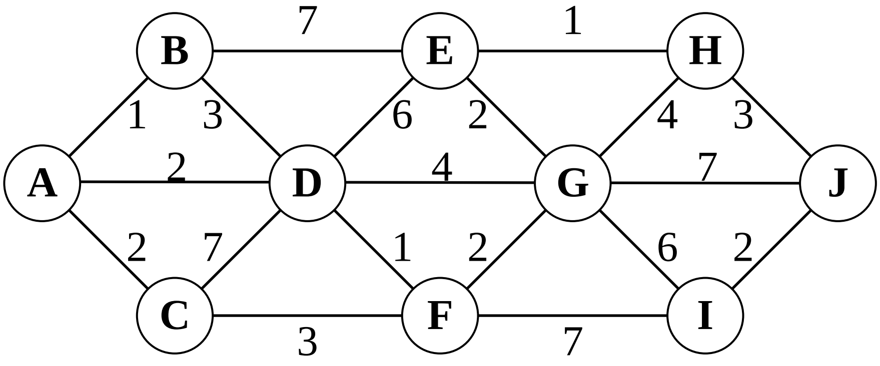
THE SHORTEST PATH

Problem: Find the shortest path from A to J

Idea: Starting at A track shortest accesses to all nodes one by one

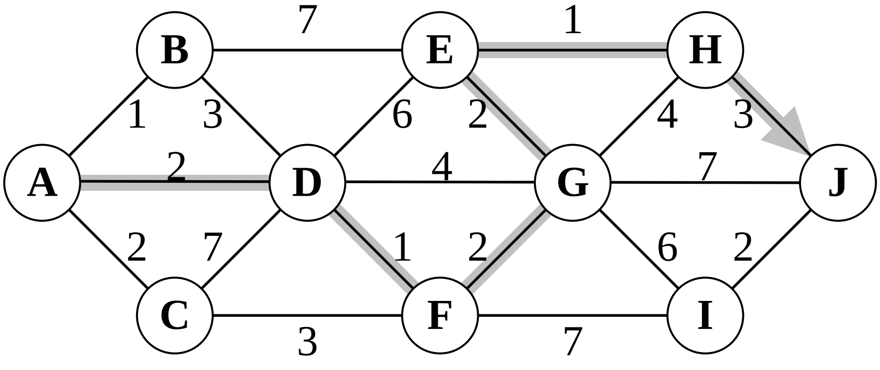


$$\left| \mathbf{P[A,J]} \right|_{\mathbf{min}} = \mathbf{?}$$

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
A		1	2	2						
B	1			3	7					
C	2			7		3				
D	2	3	7		6	1	4			
E		7		6			2	1		
F			3	1			2		7	
G				4	2	2		4	6	7
H					1		4			3
I						7	6			2
J							7	3	2	

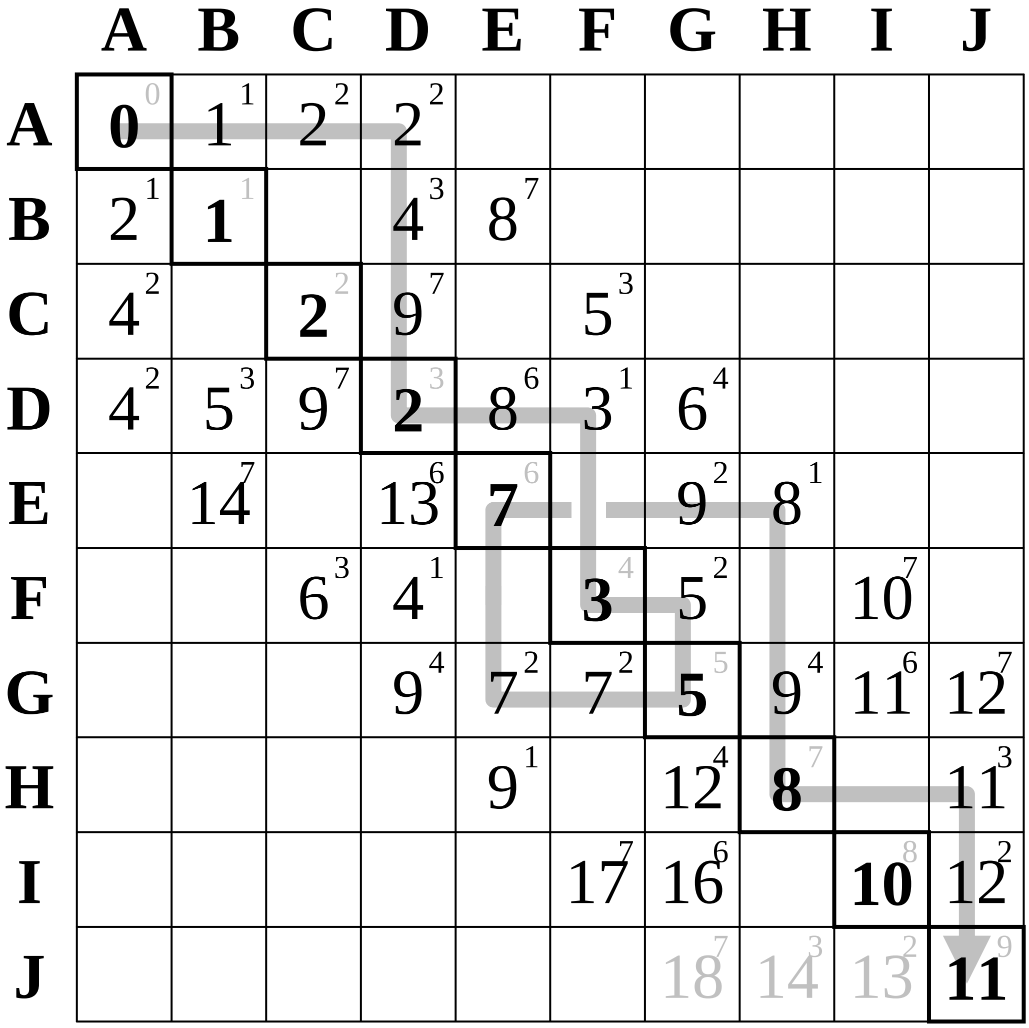
THE SHORTEST PATH

Reading values in diagonal as <u>implicite labels</u>
solution(s) can be traced back



