15.082 and 6.855J April 3, 2003

Introduction to Minimum Cost Flows

The Minimum Cost Flow Problem

u_{ij} = capacity of arc (i,j).

c_{ij} = unit cost of shipping flow from node i to node j on (i,j).

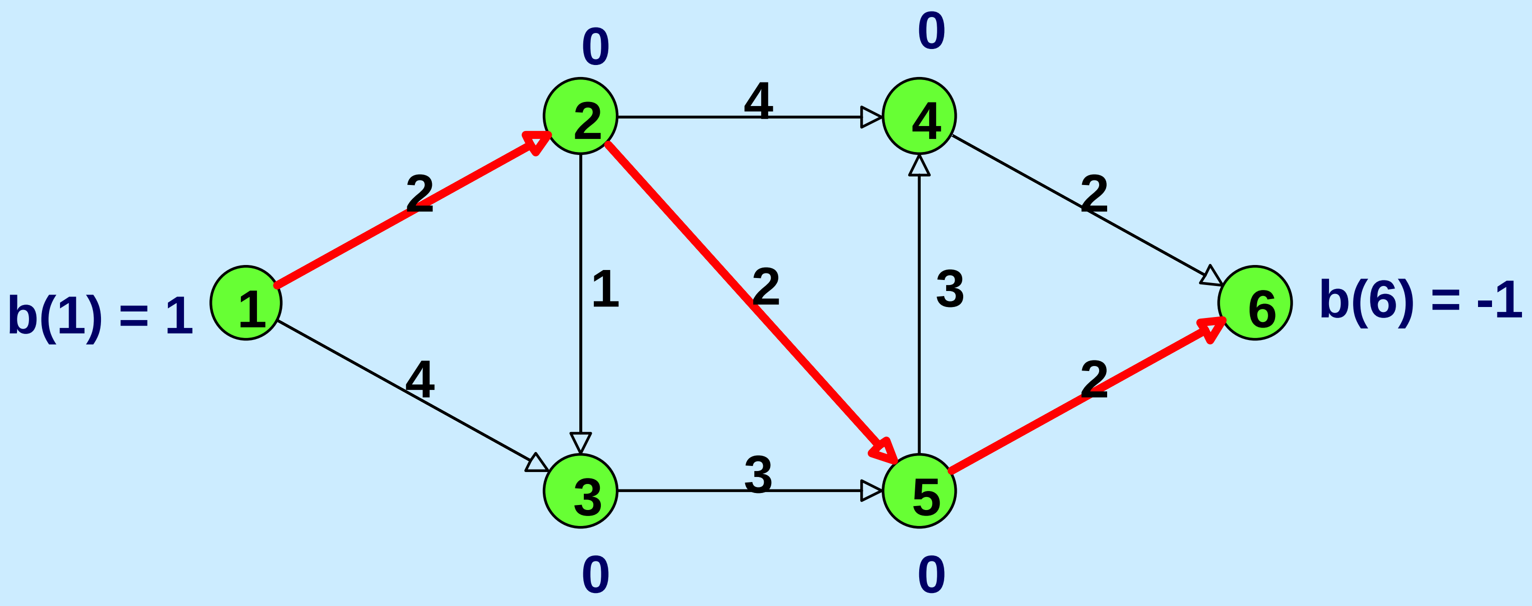
x_{ij} = amount shipped on arc (i,j)

Minimize $\sum_{(i,j)\in A} c_{ij}x_{ij}$

$\sum_j x_{ij} \quad - \quad \sum_k x_{ki} = b_i$ for all $i \in N$.

and $0 \le x_{ij} \le u_{ij}$ for all $(i,j) \in A$.

