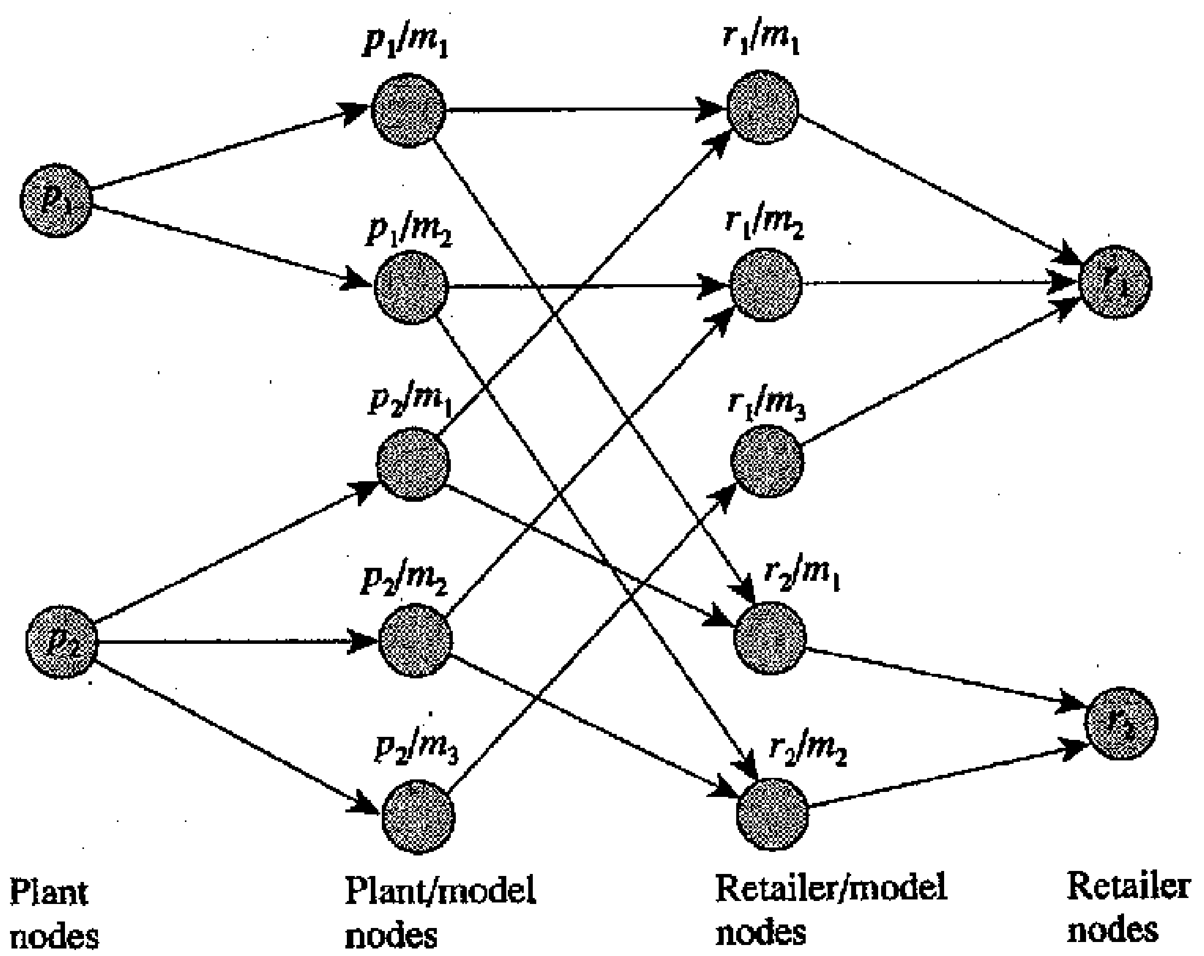
The Min Cost Flow Problem



The Min Cost Flow problem

- We want to talk about multi-source, multi-sink flows than just “flows from s to t”.

- We want to impose ***lower bounds*** as well as capacities on a given arc. Also, arcs should have ***costs***.

- Rather than maximize the value (i.e. ***amount***) of the flow through the network we want to minimize the ***cost*** of the flow.

Flow Networks with Costs

- ***Flow networks with costs*** are the ***problem instances*** of the min cost flow problem.

- A flow network with costs is given by
  1) a ***directed graph*** $G = (V,E)$
  2) ***capacities*** c: $E \rightarrow \mathbf{R}$.
  3) ***balances*** b: $V \rightarrow \mathbf{R}$.
  4) ***costs*** k: $V \times V \rightarrow \mathbf{R}$. $k(u,v) = -k(v,u)$.

- **Convention**: $c(u,v)=0$ for (u,v) not in E.

Feasible Flows

- Given a flow network with costs, a ***feasible flow*** is a ***feasible solution*** to the min cost flow problem.

- A feasible flow is a map f: $V \times V \to \mathbf{R}$ satisfying

capacity constraints: $\forall\ (u,v)$: $f(u,v) \leq c(u,v)$.

Skew symmetry: $\forall\ (u,v)$: $f(u,v) = -f(v,u)$.

Balance Constraints: $\forall\ u \in V$: $\sum_{v \in V} f(u,v) = b(u)$

Cost of Feasible Flows

- The ***cost*** of a feasible flow f is

  $$\text{cost}(f) = \tfrac{1}{2} \sum_{(u,v) \in V \times V} k(u,v)\ f(u,v)$$

- ***The Min Cost Flow Problem***: Given a flow network with costs, find the feasible flow f that minimizes cost(f).