$$P[A,J]_{min} = P\{A,D,F,G,E,H,J\}$$

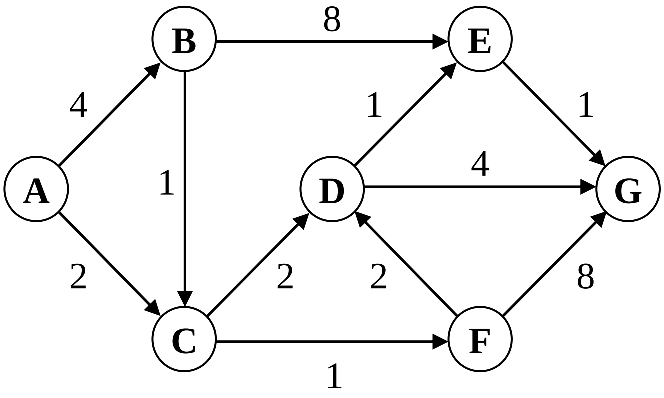
$$\left|P[A,J]\right|_{min} = 11$$



	A	B	C	D	E	F	G	H	I	J
A	$\mathbf{0}^{0}$	1^{1}	2^{2}	2^{2}						
B	2^{1}	$\mathbf{1}^{1}$		4^{3}	8^{7}					
C	4^{2}		$\mathbf{2}^{2}$	9^{7}		5^{3}				
D	4^{2}	5^{3}	9^{7}	$\mathbf{2}^{3}$	8^{6}	3^{1}	6^{4}			
E		14^{7}		13^{6}	$\mathbf{7}^{6}$		9^{2}	8^{1}		
F			6^{3}	4^{1}		$\mathbf{3}^{4}$	5^{2}		10^{7}	
G				9^{4}	7^{2}	7^{2}	$\mathbf{5}^{5}$	9^{4}	11^{6}	12^{7}
H					9^{1}		12^{4}	$\mathbf{8}^{7}$		11^{3}
I						17^{7}	16^{6}		$\mathbf{10}^{8}$	12^{2}
J							18^{7}	14^{3}	13^{2}	$\mathbf{11}^{9}$

THE LONGEST PATH

Problem: Find the longest path from A to G



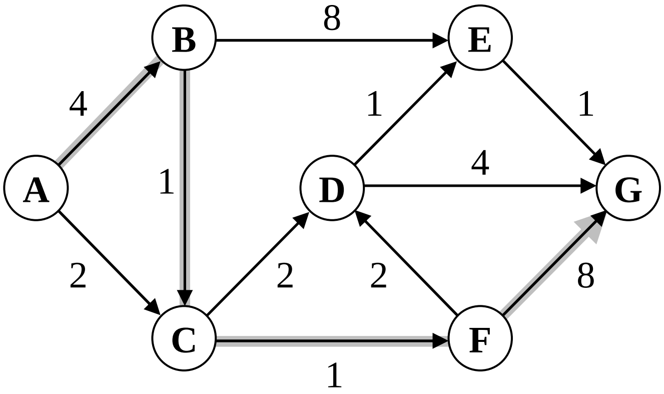
$$\left| \mathbf{P[A,G]} \right|_{\mathbf{max}} = ?$$

	A	B	C	D	E	F	G
A		4	2				
B			1		8		
C				2		1	
D					1		4
E							1
F				2			8
G							

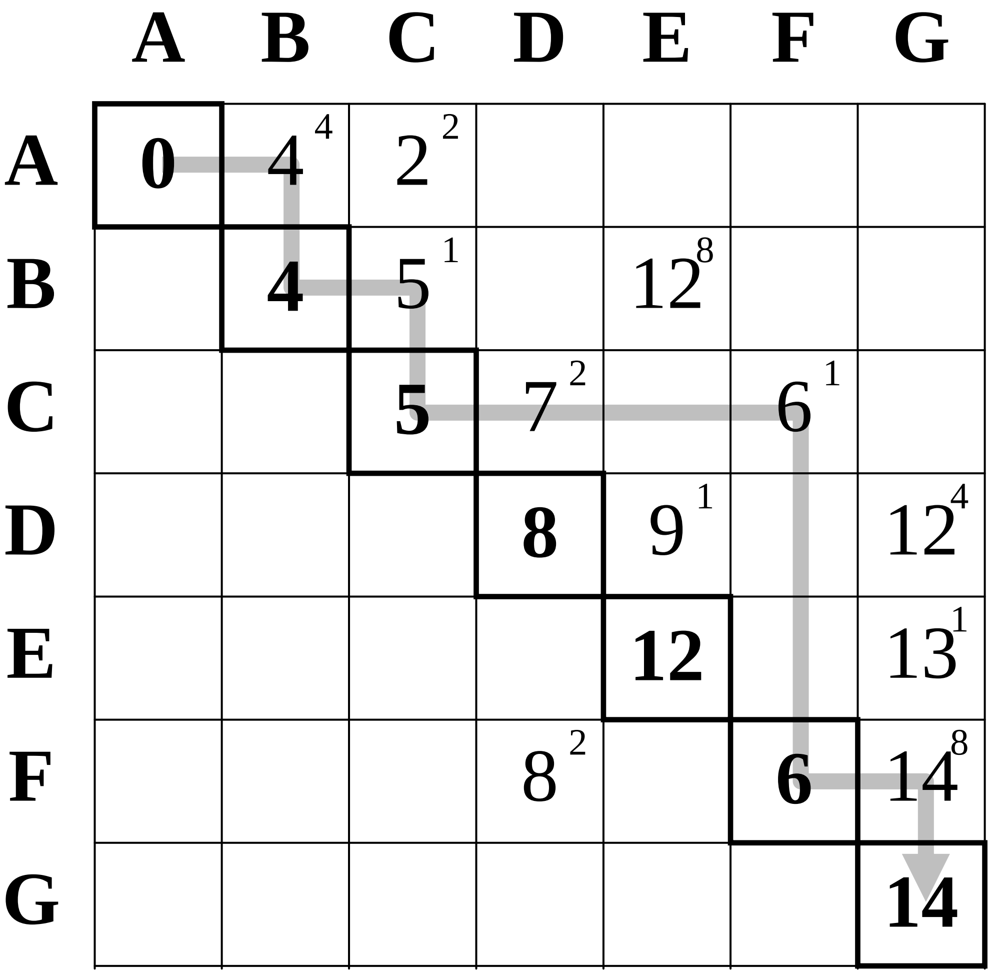
THE LONGEST PATH

Problem: Find the longest path from A to G

Idea: Starting at A track longest accesses to all nodes one by one



Reading values in diagonal as <u>implicite labels</u> solution(s) can be traced back (Calculations identify all longest paths)

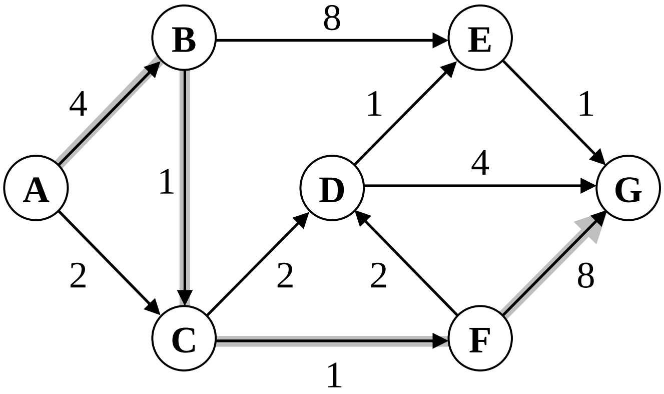


	A	B	C	D	E	F	G
A	**0**	4^{4}	2^{2}				
B		**4**	5^{1}		12^{8}		
C			**5**	7^{2}		6^{1}	
D				**8**	9^{1}		12^{4}
E					**12**		13^{1}
F				8^{2}		**6**	14^{8}
G							**14**

