $I_0^{pot} = \{1, 2, 3, 4\}$

ω_{ij}	1	2	3	4	5	6	7	ω_i	f_i	diff
1	5	2			1		3	11	5	6
2	4		6		4			14	7	7
3			5		4		1	10	5	5
4				1	1			2	6	-4

WLP: construction heuristic ADD

- **Iteration step**
  - in each iteration exactly one potential location $k \in I_0^{pot}$ becomes part of the set I_1; it is the one with the largest value for $\omega_k - f_k$
  - set $I_1 = I_1 \cup \{k\}$, $I_0^{pot} = I_0^{pot} \setminus \{k\}$ and $Z = Z - \omega_k + f_k$
  - all locations i with a negative value for $\omega_i - f_i$ (fixed costs are greater than the savings in transportation costs) can be definitely forbidden: for all $i \in I_0^{pot}$ with $\omega_i \leq f_i \rightarrow$ $I_0 = I_0 \cup \{i\}$ and $I_0^{pot} = I_0 \setminus \{i\}$
  - for each $i \in I_0^{pot}$ compute $\omega_{ij} = \max\{\omega_{ij} - \omega_{kj}, 0\}$
- **Termination criterion** The procedure ends as soon as $I_0^{pot} = \{\}$.

Result Build a warehouse at all locations $\in I_1$ and assign each customer j to the warehouse $h \in I_1$ for which $h = \arg\min_{i \in I_1}\{c_{ij}\}$

WLP: ADD - Example cont.

ω_{ij}	1	2	3	4	5	6	7	ω_i	f_i	diff
1	5	2			1		3	11	5	6
2	4		6		4			14	7	7
3			5		4		1	10	5	5
4				1	1			2	6	-4

Iteration 1

build $k = 2$ and forbid $i = 4$ ($\omega_4 < f_4$)

Z = 39 - 14 + 7 = 32

$I_0^{pot} = \{1, 3\}$, $I_1 = \{2, 5\}$, $I_0 = \{4\}$

The new ω_{ij} matrix ($\omega_{ij} = \max\{\omega_{ij} - \omega_{kj}, 0\}$):

WLP: ADD - Example cont.

ω_{ij}	1	2	3	4	5	6	7	ω_i	f_i	diff
1	5	2			1		3	11	5	6
2	4		6		4			14	7	7
3			5		4		1	10	5	5
4				1	1			2	6	-4

Iteration 1

build $k = 2$ and forbid $i = 4$ ($\omega_4 < f_4$)

Z = 39 - 14 + 7 = 32

$I_0^{pot} = \{1, 3\}$, $I_1 = \{2, 5\}$, $I_0 = \{4\}$

The new ω_{ij} matrix ($\omega_{ij} = \max\{\omega_{ij} - \omega_{kj}, 0\}$):

ω_{ij}	1	2	3	4	5	6	7	ω_i	f_i	diff
1	1	2					3	6	5	1
3							1	1	5	-4

WLP: ADD - Example cont.

ω_{ij}	1	2	3	4	5	6	7	ω_i	f_i	diff
1	1	2					3	6	5	1
3							1	1	5	-4

WLP: ADD - Example cont.

ω_{ij}	1	2	3	4	5	6	7	ω_i	f_i	diff
1	1	2					3	6	5	1
3							1	1	5	-4

Iteration 2

build $k = 1$ and forbid $i = 3$ $(\omega_3 < f_3)$

Z = 32 - 6 + 5= 31

$I_0^{pot} = \{\}$, $I_1 = \{1, 2, 5\}$, $I_0 = \{3, 4\}$

WLP: ADD - Example cont.

ω_{ij}	1	2	3	4	5	6	7	ω_i	f_i	diff
1	1	2					3	6	5	1
3							1	1	5	-4

Iteration 2

build $k = 1$ and forbid $i = 3$ ($\omega_3 < f_3$)

Z = 32 - 6 + 5= 31

$I_0^{pot} = \{\}$, $I_1 = \{1, 2, 5\}$, $I_0 = \{3, 4\}$

Result

Z = 31 and warehouses at locations 1,2, and 5 will be built

Customers 1,2,7 will be served by warehouse 1

Customers 3,5 by warehouse 2

Customers 4,6 by warehouse 5

WLP: Construction heuristic DROP

Notation

As before: $I = \{1, \dots, m\}$, I_0, I_1, Z plus

I_1^{pot} set of all potentially included locations

WLP: Construction heuristic DROP

Notation

As before: $I = \{1, \ldots, m\}$, I_0, I_1, Z plus

I_1^{pot} set of all potentially included locations

The DROP heuristic is the opposite approach to the ADD method: we start from a solution where all locations are realized.

WLP: Construction heuristic DROP

Notation

As before: $I = \{1, \ldots, m\}$, I_0, I_1, Z plus

I_1^{pot} set of all potentially included locations

The DROP heuristic is the opposite approach to the ADD method: we start from a solution where all locations are realized.

- **Initialization**
  - set $I_1^{pot} = I$, $I_0 = I_1 = \{\}$, $Z = \sum_{i=1}^{m} f_i + \sum_{j=1}^{m} \min_{i \in I_1^{pot}} c_{ij}$
- **Iteration step**
  - Forbid exactly one potential location from the set I_1^{pot} (based on total costs - select the location that causes the largest decrease in the total costs)
  - Move all locations which would cause a cost increase if forbidden from the set I_1^{pot} to the set I_1.

WLP: DROP - Example

Initialization $I_1^{pot} = \{1, 2, 3, 4, 5\}$

For each column $j = \{1, \ldots, n\}$ identify the smallest cost element $c_{h_1 j}$ and the second smallest cost element $c_{h_2 j}$

c_{ij}	1	2	3	4	5	6	7	δ_i	f_i	decr.
1	1	2	10	9	6	7	3	5	5	0
2	2	9	0	7	3	6	10	1	7	6
3	7	6	1	5	3	10	5	0	5	5
4	6	5	10	2	6	3	6	1	6	5
5	6	4	6	3	7	2	6	1	5	4

$c_{h_1 j}$		1	2	0	2	3	2	3
$c_{h_2 j}$		2	4	1	3	3	3	5
h_1		1	1	2	4	2	5	1
h_2		2	5	3	5	3	4	3

δ_i...sum of differences between $c_{h_2 j} - c_{h_1 j}$ if i is h_1 for j - transportation cost increase if i is forbidden.

WLP: DROP - Example

Iteration 1

realize warehouse 1 ($I_1 = \{1\}$, $I_1^{pot} = I_1^{pot} \setminus \{1\}$) because $\delta_i = f_i$
forbid location 2 ($I_0 = \{2\}$, $I_1^{pot} = I_1^{pot} \setminus \{2\}$) because it leads to the largest cost reduction $f_i - \delta_i$.

Iteration 2

$I_1^{pot} = \{3, 4, 5\}$, $I_1 = \{1\}$, $I_0 = \{2\}$

ij	1	2	3	4	5	6	7	δ_i	f_i	decr.
1	1	2	10	9	6	7	3	-	-	-
3	7	6	1	5	3	10	5	8	5	-3
4	6	5	10	2	6	3	6	1	6	5
5	6	4	6	3	7	2	6	1	5	4

$c_{h_1 j}$	1	2	1	2	3	2	3
$c_{h_2 j}$	6	4	6	3	6	3	5

h_1	1	1	3	4	3	5	1
h_2	4	5	5	5	1	4	3

warehouse 3 is realized, location 4 is forbidden.

WLP: DROP - Example

Iteration 3

$I_1^{pot} = \{5\}, \; I_1 = \{1,3\}, \; I_0 = \{2,4\}$

ij	1	2	3	4	5	6	7	δ_i	f_i	decr.
1	1	2	10	9	6	7	3	-	-	-
3	7	6	1	5	3	10	5	-	-	-
5	6	4	6	3	7	2	6	7	5	-2

$c_{h_1 j}$	1	2	1	3	3	2	3
$c_{h_2 j}$	6	4	6	5	6	7	5

h_1	1	1	3	5	3	5	1
h_2	5	5	5	5	1	1	3

Warehouse 5 will be built (forbidding it would lead to an increase in total cost of 2)

WLP: DROP - Example

Result

ij	1	2	3	4	5	6	7	f_i
1	1	2	10	9	6	7	3	5
3	7	6	1	5	3	10	5	5
5	6	4	6	3	7	2	6	5

Locations $I_1 = \{1, 3, 5\}$ are built.
Customers 1,2,7 are supplied by warehouse 1
Customers 3,5 by warehouse 3
Customers 4,6 by warehouse 5

Z = 30 (slightly better result than from ADD algorithm)

WLP: improvement methods

- exchange one location from set I_1 with one from the set I_0
  - swap those two that lead to the largest cost savings (best improvement)
  - swap the first two that lead to a cost decrease (first improvement)
- combine ADD and DROP
  - use the DROP-rules to forbid the location that leads to the largest cost reduction (or the smallest cost increase) then add customer locations using the ADD-algorithm until no additional cost reduction is possible.
  - use the ADD-algorithm to add the location that leads to the largest cost reduction (or with the smallest cost increase) and then use the DROP-algorithm to remove locations until no additional cost reduction is possible.