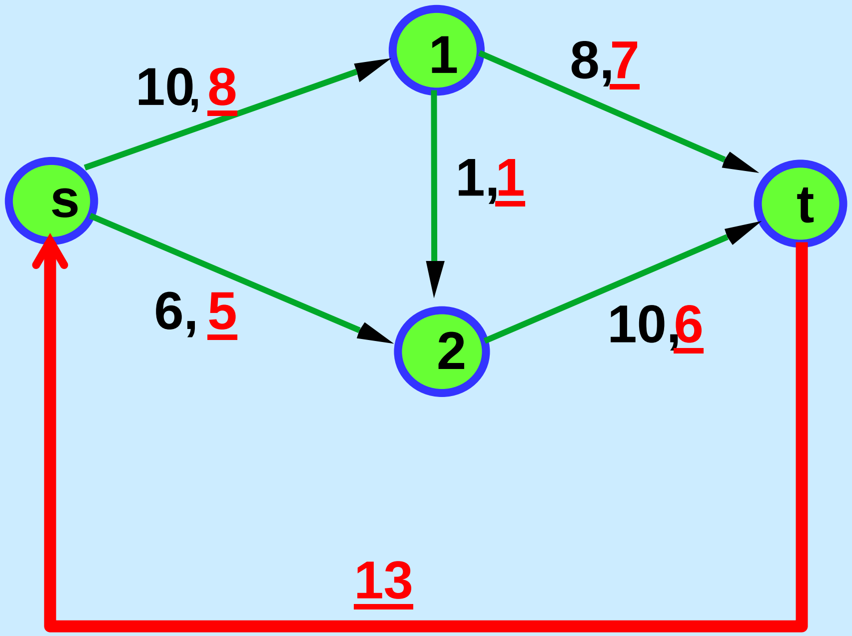
Find the shortest path from node 1 to node 6



The optimal flow is to send one unit of flow along 1-2-5-6.

This transformation works so long as there are no negative cost cycles in G.
(What if there are negative cost cycles?)

Find the Maximum Flow from s to t



b(i) = 0 for all i;

add arc (t,s) with a cost of -1 and large capacity.

The cost of every other arc is 0.

The optimal solution in the corresponding minimum cost flow problem will send as much flow in (t,s) as possible.

Transshipment Problems

Plants with given production capabilities for a product.

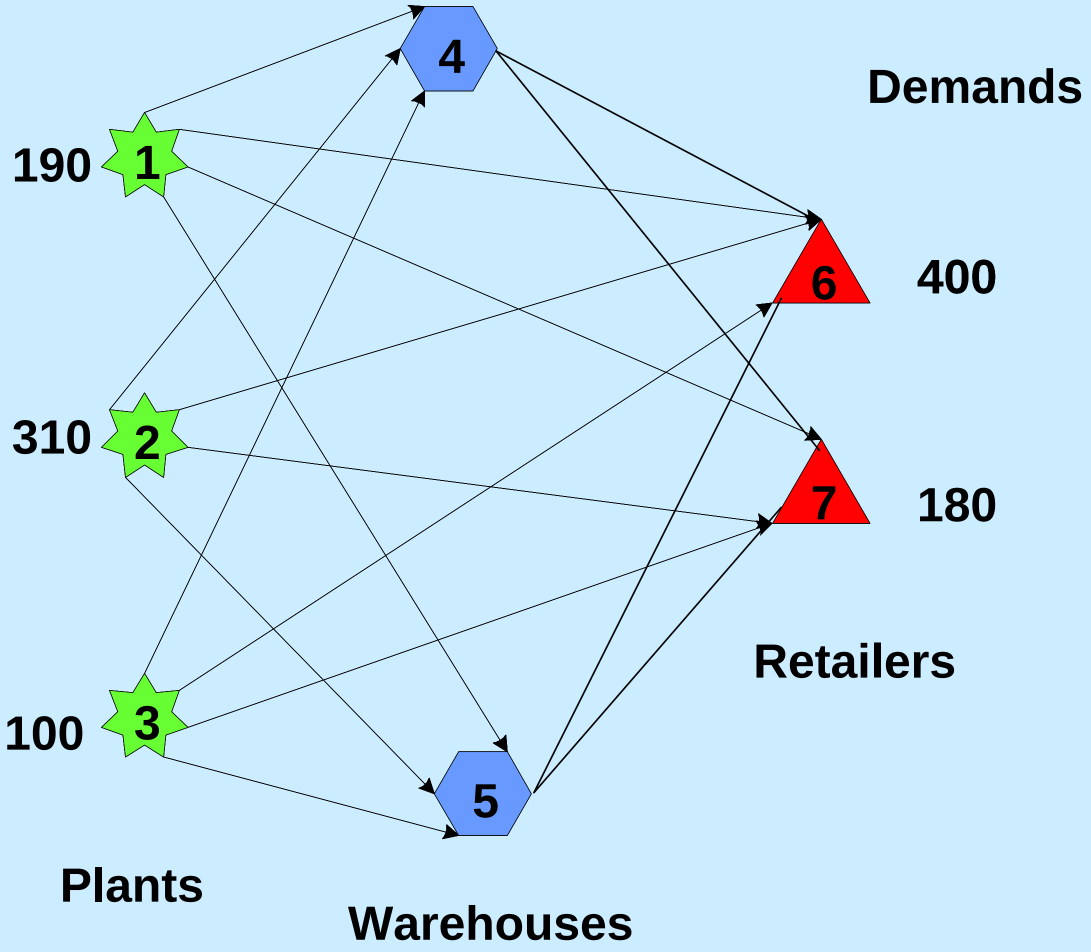
One can ship directly from the plants to retailers, or from plants to warehouses, and then from warehouses to retailers.