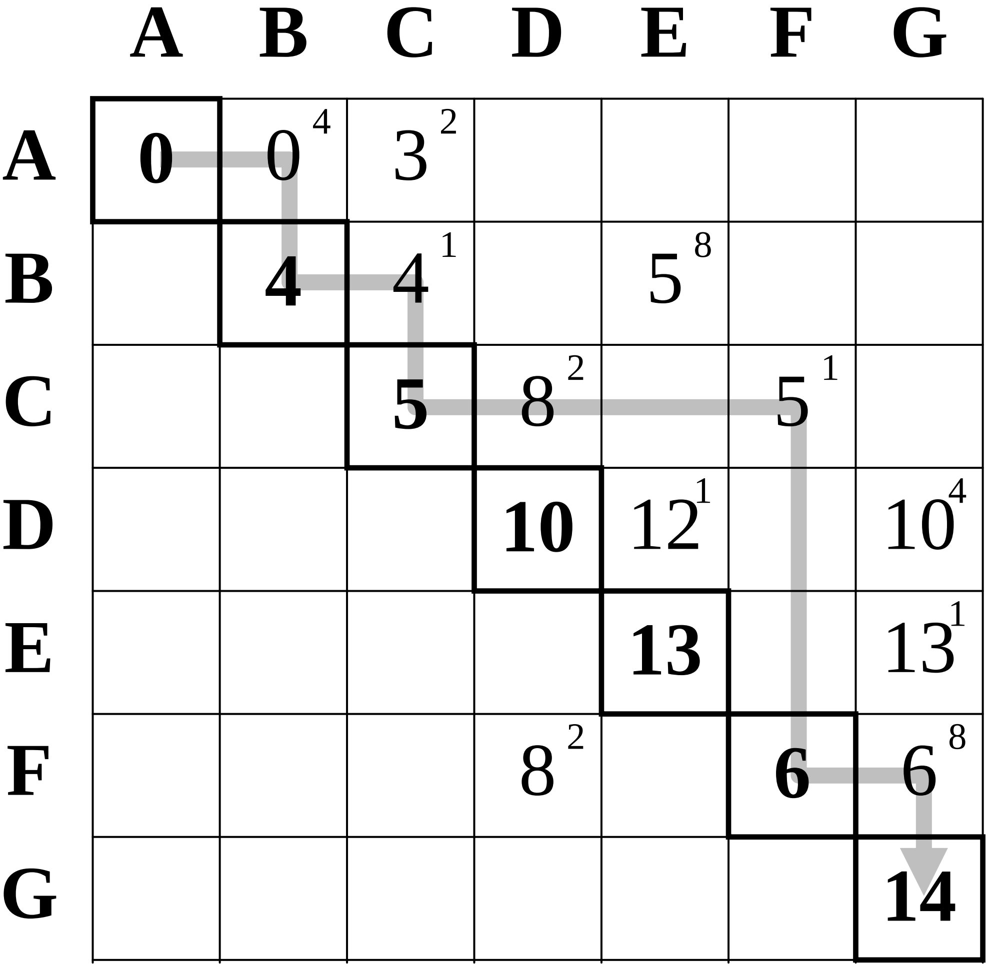
THE LONGEST PATH

Problem: Find the longest path from A to G – backward

Idea: Starting at G track longest accesses to all nodes one by one – in counter-arrow direction, and counting down



Reading values in diagonal as <u>implicite labels</u> solution(s) can be traced forward (Calculations identify the same longest path(s))

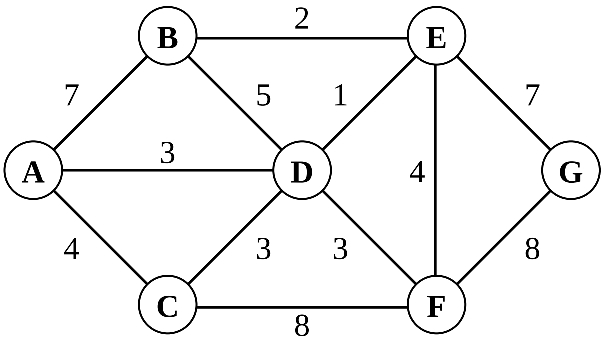


	A	B	C	D	E	F	G
A	**0**	0^{4}	3^{2}				
B		**4**	4^{1}		5^{8}		
C			**5**	8^{2}		5^{1}	
D				**10**	12^{1}		10^{4}
E					**13**		13^{1}
F				8^{2}		**6**	6^{8}
G							**14**

MAXIMUM FLOW / MINIMUM CUT

Problem: Determine the maximum flow from A to G

Idea: Recursive path-finding via edges of free capacities



$\Phi[A,G]_{max} = ?$

	A	B	C	D	E	F	G
A		7	4	3			
B	7			5	2		
C	4			3		8	
D	3	5	3		1	3	
E		2		1		4	7
F			8	3	4		8
G					7	8	

$\underline{\underline{A}}_0$: $\Delta\Phi_1 =$

	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

$\underline{\underline{A}}_1$: $\Delta\Phi_2 =$

	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

$\underline{\underline{A}}_2$: $\Delta\Phi_3 =$

	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

$\underline{\underline{A}}_3$: $\Delta\Phi_4 =$

	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

MAXIMUM FLOW / MINIMUM CUT

(maximum flow = minimum cut)

$\underline{\underline{A}}_4 : \quad \Delta\Phi_5 =$

	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

$\underline{\underline{A}}_5 : \quad \Delta\Phi_6 =$

	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

$\underline{\underline{A}}_6 : \quad \Delta\Phi_7 =$

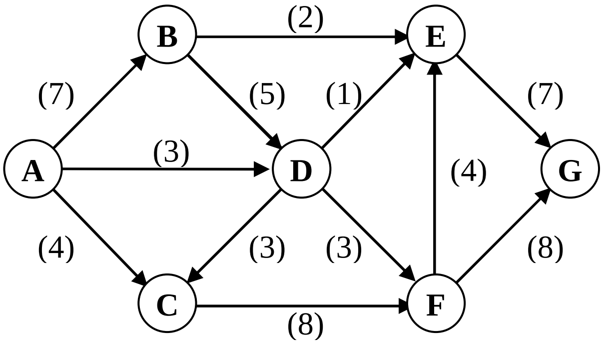
	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

$\underline{\underline{A}}_7 : \quad \Delta\Phi_8 =$

	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

$\underline{\underline{A}}_{\Phi[A,G]} = \underline{\underline{A}}_1 - \underline{\underline{A}}_n$