The design of transportation networks: the transportation problem (TP)

capacity restrictions at the production sites

The per unit transportation cost matrix:

	Customer				
production site	V1	V2	V3	V4	capacity
F1	10	5	6	11	25
F2	1	2	7	4	25
F3	9	1	4	8	50
demand	15	20	30	35	$\sum$ 100

total capacity = total demand!

TP: Model formulation

m	number of productions sites, $i \in \{1, \ldots, m\}$
s_i	supply of production site i
n	number of customers, $j \in \{1, \ldots, n\}$
d_i	demand of customer i
c_{ij}	per unit transportation cost from i to j
x_{ij}	decision variable: amount of demand of i supplied by j

TP: Model formulation

m	number of productions sites, $i \in \{1, \ldots, m\}$
s_i	supply of production site i
n	number of customers, $j \in \{1, \ldots, n\}$
d_i	demand of customer i
c_{ij}	per unit transportation cost from i to j
x_{ij}	decision variable: amount of demand of i supplied by j

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow \min \tag{6}$$

$$\sum_{j=1}^{n} x_{ij} = s_i \qquad \forall i \in \{1, \ldots, m\} \qquad (u_i) \tag{7}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad \forall j \in \{1, \ldots, n\} \qquad (v_j) \tag{8}$$

$$x_{ij} \geq 0 \qquad i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\} \tag{9}$$

supply should equal demand: $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$

TP: solution methods

Solution methods

- **heuristic**
  - Vogel's approximation method
  - Northwest corner rule
  - Column minima method
- **exact**
  - find a basic solution heuristically
  - apply transportation simplex method

TP: Vogel's approximation

1. for each row and column remaining under consideration, calculate its **difference** (= the arithmetic difference between and smallest and next-to-the-smallest unit costs c_{ij} still remaining in that row or column; if two tie, then the difference is 0)
2. in the row or column having the **largest difference** select the variable with the smallest remaining unit cost. (ties may be broken arbitrarily) Transport as much as possible between the production site and the customer of the selected variable.
3. if a resource of either row or column is fully used, eliminate the according row or column.
4. if there remains only one row or column, fill all still remaining cells of this row or column with the required amounts. Otherwise proceed with step 1.

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply
1	10	5	6	11	25
2	1	2	7	4	25
3	9	1	4	8	50
demand	15	20	30	35	100

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1	2	7	4	25	1
3	9	1	4	8	50	3
demand	15	20	30	35	100	
	8	1	2	4		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1 15	2	7	4	25	1
3	9	1	4	8	50	3
demand	15	20	30	35	100	
	8	1	2	4		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1 15	2	7	4	25	1
3	9	1	4	8	50	3
demand	15	20	30	35	100	
	8	1	2	4		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1 15	2	7	4	~~25~~ 10	~~1~~ 2
3	9	1	4	8	50	3
demand	15	20	30	35	100	
	8	1	2	4		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1	4	8	50	3
demand	15	20	30	35	100	
	8	1	2	4		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1	4	8	50	3
demand	15	20	30	35	100	
	8	1	2	4		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1	4	8	50	3
demand	15	20	30	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1 20	4	8	50	3
demand	15	20	30	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	1
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1 20	4	8	50	3
demand	15	20	30	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6	11	25	~~1~~ 5
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1 20	4	8	~~50~~ 30	~~3~~ 4
demand	15	20	30	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6 25	11	25	~~1~~ 5
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1 20	4	8	~~50~~ 30	~~3~~ 4
demand	15	20	30	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6 25	11	25	~~1~~ 5
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1 20	4	8	~~50~~ 30	~~3~~ 4
demand	15	20	30	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6 25	11	25	~~1~~ 5
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1 20	4	8	~~50~~ 30	~~3~~ 4
demand	15	20	~~30~~ 5	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6 25	11	25	~~1~~ 5
2	1 15	2	7	4 10	~~25~~ 10	~~1~~ 2
3	9	1 20	4 5	8 25	~~50~~ 30	~~3~~ 4
demand	15	20	~~30~~ 5	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

TP: Vogel's approximation - Example

i,j	1	2	3	4	supply	
1	10	5	6 / 25	11	25	~~1~~ 5
2	1 / 15	2	7	4 / 10	~~25~~ 10	~~1~~ 2
3	9	1 / 20	4 / 5	8 / 25	~~50~~ 30	~~3~~ 4
demand	15	20	~~30~~ 5	~~35~~ 25	100	
	8	~~1~~ 4	2	~~4~~ 3		

Total costs $=$ 15*1+20*1+25*6+5*4+10*4+25*8 $=$ 445

TP: Northwest corner rule

usually used to generate a starting solution for the transportation simplex method; the occupied cells correspond to a basic solution

The table is filled from the north-west corner.

- in each iteration, one cell is filled: the maximum possible value is entered, such that the complete resource of either the row or the column is consumed.
- in the case where the complete column resource is consumed, we move to the right
- in the case where the complete row resource is consumed, we move down

TP: Northwest corner rule

usually used to generate a starting solution for the transportation simplex method; the occupied cells correspond to a basic solution

The table is filled from the north-west corner.

- in each iteration, one cell is filled: the maximum possible value is entered, such that the complete resource of either the row or the column is consumed.
- in the case where the complete column resource is consumed, we move to the right
- in the case where the complete row resource is consumed, we move down

Result since supply = demand there always exists a feasible solution; exactly $m + n - 1$ occupied cells (= basic variables) x_{ij} are identified; the remaining $m * n - (m + n - 1)$ variables take value 0; they are non basic variables (NBV)

for explanation see Hillier and Lieberman (1995) 'Introduction to Operations Research' page 318.

TP: Northwest corner rule - Example 1

i,j	1	2	3	4	s_i
1					25
2					25
3					50
d_j	15	20	30	35	100

TP: Northwest corner rule - Example 1

i,j	1	2	3	4	s_i
1	15				25
2					25
3					50
d_j	15	20	30	35	100

TP: Northwest corner rule - Example 1

i,j	1	2	3	4	s_i
1	15	10			~~25~~
2					25
3					50
d_j	15	20	30	35	100

TP: Northwest corner rule - Example 1

i,j	1	2	3	4	s_i
1	15	10			~~25~~
2		10			25
3					50
d_j	~~15~~	~~20~~	30	35	100

TP: Northwest corner rule - Example 1

i,j	1	2	3	4	s_i
1	15	10			~~25~~
2		10	15		~~25~~
3					50
d_j	15	20	30	35	100

TP: Northwest corner rule - Example 1

i,j	1	2	3	4	s_i
1	15	10			25
2		10	15		25
3			15		50
d_j	15	20	30	35	100

TP: Northwest corner rule - Example 1

i,j	1	2	3	4	s_i
1	15	10			~~25~~
2		10	15		~~25~~
3			15	35	~~50~~
d_j	15	20	30	35	100

TP: Northwest corner rule - Example 2

sometimes it is necessary to move more than once to the right or down:

i,j	1	2	3	4	s_i
1	15	10	5		~~30~~
2			20		~~20~~
3			25	35	~~35~~
d_j	~~15~~	~~10~~	~~35~~	~~25~~	85

TP: Northwest corner rule - Example 2

sometimes it is necessary to move more than once to the right or down:

i,j	1	2	3	4	s_i
1	15	10	5		~~30~~
2			20		~~20~~
3			25	35	~~35~~
d_j	15	10	35	25	85

degeneracy

i,j	1	2	3	4	s_i
1	10	5			~~15~~
2		15	0		~~15~~
3			30	20	~~50~~
d_j	10	20	30	20	80

column or row could be deleted → we are only allowed to delete one (arbitrary selection)

TP: Northwest corner rule - Example 2

sometimes it is necessary to move more than once to the right or down:

i,j	1	2	3	4	s_i
1	15	10	5		~~30~~
2			20		~~20~~
3			25	35	~~35~~
d_j	~~15~~	~~10~~	~~35~~	~~25~~	85

degeneracy

i,j	1	2	3	4	s_i
1	10	5			~~15~~
2		15	0		~~15~~
3			30	20	~~50~~
d_j	~~10~~	~~20~~	~~30~~	~~20~~	80

column or row could be deleted $\rightarrow$ we are only allowed to delete one (arbitrary selection)