There is a given demand for each retailer.

Costs of shipment are given.

What is the minimum cost method for satisfying demands?

A Network Representation



The Caterer Problem

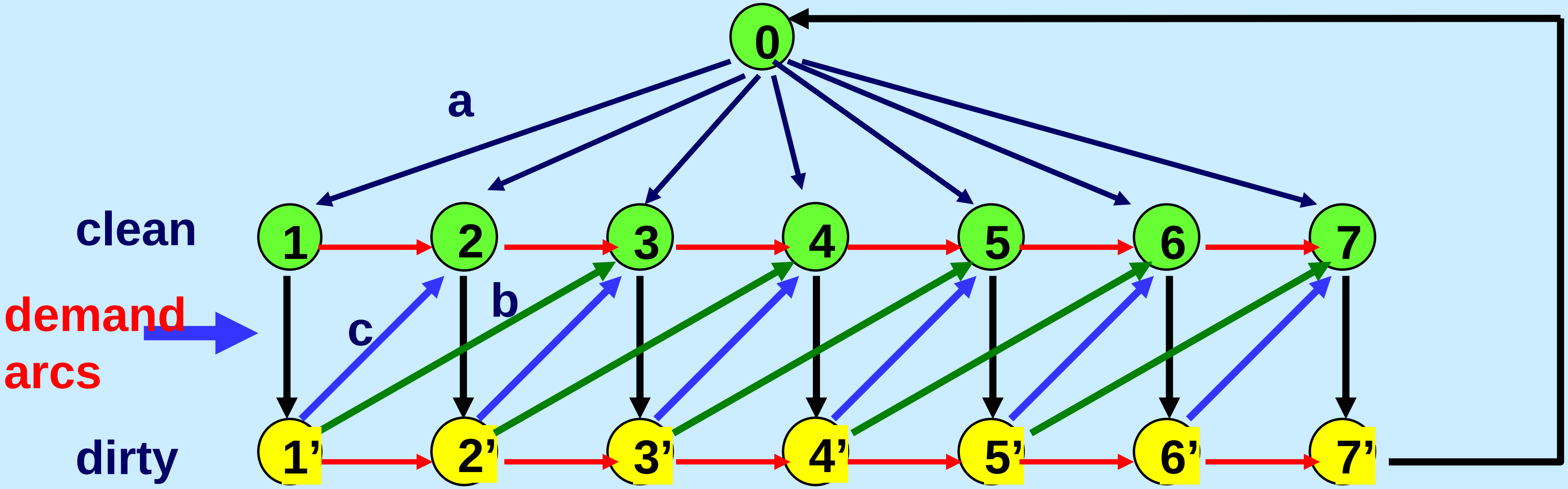
Demand for d_i napkins on day i for i = 1 to 7 (so, j $\in$ [1..7]).

Cost of new napkins: *a* cents each,

2-day laundry: b cents per napkin

1-day laundry: c cents per napkin.

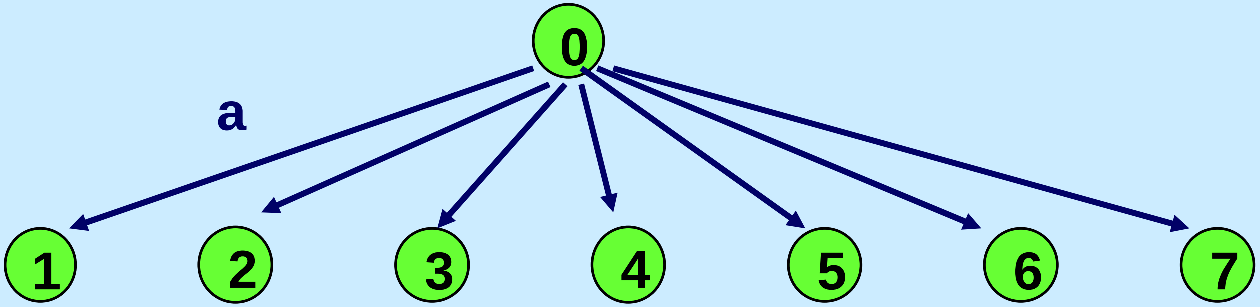
Minimize the cost of meeting demand.