	A	B	C	D	E	F	G
A							
B							
C							
D							
E							
F							
G							

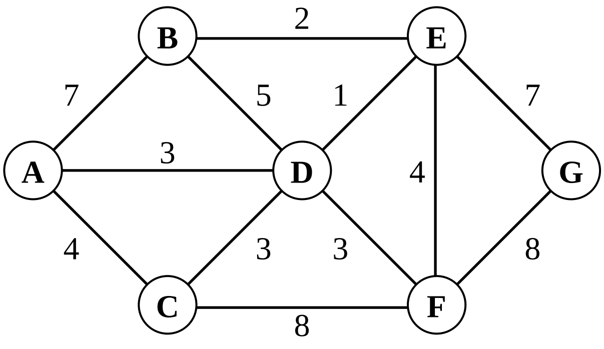


$\Phi[A,G]_{max} = \Sigma\Delta\Phi =$

MAXIMUM FLOW / MINIMUM CUT

Problem: Determine the maximum flow from A to G

Idea: Recursive path-finding via edges of free capacities



$\Phi[A,G]_{max} = ?$

	A	B	C	D	E	F	G
A		7	4	3			
B	7			5	2		
C	4			3		8	
D	3	5	3		1	3	
E		2		1		4	7
F			8	3	4		8
G					7	8	

$\underline{\underline{A}}_0$: $\Delta\Phi_1 = 4$

	A	B	C	D	E	F	G
A	+0	7	4	3			
B	7	+A		5	2		
C	4		+A	3		8	
D	3	5	3	-A	1	3	
E		2		1	-B	4	7
F			8	3	4	+C	8
G					7	8	-F

$\underline{\underline{A}}_1$: $\Delta\Phi_2 = 2$

	A	B	C	D	E	F	G
A	+0	7	0	3			
B	7	+A		5	2		
C	8		+D	3		4	
D	3	5	3	+A	1	3	
E		2		1	+B	4	7
F			12	3	4	-D	4
G					7	12	-E

$\underline{\underline{A}}_2$: $\Delta\Phi_3 = 1$

	A	B	C	D	E	F	G
A	+0	5	0	3			
B	9	+A		5	0		
C	8		+D	3		4	
D	3	5	3	+A	1	3	
E		4		1	+D	4	5
F			12	3	4	-D	4
G					9	12	-E

$\underline{\underline{A}}_3$: $\Delta\Phi_4 = 2$

	A	B	C	D	E	F	G
A	+0	5	0	2			
B	9	+A		5	0		
C	8		+D	3		4	
D	4	5	3	+A	0	3	
E		4		2	-F	4	4
F			12	3	4	+D	4
G					10	12	-F

MAXIMUM FLOW / MINIMUM CUT

(maximum flow = minimum cut)

$\underline{\underline{A}}_4$: $\Delta\Phi_5 = 1$

	A	B	C	D	E	F	G
A	+0	5	0	0			
B	9	+A		5	0		
C	8		+D	3		4	
D	6	5	3	+B	0	1	
E		4		2	-F	4	4
F			12	5	4	+D	2
G					10	14	-F

$\underline{\underline{A}}_5$: $\Delta\Phi_6 = 1$

	A	B	C	D	E	F	G
A	+0	4	0	0			
B	10	+A		4	0		
C	8		+D	3		4	
D	6	6	3	+B	0	0	
E		4		2	-F	4	4
F			12	6	4	+C	1
G					10	15	-F

$\underline{\underline{A}}_6$: $\Delta\Phi_7 = 2$

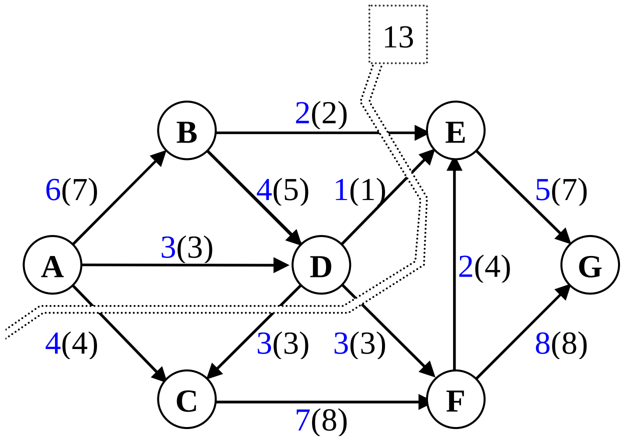
	A	B	C	D	E	F	G
A	+0	3	0	0			
B	11	+A		3	0		
C	8		+D	4		3	
D	6	7	2	+B	0	0	
E		4		2	-F	4	4
F			13	6	4	+C	0
G					10	16	-E

$\underline{\underline{A}}_7$: $\Delta\Phi_8 = 0$!

	A	B	C	D	E	F	G
A	+0	1	0	0			
B	13	+A		1	0		
C	8			6		1	
D	6	9	0	+B	0	0	
E		4		2		6	2
F			15	6	2		0
G					12	16	

$\underline{\underline{A}}_{\Phi[A,G]} = \underline{\underline{A}}_1 - \underline{\underline{A}}_7$

	A	B	C	D	E	F	G
A		6	4	3			
B	-6			4	2		
C	-4			-3		7	
D	-3	-4	3		1	3	
E		-2		-1		-2	5
F			-7	-3	2		8
G					-5	-8	

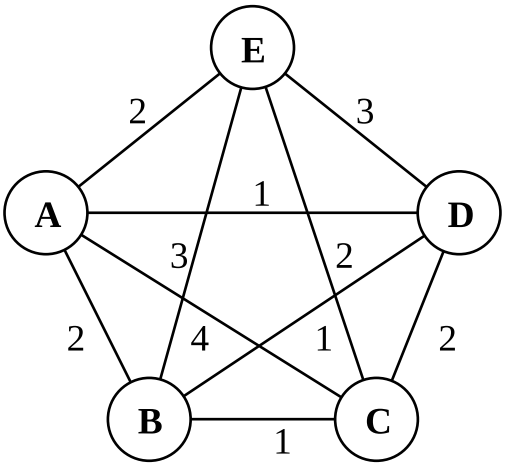


$\Phi[A,G]_{max} = \Sigma\Delta\Phi = 13$

THE SHORTEST LOOP

Problem: Find the shortest loop threading all the nodes

Idea: Try to assign the smallest elements of the rows and columns (one in each row and one in each column, forming one loop)



	A	B	C	D	E
A		2	4	1	2
B	2		1	1	3
C	4	1		2	2
D	1	1	2		3
E	2	3	2	3	

$0_{/r}$

	A	B	C	D	E
A					
B					
C					
D					
E					

$0_{/c}$

	A	B	C	D	E
A					
B					
C					
D					
E					

1.

	A	B	C	D	E
A					
B					
C					
D					
E					

2.

	A	B	C	D	E
A					
B					
C					
D					
E					

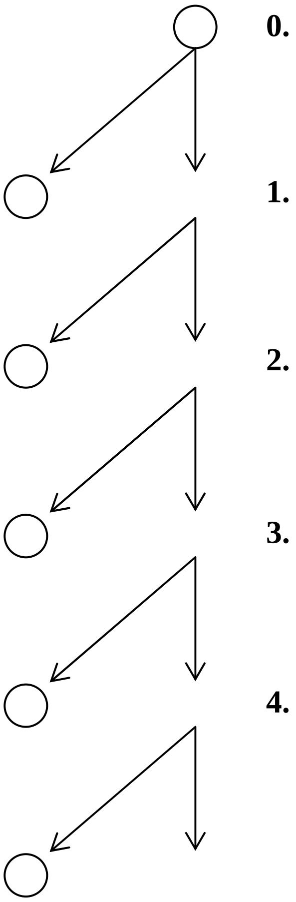
3.