Advantage: very simple and fast

Disadvantage: it neglects the cost factors $\rightarrow$ bad starting solutions

TP: Column minima method

- in each iteration, in the most left (not yet deleted) column select the smallest not yet deleted c_{ij} value and determine the maximum possible value for x_{ij}
- in the case where the column resource is consumed $\rightarrow$ delete column j
- in the case where the row resource is consumed $\rightarrow$ delete row i

it is a greedy method
usually better results than the northwest corner rule

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	10	5	6	11	25
2	1	2	7	4	25
3	9	1	4	8	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	10	5	6	11	25
2	1 15	2	7	4	25
3	9	1	4	8	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	10	5	6	11	25
2	1 15	2	7	4	25
3	9	1	4	8	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	10	5	6	11	25
2	1 15	2	7	4	25
3	9	1 20	4	8	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	10	5	6	11	25
2	1 15	2	7	4	25
3	9	1 20	4	8	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	10	5	6	11	25
2	1 15	2	7	4	25
3	9	1 20	4 30	8	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	10	5	6	11	25
2	1 15	2	7	4	25
3	9	1 20	4 30	8	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	10	5	6	11	25
2	1 15	2	7	4	25
3	9	1 20	4 30	8 0	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	^{10}	^{5}	^{6}	^{11}	25
2	^{1} 15	^{2}	^{7}	^{4} 10	25
3	^{9}	^{1} 20	^{4} 30	^{8} 0	50
d_j	15	20	30	35	100

TP: Column minima method - Example

i,j	1	2	3	4	s_i
1	^{10}	^{5}	^{6}	^{11} 25	25
2	^{1} 15	^{2}	^{7}	^{4} 10	25
3	^{9}	^{1} 20	^{4} 30	^{8} 0	50
d_j	15	20	30	35	100

TP: Exact method - MODI, stepping stone
The transportation simplex method

TP: Exact method - MODI, stepping stone
The transportation simplex method

i,j	1	2	$\cdots$	n	s_i	u_i
1	c_{11}	c_{12}	$\cdots$	c_{1n}	s_1	u_1
2	c_{21}	c_{22}	$\cdots$	c_{2n}	s_2	u_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
m	c_{m1}	c_{m2}	$\cdots$	c_{mn}	s_m	u_m
d_j	d_1	d_2	$\cdots$	d_n		
v_j	v_1	v_2	$\cdots$	v_n		

Initialization
generate a basic feasible (BF) starting tableau with a starting heuristic

TP: The transportation simplex method

- **Iteration step**
  1. for the current BF solution calculate the values of the dual variables u_i and v_j using the following rule [MODI]:
     $c_{ij} = u_i + v_j$ **whenever** x_{ij} **is an occupied cell (BV)**
     Their values are not unique.
     Rule: set the dual variable to zero that corresponds to the row or column containing the most occupied cells.
  2. For all not occupied cells (NBV), compute $c_{ij} - u_i - v_j$.
  3. Identify the entering basic variable: it is the NBV with the most negative coefficient.
  4. Increase the entering basic variable and perform the **chain reaction**: change the other occupied cells. The BV that is the first to receive the value 0 is deleted [stepping stone].
- **Termination criterion** In case the coefficients of all NBV are non-negative, the optimal solution has been reached.

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	
2	1	2 10	7 15	4	25	
3	9	1	4 15	8 35	50	
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7 15	4	25	
3	9	1	4 15	8 35	50	
d_j	15	20	30	35		
v_j						

Iteration 1 compute the values of the dual variables u_i and v_j

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	^{10} 15	^{5} 10	^{6}	^{11}	25	0
2	^{1}	^{2} 10	^{7} 15	^{4}	25	
3	^{9}	^{1}	^{4} 15	^{8} 35	50	
d_j	15	20	30	35		
v_j	10					

Iteration 1 compute the values of the dual variables u_i and v_j

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7 15	4	25	
3	9	1	4 15	8 35	50	
d_j	15	20	30	35		
v_j	10	5				

Iteration 1 compute the values of the dual variables u_i and v_j

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7 15	4	25	-3
3	9	1	4 15	8 35	50	
d_j	15	20	30	35		
v_j	10	5				

Iteration 1 compute the values of the dual variables u_i and v_j

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7 15	4	25	-3
3	9	1	4 15	8 35	50	
d_j	15	20	30	35		
v_j	10	5	10			

Iteration 1 compute the values of the dual variables u_i and v_j

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	^{10} 15	^{5} 10	^{6}	^{11}	25	0
2	^{1}	^{2} 10	^{7} 15	^{4}	25	-3
3	^{9}	^{1}	^{4} 15	^{8} 35	50	-6
d_j	15	20	30	35		
v_j	10	5	10			

Iteration 1 compute the values of the dual variables u_i and v_j

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7 15	4	25	-3
3	9	1	4 15	8 35	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

Iteration 1 compute the values of the dual variables u_i and v_j

TP: The transportation simplex method (MODI/stepping stone) - Example

Initial solution: northwest corner rule

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 -4	11 -3	25	0
2	1 -6	2 10	7 15	4 -7	25	-3
3	9 5	1 2	4 15	8 35	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

Iteration 1 compute the values of the dual variables u_i and v_j
compute the coefficients $(c_{ij} - u_i - v_j)$ of the NBV (empty cells)

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

The total costs of the initial solution are Z = 10*15 + 5*10 + 2*10 + 7*15 + 4*15 + 8*35 = 665.

In order to check the correctness of our values, in each iteration the primal and the dual objective function values can be compared. They should be equal:

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} = \sum_{i=1}^{m} u_i s_i + \sum_{j=1}^{n} v_j d_j$$

The total costs of the dual solution are given by 25*0 + 25*(-3) + 50*(-6) + 15*10 + 20*5 + 30*10 + 35*14 = 665

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

The total costs of the initial solution are Z = 10*15 + 5*10 + 2*10 + 7*15 + 4*15 + 8*35 = 665.

In order to check the correctness of our values, in each iteration the primal and the dual objective function values can be compared. They should be equal:

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} = \sum_{i=1}^{m} u_i s_i + \sum_{j=1}^{n} v_j d_j$$

The total costs of the dual solution are given by 25*0 + 25*(-3) + 50*(-6) + 15*10 + 20*5 + 30*10 + 35*14 = 665

The NBV with the most negative coefficient is variable x_{24}. It will become the new basic variable.

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

The total costs of the initial solution are Z = 10*15 + 5*10 + 2*10 + 7*15 + 4*15 + 8*35 = 665.

In order to check the correctness of our values, in each iteration the primal and the dual objective function values can be compared. They should be equal:

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} = \sum_{i=1}^{m} u_i s_i + \sum_{j=1}^{n} v_j d_j$$

The total costs of the dual solution are given by 25*0 + 25*(-3) + 50*(-6) + 15*10 + 20*5 + 30*10 + 35*14 = 665

The NBV with the most negative coefficient is variable x_{24}. It will become the new basic variable.

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 -4	11 -3	25	0
2	1 -6	2 10	7 15	4 -7	25	-3
3	9 5	1 2	4 15	8 35	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 -4	11 -3	25	0
2	1 -6	2 10	7 15	4 -7	25	-3
3	9 5	1 2	4 15	8 35	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 -4	11 -3	25	0
2	1 -6	2 10	7 15	4 -7 $+\epsilon$	25	-3
3	9 5	1 2	4 15	8 35	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 -4	11 -3	25	0
2	1 -6	2 10	7 $15-\epsilon$	4 -7 $+\epsilon$	25	-3
3	9 5	1 2	4 $15+\epsilon$	8 $35-\epsilon$	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 -4	11 -3	25	0
2	1 -6	2 10	7 $15-\epsilon$	4 -7 $+\epsilon$	25	-3
3	9 5	1 2	4 $15+\epsilon$	8 $35-\epsilon$	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

because of x_{23} we set $\epsilon = 15$

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 -4	11 -3	25	0
2	1 -6	2 10	7 $15-\epsilon$	4 -7 $+\epsilon$	25	-3
3	9 5	1 2	4 $15+\epsilon$	8 $35-\epsilon$	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

because of x_{23} we set $\epsilon = 15$
Thus, x_{23} will become a NBV, $x_{24} = 15$, $x_{33} = 30$ and $x_{34} = 20$

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 1 (cont.)