Max Flow Problem vs. Min Cost Flow Problem

Max Flow Problem

Problem Instance:
$c: E \to \mathbf{R}^+$.
Special vertices s,t.

Feasible solution:
$\forall\ u \in V-\{s,t\}:\ \sum_{v \in V} f(u,v) = 0$

Objective:
Maximize $|f(s,V)|$

Min Cost Flow Problem

Problem Instance:
$c: E \to \mathbf{R}$.
Maps b,k.

Feasible solution:
$\forall\ u \in V:\ \sum_{v \in V} f(u,v) = b(u)$

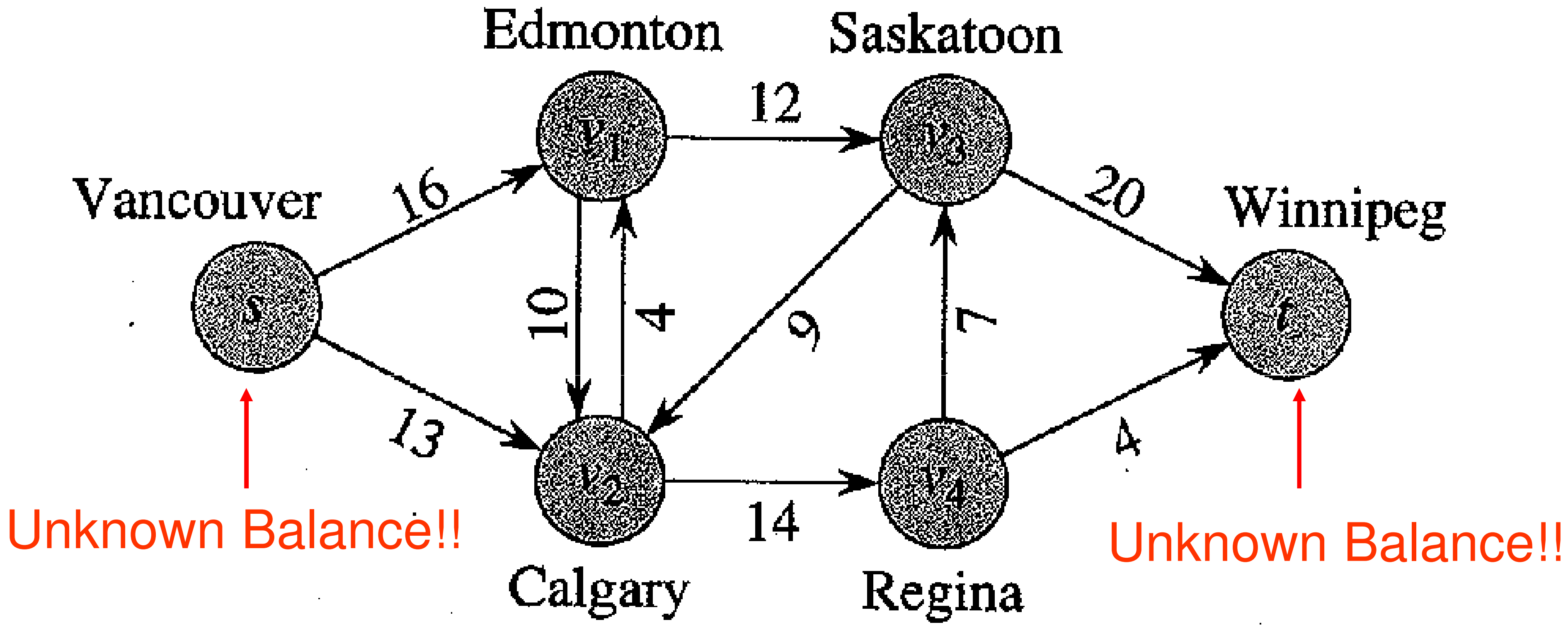
Objective:
Minimize cost(f)

Negative Capacities

- $c(u,v) < 0$.
- Let $l(v,u) = -c(u,v)$.
- $f(u,v) \leq c(u,v)$ iff $f(v,u) \geq -c(u,v) = l(v,u)$.
- l(v,u) is a ***lower bound*** on the net flow from v to u for any feasible flow.