Purchase arcs

In any period of the seven periods, one can purchase napkins, at a cost of a cents per napkin.

🟢 clean napkins

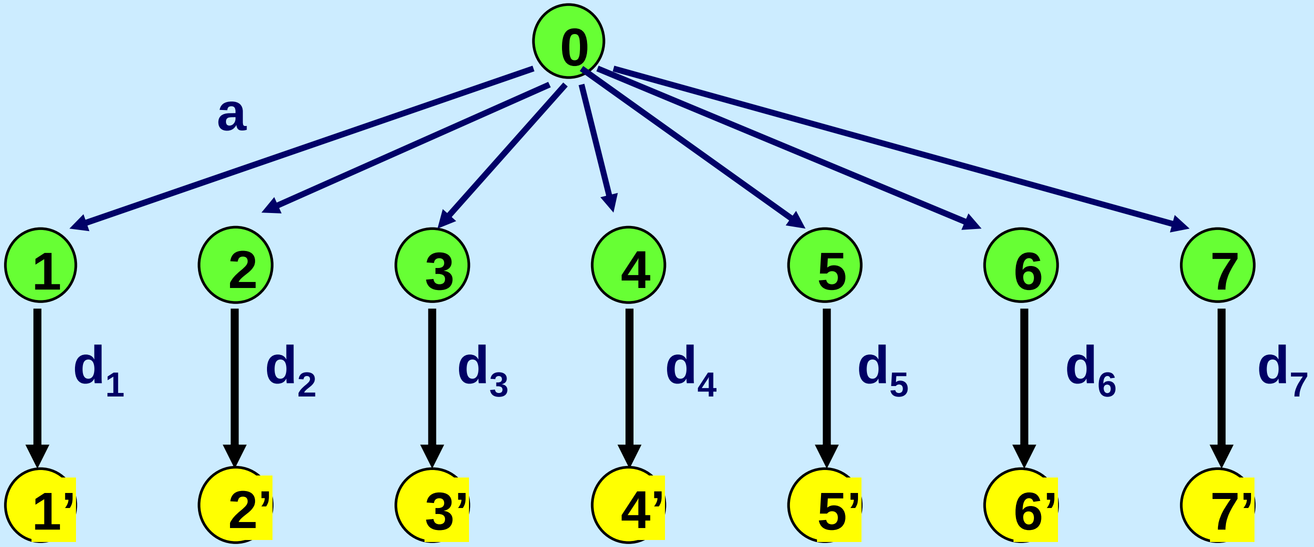


Demand Arcs

You must use d_i napkins on day i

🟡 dirty napkins

lower bound on flows



The rest of the arcs

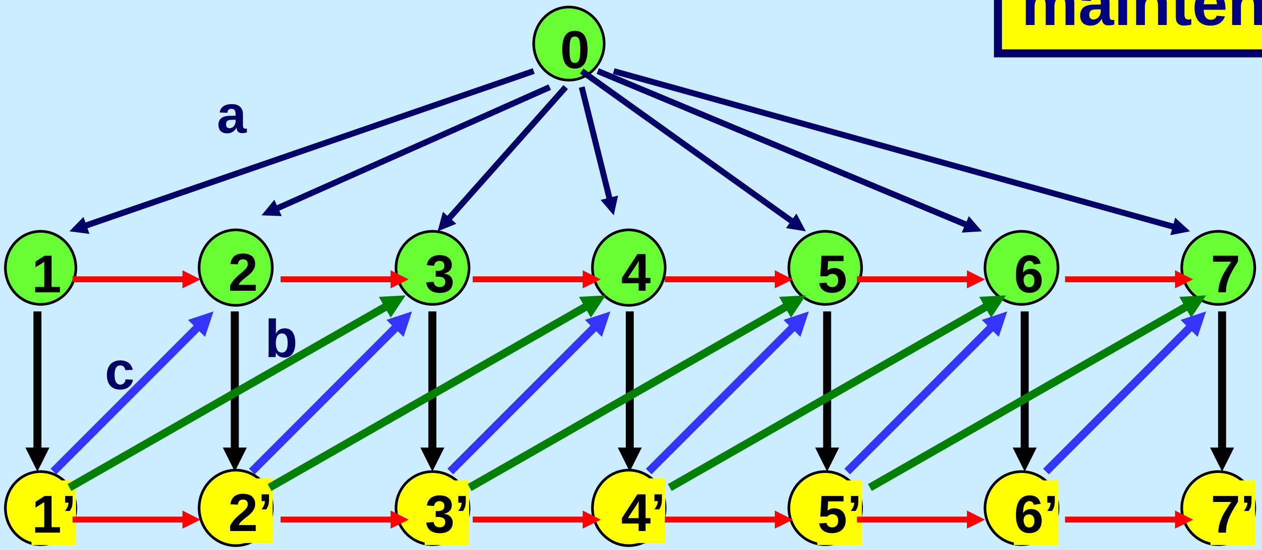
You may launder napkins in 2 days at b cents each

