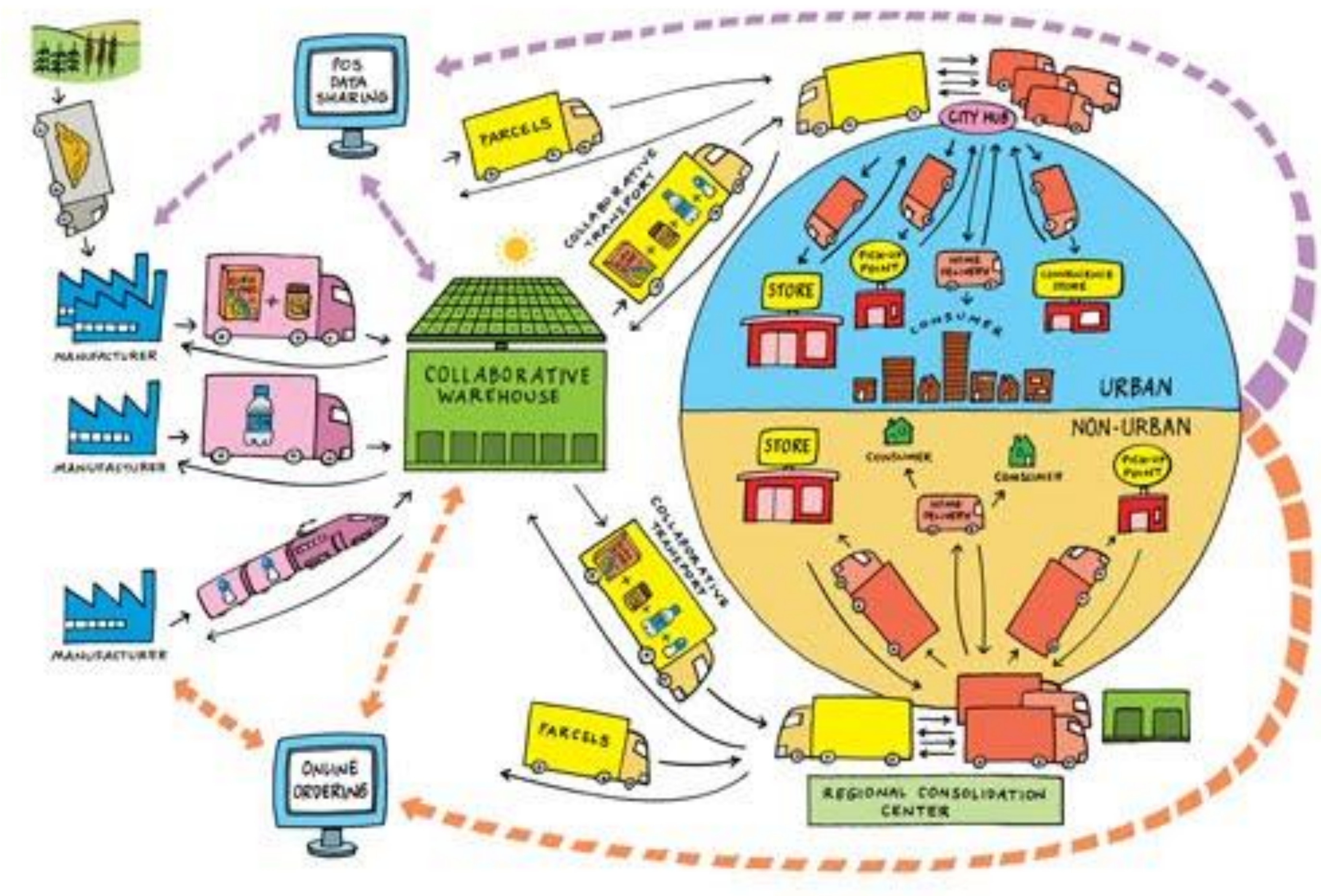
1. What do you get as the integer and non-integer parts if you factor this as we did with our cutting planes:

$x_1 = -8.75 \quad - 3.1\, x_2 + 4.2\, x_3 \; + 7\, x_5 \; -8\, x_6$

2. What constraint is added?

3. What do you need to do before adding it this contraint to the problem you just solved?

Transshipment Problems

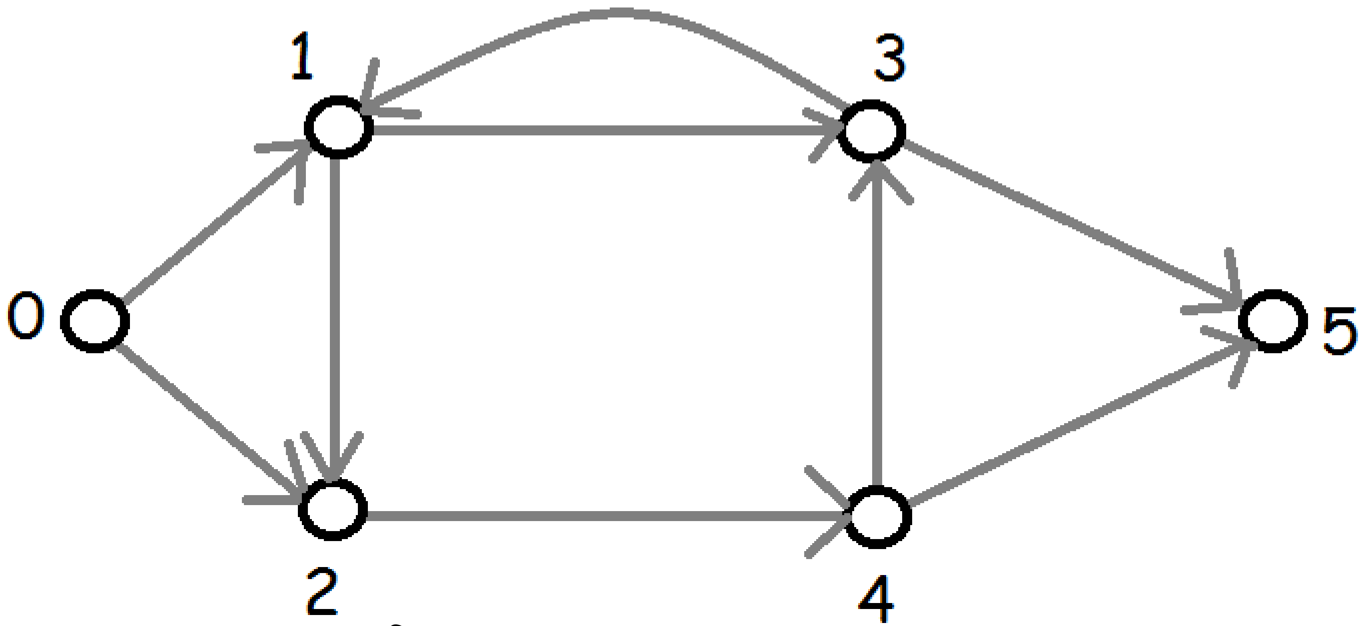


http://blog.trisakti.ac.id/informazi/2011/10/30/network-models/

Assignment #3: Due: Friday Nov. 14, beginning of class (12:30pm).
Late deadline: Tues. Nov. 18 at 12:30pm.

CSC 545 student slides:
Due date: Friday Nov. 7 at 11:55pm.
Late deadline: Tues. Nov. 11 at 11:55pm.
I plan to waive the late penalty, but NOT to extend the late deadline.
Please use a white background on the slides so that I do not waste ink and so that I have somewhere to write my feedback.

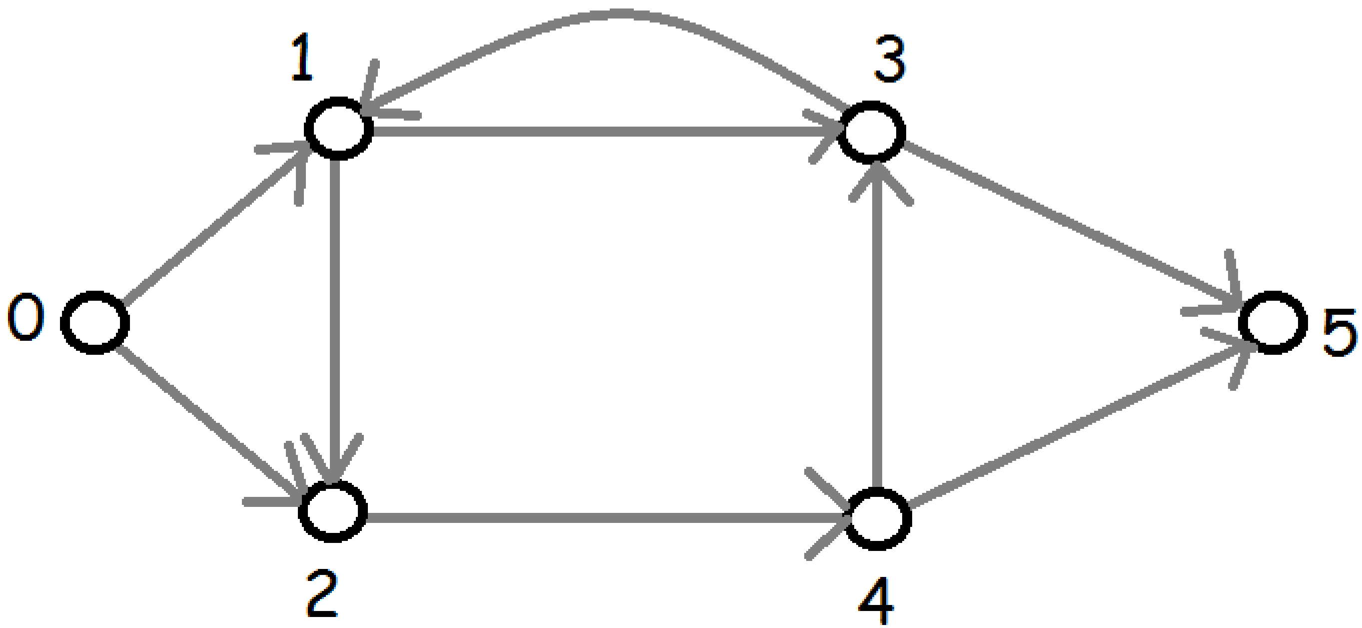
A directed graph G consists of a set V of vertices and a set E of arcs where each arc in E is associated with an ordered pair of vertices from V.



V={0, 1, 2, 3, 4, 5}

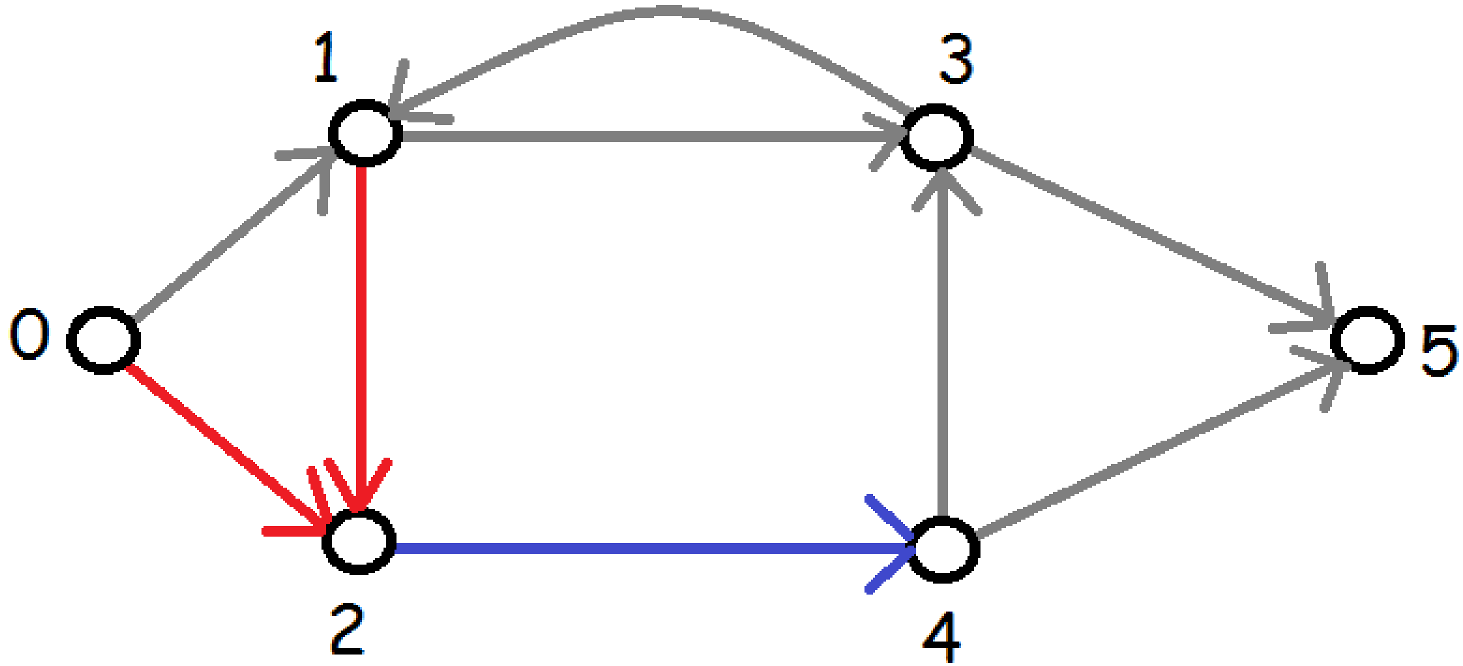
E= {(0,1), (0,2), (1,2), (1,3), (2,4),(3,1),(3,5),(4,3),(4,5)}

A directed graph G:



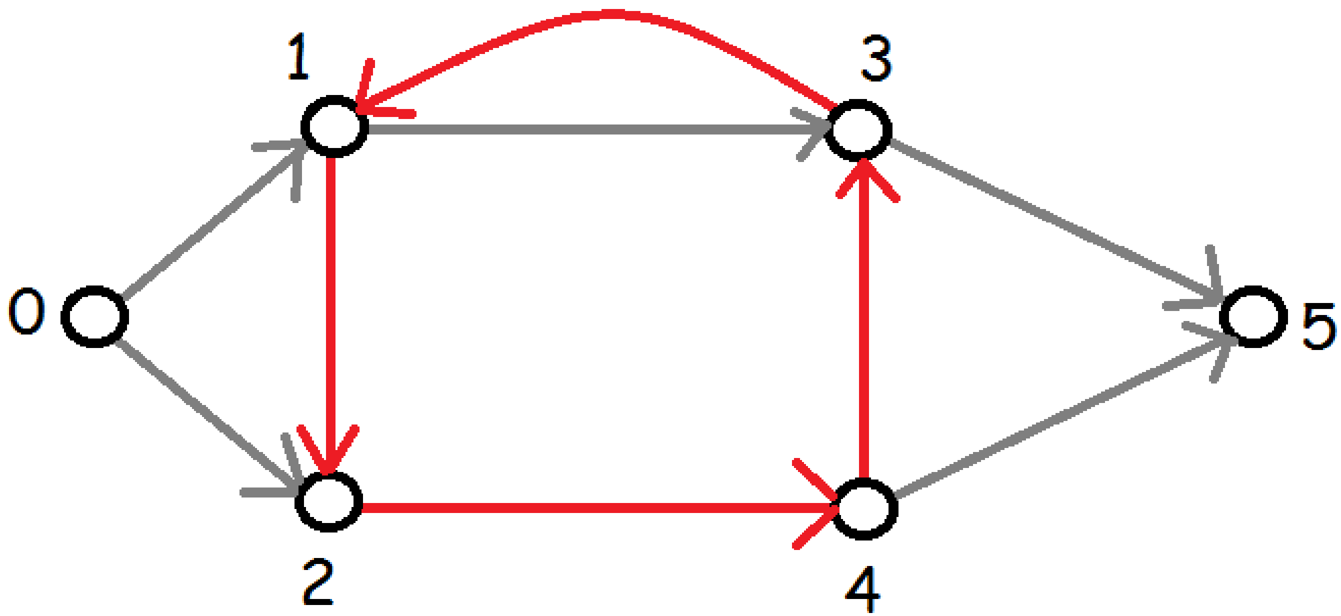
Remark: In a talk, I might just use the pictures without as many words, but I would not use words without pictures.