	A	B	C	D	E
A					
B					
C					
D					
E					

4.

	A	B	C	D	E
A					
B					
C					
D					
E					

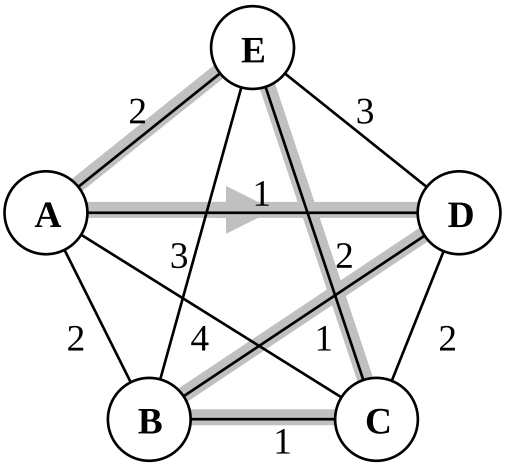
Decision Tree
Branch & Bound



THE SHORTEST LOOP

Problem: Find the shortest loop threading all the nodes

Idea: Try to assign the smallest elements of the rows and columns (one in each row and one in each column, forming one loop)



	A	B	C	D	E
A		2	4	**1**	2
B	2		**1**	1	3
C	4	1		2	**2**
D	1	**1**	2		3
E	**2**	3	2	3	

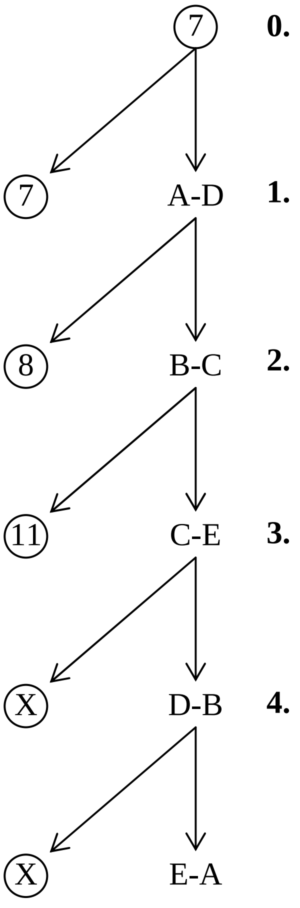
$0_{/r}$

	A	B	C	D	E	
A	X	2	4	1	2	-1
B	2	X	1	1	3	-1
C	4	1	X	2	2	-1
D	1	1	2	X	3	-1
E	2	3	2	3	X	-2
						(-6)

$0_{/c}$

	A	B	C	D	E	
A	X	1	3	0	1	
B	1	X	0	0	2	
C	3	0	X	1	1	
D	0	0	1	X	2	
E	0	1	0	1	X	
					-1	(-1)

Decision Tree
Branch & Bound



1.

	A	B	C	D	E
A	X	1	3	⓪^{0}	0^{0}
B	1	X	0^{0}	0^{0}	1
C	3	0^{0}	X	1	0^{0}
D	0^{0}	0^{0}	1	X	1
E	0^{0}	1	0^{0}	1	X

2.

	A	B	C	D	E
A	X	X	X	X	X
B	1	X	⓪^{1}	X	1
C	3	0^{0}	X	X	0^{1}
D	X	0^{1}	1	X	1
E	0^{1}	1	0^{0}	X	X

3.

	A	B	C	D	E
A	X	X	X	X	X
B	X	X	X	X	X
C	3	X	X	X	⓪^{4}
D	X	0^{2}	X	X	1
E	0^{4}	1	X	X	X

4.

	A	B	C	D	E
A	X	X	X	X	X
B	X	X	X	X	X
C	X	X	X	X	X
D	X	⓪	X	X	X
E	⓪	1	X	X	X

THE TRANSPORTATION PROBLEM

Problem: Do develope a maximum amount transportation program with minimum associating transportation cost

*Idea: Applying an LP primal-dual simplex algorithm**

c_{ij}					α_i
	1	2	2	4	180
	4	2	5	1	140
	6	3	1	2	120
β_j	150	80	60	90	

αi = maximum available capacity at source i.
βj = need at consumer j
C_{ij} = unit transportation cost from source i to consumer j

The job is:

$\Sigma_j(X_{ij}) \leq \alpha_i \ \forall i$ $\quad$ $\Sigma_i(X_{ij}) \leq \beta_j \ \forall j$
$\Sigma_i\Sigma_j(X_{ij}) = \max$ at $\Sigma_i\Sigma_j(X_{ij} \bullet C_{ij}) = \min$

THE TRANSPORTATION PROBLEM

Problem: Do develope a maximum amount transportation program with minimum associating transportation cost

*Idea: Applying an LP primal-dual simplex algorithm**

c_{ij}					α_i
	1	2	2	4	180
	4	2	5	1	140
	6	3	1	2	120
β_j	150	80	60	90	

αi = maximum available capacity at source i.
βj = need at consumer j
C_{ij} = unit transportation cost from source i to consumer j

The job is:

$\Sigma_j(X_{ij}) \le \alpha_i \quad \forall\, i \qquad \Sigma_i(X_{ij}) \le \beta_j \quad \forall\, j$

$\Sigma_i\Sigma_j(X_{ij}) = \max \quad \text{at} \quad \Sigma_i\Sigma_j(X_{ij} \bullet C_{ij}) = \min$

1. Making capacities and needs equal to each other

Here: adding a fictive consumer with β_0=440-380=60 need and with C_{0i}=0 unit transportation costs.
(So $\Sigma_i\Sigma_j(X_{ij})$= max = $\Sigma_i(\alpha_i)$ = $\Sigma_j(\beta_j)$ = 440, where β_0 represents sum of uncontracted amount at sources i)

0	1	2	2	4	180
0	4	2	5	1	140
0	6	3	1	2	120
60	150	80	60	90	440

THE TRANSPORTATION PROBLEM

Problem: Do develope a maximum amount transportation program with minimum associating transportation cost

*Idea: Applying an LP primal-dual simplex algorithm**

c_{ij}

				α_i
1	2	2	4	180
4	2	5	1	140
6	3	1	2	120
β_j 150	80	60	90	

αi = maximum available capacity at source i.
βj = need at consumer j
C_{ij} = unit transportation cost from source i to consumer j

The job is:

$\Sigma_j(X_{ij}) \leq \alpha_i \;\; \forall\, i \qquad \Sigma_i(X_{ij}) \leq \beta_j \;\; \forall\, j$
$\Sigma_i\Sigma_j(X_{ij}) = \max \quad \text{at} \quad \Sigma_i\Sigma_j(X_{ij}\bullet C_{ij}) = \min$

1. Making capacities and needs equal to each other

Here: adding a fictive consumer with β_0=440-380=60 need and with C_{0i}=0 unit transportation costs.
(So $\Sigma_i\Sigma_j(X_{ij})$= max = $\Sigma_i(\alpha_i) = \Sigma_j(\beta_j)$ = 440, where β_0 represents sum of uncontracted amount at sources i)