You may launder napkins in 1 day at c cents each

You may store clean napkins for free

You may store dirty napkins for free

Application to airplane maintenance.



Some Assumptions

1. **All data is integral. (Needed for some proofs, and some running time analysis).**

2. **The network is directed.**

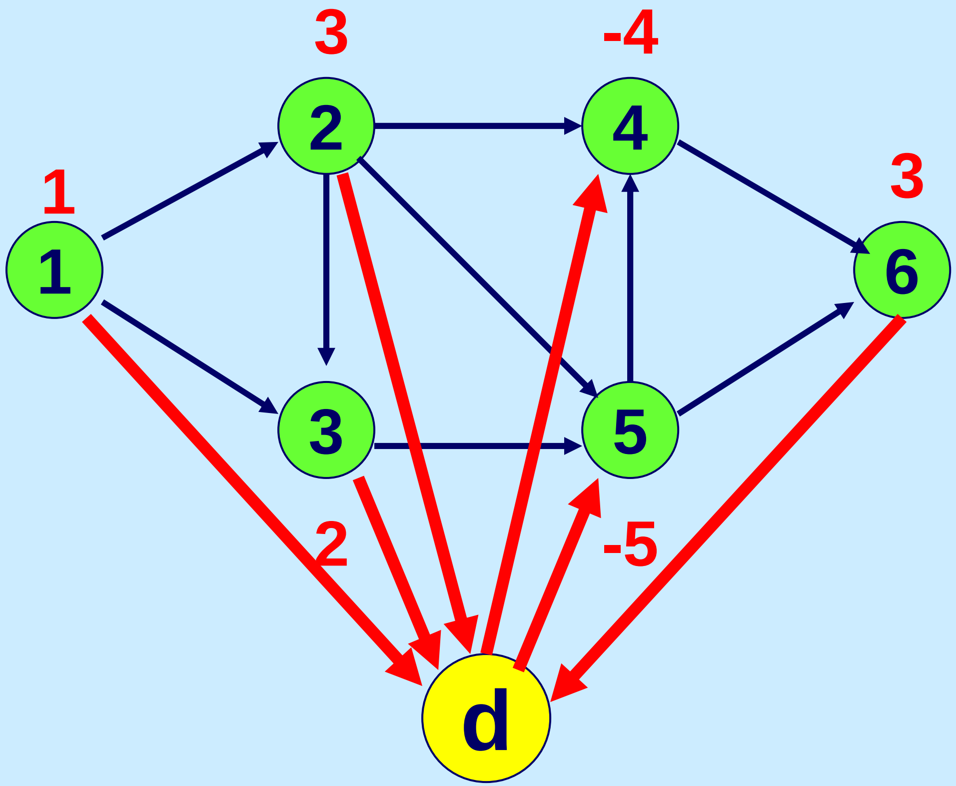
3. **$\sum_{i=1 \text{ to } n} b(i) = 0$.**
   (Otherwise, there cannot be a feasible solution)

4. **There is a feasible solution (see next slide)**

Artificial Solutions

To create a feasible solution, add a dummy node d.

Add an arc from d to each demand node, each with a large cost M, and large capacity.



Add an arc to d from each supply node, each with a large cost M, and a large capacity.

In an optimal solution, arcs with large cost will have a flow of 0.

Overview of the solution procedure

Reduced costs

- recall replacing $c_{ij} - \pi_i + \pi_j$ for the shortest path problem. The same transformation is very useful for min cost flow algorithms.

Optimality conditions

- Most iterative optimization algorithms stop when "optimality conditions are satisfied". We describe optimality conditions for the min cost flow problem.

Residual network

- Just as in max flows, we run most algorithms on the residual network.