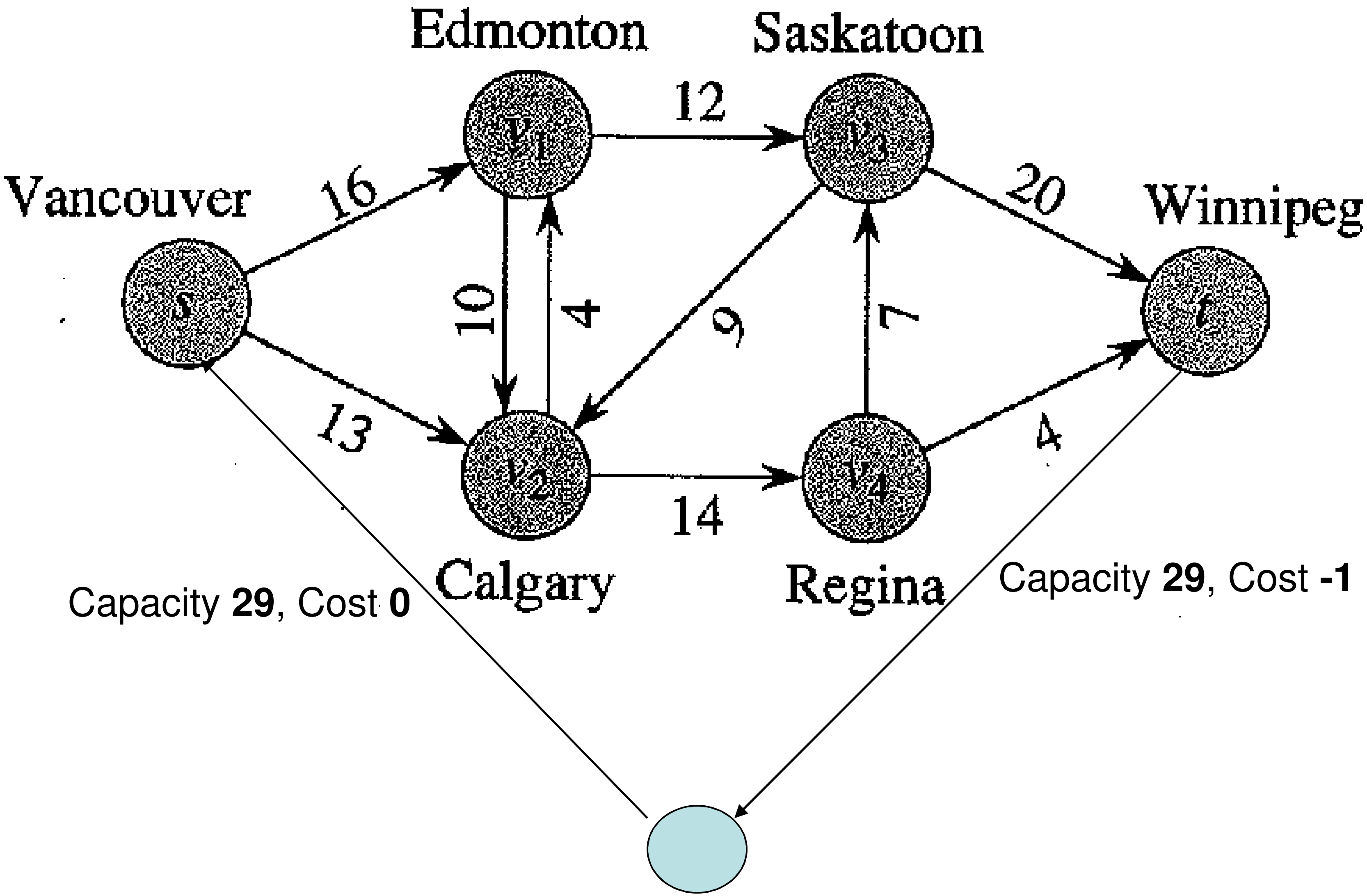
Balance Constraints

- Vertices u with $b(u)>0$ are ***producing*** flow.
- Vertices u with $b(u)<0$ are ***consuming*** flow
- Vertices u with $b(u)=0$ are ***shipping*** flow.



Can we solve the max-flow problem using software for the min-cost flow problem?

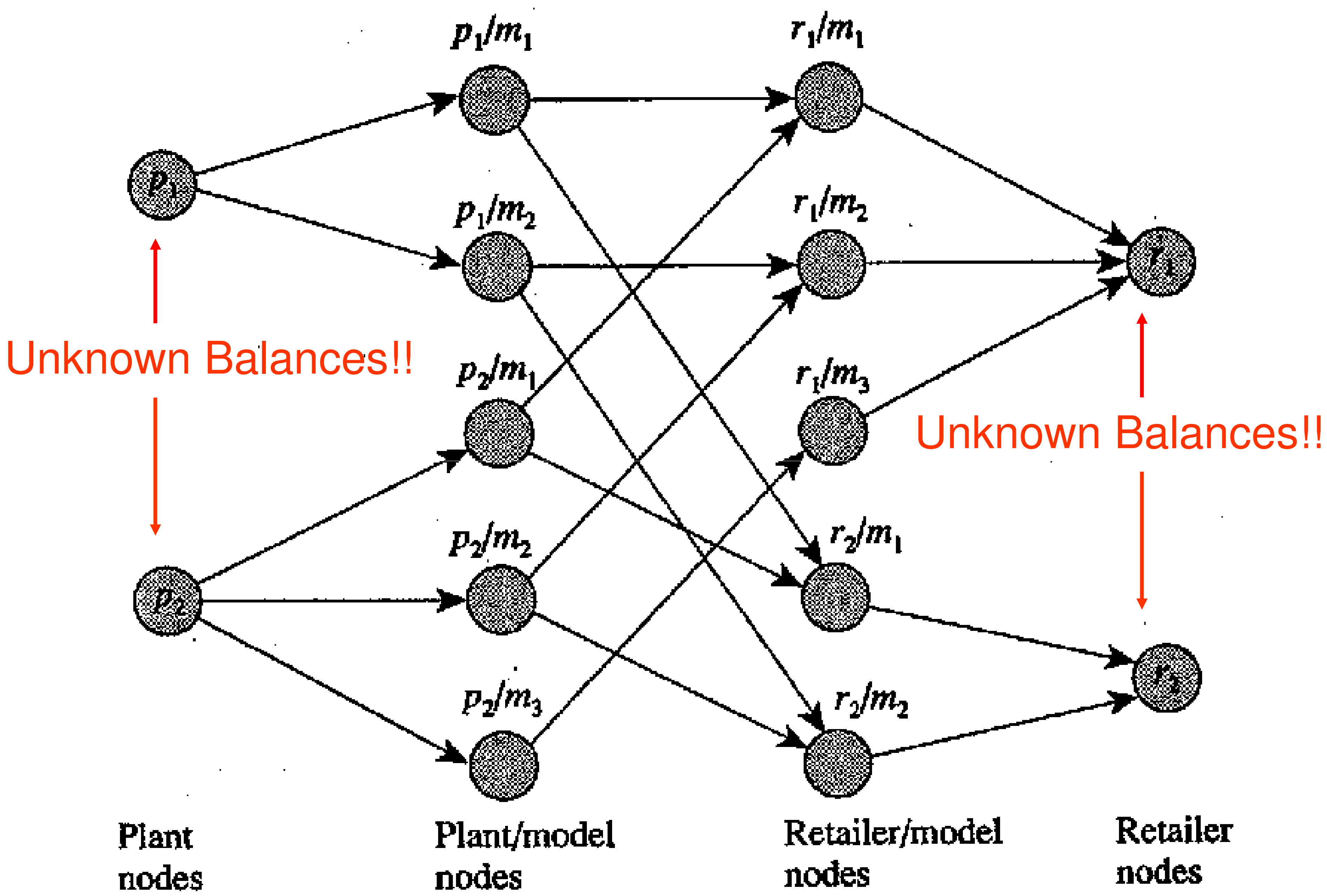
All other costs 0, all balances 0.