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 -4	11 -3	25	0
2	1 -6	2 10	7 $15-\epsilon$	4 -7 $+\epsilon$	25	-3
3	9 5	1 2	4 $15+\epsilon$	8 $35-\epsilon$	50	-6
d_j	15	20	30	35		
v_j	10	5	10	14		

because of x_{23} we set $\epsilon = 15$
Thus, x_{23} will become a NBV, $x_{24} = 15$, $x_{33} = 30$ and $x_{34} = 20$

Z $=$ 665 $+$ ϵ*(-7) $=$ 665 - 7*15 $=$ 560

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	
2	1	2 10	7	4 15	25	
3	9	1	4 30	8 20	50	
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 / 15	5 / 10	6	11	25	0
2	1	2 / 10	7	4 / 15	25	
3	9	1	4 / 30	8 / 20	50	
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 / 15	5 / 10	6	11	25	0
2	1	2 / 10	7	4 / 15	25	
3	9	1	4 / 30	8 / 20	50	
d_j	15	20	30	35		
v_j	10					

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7	4 15	25	
3	9	1	4 30	8 20	50	
d_j	15	20	30	35		
v_j	10	5				

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7	4 15	25	-3
3	9	1	4 30	8 20	50	
d_j	15	20	30	35		
v_j	10	5				

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7	4 15	25	-3
3	9	1	4 30	8 20	50	
d_j	15	20	30	35		
v_j	10	5		7		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7	4 15	25	-3
3	9	1	4 30	8 20	50	1
d_j	15	20	30	35		
v_j	10	5		7		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6	11	25	0
2	1	2 10	7	4 15	25	-3
3	9	1	4 30	8 20	50	1
d_j	15	20	30	35		
v_j	10	5	3	7		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 3	11 4	25	0
2	1 -6	2 10	7 7	4 15	25	-3
3	9 -2	1 -5	4 30	8 20	50	1
d_j	15	20	30	35		
v_j	10	5	3	7		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 3	11 4	25	0
2	1 -6	2 10	7 7	4 15	25	-3
3	9 -2	1 -5	4 30	8 20	50	1
d_j	15	20	30	35		
v_j	10	5	3	7		

x_{21} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 15	5 10	6 3	11 4	25	0
2	1 -6 $+\epsilon$	2 10	7 7	4 15	25	-3
3	9 -2	1 -5	4 30	8 20	50	1
d_j	15	20	30	35		
v_j	10	5	3	7		

x_{21} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 $15-\epsilon$	5 $10+\epsilon$	6 3	11 4	25	0
2	1 -6 $+\epsilon$	2 $10-\epsilon$	7 7	4 15	25	-3
3	9 -2	1 -5	4 30	8 20	50	1
d_j	15	20	30	35		
v_j	10	5	3	7		

x_{21} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 / $15-\epsilon$	5 / $10+\epsilon$	6 / 3	11 / 4	25	0
2	1 / -6 / $+\epsilon$	2 / $10-\epsilon$	7 / 7	4 / 15	25	-3
3	9 / -2	1 / -5	4 / 30	8 / 20	50	1
d_j	15	20	30	35		
v_j	10	5	3	7		

x_{21} is the entering BV; because of x_{22} we set $\epsilon = 10$;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 $15-\epsilon$	5 $10+\epsilon$	6 3	11 4	25	0
2	1 -6 $+\epsilon$	2 $10-\epsilon$	7 7	4 15	25	-3
3	9 -2	1 -5	4 30	8 20	50	1
d_j	15	20	30	35		
v_j	10	5	3	7		

x_{21} is the entering BV; because of x_{22} we set $\epsilon = 10$; thus, x_{22} will become a NBV, $x_{21} = 10$, $x_{11} = 5$ and $x_{12} = 20$

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 2

i,j	1	2	3	4	s_i	u_i
1	10 / $15-\epsilon$	5 / $10+\epsilon$	6 / 3	11 / 4	25	0
2	1 / -6 / $+\epsilon$	2 / $10-\epsilon$	7 / 7	4 / 15	25	-3
3	9 / -2	1 / -5	4 / 30	8 / 20	50	1
d_j	15	20	30	35		
v_j	10	5	3	7		

x_{21} is the entering BV; because of x_{22} we set $\epsilon = 10$; thus, x_{22} will become a NBV, $x_{21} = 10$, $x_{11} = 5$ and $x_{12} = 20$

Z = 560 + ϵ*(-6) = 560 - 6*10 = 500

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 / 5	5 / 20	6	11	25	
2	1 / 10	2	7	4 / 15	25	
3	9	1	4 / 30	8 / 20	50	
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 / 5	5 / 20	6	11	25	0
2	1 / 10	2	7	4 / 15	25	
3	9	1	4 / 30	8 / 20	50	
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 / 5	5 / 20	6	11	25	0
2	1 / 10	2	7	4 / 15	25	
3	9	1	4 / 30	8 / 20	50	
d_j	15	20	30	35		
v_j	10					

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 5	5 20	6	11	25	0
2	1 10	2	7	4 15	25	
3	9	1	4 30	8 20	50	
d_j	15	20	30	35		
v_j	10	5				

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 / 5	5 / 20	6	11	25	0
2	1 / 10	2	7	4 / 15	25	-9
3	9	1	4 / 30	8 / 20	50	
d_j	15	20	30	35		
v_j	10	5				

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 / 5	5 / 20	6	11	25	0
2	1 / 10	2	7	4 / 15	25	-9
3	9	1	4 / 30	8 / 20	50	
d_j	15	20	30	35		
v_j	10	5		13		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 / 5	5 / 20	6	11	25	0
2	1 / 10	2	7	4 / 15	25	-9
3	9	1	4 / 30	8 / 20	50	-5
d_j	15	20	30	35		
v_j	10	5		13		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 5	5 20	6	11	25	0
2	1 10	2	7	4 15	25	-9
3	9	1	4 30	8 20	50	-5
d_j	15	20	30	35		
v_j	10	5	9	13		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 / 5	5 / 20	6 / -3	11 / -2	25	0
2	1 / 10	2 / 6	7 / 7	4 / 15	25	-9
3	9 / 4	1 / 1	4 / 30	8 / 20	50	-5
d_j	15	20	30	35		
v_j	10	5	9	13		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 / 5	5 / 20	6 / -3	11 / -2	25	0
2	1 / 10	2 / 6	7 / 7	4 / 15	25	-9
3	9 / 4	1 / 1	4 / 30	8 / 20	50	-5
d_j	15	20	30	35		
v_j	10	5	9	13		

x_{13} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 5	5 20	6 -3 $+\epsilon$	11 -2	25	0
2	1 10	2 6	7 7	4 15	25	-9
3	9 4	1 1	4 30	8 20	50	-5
d_j	15	20	30	35		
v_j	10	5	9	13		

x_{13} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 $5-\epsilon$	5 20	6 -3 $+\epsilon$	11 -2	25	0
2	1 $10+\epsilon$	2 6	7 7	4 $15-\epsilon$	25	-9
3	9 4	1 1	4 $30-\epsilon$	8 $20+\epsilon$	50	-5
d_j	15	20	30	35		
v_j	10	5	9	13		

x_{13} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 $5-\epsilon$	5 20	6 -3 $+\epsilon$	11 -2	25	0
2	1 $10+\epsilon$	2 6	7 7	4 $15-\epsilon$	25	-9
3	9 4	1 1	4 $30-\epsilon$	8 $20+\epsilon$	50	-5
d_j	15	20	30	35		
v_j	10	5	9	13		

x_{13} is the entering BV; because of x_{11} we set $\epsilon = 5$;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 $5-\epsilon$	5 20	6 -3 $+\epsilon$	11 -2	25	0
2	1 $10+\epsilon$	2 6	7 7	4 $15-\epsilon$	25	-9
3	9 4	1 1	4 $30-\epsilon$	8 $20+\epsilon$	50	-5
d_j	15	20	30	35		
v_j	10	5	9	13		

x_{13} is the entering BV; because of x_{11} we set $\epsilon = 5$; thus, x_{11} will become a NBV, $x_{21} = 15$, $x_{23} = 10$, $x_{33} = 25$, $x_{34} = 25$

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 3

i,j	1	2	3	4	s_i	u_i
1	10 $5-\epsilon$	5 20	6 -3 $+\epsilon$	11 -2	25	0
2	1 $10+\epsilon$	2 6	7 7	4 $15-\epsilon$	25	-9
3	9 4	1 1	4 $30-\epsilon$	8 $20+\epsilon$	50	-5
d_j	15	20	30	35		
v_j	10	5	9	13		

x_{13} is the entering BV; because of x_{11} we set $\epsilon = 5$;
thus, x_{11} will become a NBV, $x_{21} = 15$, $x_{23} = 10$, $x_{33} = 25$, $x_{34} = 25$

Z $= 500 + \epsilon$*(-3) $=$ 500 - 3*5 $=$ 485

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10	5 20	6 5	11	25	
2	1 15	2	7	4 10	25	
3	9	1	4 25	8 25	50	
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10	5 20	6 5	11	25	0
2	1 15	2	7	4 10	25	
3	9	1	4 25	8 25	50	
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10	5 20	6 5	11	25	0
2	1 15	2	7	4 10	25	
3	9	1	4 25	8 25	50	
d_j	15	20	30	35		
v_j		5				