A directed graph G:



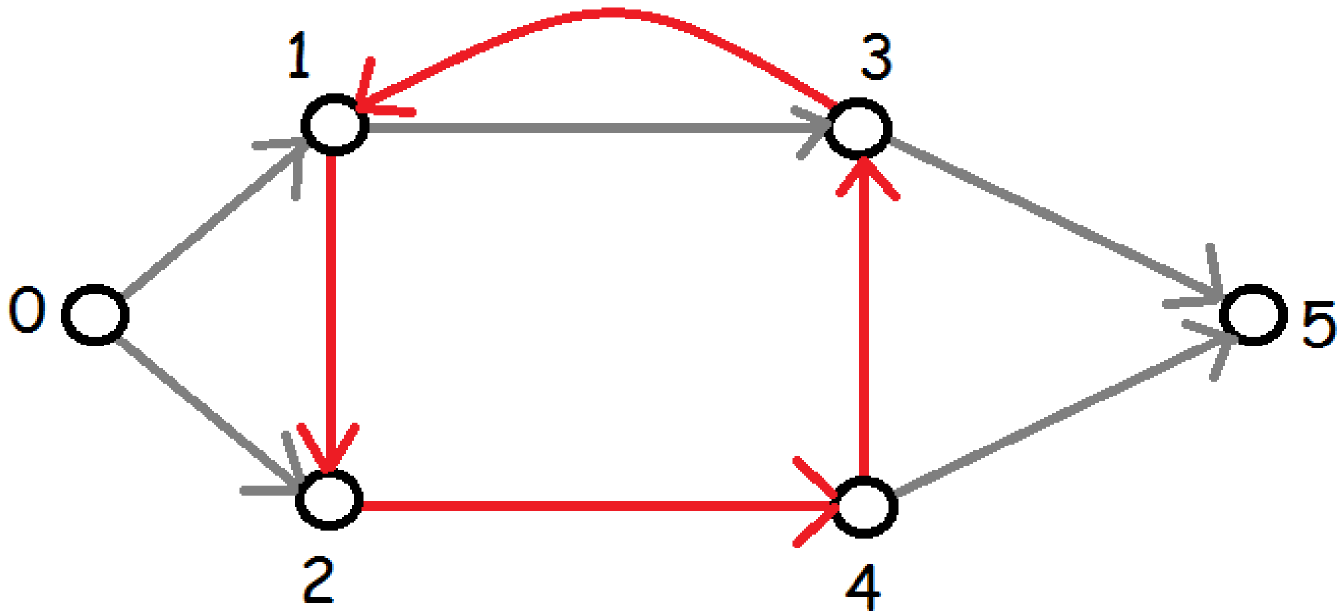
Vertex 2 has in-degree 2 and out-degree 1.

A directed cycle of length k consists of an alternating sequence of vertices and arcs of the form: $v_0, e_1, v_1, e_2, \ldots, v_{k-1}, e_k, v_k$ where $v_0 = v_k$ but otherwise the vertices are distinct and where e_i is the arc (v_i, v_{i+1}) for i= 0, 1, 2, …, k-1.

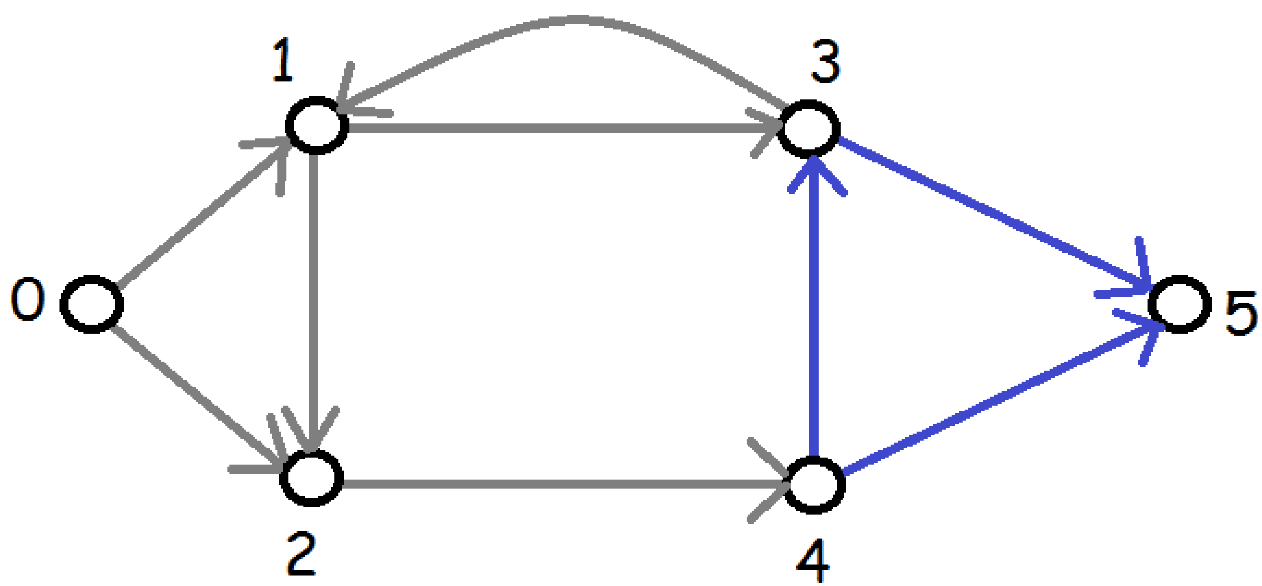


1, (1,2), 2, (2,4), 4, (4,3), 3, (3,1), 1

A directed cycle of length 4:

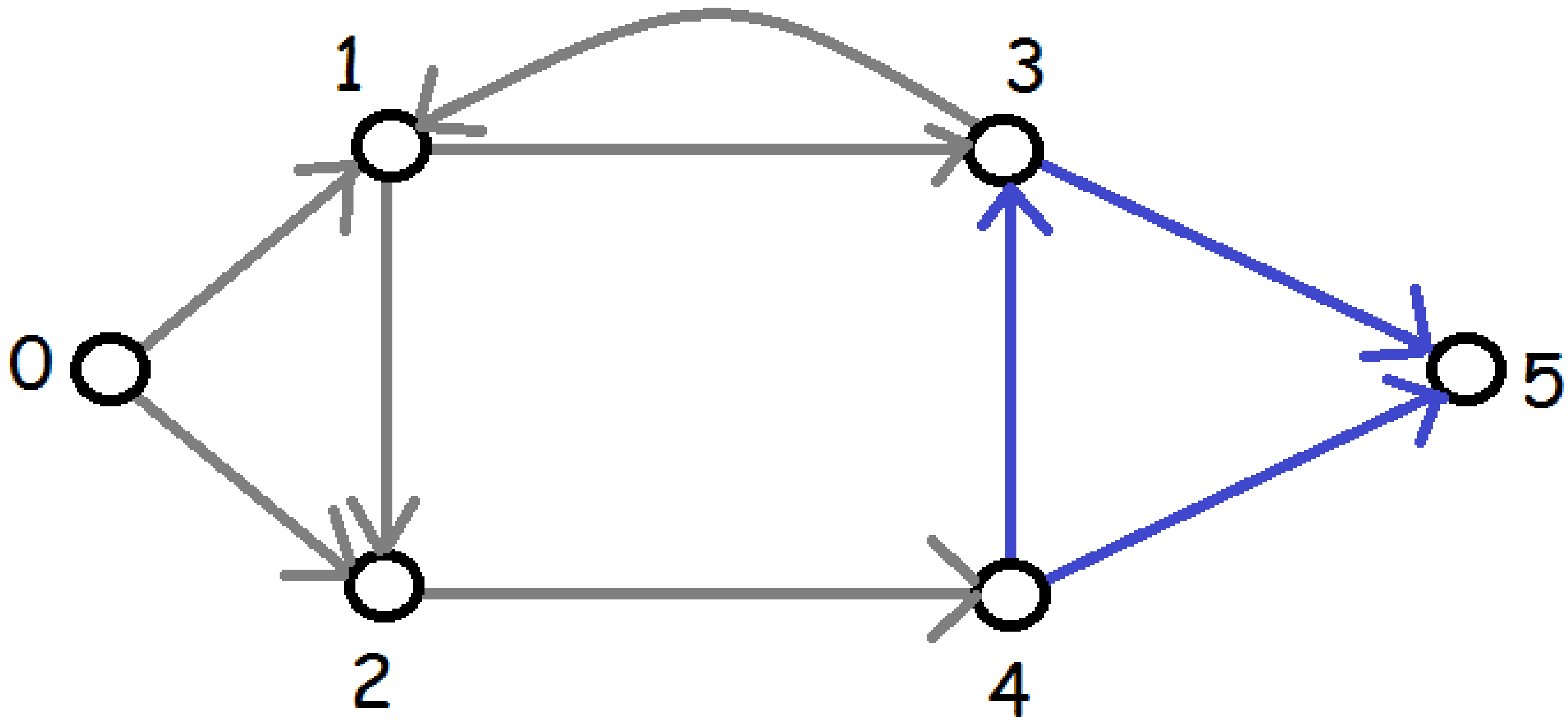


A cycle of length k consists of an alternating sequence of vertices and arcs of the form: v_0, e_1, v_1, e_2, … ,v_{k-1}, e_k, v_k where $v_0 = v_k$ but otherwise the vertices are distinct and where e_i is either the arc (v_i, v_{i+1}) or (v_{i+1}, v_i) for i= 0, 1, 2, …, k-1.

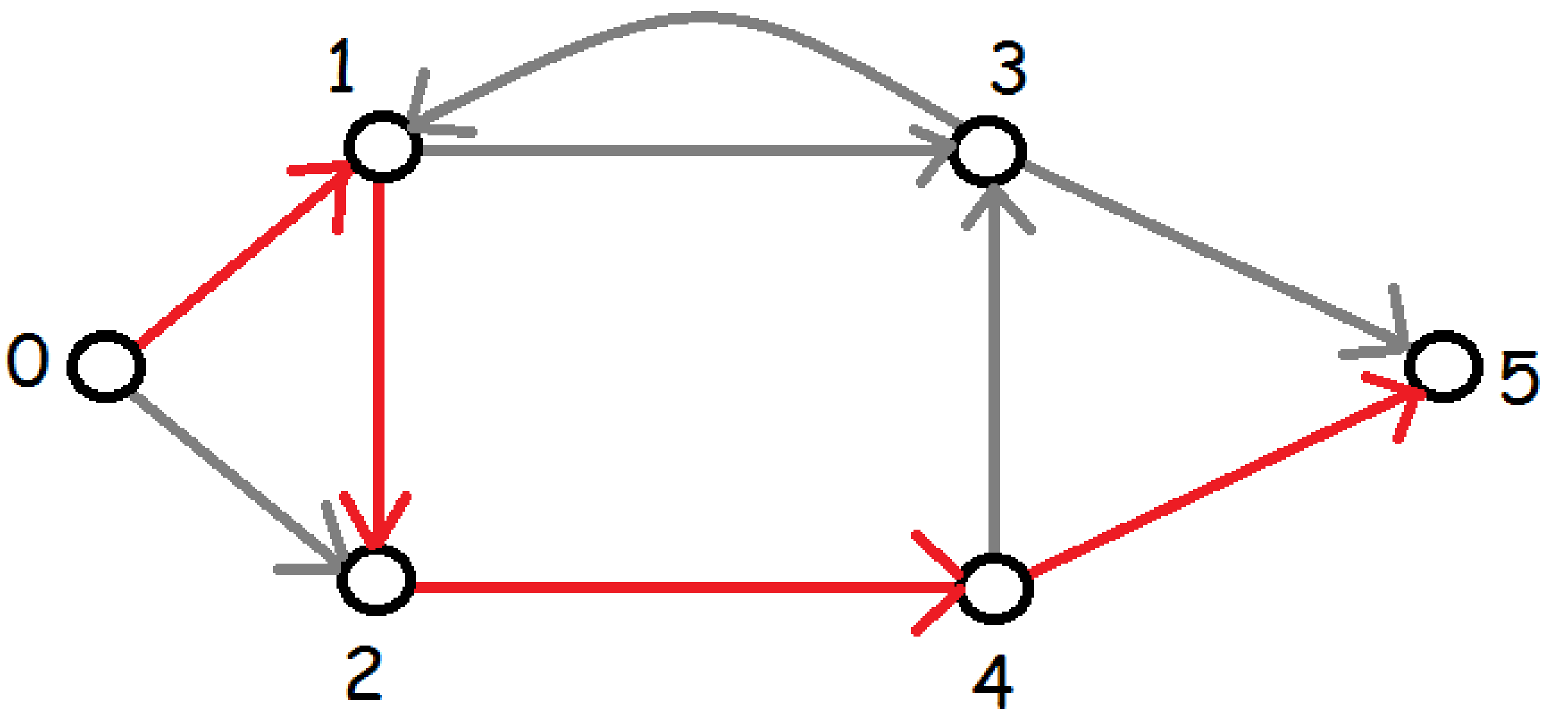


3, (3,5), 5, (4,5), 4, (4,3), 3

A cycle of length 3 which is not a directed cycle (arcs can be traversed in either direction):