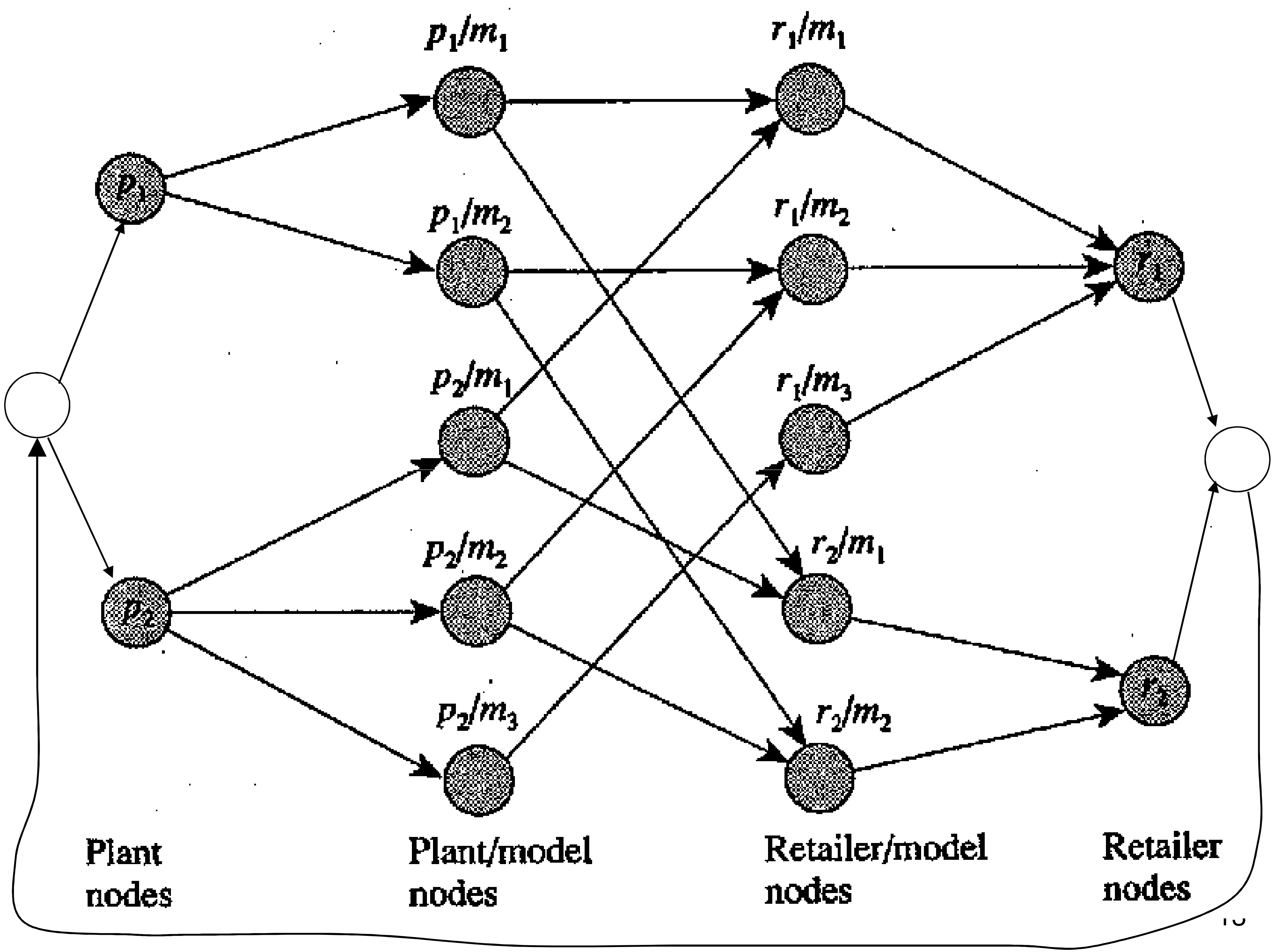
Circulation networks

- Flow networks with $b \equiv 0$ are called ***circulation networks***.