0	1	2	2	4	180
0	4	2	5	1	140
0	6	3	1	2	120
60	150	80	60	90	440

2. Creating an initial solution

Here: using Danzig's method *

x_{ij}

0 60	1 120	2	2	4	~~180~~ ~~120~~
0	4	2 50	5	1 90	~~140~~ ~~50~~
0	6 30	3 30	1 60	2	~~120~~ ~~60~~ 30
~~60~~	~~150~~ 30	~~80~~ ~~30~~	~~60~~	~~90~~	440

$\Sigma_i\Sigma_j(X_{ij})$ = 60+120+50+ 90+30+30+60=440

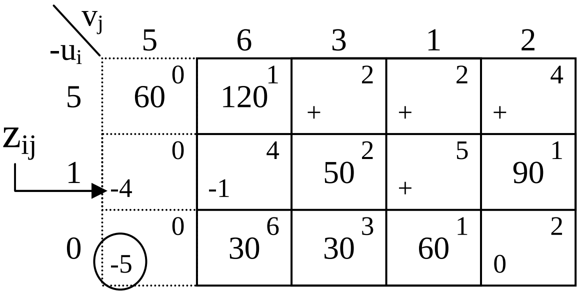
$\Sigma_i\Sigma_j(X_{ij}\bullet C_{ij})$ = 60•0+120•1+50•2+90•1+30•6+30•3+60•1=640

THE TRANSPORTATION PROBLEM

Problem: Do develope a maximum amount transportation program with minimum associating transportation cost

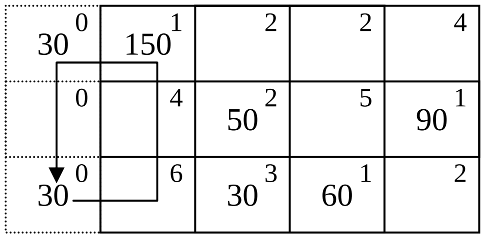
*Idea: Applying an LP primal-dual simplex algorithm**

3. Checking optimality
Assigning -u_i and v_j dual variables to rows and columns via assigned fields and calculating z_{ij} shadow prices at unassigned fields.
Here Z_{20} =-4, Z_{21} =-1 and Z_{30} =-5 values indicate possibility of decreasing overall transportation cost, so our program is still not optimal.



	5	6	3	1	2
5	0 60	1 120	2 +	2 +	4 +
1	0 -4	4 -1	2 50	5 +	1 90
0	0 -5	6 30	3 30	1 60	2 0

4. Improving the program
Assigning maximum possible amount to field of $z_{ij}^{min}<0$ value and deriving all necessary modifications along the corresponding loop.



0 30	1 150	2	2	4
0	4	2 50	5	1 90
0 30	6	3 30	1 60	2

$\delta X_{ij} = 30$ $\Sigma_i\Sigma_j(X_{ij}) = 30+150+50+90+30+30+60=440$

$\Sigma_i\Sigma_j(X_{ij}\bullet C_{ij}) = 30\bullet0+150\bullet1+50\bullet2+90\bullet1+30\bullet0+30\bullet3+60\bullet1=490$

Look: $\Delta C = 640-490 = Z_{30}\bullet \delta X_{ij} = (-5)\bullet30 = 150$

Repeat steps 3 and 4 until no negative z_{ij} values can be found
Z_{ij}=0 values - if any - indicate existence of other transportation programs with overall transportation cost the same.

THE TRANSPORTATION PROBLEM

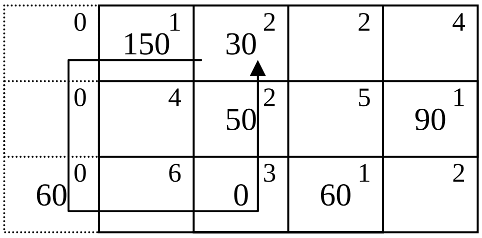
Problem: Do develope a maximum amount transportation program with minimum associating transportation cost

Idea: Applying an LP primal-dual simplex algorithm[*]

3. Checking optimality
Assigning -u_i and v_j dual variables to rows and columns via assigned fields and calculating z_{ij} shadow prices at unassigned fields.
Here Z_{12} =-1 value indicates possibility of decreasing overall transportation cost, so our program is still not optimal.

	0	1	3	1	2
0	0 30	1 150	2 -1	2 +	4 +
1	0 +	4 +	2 50	5 +	1 90
0	0 30	6 +	3 30	1 60	2 0

4. Improving the program
Assigning maximum possible amount to field of z_{ij}^{min}<0 value and deriving all necessary modifications along the corresponding loop.



0	1 150	2 30	2	4
0	4	2 50	5	1 90
0 60	6	3 0	1 60	2

$\delta X_{ij} = 30$ $\quad \Sigma_i\Sigma_j(X_{ij}) = 150+30+50+ 90+60+0+60=440$

$\Sigma_i\Sigma_j(X_{ij}\bullet C_{ij}) = 150\bullet1+30\bullet2+50\bullet2+90\bullet1+60\bullet0+0\bullet3+60\bullet1=460$

Look: $\Delta C = 490-460 = Z_{12}\bullet \delta X_{ij} = (-1)\bullet30 = 30$

Remark:
Fort to provide functionality of the algorithm one entering and one disappearing assigned field can be at each improvement, so here X_{32}=0 value - as a dummy assignment - must stay in the program.

THE TRANSPORTATION PROBLEM

Problem: Do develope a maximum amount transportation program with minimum associating transportation cost

Idea: Applying an LP primal-dual simplex algorithm[*]

3. Checking optimality
Assigning -u_i and v_j dual variables to rows and columns via assigned fields and calculating z_{ij} shadow prices at unassigned fields.
Here: no $Z_{ij}^{min}<0$ value can be found, so our program is optimal.

	0	2	3	1	2
1	0 +	1 150	2 30	2 +	4 +
1	0 +	4 +	2 50	5 +	1 90
0	0 60	6 +	3 0	1 60	2 0

Remarks:
Returning to original capacities and needs, X_{30}=60 value represents unassigned capacity at source 3.
Z_{34}=0 value indicates existence of an alternative program with overall transportation cost the same.