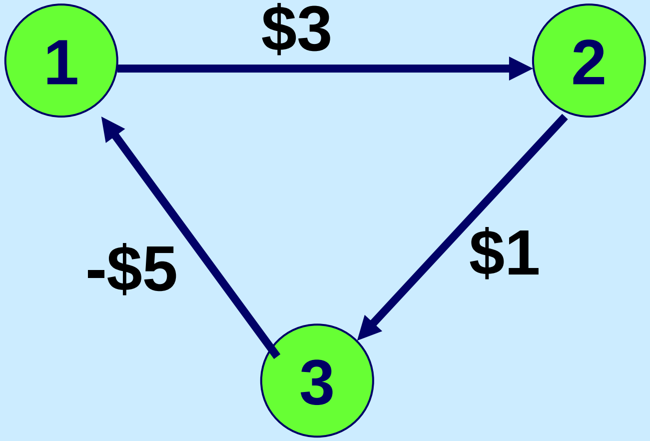
Reduced Costs

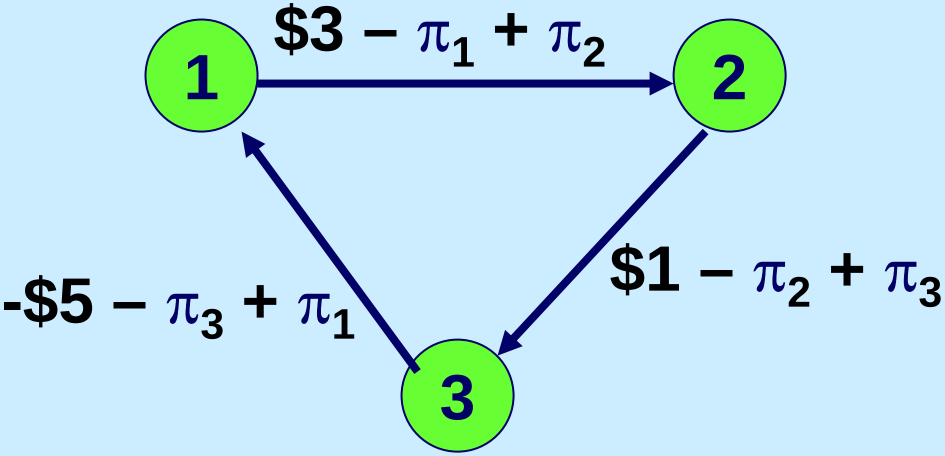
Let π_i denote the node potential (or dual price) for node i.

$$c_{ij}^{\pi} = c_{ij} - \pi_i + \pi_j$$

For unit of flow out of node i, subtract π_i from the cost

For unit of flow into node j, add π_j to the cost.





More on Using Reduced Costs

Claim: **For any feasible solution x, cx - c^πx = πb. Thus a feasible flow x that minimizes cx will also minimize c^πx.**

Proof:

$$cx - c^\pi x = \sum_{(i,j)\in A} (\pi_i - \pi_j) x_{ij}$$

$$= \sum_i \pi_i \left(\sum_j x_{ij} - \sum_j x_{ji}\right)$$

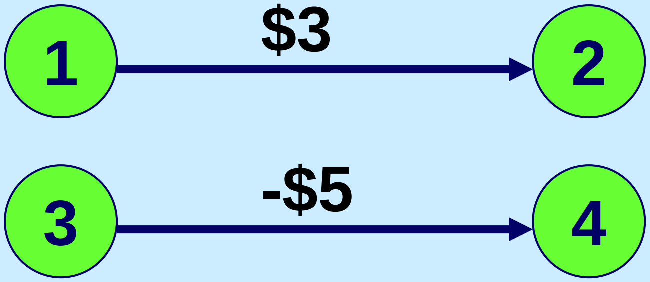
$$= \sum_i \pi_i b_i = \pi b$$

Optimality Conditions

Let x be a flow and let π be a vector of node potentials.