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10	5 20	6 5	11	25	0
2	1 15	2	7	4 10	25	
3	9	1	4 25	8 25	50	
d_j	15	20	30	35		
v_j		5	6			

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10	5 20	6 5	11	25	0
2	1 15	2	7	4 10	25	
3	9	1	4 25	8 25	50	-2
d_j	15	20	30	35		
v_j		5	6			

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10	5 20	6 5	11	25	0
2	1 15	2	7	4 10	25	
3	9	1	4 25	8 25	50	-2
d_j	15	20	30	35		
v_j		5	6	10		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10	5 20	6 5	11	25	0
2	1 15	2	7	4 10	25	-6
3	9	1	4 25	8 25	50	-2
d_j	15	20	30	35		
v_j		5	6	10		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10	5 **20**	6 **5**	11	25	0
2	1 **15**	2	7	4 **10**	25	-6
3	9	1	4 **25**	8 **25**	50	-2
d_j	15	20	30	35		
v_j	7	5	6	10		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10 3	5 20	6 5	11 1	25	0
2	1 15	2 3	7 7	4 10	25	-6
3	9 4	1 -2	4 25	8 25	50	-2
d_j	15	20	30	35		
v_j	7	5	6	10		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10 3	5 20	6 5	11 1	25	0
2	1 15	2 3	7 7	4 10	25	-6
3	9 4	1 -2	4 25	8 25	50	-2
d_j	15	20	30	35		
v_j	7	5	6	10		

x_{32} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10 3	5 20	6 5	11 1	25	0
2	1 15	2 3	7 7	4 10	25	-6
3	9 4	1 -2 $+\epsilon$	4 25	8 25	50	-2
d_j	15	20	30	35		
v_j	7	5	6	10		

x_{32} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10 3	5 $20-\epsilon$	6 $5+\epsilon$	11 1	25	0
2	1 15	2 3	7 7	4 10	25	-6
3	9 4	1 -2 $+\epsilon$	4 $25-\epsilon$	8 25	50	-2
d_j	15	20	30	35		
v_j	7	5	6	10		

x_{32} is the entering BV;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10 3	5 20$-\epsilon$	6 5$+\epsilon$	11 1	25	0
2	1 15	2 3	7 7	4 10	25	-6
3	9 4	1 -2 $+\epsilon$	4 25$-\epsilon$	8 25	50	-2
d_j	15	20	30	35		
v_j	7	5	6	10		

x_{32} is the entering BV; because of x_{12} we set $\epsilon = 20$;

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10 3	5 $20-\epsilon$	6 $5+\epsilon$	11 1	25	0
2	1 15	2 3	7 7	4 10	25	-6
3	9 4	1 -2 $+\epsilon$	4 $25-\epsilon$	8 25	50	-2
d_j	15	20	30	35		
v_j	7	5	6	10		

x_{32} is the entering BV; because of x_{12} we set $\epsilon = 20$; thus, x_{12} will become a NBV, $x_{13} = 25$ and $x_{33} = 5$

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 4

i,j	1	2	3	4	s_i	u_i
1	10 3	5 $20-\epsilon$	6 $5+\epsilon$	11 1	25	0
2	1 15	2 3	7 7	4 10	25	-6
3	9 4	1 -2 $+\epsilon$	4 $25-\epsilon$	8 25	50	-2
d_j	15	20	30	35		
v_j	7	5	6	10		

x_{32} is the entering BV; because of x_{12} we set $\epsilon = 20$; thus, x_{12} will become a NBV, $x_{13} = 25$ and $x_{33} = 5$

Z = 485 + ϵ*(-2) = 485- 2*20 = 445

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10	5	6 25	11	25	
2	1 15	2	7	4 10	25	
3	9	1 20	4 5	8 25	50	
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10	5	6 25	11	25	
2	1 15	2	7	4 10	25	
3	9	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j						

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10	5	6 25	11	25	
2	1 15	2	7	4 10	25	
3	9	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j		1				

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10	5	6 25	11	25	
2	1 15	2	7	4 10	25	
3	9	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j		1	4			

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10	5	6 25	11	25	
2	1 15	2	7	4 10	25	
3	9	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j		1	4	8		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10	5	6 25	11	25	2
2	1 15	2	7	4 10	25	
3	9	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j		1	4	8		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10	5	6 25	11	25	2
2	1 15	2	7	4 10	25	-4
3	9	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j		1	4	8		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10	5	6 25	11	25	2
2	1 15	2	7	4 10	25	-4
3	9	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j	5	1	4	8		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10 3	5 2	6 25	11 1	25	2
2	1 15	2 5	7 7	4 10	25	-4
3	9 4	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j	5	1	4	8		

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10 3	5 2	6 25	11 1	25	2
2	1 15	2 5	7 7	4 10	25	-4
3	9 4	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j	5	1	4	8		

no negative coefficients
$\rightarrow$ **STOP optimal solution found**

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10 3	5 2	6 25	11 1	25	2
2	1 15	2 5	7 7	4 10	25	-4
3	9 4	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j	5	1	4	8		

no negative coefficients
$\rightarrow$ **STOP optimal solution found**
Z = 445

TP: The transportation simplex method (MODI/stepping stone) - Example

Iteration 5

i,j	1	2	3	4	s_i	u_i
1	10 3	5 2	6 25	11 1	25	2
2	1 15	2 5	7 7	4 10	25	-4
3	9 4	1 20	4 5	8 25	50	0
d_j	15	20	30	35		
v_j	5	1	4	8		

no negative coefficients
→ **STOP optimal solution found**
Z = 445
basic variables:
$x_{13} = 25$
$x_{21} = 15$
$x_{24} = 10$
$x_{32} = 20$
$x_{33} = 5$
$x_{34} = 25$

TP: Sensitivity analysis

The transportation problem is an LP with equality constraints (=). Therefore, the dual variables are unrestricted in sign.

From duality theory we can derive the following. A data change of the following form

$$s_i \to s_i + \Delta \text{ for an } i \text{ and}$$
$$d_j \to d_j + \Delta \text{ for a } j$$

(i.e. small changes in the right hand side RHS) does not change the values of the dual variables u_i and v_j (basis does not change, solution remains optimal). In this case, the objective function value is only changed by $\Delta(u_i + v_j)$:

$$Z \to Z + \Delta(u_i + v_j)$$

Obviously, s_i and d_j have to be changed **simultaneously**, otherwise total demand and total supply are no longer equal.

TP: Sensitivity analysis - Example

i,j	1	2	3	4	s_i	u_i
1	1 10	4	2 12	5	$22 + \Delta$	0
2	6	1 13	5	6 11	24	2
3	7	5	3 10	5 6	16	1
d_j	10	$13 + \Delta$	22	17		
v_j	1	-1	2	4		

The solution given is the optimal solution. (Z = 173)

What happens if the data is changed as follows:

$s_1 \rightarrow s_1 + \Delta$

$d_2 \rightarrow d_2 + \Delta$

TP: Sensitivity analysis - Example

i,j	1	2	3	4	s_i	u_i
1	1 10	4	2 12	5	$22+\Delta$	0
2	6	1 13	5	6 11	24	2
3	7	5	3 10	5 6	16	1
d_j	10	$13+\Delta$	22	17		
v_j	1	-1	2	4		

The solution given is the optimal solution. (Z $=$ 173)

What happens if the data is changed as follows:

$s_1 \to s_1 + \Delta$

$d_2 \to d_2 + \Delta$

The objective value of the optimal solution changes to Z $= 173 + \Delta(u_1 + v_2)$ $=$ 173 - Δ; this means that the **costs are reduced**, if the amount transported is increased. (This can happen in the case of negative u_i and v_j values; usually, the costs are more likely to increase.)

TP: Sensitivity analysis - Example

i,j	1	2	3	4	s_i	u_i
1	1 10	4	2 $12+\Delta$	5	$22+\Delta$	0
2	6	1 $13+\Delta$	5	6 $11-\Delta$	24	2
3	7	5	3 $10-\Delta$	5 $6+\Delta$	16	1
d_j	10	$13+\Delta$	22	17		
v_j	1	-1	2	4		

What's the maximum value Δ may take such that the current basis remains optimal?

TP: Sensitivity analysis - Example

i,j	1	2	3	4	s_i	u_i
1	1 10	4	2 12+Δ	5	$22+\Delta$	0
2	6	1 13+Δ	5	6 11-Δ	24	2
3	7	5	3 10-Δ	5 6+Δ	16	1
d_j	10	$13+\Delta$	22	17		
v_j	1	-1	2	4		

What's the maximum value Δ may take such that the current basis remains optimal?

This value can be identified in a similar way as in the stepping stone step (chain reaction).