Sub-Problem: Creating a Feasible Initial Soultion – by Danzig'

Idea: in increasing order of c_{ij} values – and as English reads (left-right, top-down), do assign for transport minimum of still free capacities and still unsatisfied needs[*]

2/1. $c_{ij}^{min}=c_{10}=0$; $X_{10}=\min\{180,60\}$
$\alpha_1'=180-60=120$; column 0 canceled

0 60	1	2	2	4	~~180~~ 120
0	4	2	5	1	140
0	6	3	1	2	120
~~60~~	150	80	60	90	

2/2. $c_{ij}^{min}=c_{11}=1$; $X_{11}=\min\{120,150\}$
$\beta_1'=150-120=30$; row 1 canceled

0 60	1 120	2	2	4	~~180~~ ~~120~~
0	4	2	5	1	140
0	6	3	1	2	120
~~60~~	~~150~~ 30	80	60	90	

2/3. $c_{ij}^{min}=c_{24}=1$; $X_{24}=\min\{140,90\}$
$\alpha_2'=140-90=50$; column 4 canceled

0 60	1 120	2	2	4	~~180~~ ~~120~~
0	4	2	5	1 90	~~140~~ 50
0	6	3	1	2	120
~~60~~	~~150~~ 30	80	60	~~90~~	

2/4. $c_{ij}^{min}=c_{33}=1$; $X_{33}=\min\{120,60\}$
$\alpha_3'=120-60=60$; column 3 canceled

0 60	1 120	2	2	4	~~180~~ ~~120~~
0	4	2	5	1 90	~~140~~ 50
0	6	3	1 60	2	~~120~~ 60
~~60~~	~~150~~ 30	80	~~60~~	~~90~~	

2/5. $c_{ij}^{min}=c_{22}=2$; $X_{22}=\min\{50,80\}$
$\beta_2'=80-50=30$; row 2 canceled

0 60	1 120	2	2	4	~~180~~ ~~120~~
0	4	2 50	5	1 90	~~140~~ ~~50~~
0	6	3	1 60	2	~~120~~ 60
~~60~~	~~150~~ 30	~~80~~ 30	~~60~~	~~90~~	

2/6. $c_{ij}^{min}=c_{32}=3$; $X_{32}=\min\{60,30\}$
$\alpha_3''=60-30=30$; column 2 canceled

0 60	1 120	2	2	4	~~180~~ ~~120~~
0	4	2 50	5	1 90	~~140~~ ~~50~~
0	6	3 30	1 60	2	~~120~~ ~~60~~ 30
~~60~~	~~150~~ 30	~~80~~ ~~30~~	~~60~~	~~90~~	

2/7. last: $c_{ij}^{min}=c_{31}=6$; $X_{31}=\alpha_3''=\beta_1'=30$
(Both row 3 and column 1 run out.)

0 60	1 120	2	2	4	~~180~~ ~~120~~
0	4	2 50	5	1 90	~~140~~ ~~50~~
0	6 30	3 30	1 60	2	~~120~~ ~~60~~ ~~30~~
~~60~~	~~150~~ ~~30~~	~~80~~ ~~30~~	~~60~~	~~90~~	

**Remarks:*

At each step but the last you may cancel either the row or the column otherways you should keep either the row or the column in the program with 0 value.

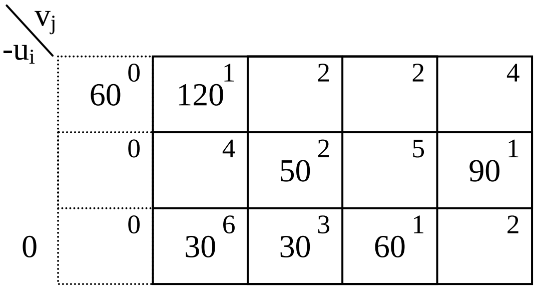
Number of assigned fields (X_{ij} values) must be one less than sum of number of rows and of columns of the table (Here: 3+5-1=7)

$C_0 = 60\bullet0+120\bullet1+50\bullet2+90\bullet1+30\bullet6+30\bullet3+60\bullet1=640$

Sub-Problem: Assign dual variables to rows and columns

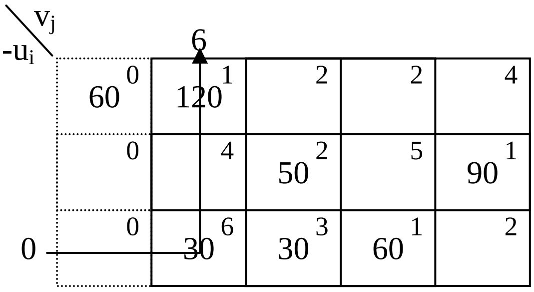
*Idea: Starting with $-u_i=0$ value at row where c_{ij}=max at assigned fields is, do assign further v_j and $-u_i$ values via assigned fields, applying relation $v_j=c_{ij}-u_i$ **

3/1. $c_{ij}^{max}=c_{31}=6$; $-u_3=0$



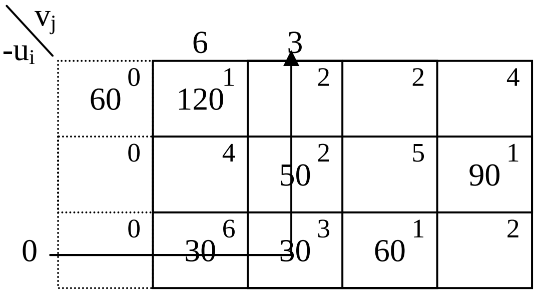
v_j / $-u_i$					
	60^{0}	120^{1}	^{2}	^{2}	^{4}
	^{0}	^{4}	50^{2}	^{5}	90^{1}
0	^{0}	30^{6}	30^{3}	60^{1}	^{2}

3/2. $v_1=c_{31}+(-u_3)$; $v_1=6+0=6$



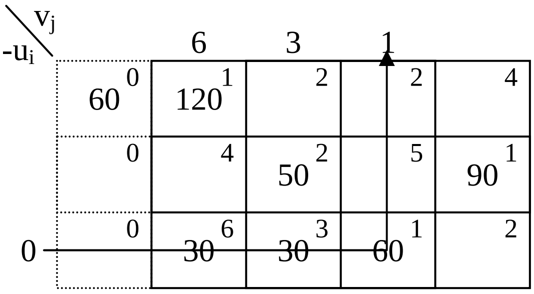
v_j / $-u_i$		6			
	60^{0}	120^{1}	^{2}	^{2}	^{4}
	^{0}	^{4}	50^{2}	^{5}	90^{1}
0	^{0}	30^{6}	30^{3}	60^{1}	^{2}

3/3. $v_2=c_{32}+(-u_3)$; $v_2=3+0=3$



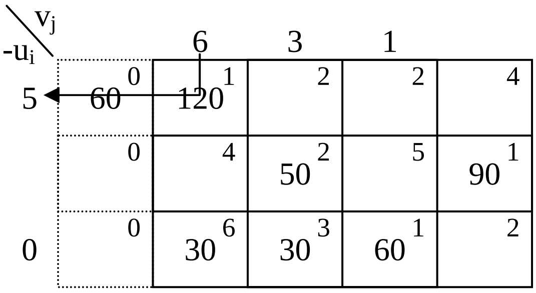
v_j / $-u_i$		6	3		
	60^{0}	120^{1}	^{2}	^{2}	^{4}
	^{0}	^{4}	50^{2}	^{5}	90^{1}
0	^{0}	30^{6}	30^{3}	60^{1}	^{2}