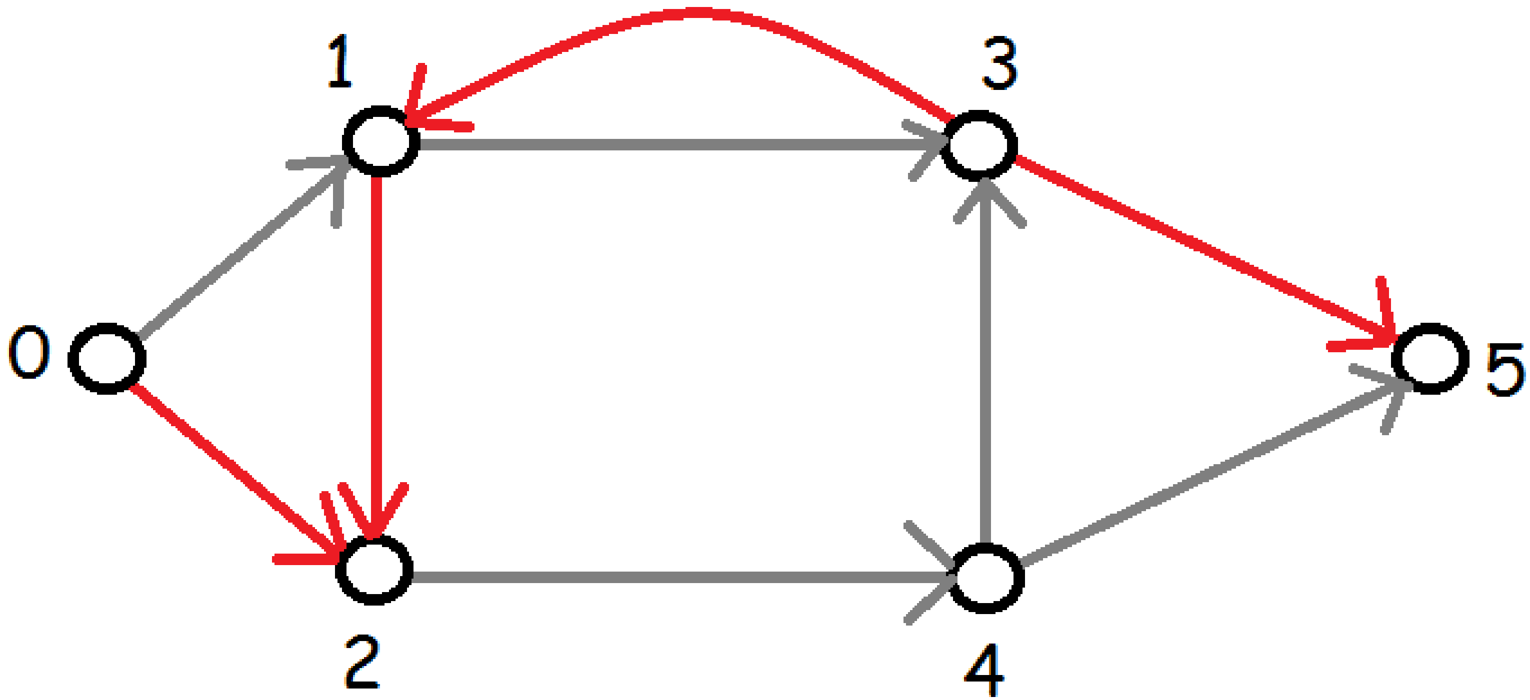


A directed path of length 4 from vertex 0 to vertex 5:



0, (0,1), 1, (1,2), 2, (2,4), 4, (4,5), 5

A path of length 4 which is not a directed path (arcs can be traversed in either direction) from vertex 0 to vertex 5:



0, (0,2), 2, (1,2), 1, (3,1), 3, (3,5), 5

Transshipment Problem

Ship prescribed amounts of a commodity from specified origins to specified destinations through a concrete transportation network.

Sources- nodes with a supply of a commodity

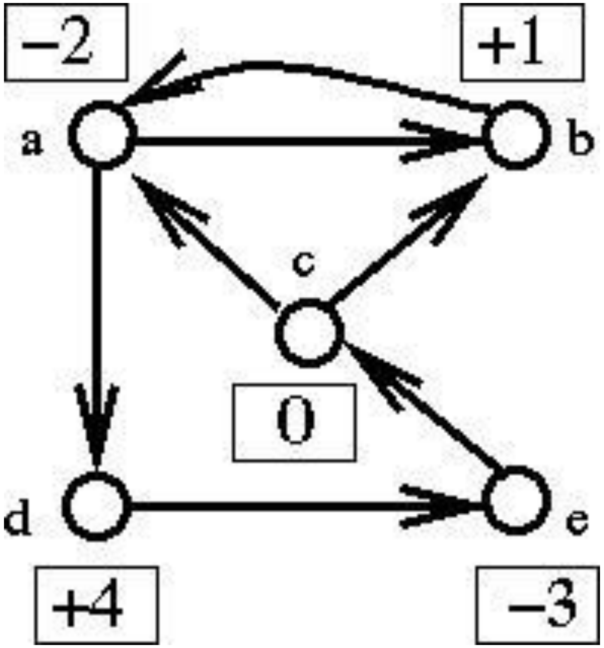
Sinks- nodes with a demand for a commodity

Intermediate Nodes- nodes which are not a source or a sink

Node	Status	Type
a	supply of 2 units	source
b	demand for 1 unit	sink
c	no supply or demand	intermediate node
d	demand for 4 units	sink
e	supply of 3 units	source

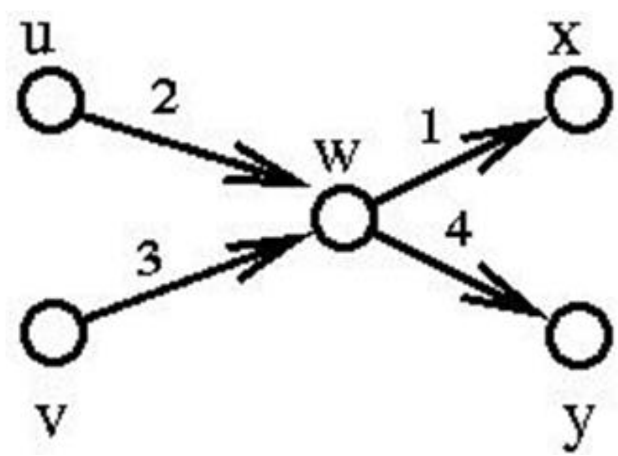
Example:

The demands are shown in the boxes. Sources have negative demands equal to -1 * (# units available).



Assumption: total supply = total demand.
Schedule: specifies amount shipped from node i to node j along each arc.

The following schedule is ambiguous:



However, it is assumed that units are interchangeable so the trajectory of individual units is irrelevant.

$f^+(v)$ = the number of units entering node v

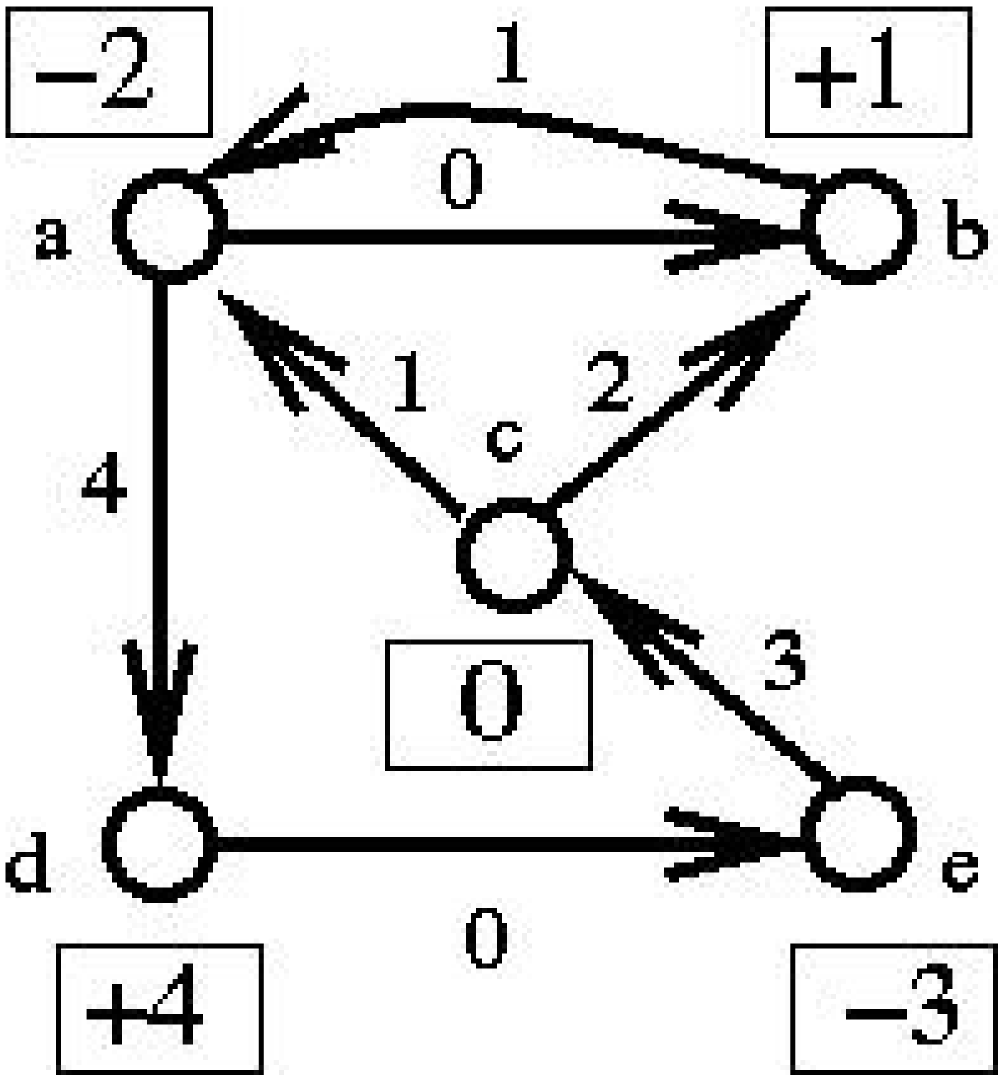
$f^-(v)$ = the number of units leaving node v

Conservation of flow means:

1. For each intermediate node v, $f^+(v) = f^-(v)$
2. For each source node v,
   $f^-(v) - f^+(v)$ = supply at source node v
3. For each sink node v,
   $f^+(v) - f^-(v)$ = demand at sink node v

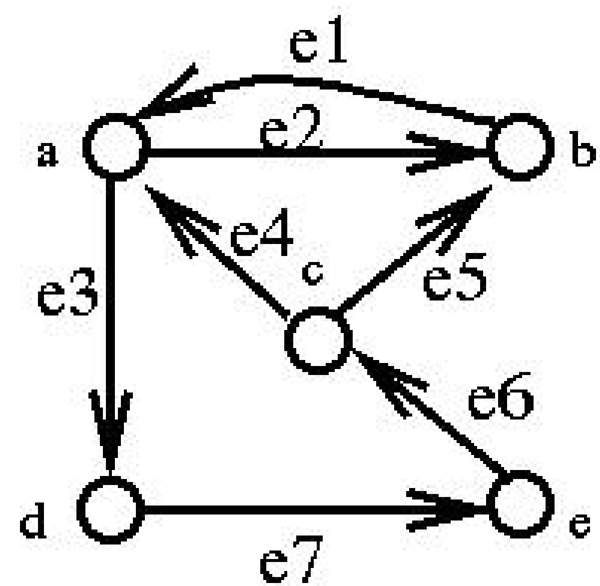
In addition, it only makes sense if the amounts shipped along each are are non-negative.

Is flow conserved with this schedule?



Let A be a matrix whose rows correspond to the nodes of the network and whose columns correspond to the arcs. For each arc $e_i = (v_j, v_k)$:

- entry $a_{ji} = -1$
- entry $a_{ki} = +1$
- for r not equal to j or k, a_{ri} is zero.



Incidence Matrix A:

$$\begin{array}{cc} & \begin{array}{ccccccc} e1 & e2 & e3 & e4 & e5 & e6 & e7 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} +1 & -1 & -1 & +1 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & +1 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & +1 \end{bmatrix} \end{array}$$

The X vector for this problem has one variable for each of the m arcs of the graph.

m= the number of arcs.

Note: This deviates from our standard convention for the use of variable m. We do this because for a digraph, n is conventionally the number of nodes, and m is the number of arcs.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_m \end{bmatrix}$$

Requirements

We must have $A x = b$ where $x \geq 0$, and

- $b_i = 0$ if node i is an intermediate node,
- $b_i = -1 * (\text{supply at node } i)$ if node i is a source, and
- $b_i = \text{demand at node } i$ if node i is a sink.

Also, the flow on each arc is non-negative:

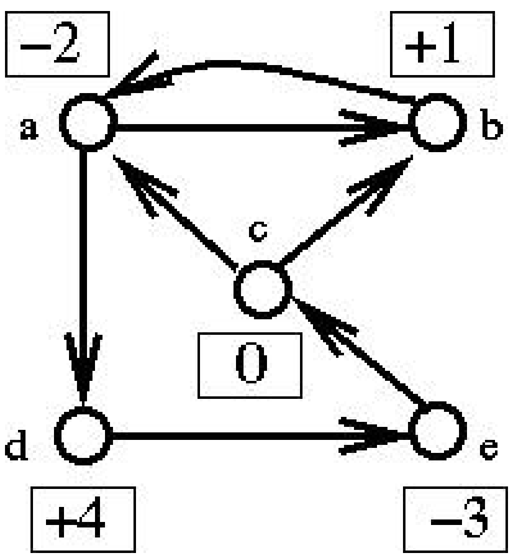
$x \geq 0$

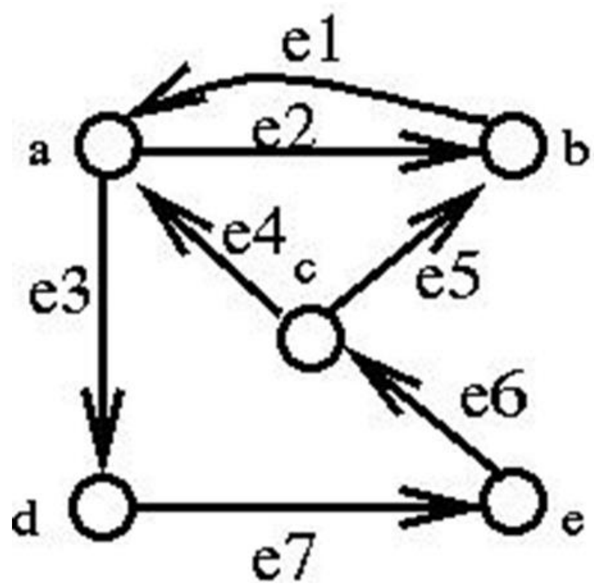
Incidence Matrix A:

$$
\begin{array}{c|ccccccc}
 & e1 & e2 & e3 & e4 & e5 & e6 & e7 \\
\hline
a & +1 & -1 & -1 & +1 & 0 & 0 & 0 \\
b & -1 & +1 & 0 & 0 & +1 & 0 & 0 \\
c & 0 & 0 & 0 & -1 & -1 & +1 & 0 \\
d & 0 & 0 & +1 & 0 & 0 & 0 & -1 \\
e & 0 & 0 & 0 & 0 & 0 & -1 & +1
\end{array}
$$

$$
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} -2 \\ +1 \\ 0 \\ +4 \\ -3 \end{bmatrix}
$$