TP: Sensitivity analysis - Example

i,j	1	2	3	4	s_i	u_i
1	1 10	4	2 $12+\Delta$	5	$22+\Delta$	0
2	6	1 $13+\Delta$	5	6 $11-\Delta$	24	2
3	7	5	3 $10-\Delta$	5 $6+\Delta$	16	1
d_j	10	$13+\Delta$	22	17		
v_j	1	-1	2	4		

What's the maximum value Δ may take such that the current basis remains optimal?

This value can be identified in a similar way as in the stepping stone step (chain reaction).

x_{33} is the first variable to become 0, if Δ increases: the **upper bound** is ≤ 10

TP: Sensitivity analysis - Example

i,j	1	2	3	4	s_i	u_i
1	1 10	4	2 12+Δ	5	$22+\Delta$	0
2	6	1 13+Δ	5	6 11-Δ	24	2
3	7	5	3 10-Δ	5 6+Δ	16	1
d_j	10	$13+\Delta$	22	17		
v_j	1	-1	2	4		

What's the maximum value Δ may take such that the current basis remains optimal?

This value can be identified in a similar way as in the stepping stone step (chain reaction).

x_{33} is the first variable to become 0, if Δ increases: the **upper bound** is ≤ 10
x_{34} is the first variable to become 0, if Δ decreases: the **lower bound** is ≥ -6

TP: Sensitivity analysis - Example

i,j	1	2	3	4	s_i	u_i
1	1 10	4	2 $12+\Delta$	5	$22+\Delta$	0
2	6	1 $13+\Delta$	5	6 $11-\Delta$	24	2
3	7	5	3 $10-\Delta$	5 $6+\Delta$	16	1
d_j	10	$13+\Delta$	22	17		
v_j	1	-1	2	4		

What's the maximum value Δ may take such that the current basis remains optimal?

This value can be identified in a similar way as in the stepping stone step (chain reaction).

x_{33} is the first variable to become 0, if Δ increases: the **upper bound** is ≤ 10
x_{34} is the first variable to become 0, if Δ decreases: the **lower bound** is ≥ -6

$-6 \leq \Delta \leq 10$

TP: Sensitivity analysis - Example

i,j	1	2	3	4	s_i	u_i
1	1 10	4	2 12+Δ	5	$22+\Delta$	0
2	6	1 13+Δ	5	6 11-Δ	24	2
3	7	5	3 10-Δ	5 6+Δ	16	1
d_j	10	$13+\Delta$	22	17		
v_j	1	-1	2	4		

What's the maximum value Δ may take such that the current basis remains optimal?

This value can be identified in a similar way as in the stepping stone step (chain reaction).

x_{33} is the first variable to become 0, if Δ increases: the **upper bound** is ≤ 10
x_{34} is the first variable to become 0, if Δ decreases: the **lower bound** is ≥ -6

$-6 \leq \Delta \leq 10$

Check for correctness: Z =

1*10+2*(12+Δ)+1*(13+Δ)+6*(11-Δ)+3*(10-Δ)+5*(6+Δ) = 173 - Δ

The capacitated warehouse location problem (CWLP)

The single-level CWLP differs from the uncapacitated WLP only in the assumption that

- the capacity on the potential locations $i = 1, \dots, m$ is bounded by $s_1, ..., s_m$ units (per period)
- the transportation costs c_{ij} are the costs **per transported unit**
- the demand of the customer is given with $d_1, ..., d_n$ units
- x_{ij} is the amount of goods transported from the warehouse at location i to customer j

CWLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{10}$$

$$\sum_{j=1}^{n} x_{ij} \leq s_i y_i \qquad \forall i \in \{1,\dots,m\} \tag{11}$$

$$x_{ij} \leq d_j y_i \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{12}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad \forall j \in \{1,\dots,n\} \tag{13}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\dots,m\} \tag{14}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{15}$$

CWLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{10}$$

$$\sum_{j=1}^{n} x_{ij} \leq s_i y_i \qquad \forall i \in \{1,\ldots,m\} \tag{11}$$

$$x_{ij} \leq d_j y_i \qquad \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{12}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad \forall j \in \{1,\ldots,n\} \tag{13}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\ldots,m\} \tag{14}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{15}$$

The objective function (10) minimizes total transportation and site costs.

CWLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{10}$$

$$\sum_{j=1}^{n} x_{ij} \leq s_i y_i \qquad \forall i \in \{1,\dots,m\} \tag{11}$$

$$x_{ij} \leq d_j y_i \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{12}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad \forall j \in \{1,\dots,n\} \tag{13}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\dots,m\} \tag{14}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{15}$$

The objective function (10) minimizes total transportation and site costs.
Constraints (11) ensure that a customer can only be served by a warehouse that is built; the total amount may not exceed the capacity.

CWLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{10}$$

$$\sum_{j=1}^{n} x_{ij} \leq s_i y_i \qquad \forall i \in \{1,\dots,m\} \tag{11}$$

$$x_{ij} \leq d_j y_i \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{12}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad \forall j \in \{1,\dots,n\} \tag{13}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\dots,m\} \tag{14}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{15}$$

The objective function (10) minimizes total transportation and site costs. Constraints (11) ensure that a customer can only be served by a warehouse that is built; the total amount may not exceed the capacity. Constraints (12) ensure that the quantity transported from i to j may not exceed the demand.

CWLP: MIP formulation

$$Z(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \quad \rightarrow \quad \min \tag{10}$$

$$\sum_{j=1}^{n} x_{ij} \leq s_i y_i \qquad \forall i \in \{1,\dots,m\} \tag{11}$$

$$x_{ij} \leq d_j y_i \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{12}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad \forall j \in \{1,\dots,n\} \tag{13}$$

$$y_i \in \{0,1\} \qquad \forall i \in \{1,\dots,m\} \tag{14}$$

$$x_{ij} \geq 0 \qquad \forall i \in \{1,\dots,m\}, j \in \{1,\dots,n\} \tag{15}$$

The objective function (10) minimizes total transportation and site costs. Constraints (11) ensure that a customer can only be served by a warehouse that is built; the total amount may not exceed the capacity. Constraints (12) ensure that the quantity transported from i to j may not exceed the demand. Constraints (13) ensure that the entire demand of customer j is delivered.

CWLP: ADD and DROP

Solution process:

- In general identical to uncapacitated problems
- When evaluating the solutions a small transportation problem has to be solved $\rightarrow$ in every iteration step a sequence of different transportation problems has to be solved.
- A dummy-customer/warehouse is introduced to balance excess capacity or missing capacities (e.g. at the beginning of the ADD-algorithm – its transportation costs are set to a big constant M).

CWLP: DROP for capacitated problems - Example

We have 4 possible locations with the capacities 20, 20, 10 and 10 and 4 customers with the demand 8, 9, 10, 11.
We introduce a dummy customer 5 with demand 22.
To work with smaller numbers, we reduce the costs by subtracting the row and column minimum.

i,j	1	2	3	4	5	s_i	f_i
1	8	3	5	4	0	20	10
2	1	2	3	4	0	20	10
3	6	5	7	3	0	10	7
4	8	4	7	5	0	10	7
d_j	8	9	10	11	22	60	

CWLP: DROP for capacitated problems - Example

We have 4 possible locations with the capacities 20, 20, 10 and 10 and 4 customers with the demand 8, 9, 10, 11.
We introduce a dummy customer 5 with demand 22.
To work with smaller numbers, we reduce the costs by subtracting the row and column minimum.

i,j	1	2	3	4	5	s_i	f_i
1	~~8~~ 7	~~3~~ 1	~~5~~ 2	~~4~~ 1	0	20	10
2	~~1~~ 0	~~2~~ 0	~~3~~ 0	~~4~~ 1	0	20	10
3	~~6~~ 5	~~5~~ 3	~~7~~ 4	~~3~~ 0	0	10	7
4	~~8~~ 7	~~4~~ 2	~~7~~ 4	~~5~~ 2	0	10	7
d_j	8	9	10	11	22	60	

Reduction constant $= 8*1+9*2+10*3+11*3 = 89$

CWLP: DROP for capacitated problems - Example

Initialization all locations are realized

$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} = 97, \sum_{i=1}^{m} f_i = 34$, Z = 131

i,j	1	2	3	4	5	s_i	f_i
1	7	1 7	2	1 1	0 12	20	10
2	0 8	0 2	0 10	1	0	20	10
3	5	3	4	0 10	0	10	7
4	7	2	4	2	0 10	10	7
d_j	8	9	10	11	22	60	

CWLP: DROP for capacitated problems - Example

Iteration step forbid one of the four locations in turn and solve a TP for each configuration

CWLP: DROP for capacitated problems - Example

Iteration step forbid one of the four locations in turn and solve a TP for each configuration