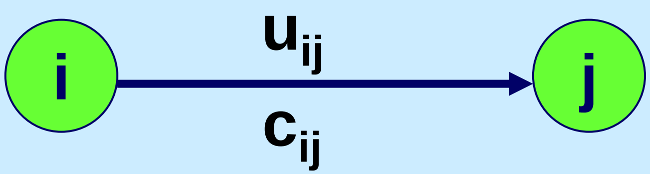
The pair (x, π) is optimal if it satisfies the following:

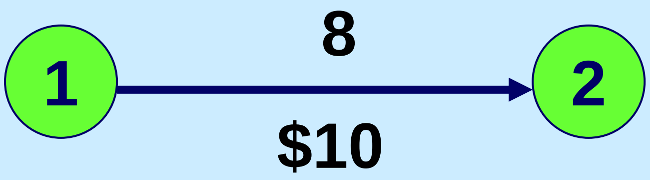
1. x is a feasible flow
2. If $c^{\pi}_{ij} > 0$, then $x_{ij} = 0$
3. If $c^{\pi}_{ij} < 0$, then $x_{ij} = u_{ij}$



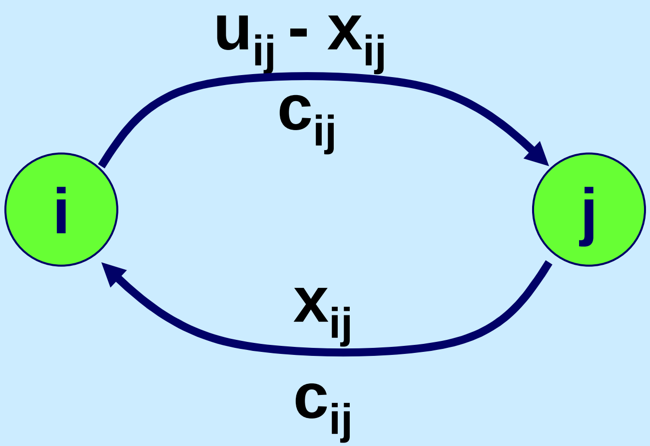
If x, π satisfy (1) – (3), we say that x is an optimal flow, and π is an optimal set of node potentials.

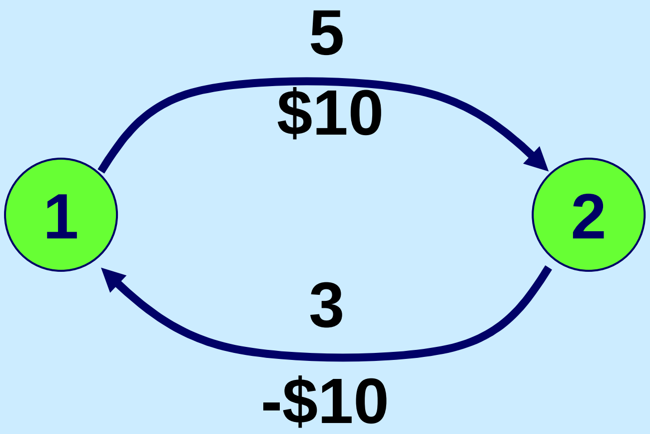
The Residual Network G(x)





Suppose $x_{12} = 3$



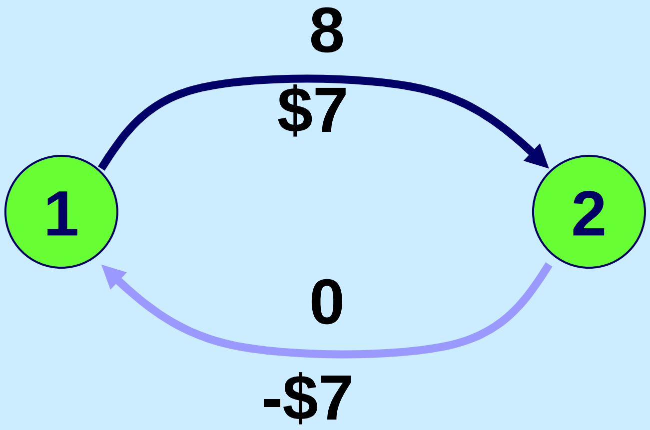


Optimality Conditions for G(x)

Every arc in the residual network has a non-negative reduced cost.

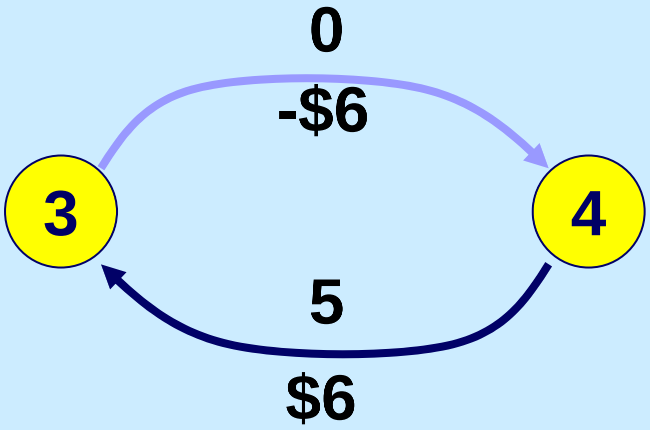
If $c^{\pi}_{ij} > 0$, then $x_{ij} = 0$

Then $x_{12} = 0$



If $c^{\pi}_{ij} < 0$, then $x_{ij} = u_{ij}$

Then $x_{34} = 5$



Other optimality conditions for G(x)

A feasible flow x is optimal for the minimum cost flow problem if and only if there is no negative cost cycle in the residual network G(x).