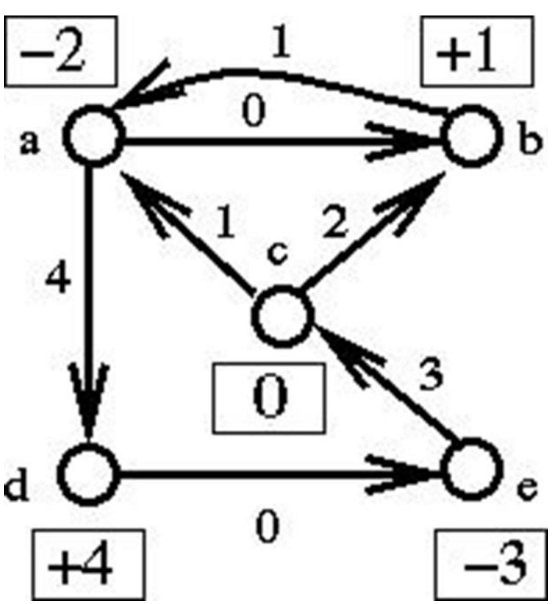
$x \geq 0$

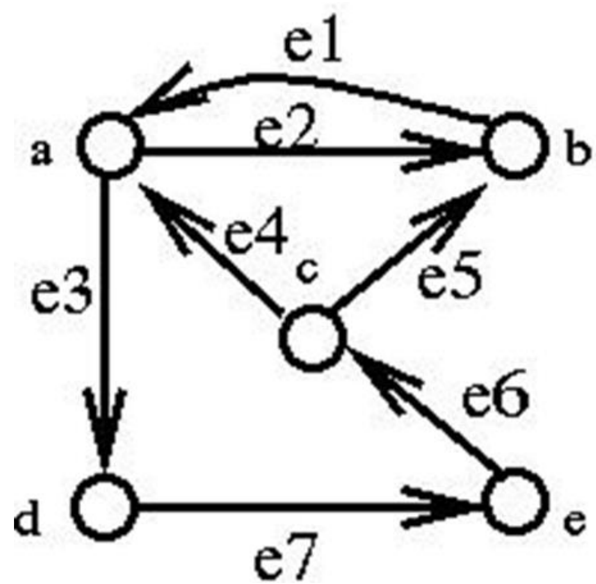




Incidence Matrix A:

$$\begin{array}{cc} & \begin{array}{ccccccc} e1 & e2 & e3 & e4 & e5 & e6 & e7 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} +1 & -1 & -1 & +1 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & +1 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & +1 \end{bmatrix} \end{array} \begin{bmatrix} 1 \\ 0 \\ 4 \\ 1 \\ 2 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ +1 \\ 0 \\ +4 \\ -3 \end{bmatrix}$$





$x \geq 0$

Let c_{ij} be the cost of shipping one unit from i to j along arc (i, j).
The cost of a schedule is $c^T x$.
Find the cheapest schedule.
The LP: Minimize $c^T x$
subject to $A x = b, x \geq 0$.

A Transshipment Problem is formulated as above where A is the incidence matrix of some network, and the sum of the entries of b is zero (the total supply equals the total demand).

Note that the sum of all the equations is equal to 0:

$$\begin{array}{c|ccccccc|}
 & e1 & e2 & e3 & e4 & e5 & e6 & e7 \\
a & +1 & -1 & -1 & +1 & 0 & 0 & 0 \\
b & -1 & +1 & 0 & 0 & +1 & 0 & 0 \\
c & 0 & 0 & 0 & -1 & -1 & +1 & 0 \\
d & 0 & 0 & +1 & 0 & 0 & 0 & -1 \\
e & 0 & 0 & 0 & 0 & 0 & -1 & +1
\end{array}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} -2 \\ +1 \\ 0 \\ +4 \\ -3 \end{bmatrix}$$

Remove the last equation to get rid of the dependence:

$A^* x = b$:

$$
\begin{array}{c c}
 & \begin{array}{ccccccc} e1 & e2 & e3 & e4 & e5 & e6 & e7 \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} +1 & -1 & -1 & +1 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & +1 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & -1 \end{bmatrix}
\end{array}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} -2 \\ +1 \\ 0 \\ +4 \end{bmatrix}
$$

Matrix A* is called the truncated incidence matrix.

A network is *connected* if there is a path between every pair of nodes.

A network is *acyclic* if there are no cycles.

A *tree* is a connected, acyclic network.

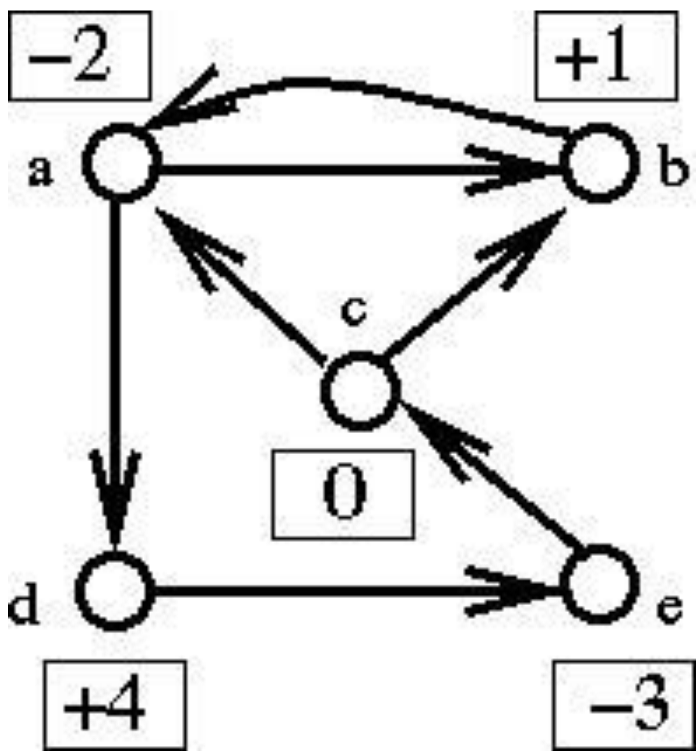
A *subgraph* H of network D is a directed graph such that V(H) is a subset of V(D) and E(H) is a subset of E(D).

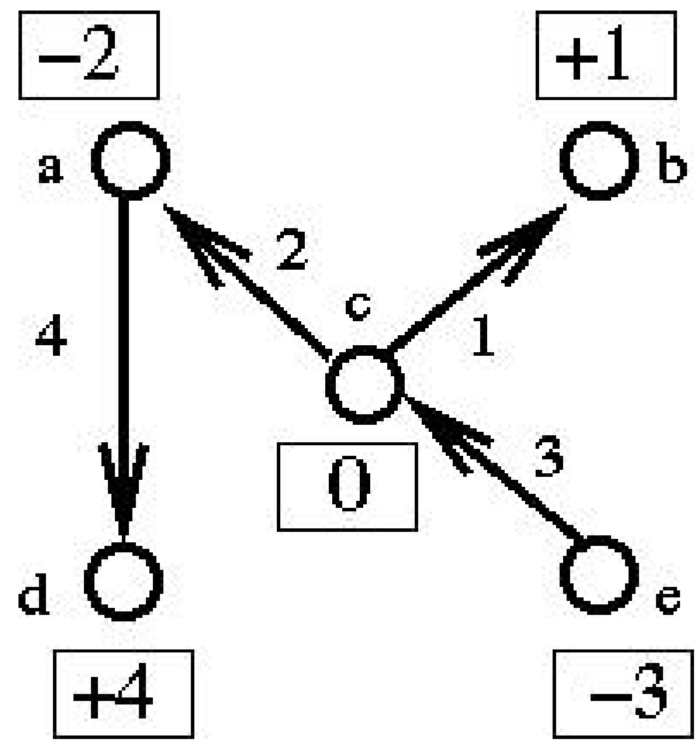
A subgraph H is a *spanning* subgraph of D if V(H) = V(D).

A *spanning tree* of network D is a spanning subgraph of D that is a tree.

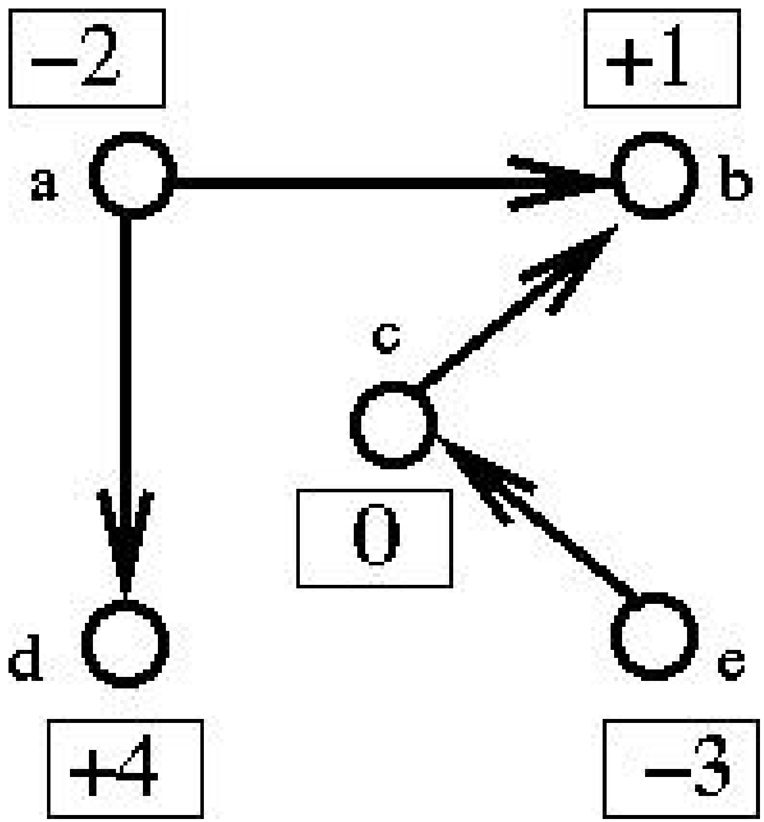
Trees and Feasible Solutions

The following picture shows a *feasible solution associated with a tree*:





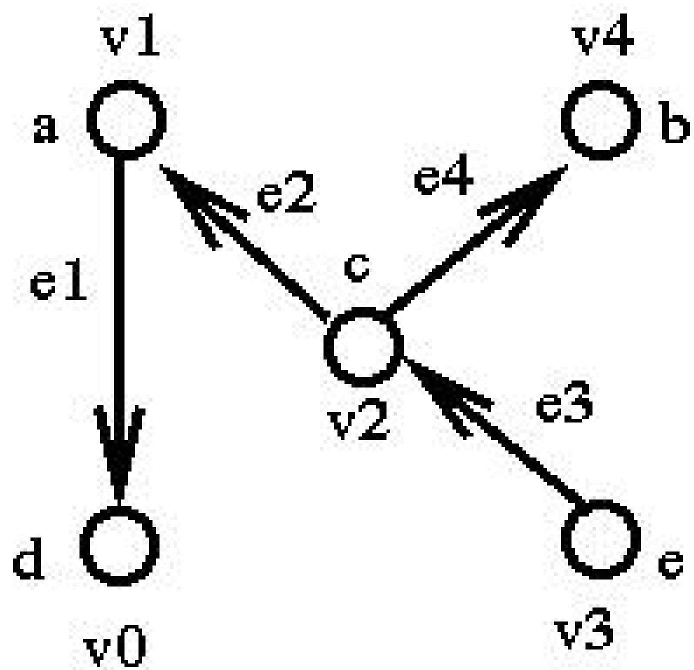
Not all trees correspond to feasible solutions. Consider for example:



Labeling the Tree

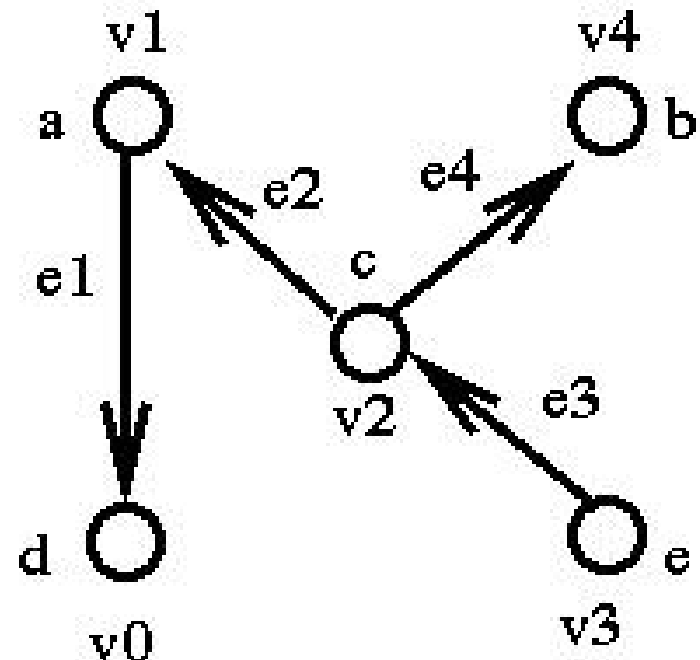
Pick any node of T and label it v_0.

Use BFS (breadth first search) to number the other nodes by $v_1, v_2, \ldots, v_{n-1}$ and to number the arcs of the tree by $e_1, e_2, \ldots, e_{n-1}$.



Let B be the truncated incidence matrix for T with the row corresponding to v_0 deleted.

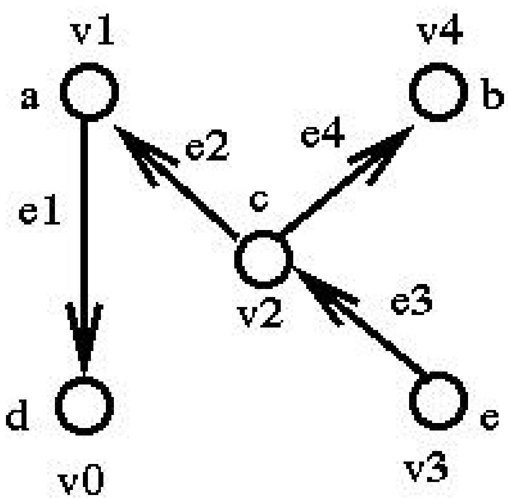
$$\begin{array}{c|c} & \begin{array}{cccc} e1 & e2 & e3 & e4 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{bmatrix} -1 & +1 & 0 & 0 \\ 0 & -1 & +1 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & +1 \end{bmatrix} \end{array}$$

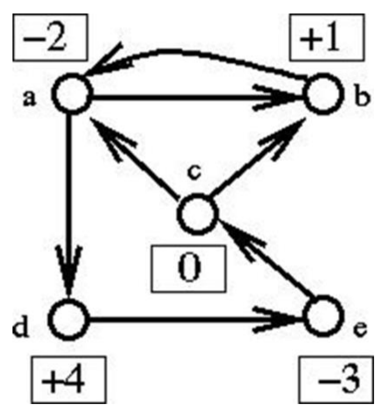


The matrix B is upper triangular.
The diagonal has +1/-1 entries so B has full rank.

Conclusion: $B x^* = b^*$ has a unique solution (the * means delete the row which corresponds to v_0).

$$\begin{array}{c|cccc} & e1 & e2 & e3 & e4 \\ \hline v_1 & -1 & +1 & 0 & 0 \\ v_2 & 0 & -1 & +1 & -1 \\ v_3 & 0 & 0 & -1 & 0 \\ v_4 & 0 & 0 & 0 & +1 \end{array} \quad \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ -3 \\ +1 \end{bmatrix}$$





The Network Simplex Method considers only feasible solutions that correspond to trees.

We saw before: the dual variables have an economic interpretation.

y_i = the cost of 1 unit of the commodity at node i.