- A feasible flow in a circulation network is called a feasible ***circulation***.



p_1/m_1
r_1/m_1
p_1
p_1/m_2
r_1/m_2
r_1
Unknown Balances!!
p_2/m_1
r_1/m_3
Unknown Balances!!
p_2/m_2
r_2/m_1
p_2
r_2
p_2/m_3
r_2/m_2
Plant nodes
Plant/model nodes
Retailer/model nodes
Retailer nodes



p_1/m_1
r_1/m_1
p_1
p_1/m_2
r_1/m_2
r_1
p_2/m_1
r_1/m_3
r_2/m_1
p_2/m_2
p_2
r_2
p_2/m_3
r_2/m_2
Plant nodes
Plant/model nodes
Retailer/model nodes
Retailer nodes

Integrality theorem for min cost flow

If a flow network with costs has integral capacities and balances and a feasible flow in the network exists, then there is a minimum cost feasible flow which is integral on every arc.

(shown later by “type checking”)

Assignment problem

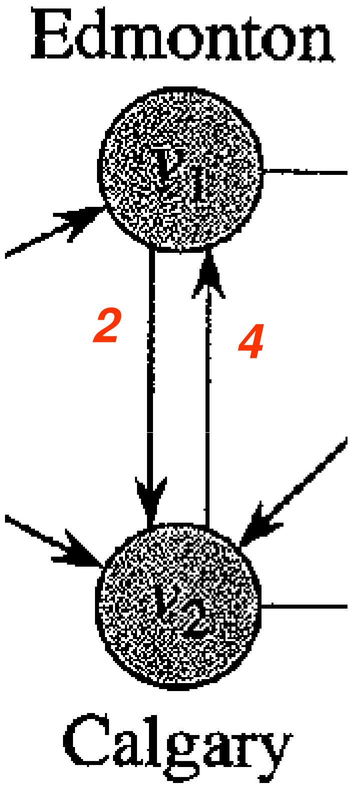
- Given integer weight matrix
  $(w(i,j))$, $1 \leq i,j \leq n$.

- Find a permutation π on $\{1,..,n\}$ maximizing
  $\sum_i w(i,\pi(i))$.

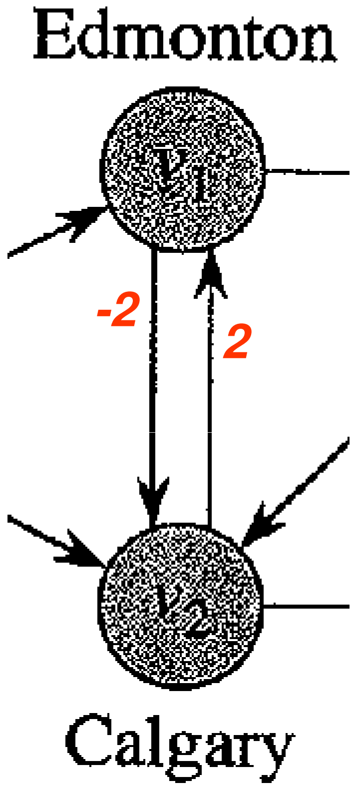
Min cost flow model of *Ahuja et al*

- Ahuja operates with non-reduced flows, we work with reduced flows (net flows). They do not require flows and costs to be skew-symmetric.

Unreduced vs net flows



Unreduced flow



Net flow

Min cost flow model of Ahuja

- Ahuja operates with non-reduced flows, we work with reduced flows (net flows). They do not require flows and costs to be skew symmetric.
- The difference matters only bidirectional arcs (an arc from u to v and an arc from v to u) with positive capcity in each direction.
- One can translate (reduce) the Ahuja version to our version (exercise).

Tanker Scheduling Problem

Shipment	Origin	Destination	Delivery date
1	Port *A*	Port *C*	3
2	Port *A*	Port *C*	8
3	Port *B*	Port *D*	3
4	Port *B*	Port *C*	6

(a)