3/4. $v_3=c_{33}+(-u_3)$; $v_3=1+0=1$



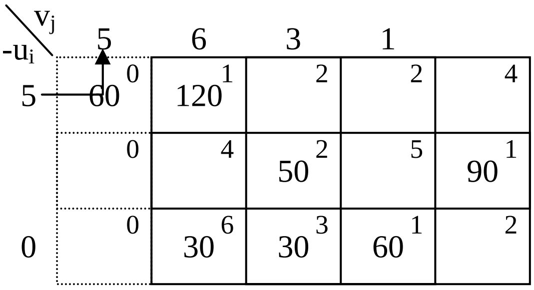
v_j / $-u_i$		6	3	1	
	60^{0}	120^{1}	^{2}	^{2}	^{4}
	^{0}	^{4}	50^{2}	^{5}	90^{1}
0	^{0}	30^{6}	30^{3}	60^{1}	^{2}

3/5. $-u_1=v_1-c_{11}$; $-u_1=6-1=5$



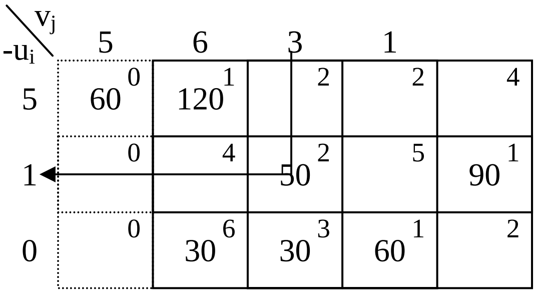
v_j / $-u_i$		6	3	1	
5	60^{0}	120^{1}	^{2}	^{2}	^{4}
	^{0}	^{4}	50^{2}	^{5}	90^{1}
0	^{0}	30^{6}	30^{3}	60^{1}	^{2}

3/6. $v_0=c_{10}+(-u_1)$; $v_0=0+5=5$



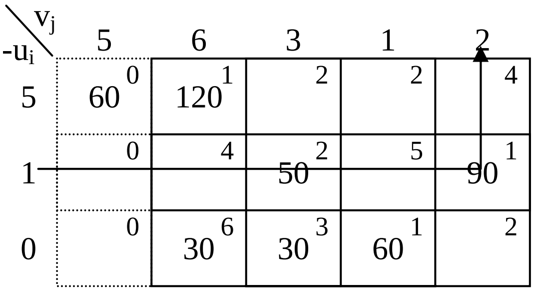
v_j / $-u_i$	5	6	3	1	
5	60^{0}	120^{1}	^{2}	^{2}	^{4}
	^{0}	^{4}	50^{2}	^{5}	90^{1}
0	^{0}	30^{6}	30^{3}	60^{1}	^{2}

3/7. $-u_2=v_2-c_{22}$; $-u_2=3-2=1$



v_j / $-u_i$	5	6	3	1	
5	60^{0}	120^{1}	^{2}	^{2}	^{4}
1	^{0}	^{4}	50^{2}	^{5}	90^{1}
0	^{0}	30^{6}	30^{3}	60^{1}	^{2}

3/8. $v_4=c_{24}+(-u_2)$; $v_4=1+1=2$



v_j / $-u_i$	5	6	3	1	2
5	60^{0}	120^{1}	^{2}	^{2}	^{4}
1	^{0}	^{4}	50^{2}	^{5}	90^{1}
0	^{0}	30^{6}	30^{3}	60^{1}	^{2}

* See: Linear Programming / Simplex Method / Primal-Dual Algorithms

Sub-Problem: Improving the program

Idea: Assigning maximum possible amount to field of $z_{ij}^{min}<0$ value and deriving all necessary modifications along the corresponding loop.

4/1. Calculate $z_{ij}=c_{ij}-u_i-v_j$ values *
(„shadow prices”) for non-assigned fields
Check $z_{ij}^{min}<0$ values. Here: $z_{ij}^{min}=z_{30}=-5$

	5	6	3	1	2
5	60 ^{0}	120 ^{1}	^{2} +	^{2} +	^{4} +
1	^{0} -4	^{4} -1	50 ^{2}	^{5} +	90 ^{1}
0	^{0} -5	30 ^{6}	30 ^{3}	60 ^{1}	^{2} 0

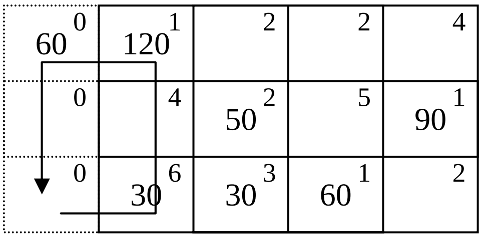
Remarks:

$z_{ij}<0$ values indicate possibility of improving the transportation program (decreasing the overall transportation cost)

$z_{ij}=0$ values indicate possibility of alternative transportation programs (with overall transportation costs the same)

$z_{ij}>0$ values are now out of interest, simply indicate them by + signs

4/2. Find the only loop for the new (would be) assigned field (X_{30})
Here: $X_{30}-X_{31}-X_{11}-X_{10}-X_{30}$



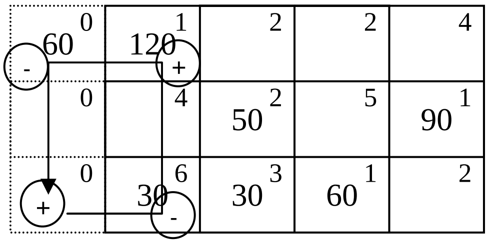
60 ^{0}	120 ^{1}	^{2}	^{2}	^{4}
^{0}	^{4}	50 ^{2}	^{5}	90 ^{1}
^{0}	30 ^{6}	30 ^{3}	60 ^{1}	^{2}

Remarks:

Due to the algorithm for each non-assigned field there exists a single loop other wedges of which are at assigned fields.

Remember: Sum of X_{ij} values are fixed in each row and in each column. If I modified somewhere I have to make counter-measures in its row and in its column too. And so on - around.

4/3. Assign +/- signs to wedges alternately and define δX_{ij} value for improving
Here: $\delta X_{ij}=\min\{30,60\}=30$



60 ^{0} -	120 ^{1} +	^{2}	^{2}	^{4}
^{0}	^{4}	50 ^{2}	^{5}	90 ^{1}
^{0} +	30 ^{6} -	30 ^{3}	60 ^{1}	^{2}

Remarks:

δX_{ij} value (maximum amount that can be re-assigned) is the X_{ij}^{min} value at - wedges.

Remember: For the sake of correctness of the (simplex) algorithm, number of assigned fields must be one less than sum of rows and of columns of the table (here: 3+5-1=7) after re-assigning too. Negative X_{ij} values would have no meaning.

A new assigned field enters an old assigned field disappears. If more than one assigned field would disappear one should be selected for to disappear and the others should be kept in the program with $X_{ij}=0$ value.

Look: The overall transportation cost got be $C=30\bullet0+150\bullet1+50\bullet2+90\bullet1+30\bullet0+30\bullet3+$ $+60\bullet1=490$ that is $\Delta=z_{30}\bullet\delta X_{ij}=-5\bullet30=-150$ less than was before.

4/4. Re-assign the program along the loop

30 ^{0}	150 ^{1}	^{2}	^{2}	^{4}
^{0}	^{4}	50 ^{2}	^{5}	90 ^{1}
30 ^{0}	^{6}	30 ^{3}	60 ^{1}	^{2}

* See: Linear Programming / Simplex Method / Primal-Dual Algorithms