Forbid location 1? Z = 129 (improvement)

i,j	1	2	3	4	5	s_i	f_i
2	0 8	0 2	0 10	1	0	20	10
3	5	3	4	0 10	0	10	7
4	7	2 7	4	2 1	0 2	10	7
d_j	8	9	10	11	2	60	

CWLP: DROP for capacitated problems - Example

Iteration step forbid one of the four locations in turn and solve a TP for each configuration

Forbid location 2? $Z = 199$ (deterioration)

i,j	1	2	3	4	5	s_i	f_i
1	^{7}	$^{1}9$	$^{2}10$	$^{1}1$	^{0}	20	10
3	^{5}	^{3}	^{4}	$^{0}10$	^{0}	10	7
4	$^{7}8$	^{2}	^{4}	^{2}	$^{0}2$	10	7
d_j	8	9	10	11	2	60	

CWLP: DROP for capacitated problems - Example

Iteration step forbid one of the four locations in turn and solve a TP for each configuration

Forbid location 3? Z = 134 (deterioration)

i,j	1	2	3	4	5	s_i	f_i
1	^{7}	^{1}7	^{2}	^{1}11	^{0}2	20	10
2	^{0}8	^{0}2	^{0}10	^{1}	^{0}	20	10
4	^{7}	^{2}	^{4}	^{2}	^{0}10	10	7
d_j	8	9	10	11	12	60	

CWLP: DROP for capacitated problems - Example

Iteration step forbid one of the four locations in turn and solve a TP for each configuration

Forbid location 4? $Z = 124$ (improvement)

i,j	1	2	3	4	5	s_i	f_i
1	^{7}	$^{1}7$	^{2}	$^{1}11$	$^{0}2$	20	10
2	$^{0}8$	$^{0}2$	$^{0}10$	^{1}	^{0}	20	10
3	^{5}	^{3}	^{4}	^{0}	$^{0}10$	10	7
d_j	8	9	10	11	12	60	

CWLP: DROP for capacitated problems - Example

Result from Iteration 2:

Warehouses at locations 2 and 3 are build

Location 4 is forbidden

$I_0 = \{4\}$, $I_1 = \{2,3\}$, $I_1^{pot} = \{1\}$

Since location 1 cannot be forbidden, because of capacity reasons, the result of the DROP method is $I_1 = \{1,2,3\}$ with Z $= 124$.

Linear assignment problem (LAP)

The LAP is the fundamental optimization problem for internal location planning. It is related to the transportation problem.

Input

m	machines (activities, workers, **drivers**)
n	potential locations (dates, projects, **vehicles**)
c_{ij}	costs to build or run machine i at location j

Each driver has to be assigned to exactly one vehicle and to each vehicle at most one driver can be assigned. We want to minimize the total costs.

LAP - Example

3 drivers have to be assigned to 4 vehicles, the costs c_{ij} are given in the following matrix:

		Vehicles j			
	i,j	1	2	3	4
Drivers i	1	13	10	12	11
	2	15	M	13	20
	3	5	7	10	6

Driver 2 is not able to drive vehicle 2. Therefore the assignment costs are set to M (a large constant).

LAP - Example

If the number of vehicles $\neq$ the number of drivers we add **dummy rows** (drivers) or **dummy columns** (vehicles) with 0 cost.

		Vehicles j			
	i,j	1	2	3	4
Drivers i	1	13	10	12	11
	2	15	M	13	20
	3	5	7	10	6
Dummy	4	0	0	0	0

The vehicle the dummy driver is assigned to remains unused.

LAP - LP formulation

Decision variables

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

LAP - LP formulation

Decision variables

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

Formulation

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow \min \tag{16}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \in \{1, \ldots, n\} \tag{17}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j \in \{1, \ldots, n\} \tag{18}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \{1, \ldots, n\}, j \in \{1, \ldots, n\} \tag{19}$$

LAP - LP formulation

Decision variables

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

Formulation

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow \min \tag{16}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \in \{1, \ldots, n\} \tag{17}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j \in \{1, \ldots, n\} \tag{18}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \{1, \ldots, n\}, j \in \{1, \ldots, n\} \tag{19}$$

We want to minimize the total assignment costs such that every driver is assigned to exactly one vehicle and each vehicle receives exactly one driver.

LAP - Formulation as TP

- Compare the LP formulation of the TP and LAP!

LAP - Formulation as TP

- Compare the LP formulation of the TP and LAP!
- A LAP can be seen as a special case of the TP: each driver can be seen as a supplier with a capacity of 1 and each vehicle can be seen as a customer with a demand of 1.

LAP - Formulation as TP

- Compare the LP formulation of the TP and LAP!
- A LAP can be seen as a special case of the TP: each driver can be seen as a supplier with a capacity of 1 and each vehicle can be seen as a customer with a demand of 1.
- In general, the TP would allow non-integer x_{ij}. However, optimal solutions to the TP have the property that exactly n variables take value 1 while all other variables take value 0. Therefore, we obtain a feasible assignment.

LAP as TP - Example

i,j	1	2	3	4	s_i
1	13	10	12	11	1
2	15	M	13	20	1
3	5	7	10	6	1
4	0	0	0	0	1
d_j	1	1	1	1	

reduced cost matrix:
Subtract the smallest cost coefficient of each row/column from every cost coefficient of this row/column; the optimal solution remains the same but its cost change.

LAP as TP - Example

i,j	1	2	3	4	s_i	
1	13	10	12	11	1	-10
2	15	M	13	20	1	-13
3	5	7	10	6	1	-5
4	0	0	0	0	1	
d_j	1	1	1	1		

reduced cost matrix:
Subtract the smallest cost coefficient of each row/column from every cost coefficient of this row/column; the optimal solution remains the same but its cost change.

LAP as TP - Example

Reduced cost matrix

i,j	1	2	3	4	s_i
1	3	0	2	1	1
2	2	M	0	7	1
3	0	2	5	1	1
4	0	0	0	0	1
d_j	1	1	1	1	

LAP as TP - Example

Reduced cost matrix

i,j	1	2	3	4	s_i
1	3	0	2	1	1
2	2	M	0	7	1
3	0	2	5	1	1
4	0	0	0	0	1
d_j	1	1	1	1	

Column minima method gives the optimal solution

For the solution of larger problems, additional MODI-steps have to be performed, until the optimal solution is reached.

LAP: The Hungarian Method (Kuhn's algorithm)

- **Step 1** generate a **reduced matrix**, i.e. a matrix that contains at least one zero in each row and in each column.
- **Step 2** For the first assignment, choose a row having only one zero and box this zero; cross all other zeros of the column in which the boxed zero lies. Repeat this step for all rows containing a single zero. Then, repeat the same procedure for the columns.
- **Step 3** If each zero of the reduced matrix is either boxed or crossed, and each row and column contains exactly one boxed zero. The optimal solution has been found. Otherwise proceed with step 4.
- **Step 4** Draw a minimum number of horizontal and vertical lines such that all zeros are covered. (start with the row or column that contains the maximum number of zeros; ties can be broken arbitrarily)
- **Step 5** Identify the smallest value in all uncovered cells. Subtract this value from all values in uncovered cells and add this value to all entries in cells where two lines intersect. Proceed with step 2.

Source: Kasana and Kumar (2004) 'Introductory Operations Research'

LAP: The Hungarian Method (Kuhn's algorithm)

Step 4 Algorithm for finding the minimum number of horizontal and vertical lines:

1. Mark all rows, which contain no boxed 0.
2. Mark all columns, which contain one crossed 0 on a marked row.
3. Mark all rows, which contain a framed 0 on a marked column.
4. Repeat 2 and 3 until no additional column or row can be marked.
5. Mark with a line the non-marked rows and every marked column. All framed and crossed 0 should now be covered by at least one line.