	C	*D*
A	3	2
B	2	3

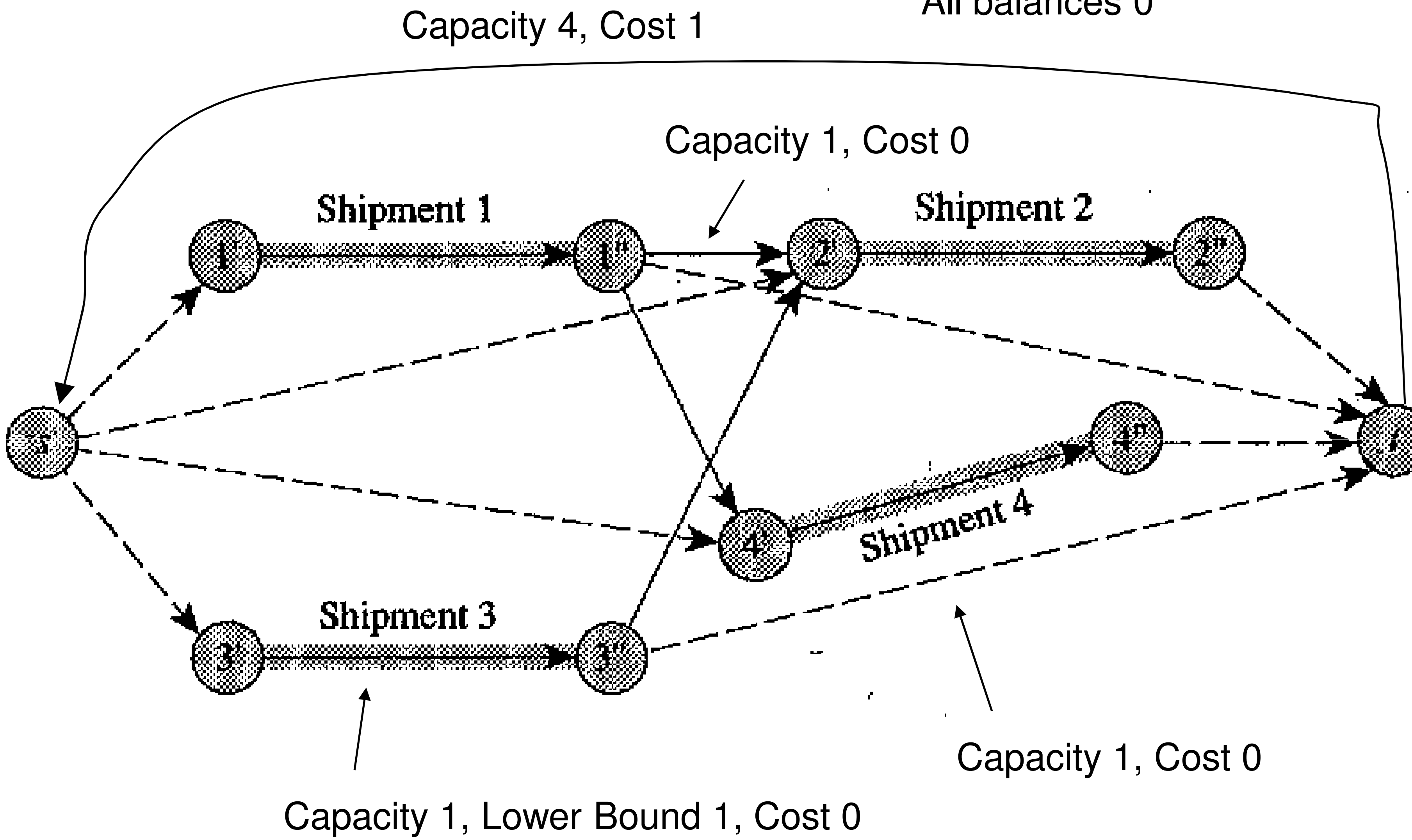
(b)

	A	*B*
C	2	1
D	1	2

(c)

Figure 6.8 Data for the tanker scheduling problem: (a) shipment characteristics; (b) shipment transit times; (c) return times.

All balances 0

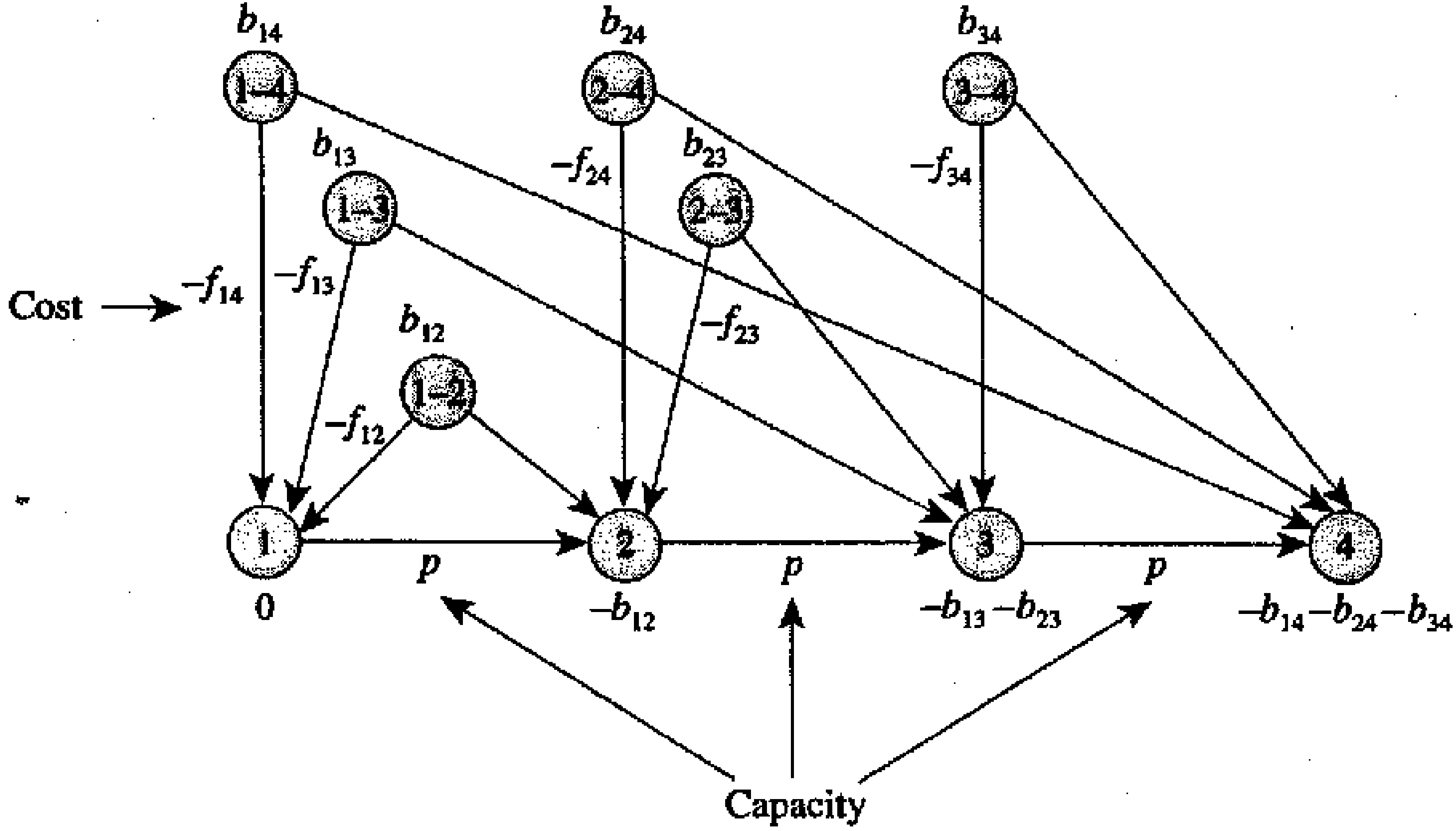


Hopping Airplane Problem

- An airplane must travel from city 1, to city 2, to city 3, .., to city n. At most p passengers can be carried at any time.