Negative Cycle Algorithm

```
Algorithm NEGATIVE CYCLE;
begin
    establish a feasible flow x in the network;
    while G(x) contains a negative cost cycle do
    begin
        use some algorithm to identify a negative cost cycle C;
        let $\delta := \min\{ r_{ij} : (i, j) \in C \}$;
        augment $\delta$ units of flow in cycle C and update G(x);
    end;
end;
```

Negative Cycle Algorithm

More on the Negative Cycle Algorithm

Suppose that all supplies/demands are integral, and capacities are integral. Then the negative cycle algorithm maintains an integral solution at each iteration.

Finiteness

Theorem. The Negative Cycle Algorithm is finite if all data are finite and integral.

Proof. By flow decomposition, we can express the min cost flow as the sum of n+m paths and cycles. Each path and cycle has a cost bounded by nC, where C = max ($|c_{ij}|$: $(i,j) \in A$). The cost of the flow is at most (nC)(n+m)U, where U is the largest capacity.

At each iteration of cycle canceling, the cost improves by at least one.

Optimality

Theorem. *The Negative Cycle Algorithm terminates with an optimal flow.*

Proof. Consider the final residual network G(x*). There is no negative cost cycle. Let d(j) denote the shortest path from node 1 to node j for each j. Let $\pi^*_j = -d(j)$ for each j.

Suppose that $(i,j) \in G(x^*)$. Then $d(j) \leq d(i) + c_{ij}$.

Therefore, $c_{ij} - \pi^*_i + \pi^*_j \geq 0$, and thus $c^{\pi^*}_{ij} \geq 0$.

Review of Cycle Canceling

Given a flow x, we look for negative cost cycles in G(x).

- If we find a negative cost cycle, we sent flow around the cycle
- If we don’t find a negative cost cycle, we establish optimality.

It is a very generic algorithm for solving minimum cost flows.

Key subroutine: finding a negative cost cycle. It can be done in different ways.

How to Find a Negative Cycle

POSSIBILITY 1. Use a shortest path algorithm to determine a negative cost cycle.

POSSIBILITY 2. Find the most negative cost cycle.

POSSIBILITY 3. Augment along the cycle that minimizes COST(C)/|C|. (The cost divided by the number of arcs.)

POSSIBILITY 4. Simplex method.

Summary

Some applications of the minimum cost flow problem

Optimality Conditions

Cycle Canceling Algorithm

Integrality Property for Minimum Cost Flows

Opt. Conditions When Capacities Are Infinite

Suppose $u_{ij} = \infty$ for all (i,j).

Node potential vector π of node potentials π is **feasible** if for every arc $(i,j) \in A$, $c^{\pi}_{ij} \geq 0$.

Lemma. If x is a feasible flow, and if π is a feasible vector of node potentials, then $cx \geq \pi b$.

Proof. We have shown that $cx - c^{\pi}x = \pi b$.

Therefore, $cx - \pi b = c^{\pi}x$. By assumption, $c^{\pi} \geq 0$ and $x \geq 0$, so the result is true.

Strong Duality When Capacities are Infinite

Strong Duality Theorem. Suppose there is a feasible flow and a feasible vector of node potentials for the uncapacitated minimum cost flow problem. Let x* be a minimum cost flow , and let π^* be a maximum cost feasible vector of node potentials. (That is, it maximizes πb.)

Then $cx^* = \pi^* b$.

Proof. We already know that $cx^* \geq \pi b$ for any feasible vector π of node potentials. Let π' be the vector of node potentials satisfying the optimality conditions. We claim that $cx^* = \pi' b$.

First, $cx^* - \pi' b = c^{\pi'} x$. Next $c^{\pi'} \geq 0$ and $x \geq 0$.

If $c^{\pi}_{ij} > 0$, then $x_{ij} = 0$

Therefore, $c^{\pi'} x = 0$.