LAP: The Hungarian Method - Example

Step 1 generate a **reduced matrix**

i,j	1	2	3	4	5
1	5	5	7	4	8
2	6	5	8	3	7
3	6	8	9	5	10
4	7	6	6	3	6
5	6	7	10	6	11

LAP: The Hungarian Method - Example

Step 1 generate a **reduced matrix**

i,j	1	2	3	4	5
1	5	5	7	4	8
2	6	5	8	3	7
3	6	8	9	5	10
4	7	6	6	3	6
5	6	7	10	6	11

i,j	1	2	3	4	5	
1	5	5	7	4	8	-4
2	6	5	8	3	7	-3
3	6	8	9	5	10	-5
4	7	6	6	3	6	-3
5	6	7	10	6	11	-6

Source: Kasana and Kumar (2004) 'Introductory Operations Research'

LAP: The Hungarian Method - Example

Step 1 generate a **reduced matrix**

i,j	1	2	3	4	5
1	5	5	7	4	8
2	6	5	8	3	7
3	6	8	9	5	10
4	7	6	6	3	6
5	6	7	10	6	11

i,j	1	2	3	4	5	
1	1	1	3	0	4	-4
2	3	2	5	0	4	-3
3	1	3	4	0	5	-5
4	4	3	3	0	3	-3
5	0	1	4	0	5	-6

Source: Kasana and Kumar (2004) 'Introductory Operations Research'

LAP: The Hungarian Method - Example

Step 1 generate a **reduced matrix**

i,j	1	2	3	4	5
1	5	5	7	4	8
2	6	5	8	3	7
3	6	8	9	5	10
4	7	6	6	3	6
5	6	7	10	6	11

i,j	1	2	3	4	5
1	1	1	3	0	4
2	3	2	5	0	4
3	1	3	4	0	5
4	4	3	3	0	3
5	0	1	4	0	5
		-1	-3		-3

Source: Kasana and Kumar (2004) 'Introductory Operations Research'

LAP: The Hungarian Method - Example

Step 1 generate a **reduced matrix**

i,j	1	2	3	4	5
1	5	5	7	4	8
2	6	5	8	3	7
3	6	8	9	5	10
4	7	6	6	3	6
5	6	7	10	6	11

i,j	1	2	3	4	5
1	1	0	0	0	1
2	3	1	2	0	1
3	1	2	1	0	2
4	4	2	0	0	0
5	0	0	1	0	2
		-1	-3		-3

Source: Kasana and Kumar (2004) 'Introductory Operations Research'

LAP: The Hungarian Method - Example

Step 1 generate a **reduced matrix**

i,j	1	2	3	4	5
1	5	5	7	4	8
2	6	5	8	3	7
3	6	8	9	5	10
4	7	6	6	3	6
5	6	7	10	6	11

i,j	1	2	3	4	5
1	1	0	0	0	1
2	3	1	2	0	1
3	1	2	1	0	2
4	4	2	0	0	0
5	0	0	1	0	2
		-1	-3		-3

cost reduction constant = 4+3+5+3+6+1+3+3 = 28

Source: Kasana and Kumar (2004) 'Introductory Operations Research'

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	0	0	0	1
2	3	1	2	0	1
3	1	2	1	0	2
4	4	2	0	0	0
5	0	0	1	0	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	0	0	0	1
2	3	1	2	0	1
3	1	2	1	0	2
4	4	2	0	0	0
5	0	0	1	0	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	0	0	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	0	$\cancel{0}$	0
5	0	0	1	$\cancel{0}$	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	0	0	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	0	$\cancel{0}$	0
5	$\boxed{0}$	0	1	$\cancel{0}$	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	0	0	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	0	$\cancel{0}$	0
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	0	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	0	$\cancel{0}$	0
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	0	$\cancel{0}$	0
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	$\boxed{0}$	$\cancel{0}$	0
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

Step 3 Every zero is either boxed or crossed but each row and column does not contain exactly one boxed zero $\rightarrow$ optimality not reached yet.

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	~~0~~	~~0~~	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	~~0~~	2
4	4	2	$\boxed{0}$	~~0~~	~~0~~
5	$\boxed{0}$	~~0~~	1	~~0~~	2

Step 3 Every zero is either boxed or crossed but each row and column does not contain exactly one boxed zero → optimality not reached yet.

Step 4 draw horizontal and vertical lines

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

Step 3 Every zero is either boxed or crossed but each row and column does not contain exactly one boxed zero $\rightarrow$ optimality not reached yet.

Step 4 draw horizontal and vertical lines

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

LAP: The Hungarian Method - Example

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

Step 3 Every zero is either boxed or crossed but each row and column does not contain exactly one boxed zero $\rightarrow$ optimality not reached yet.

Step 4 draw horizontal and vertical lines

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1
2	3	1	2	$\boxed{0}$	1
3	1	2	1	$\cancel{0}$	2
4	4	2	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$
5	$\boxed{0}$	$\cancel{0}$	1	$\cancel{0}$	2

Step 5 identify the smallest value of all uncovered cells: 1. Subtract this value from all uncovered cells and add it to cells at intersections of two lines.

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	~~0~~	1	1
2	2	~~0~~	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	~~0~~	1	1	2

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	1	1
2	2	$\cancel{0}$	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	$\boxed{0}$	$\cancel{0}$	1	1	2

$\rightarrow$ There is no row or column with a single 0. We will choose a row or column with 2 zeros. We decide to box cell (1,2).

$\rightarrow$Now there a row with a single 0.

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	1	1
2	2	$\cancel{0}$	1	0	0
3	$\cancel{0}$	1	0	0	1
4	4	2	0	1	0
5	$\boxed{0}$	$\cancel{0}$	1	1	2

$\rightarrow$ There is no row or column with a single 0. We will choose a row or column with 2 zeros. We decide to box cell (1,2).

$\rightarrow$Now there a row with a single 0.

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	1	1
2	2	$\cancel{0}$	1	$\boxed{0}$	0
3	$\cancel{0}$	1	0	0	1
4	4	2	0	1	0
5	$\boxed{0}$	$\cancel{0}$	1	1	2

$\rightarrow$ There is no row or column with a single 0. We will choose a row or column with 2 zeros. We decide to box cell (1,2).

$\rightarrow$Now there a row with a single 0.

$\rightarrow$There is again no row or column with a single 0. We choose a row with two zeros. We decide to box cell (2,4).

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	~~0~~	1	1
2	2	~~0~~	1	$\boxed{0}$	~~0~~
3	~~0~~	1	0	~~0~~	1
4	4	2	0	1	0
5	$\boxed{0}$	~~0~~	1	1	2

$\rightarrow$ There is no row or column with a single 0. We will choose a row or column with 2 zeros. We decide to box cell (1,2).

$\rightarrow$Now there a row with a single 0.

$\rightarrow$There is again no row or column with a single 0. We choose a row with two zeros. We decide to box cell (2,4).

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	$\cancel{0}$	1	1
2	2	$\cancel{0}$	1	$\boxed{0}$	$\cancel{0}$
3	$\cancel{0}$	1	0	$\cancel{0}$	1
4	4	2	0	1	$\boxed{0}$
5	$\boxed{0}$	$\cancel{0}$	1	1	2

→ There is no row or column with a single 0. We will choose a row or column with 2 zeros. We decide to box cell (1,2).
→Now there a row with a single 0.
→There is again no row or column with a single 0. We choose a row with two zeros. We decide to box cell (2,4).
→There is a single zero in column 5.

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	~~0~~	1	1
2	2	~~0~~	1	$\boxed{0}$	~~0~~
3	~~0~~	1	0	~~0~~	1
4	4	2	~~0~~	1	$\boxed{0}$
5	$\boxed{0}$	~~0~~	1	1	2

→ There is no row or column with a single 0. We will choose a row or column with 2 zeros. We decide to box cell (1,2).
→Now there a row with a single 0.
→There is again no row or column with a single 0. We choose a row with two zeros. We decide to box cell (2,4).
→There is a single zero in column 5.

LAP: The Hungarian Method - Example

Step 5 cont. Substract 1 from all uncovered cells and add it to cells at intersections of two lines.

i,j	1	2	3	4	5
1	1	0	0	1	1
2	2	0	1	0	0
3	0	1	0	0	1
4	4	2	0	1	0
5	0	0	1	1	2

Step 2 box and cross zeros

i,j	1	2	3	4	5
1	1	$\boxed{0}$	~~0~~	1	1
2	2	~~0~~	1	$\boxed{0}$	~~0~~
3	~~0~~	1	$\boxed{0}$	~~0~~	1
4	4	2	~~0~~	1	$\boxed{0}$
5	$\boxed{0}$	~~0~~	1	1	2

→ There is no row or column with a single 0. We will choose a row or column with 2 zeros. We decide to box cell (1,2).

→Now there a row with a single 0.

→There is again no row or column with a single 0. We choose a row with two zeros. We decide to box cell (2,4).

→There is a single zero in column 5.

LAP: The Hungarian Method - Example

i,j	1	2	3	4	5
1	1	0	$\cancel{0}$	1	1
2	2	$\cancel{0}$	1	0	$\cancel{0}$
3	$\cancel{0}$	1	0	$\cancel{0}$	1
4	4	2	$\cancel{0}$	1	0
5	0	$\cancel{0}$	1	1	2

Optimality reached. This problem has alternative optimal solutions.

Total costs $Z = 28 + 1 = 29$

Assignments: driver 1 on vehicle 2, driver 2 on vehicle 4, driver 3 on vehicle 3, driver 4 on vehicle 5, driver 5 on vehicle 1.

References

F. Hillier, G. Lieberman (1995) Introduction to Operations Research, McGraw-Hill. (In German published by Oldenbourg)

H.S. Kasana, K.D. Kumar (2004) Introductory Operations Research, Springer.