- b_{ij} passengers want to go from city i to city j and will pay f_{ij} for the trip.

- How many passengers should be picked up at each city in order to maximize profits?



b_{14}
1–4
b_{24}
2–4
b_{34}
3–4
b_{13}
1–3
b_{23}
2–3
b_{12}
1–2
$-f_{24}$
$-f_{34}$
Cost
$-f_{14}$
$-f_{13}$
$-f_{23}$
$-f_{12}$
1
2
3
4
p
p
p
0
$-b_{12}$
$-b_{13}-b_{23}$
$-b_{14}-b_{24}-b_{34}$
Capacity

Local Search Pattern

LocalSearch(ProblemInstance x)

y := feasible solution to x;

while $\exists\ z \in N(y)$: $v(z)<v(y)$ **do**

    $y := z$;

od;

return y;

$N(y)$ is a ***neighborhood*** of y.

Local search checklist

Design:

- How do we find the first feasible solution?
- Neighborhood design?
- Which neighbor to choose?

Analysis:

- Partial correctness? (termination $\Rightarrow$correctness)
- Termination?
- Complexity?

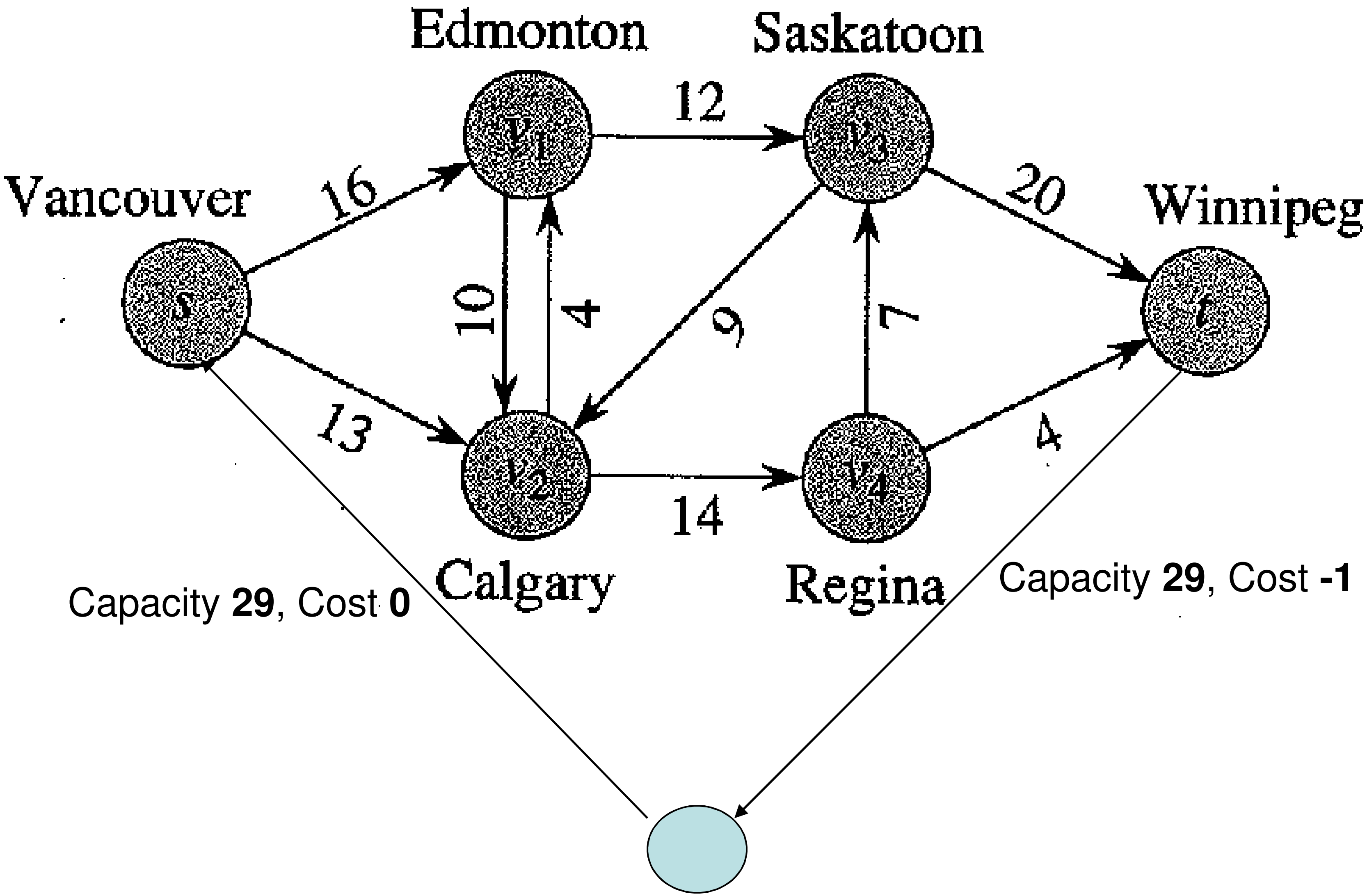
The first feasible flow?