If 1 unit is bought at node i and then shipped along arc (i, j) of cost c_{ij} to node j, then the fair price at node j is $y_i + c_{ij}$.

y_i = the cost of 1 unit of the commodity at node i.

If one unit is bought at node i and then shipped along arc (i, j) of cost c_{ij} to node j, then the fair price at node j is $y_i + c_{ij}$.

If the cost of doing this is less than the current cost of the commodity at node j, it pays to ship along arc (i,j).

If the cost of doing this is more than the current cost of the commodity at node j, it pays to not ship along arc (i,j).

The Network Simplex Method

Step 1: Assume the price of the commodity at v_0 is 0 and use the current feasible tree solution to determine fair prices at all the other nodes.

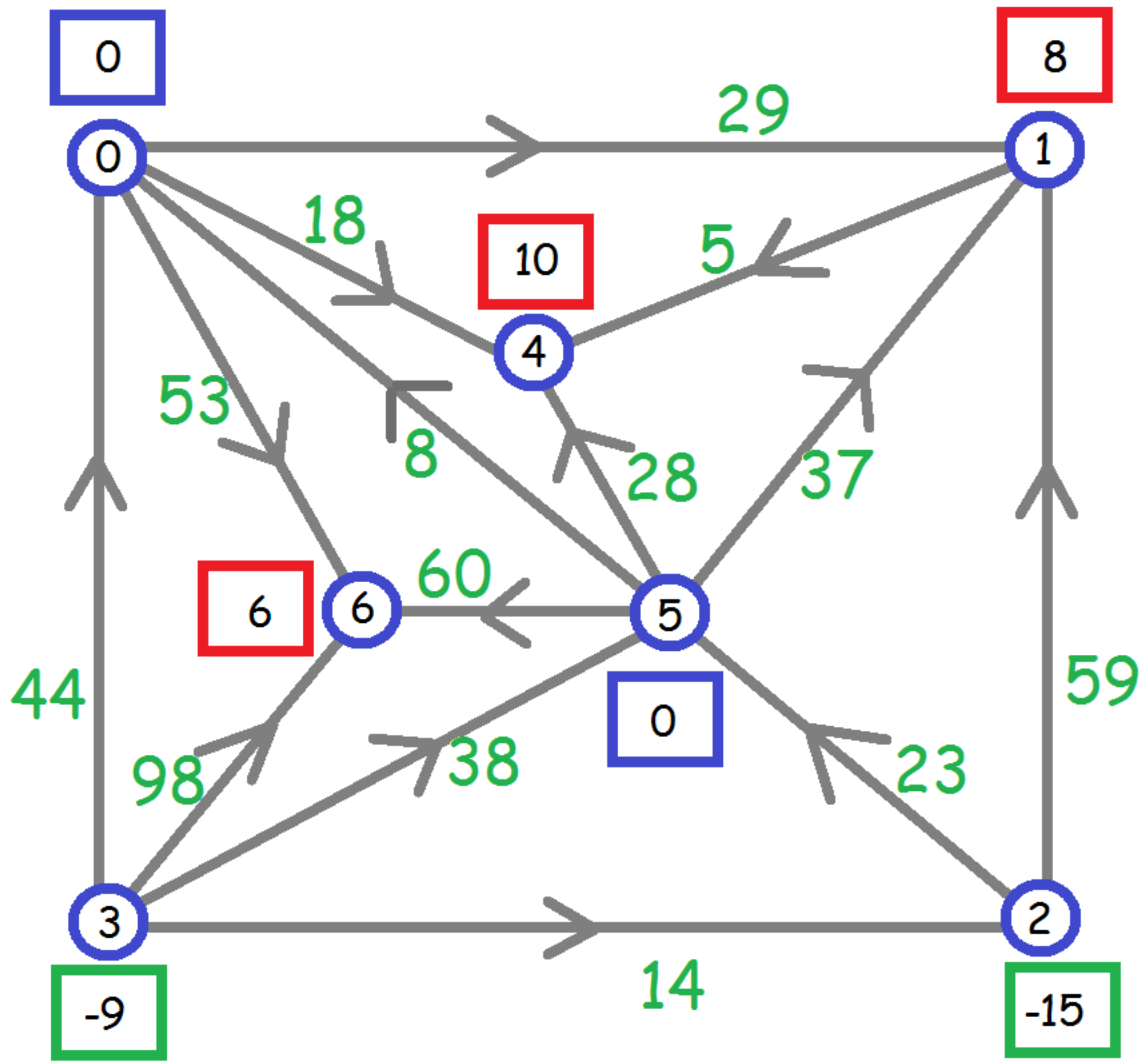
Note: there are n-1 equations and n unknowns so the fair prices are not unique. If y is a solution, then so is the vector obtained by adding a constant d to each entry of y. We arbitrarily set $y_0=0$ to get a unique solution.

Step 2: Look for an arc not in T where it pays to buy and ship. If there are none- stop, the solution is optimal.

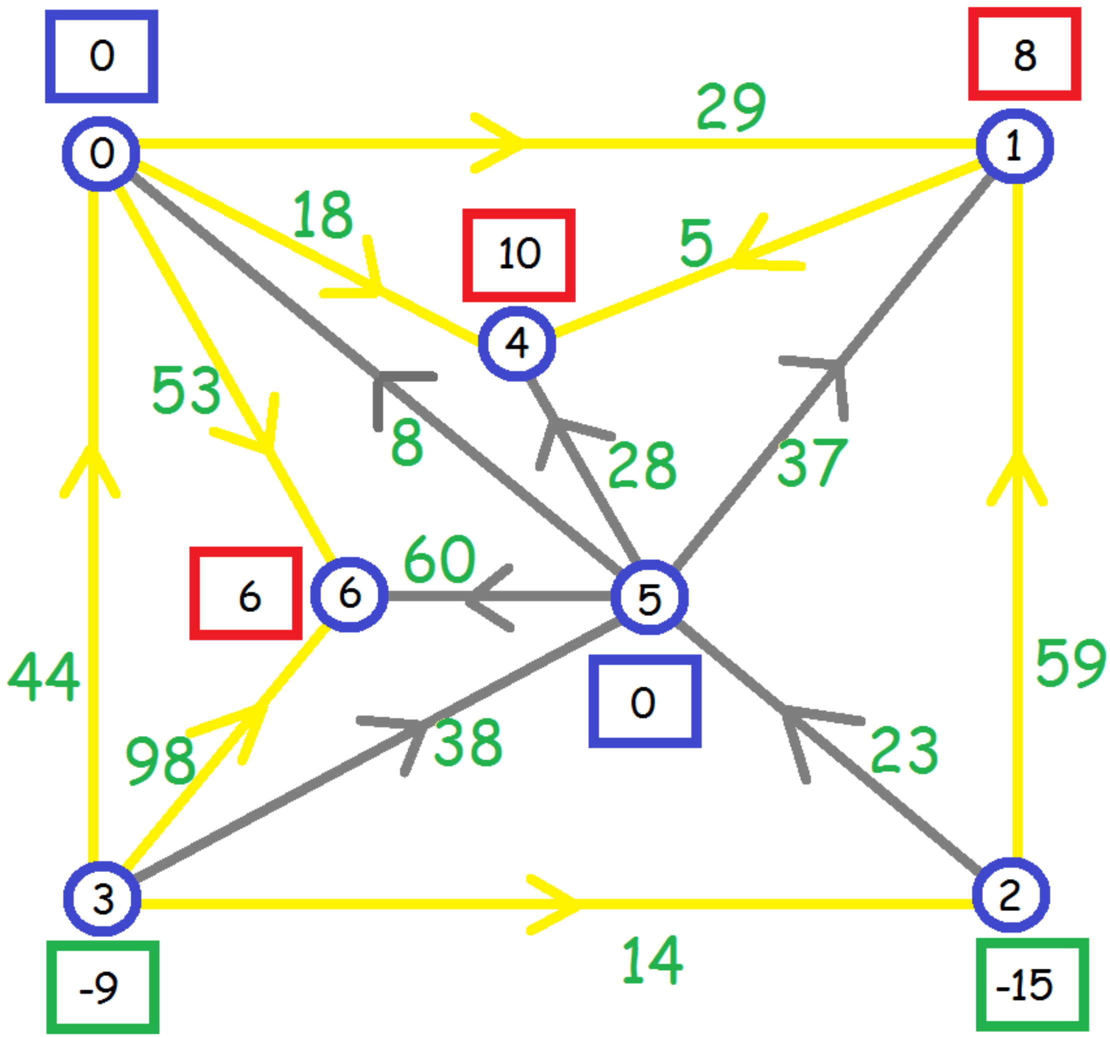
Step 3: Update the tree solution.
The new arc forms a cycle with T. Assume t units are shipped along that arc. Traverse the cycle in the direction that corresponds to the new arc adding or subtracting t from the amount shipped on each other arc to ensure conservation of flow.

Find the maximum value of t so that none of the shipped amounts are negative. The new arc is added and an arc whose shipped amount is decreased to zero is deleted.

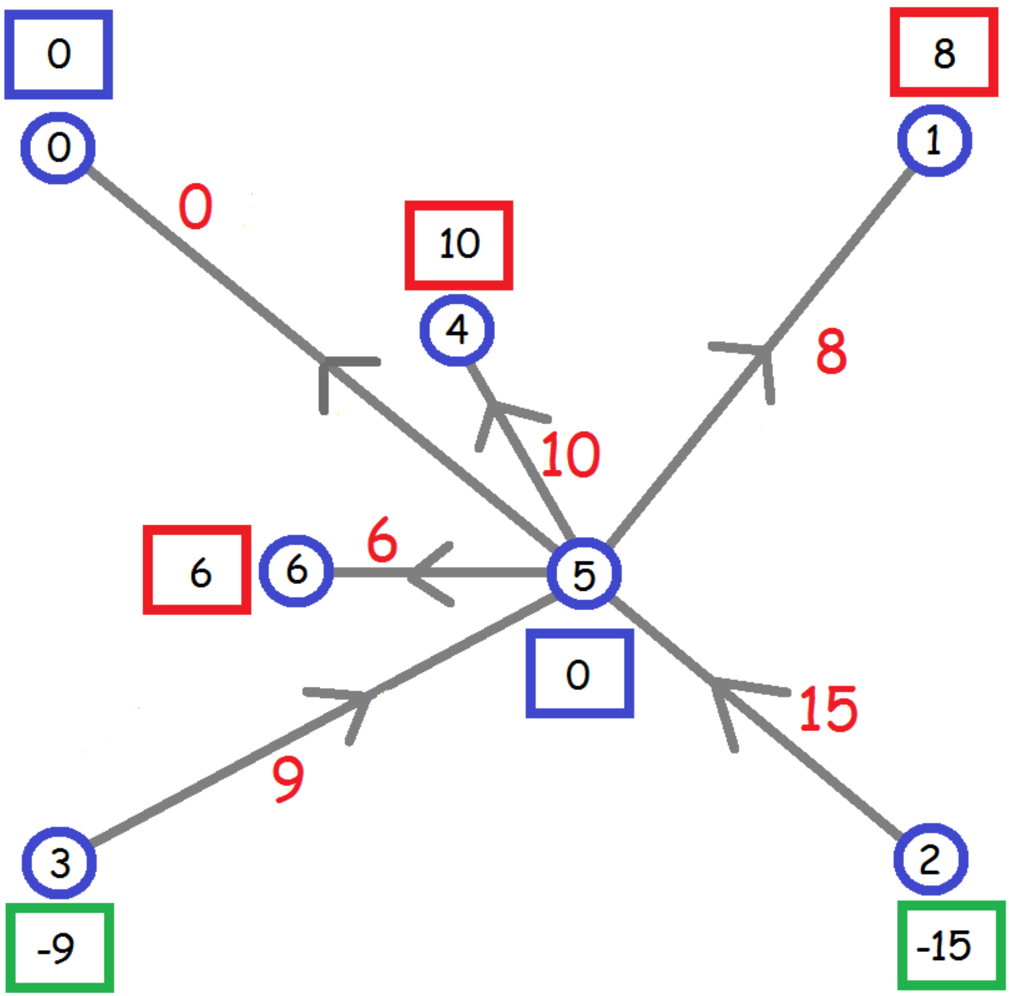
Sample problem:
The green numbers are the costs.



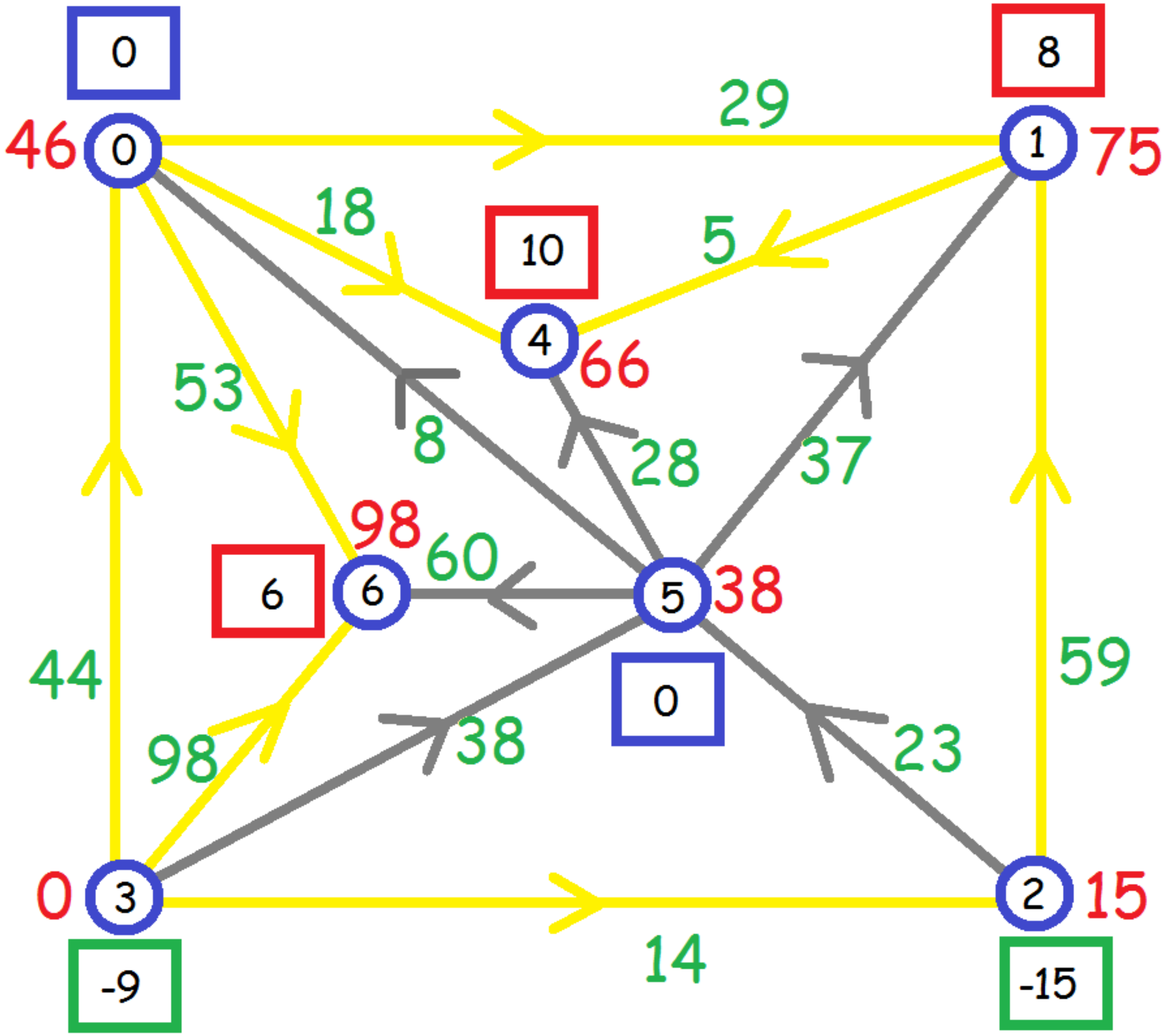
An initial tree solution.
How many units are shipped along each arc?