- Because of negative capacities and balance constraints, finding the first flow is non-trivial.
- The zero flow may not work and there may not be ***any*** feasible flow for a given instance.
- We can find the first feasible flow, if one exists, by reducing this problem to a max flow problem.

Neighborhood design

- Given a feasible flow, how can we find a slightly different (and hopefully slightly better) flow?

All other costs 0, all balances 0.



The residual network

- Let $G=(V,E,c,b,k)$ be a flow network with costs and let f be a flow in G.

- The ***residual network*** G_f is the flow network with costs inherited from G and edges, capacities and balances given by:

  $E_f = \{(u,v) \in V \times V | \; f(u,v) < c(u,v)\}$

  $c_f(u,v) = c(u,v) - f(u,v) \geq 0$

  $b_f(\mathrm{u})=0$

Lemma 3

Let

- $G=(V,E,c,b,k)$ be a flow network with costs
- f be a feasible flow in G
- G_f be the residual network
- f' be a feasible flow in G_f

Then

- $f+f'$ is a feasible flow in G with $\text{cost}(f+f')=\text{cost}(f)+\text{cost}(f')$

Lemma 4

Let

- $G=(V,E,c,b,k)$ be a flow network with costs
- f be a feasible flow in G
- f' be a feasible flow in G

Then

- $f'-f$ is a feasible flow in G_f

Cycle Flows

- Let C = ($u_1 \rightarrow u_2 \rightarrow u_3 \ldots \rightarrow u_k = u_1$) be a ***simple cycle*** in G.
- The ***cycle flow*** γ_C^{δ} is the circulation defined by

$$\gamma_C^{\delta}(u_i, u_{i+1}) = \delta$$

$$\gamma_C^{\delta}(u_{i+1}, u_i) = - \delta$$

$$\gamma_C^{\delta}(u,v) = 0 \text{ otherwise.}$$

Augmenting cycles

- Let G be a flow network with costs and G_f the residual network.

- An ***augmenting cycle*** $C=(u_1, u_2, \ldots, u_r = u_1)$ is a simple cycle in G_f for which $\mathrm{cost}(\gamma_C^{\delta})<0$ where δ is the minimum capacity $c_f(u_i, u_{i+1})$, $i = 1\ldots r\text{-}1$

Klein's algorithm for min cost flow

MinCostFlow(G)

Using max flow algorithm, find feasible flow f in G (if no such flow exist, abort).

while($\exists$ augmenting cycle C in G_f){

$\delta = \min\{c_f(e) \mid e \text{ on } C\}$

$f := f + \gamma_C^{\delta}$

}

output f

Klein’s algorithm

- If Klein’s algorithm terminates it produces a feasible flow in G (by Lemma 3).
- Is it partially correct?
- Does it terminate?

Circulation Decomposition Lemma

Let G be a circulation network with no negative capacities. Let f be a feasible circulation in G. Then, f may be written as a sum of cycle flows:

$$f = \gamma_{C_1}{}^{\delta_1} + \gamma_{C_2}{}^{\delta_2} + \ldots + \gamma_{C_m}{}^{\delta_m}$$

where each cycle flow is a feasible circulation in G.

CDL $\Rightarrow$ Partial Correctness of Klein

- Suppose f is ***not*** an minimum cost flow in G. We should show that G_f has an augmenting cycle.
- Let f^* be a minimum cost flow in G.
- $f^* - f$ is a feasible circulation in G_f of strictly negative cost (Lemma 4).
- $f^* - f$ is a sum of cycle flows, feasible in G_f (by CDL).
- At least one of them must have strictly negative cost.
- The corresponding cycle is an augmenting cycle.

Termination

- Assume integer capacities and balances.
- For any feasible flow f occuring in Klein's algorithm and any u,v, the flow $f(u,v)$ is an integer between $-c(v,u)$ and $c(u,v)$.
- Thus there are only finitely many possibilities for f.
- In each iteration, f is improved – thus we never see an old f again.
- Hence we terminate.

Integrality theorem for min cost flow

If a flow network with costs has integral capacities and balances and a feasible flow in the network exists, then there is a minimum cost feasible flow which is integral on every arc.

Proof by "type checking" Klein's algorithm

Complexity

- How fast can we perform a single iteration of the local search?
- How many iterations do we have?

Complexity of a single iteration

- An iteration is dominated by finding an augmenting cycle.

- An augmenting cycle is a cycle $(u_1, u_2, \ldots u_r = u_1)$ in G_f with

  $$\sum_i k(u_i, u_{i+1}) < 0$$

- How to find one efficiently? Exercise 7.

Number of iterations

- As Ford-Fulkerson, Klein's algorithm may use an exponential number of iterations, if care is not taken choosing the augumentation (Exercise 6).

- ***Fact:*** If the cycle with minimum ***average*** edge cost is chosen, there can be at most $O(|E|^2\,|V|\log|V|)$ iterations.

Generality of Languages

Linear Programs	*Algorithm* →	Simplex Algorithm, Interior Point Algorithms
↑ *Reduction*		
Min Cost Flow	*Algorithm* →	Klein's algorithm
↑ *Reduction*		
Max (s,t)-Flow	*Algorithm* →	Edmonds-Karp
↑ *Reduction*		
Maximum Cardinality Matching	*Algorithm* →	Hopcroft-Karp (dADS)

Increasing generality

Decreasing efficiency