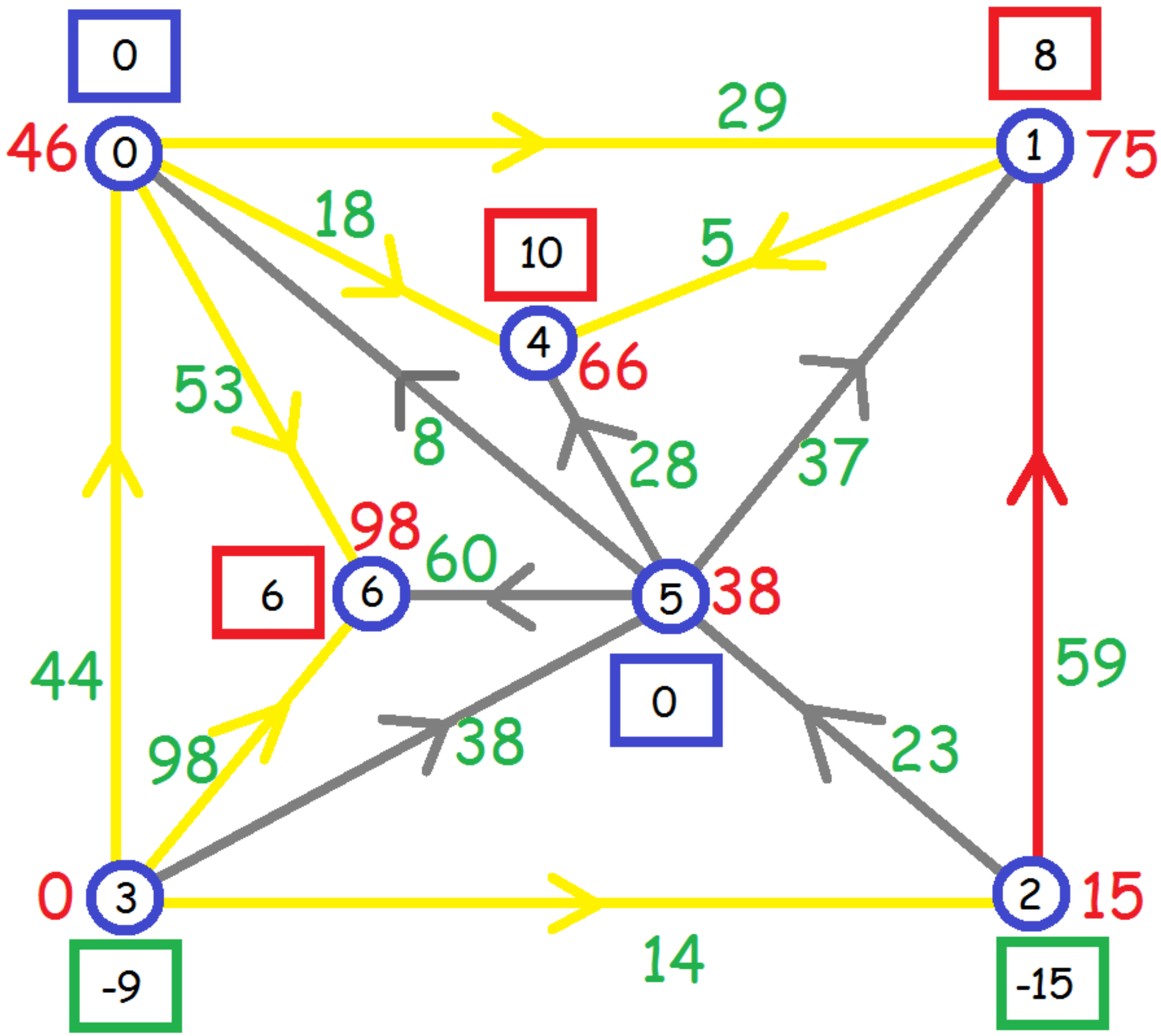
An initial tree solution.
How many units are shipped along each arc?



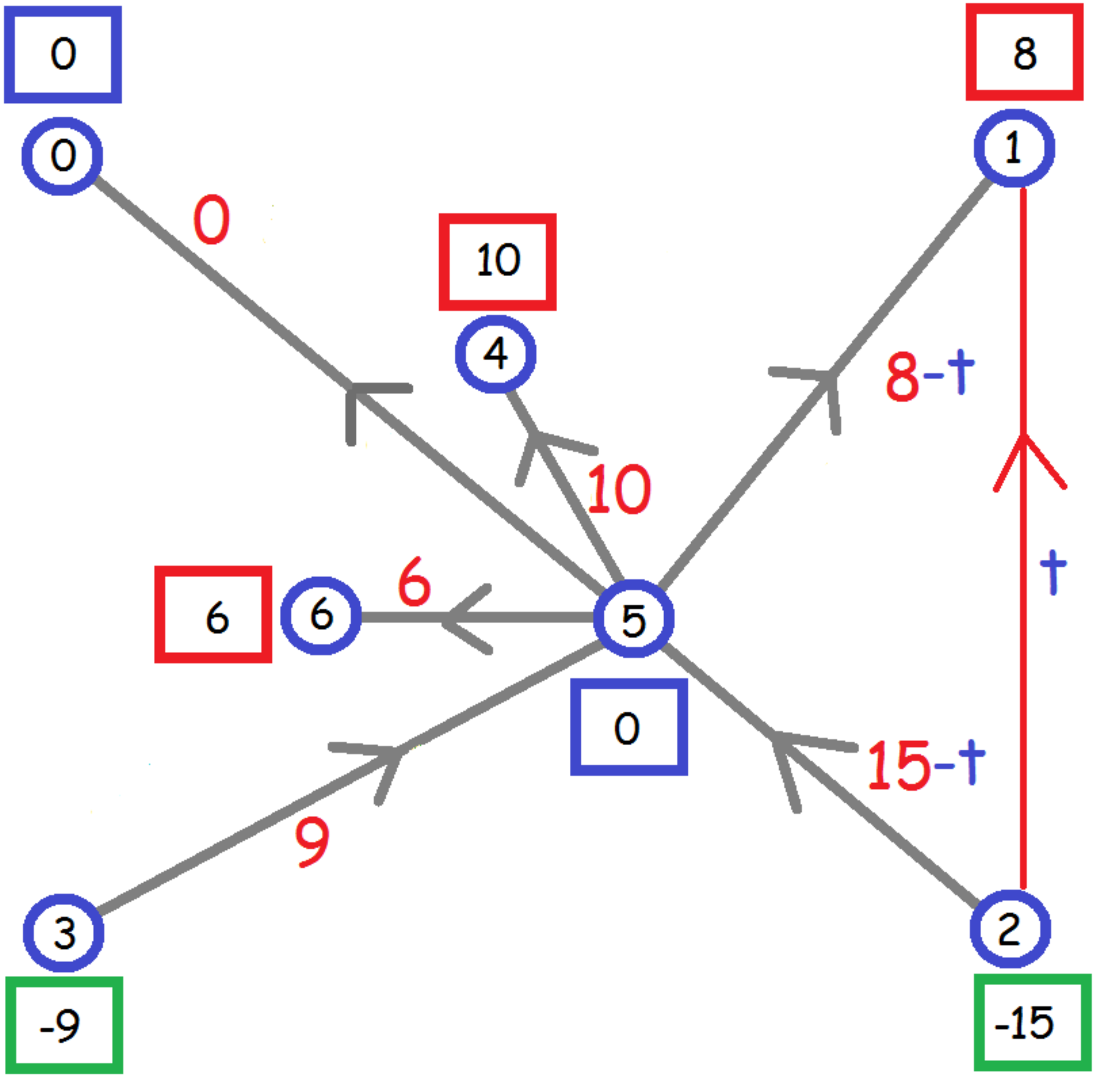
Compute fair prices (started at node 3):



Red arc: pays to buy and ship.

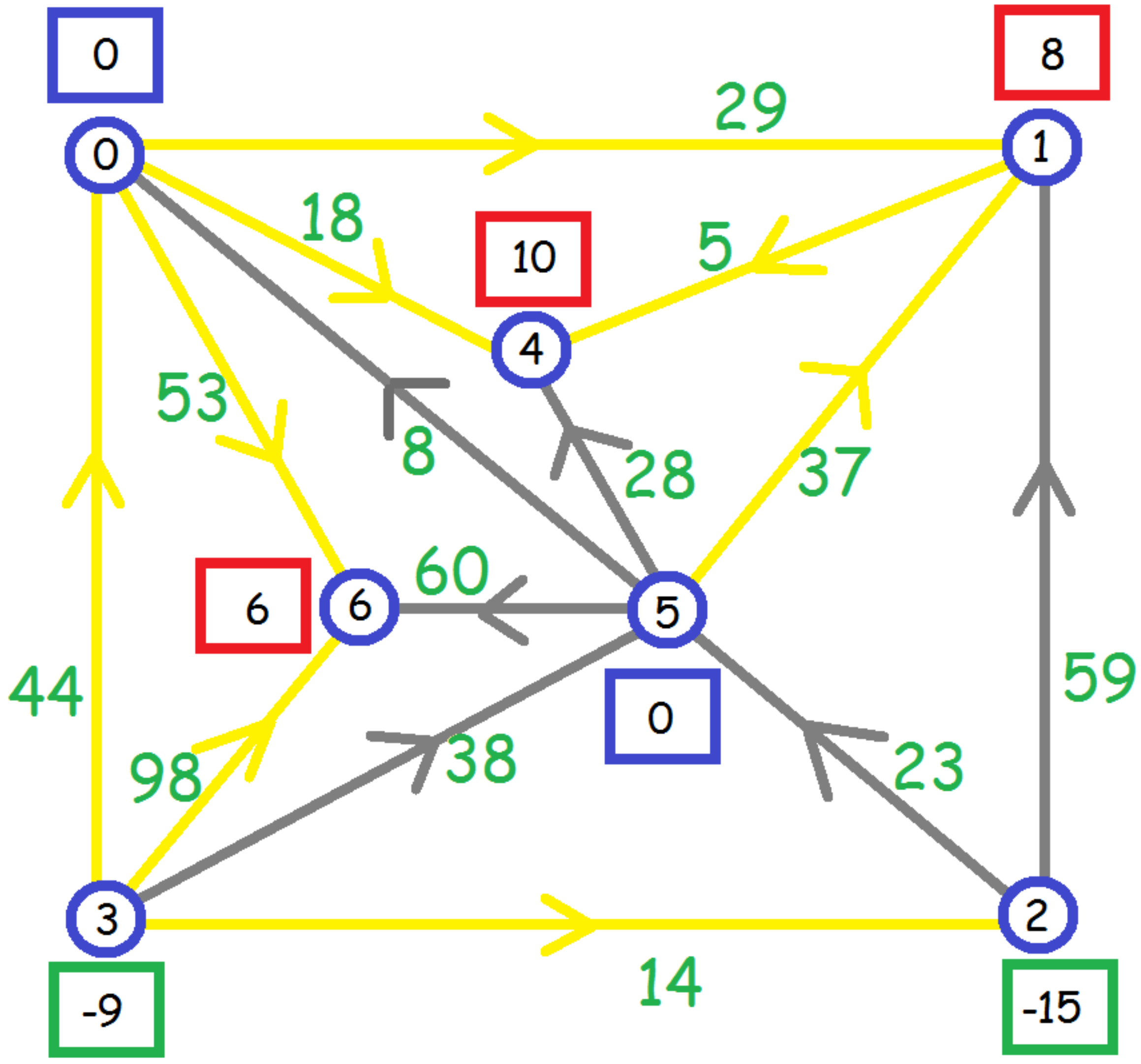


Solve for t so no negative shipments:

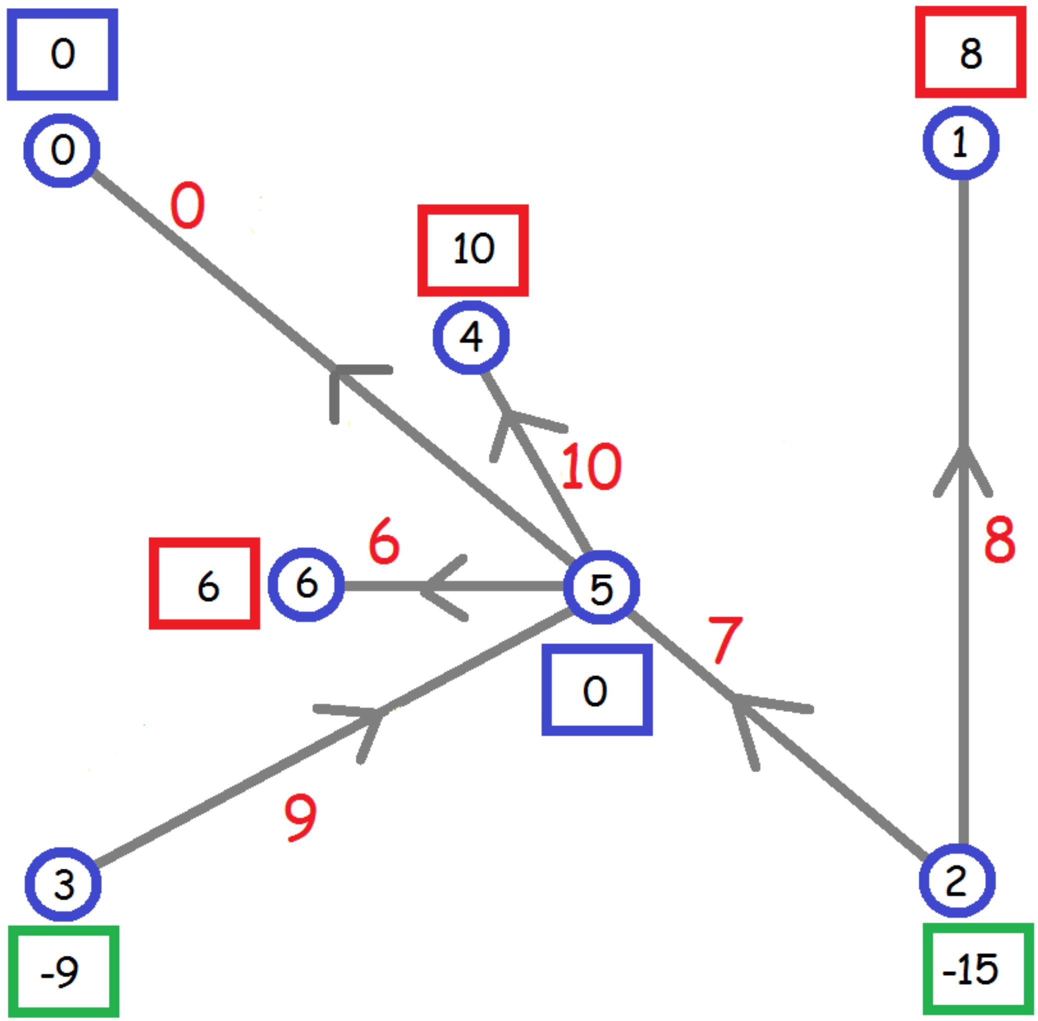


$t \le 8$

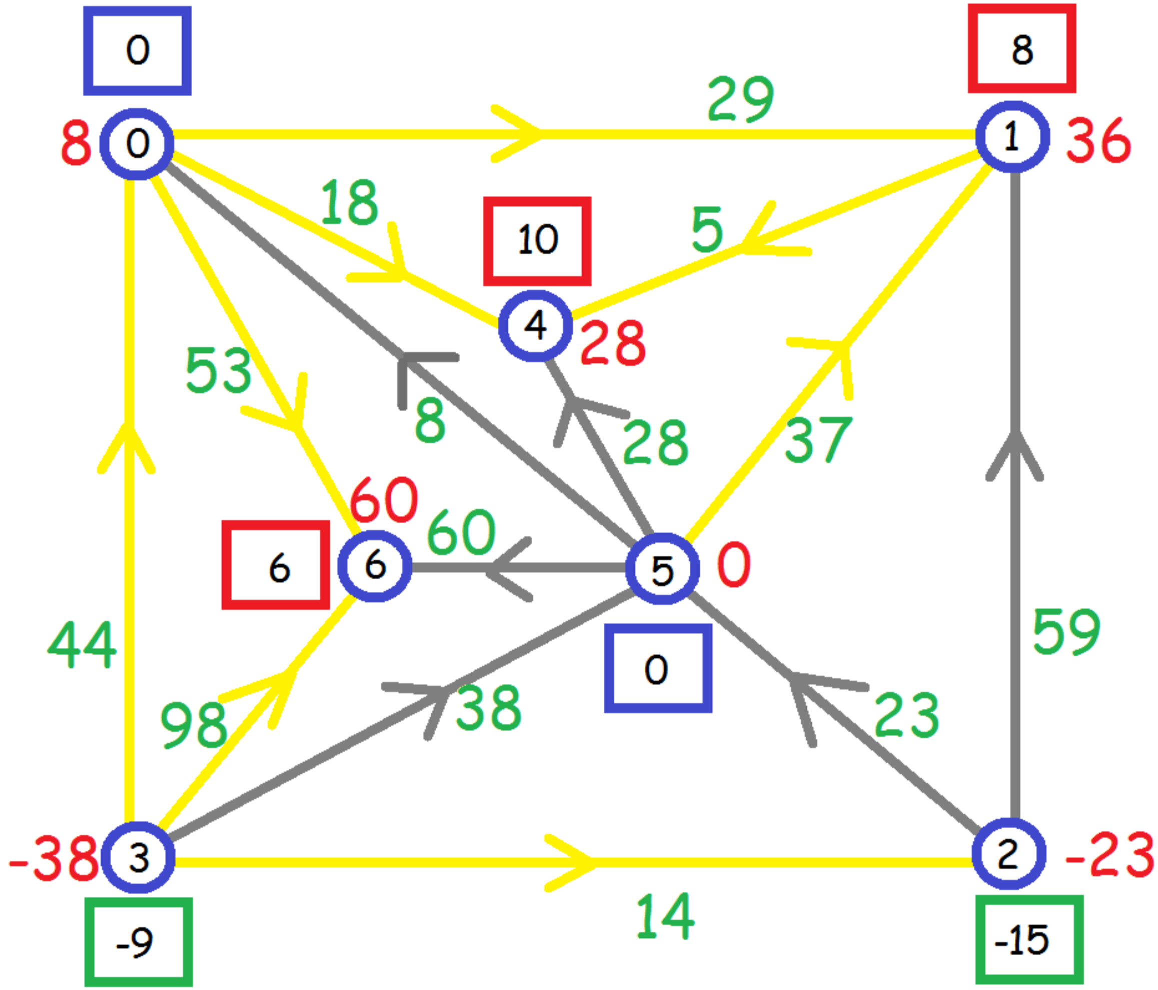
The next tree solution:



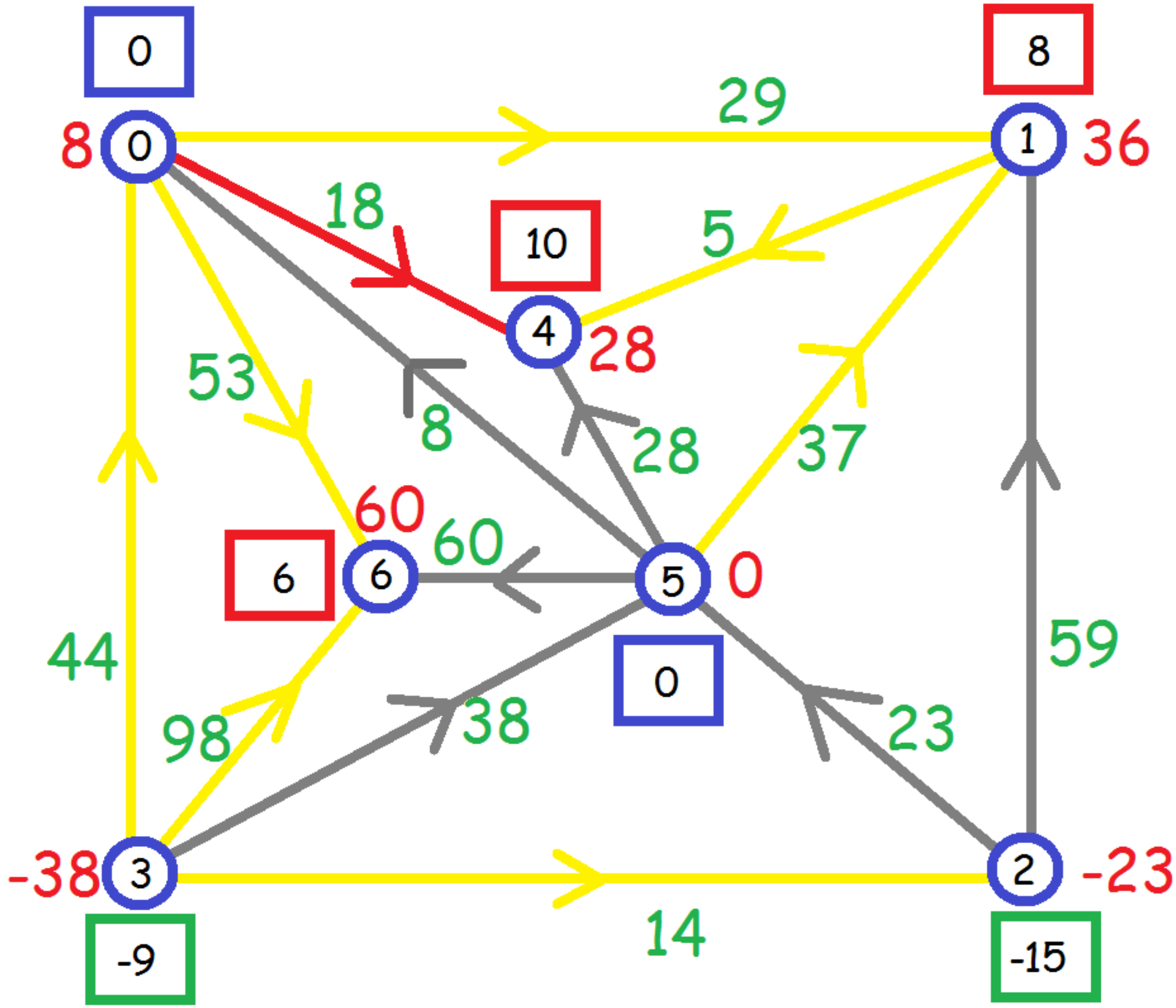
The amounts shipped:



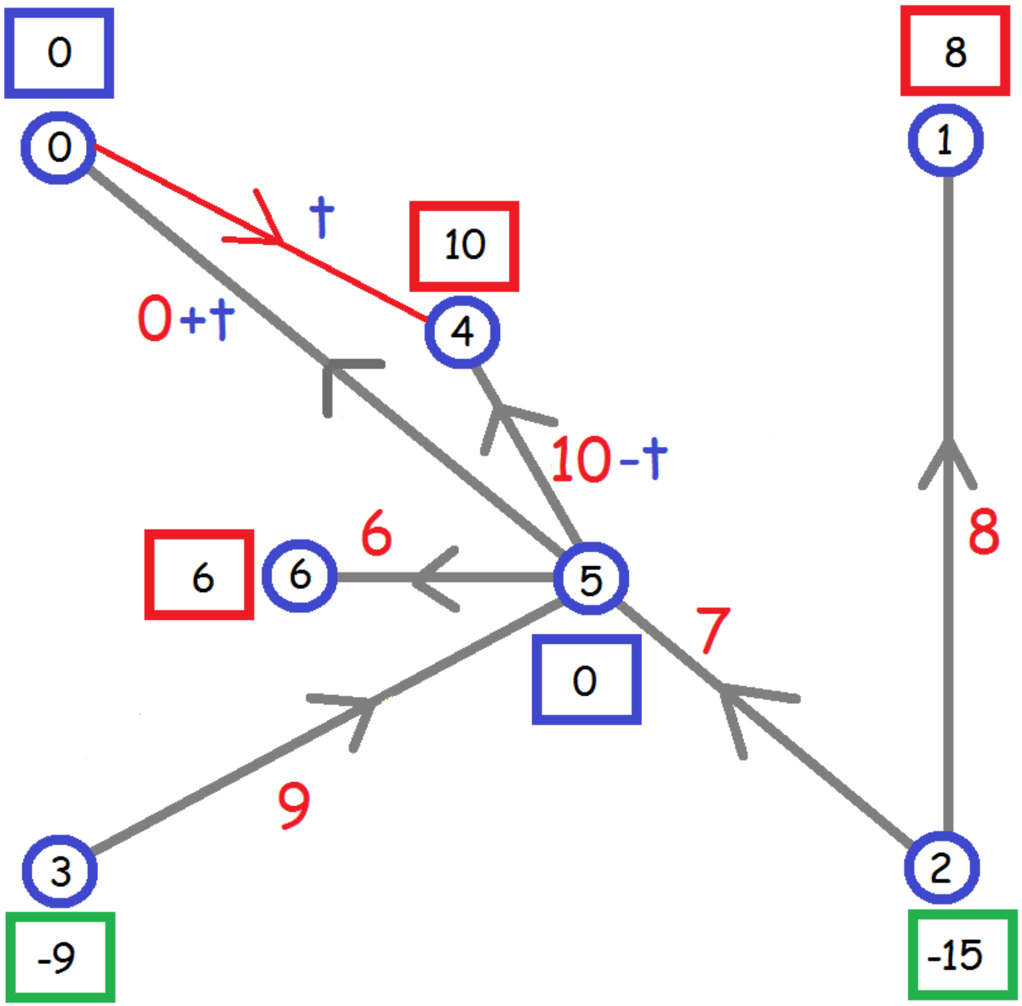
Fair prices: it does not matter where we start so just for the fun of it, I started at node 5 this time.



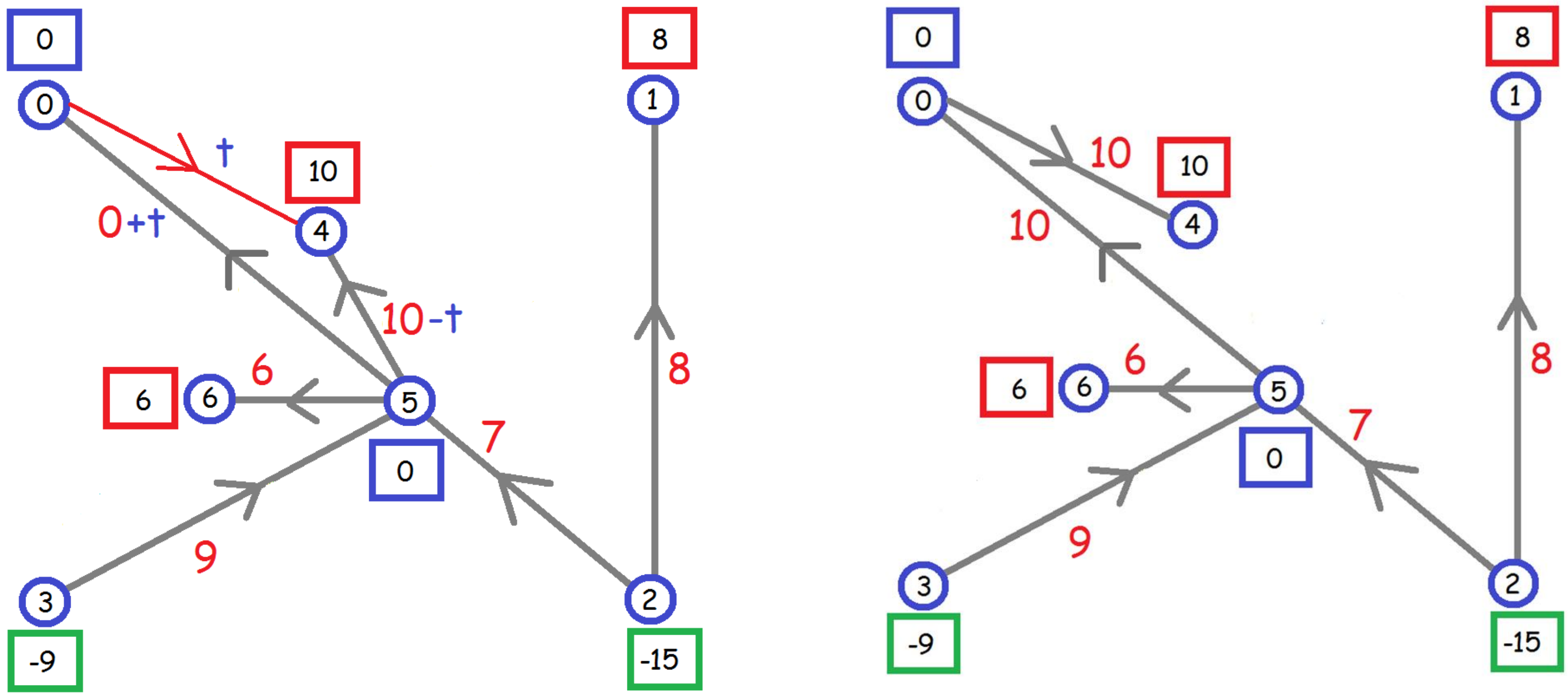
It pays to buy and ship on red arc:



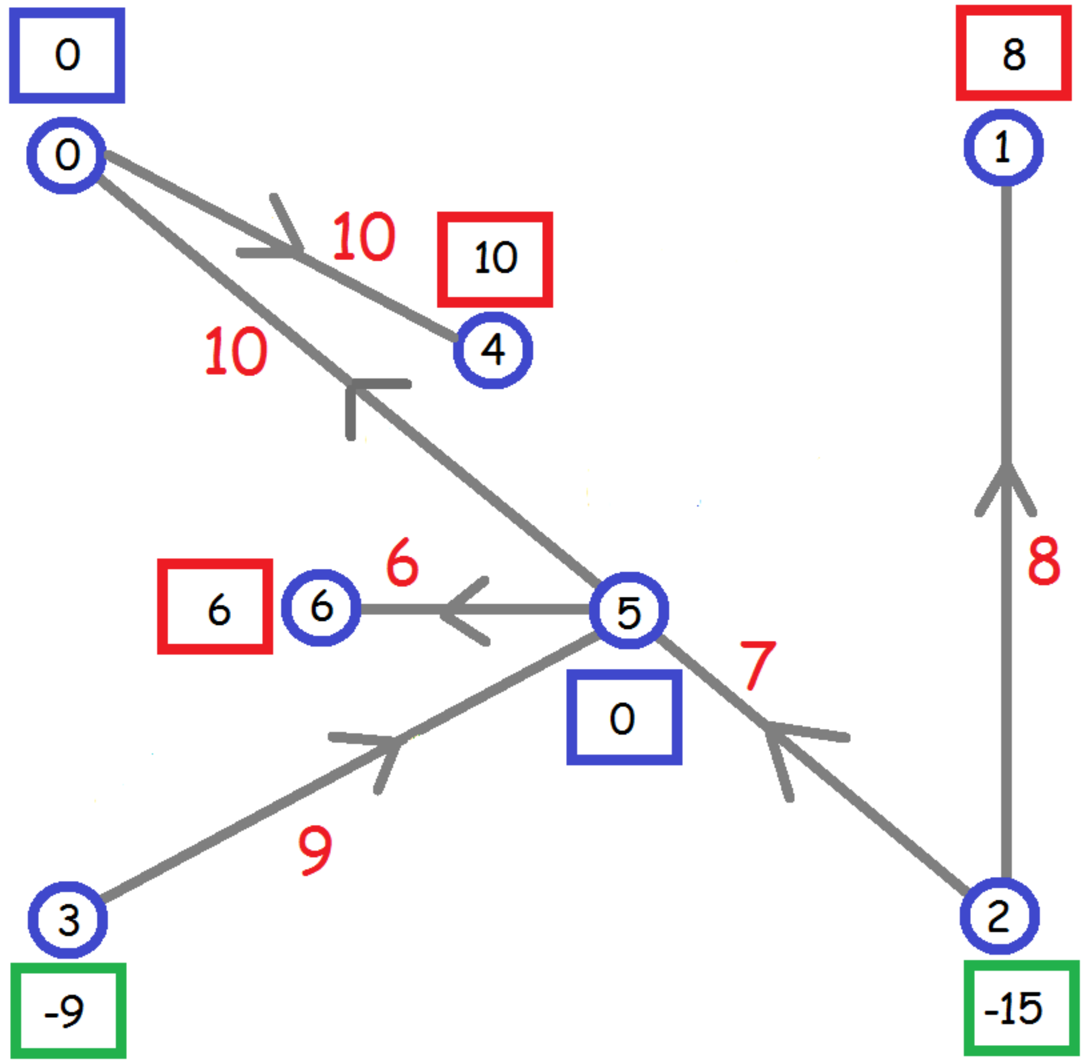
Solve for t so shipped amounts are non-negative: t ≤ 10.



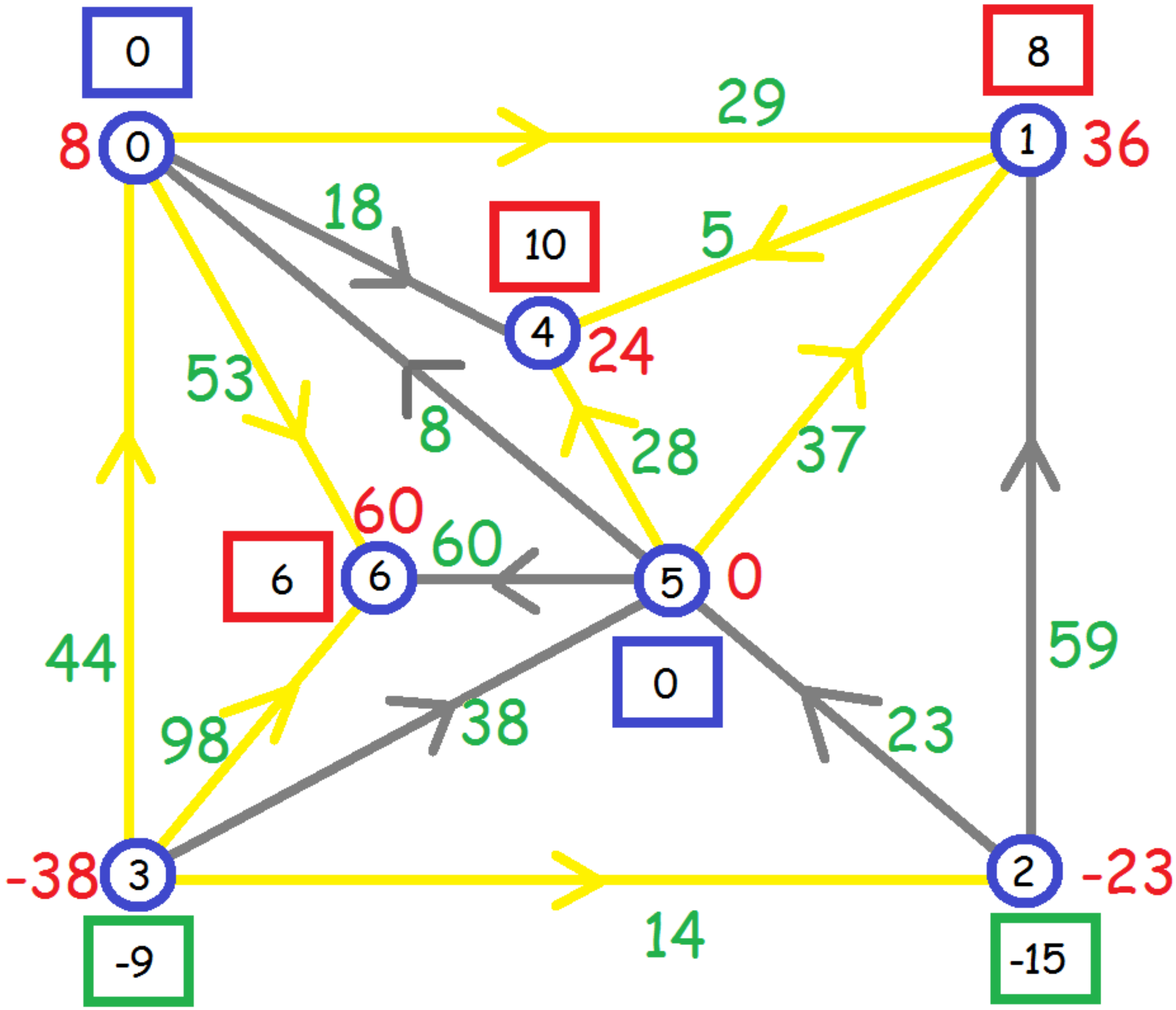
Solve for t so shipped amounts are non-negative: t ≤ 10.



The amounts shipped:

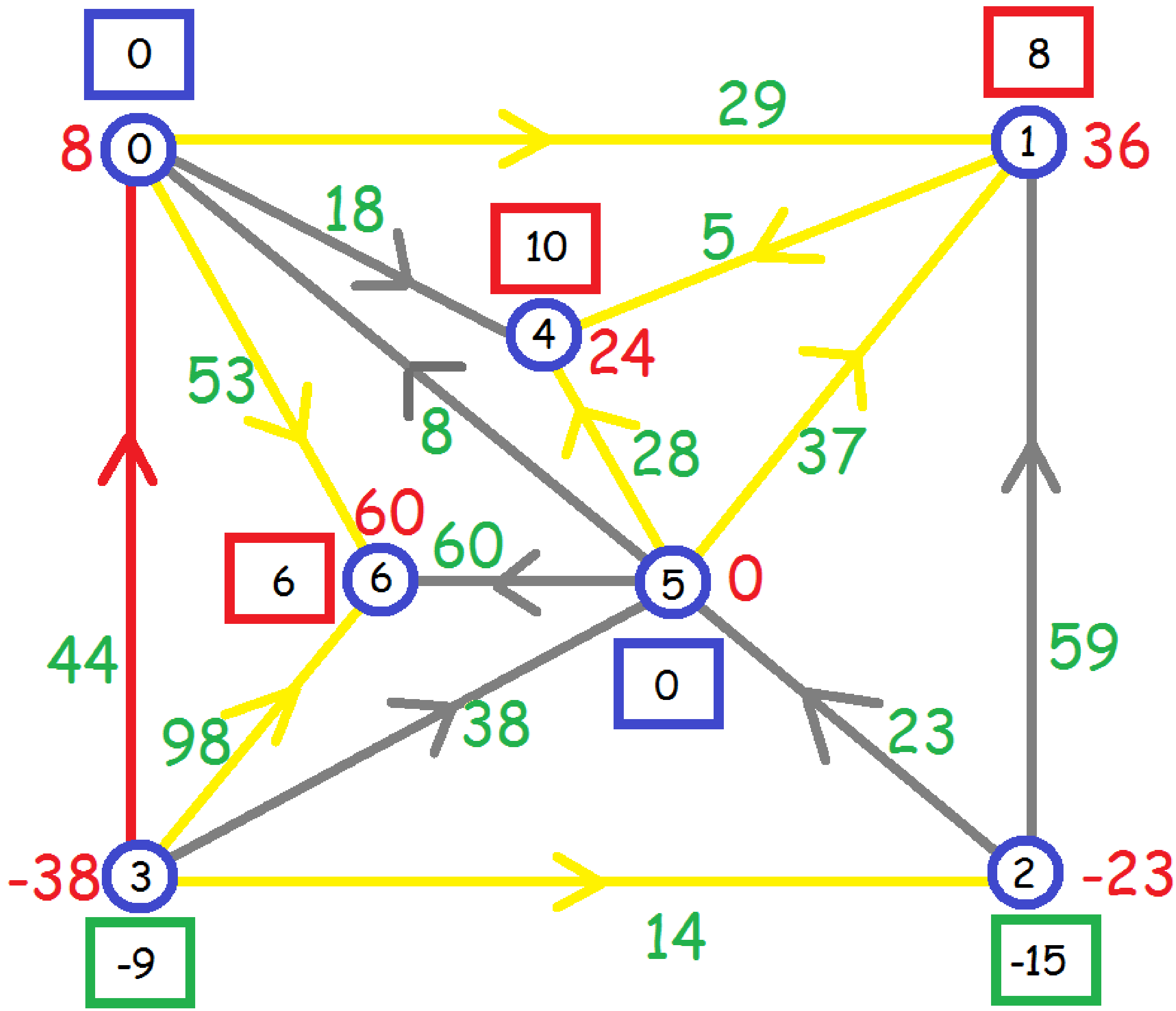


Fair prices

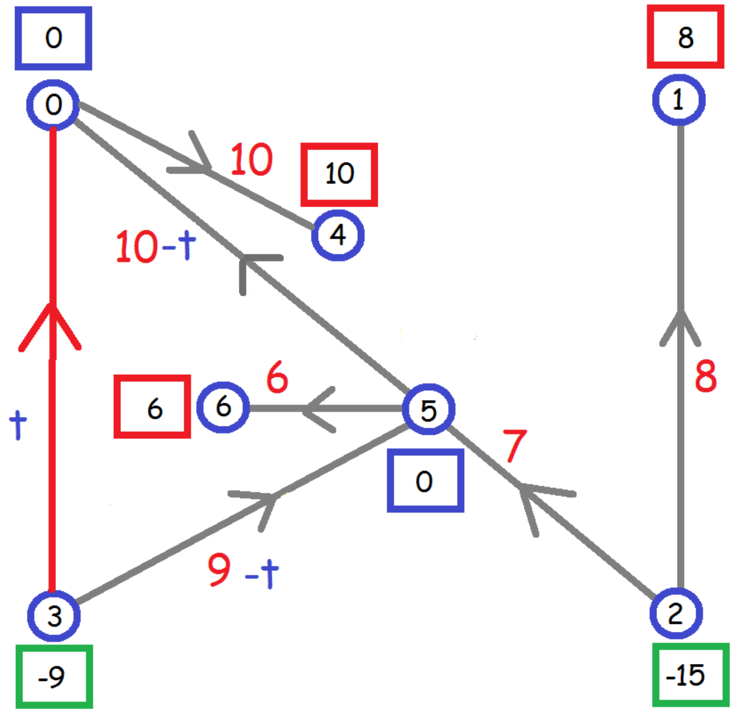


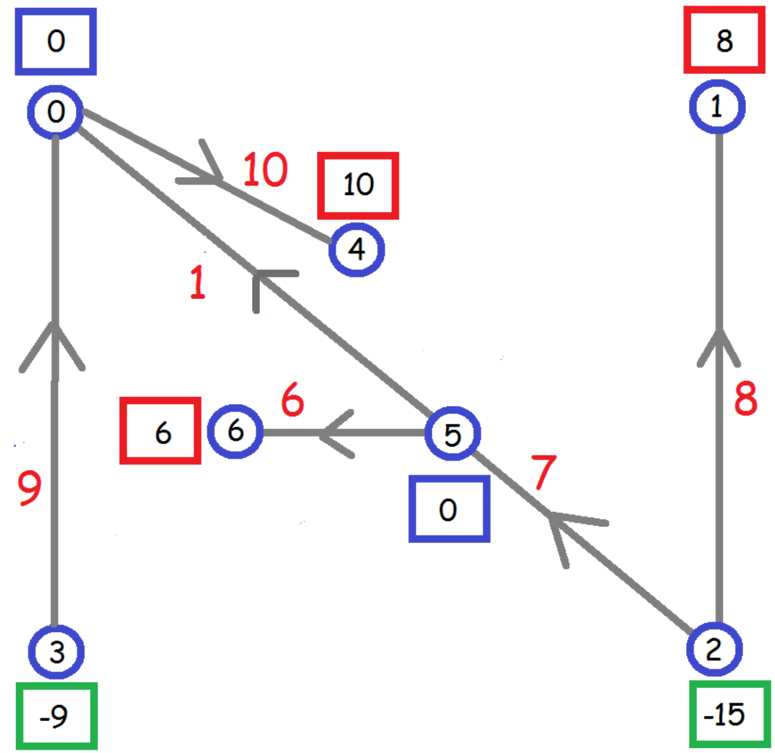
0
8
10
6
0
-9
-15
29
18
5
53
8
28
37
60
44
98
38
23
59
14
8
36
24
60
0
-38
-23
0
1
2
3
4
5
6

It pays to buy and ship on red arc:



Solve for t so shipped amounts are non-negative: $t \le 9$.





We can't undercut these prices: Optimal!