Network Models and Maximum Flow

A flow describes how much material, traffic, or data travels on each directed connection.

Model: translate a network into conservation equations and capacity bounds.

Solve: increase source-to-destination flow, including undoing an earlier routing decision.

Certify: identify a cut whose capacity matches the flow value.

Work through the pause questions before opening the following answer page.

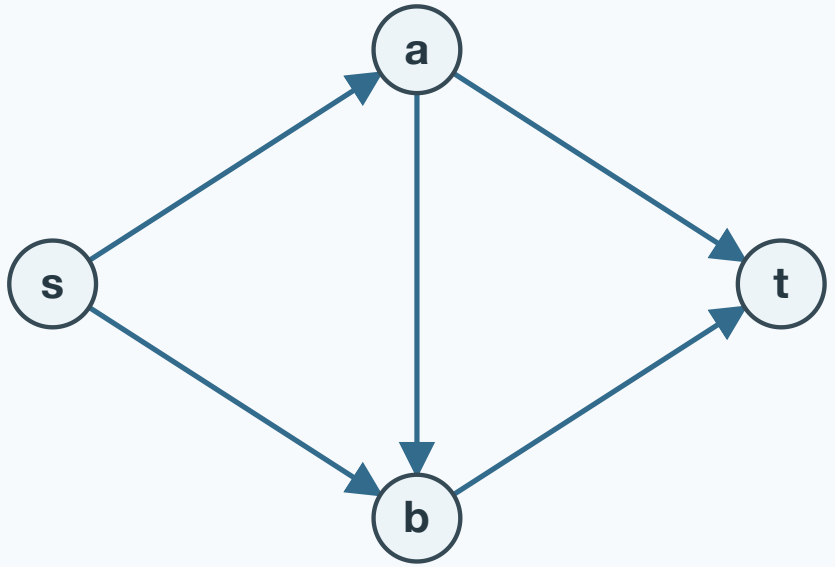
Nodes, arcs, and direction

A **directed graph** has nodes V and directed arcs E. An arc (i, j) goes from node i to node j.

Here $V = \{s, a, b, t\}$ and

$$E = \{(s, a), (s, b), (a, b), (a, t), (b, t)\}.$$

The variable $x_{ij} \geq 0$ is the amount sent on arc (i, j). The number u_{ij} is its **capacity**: $x_{ij} \leq u_{ij}$.



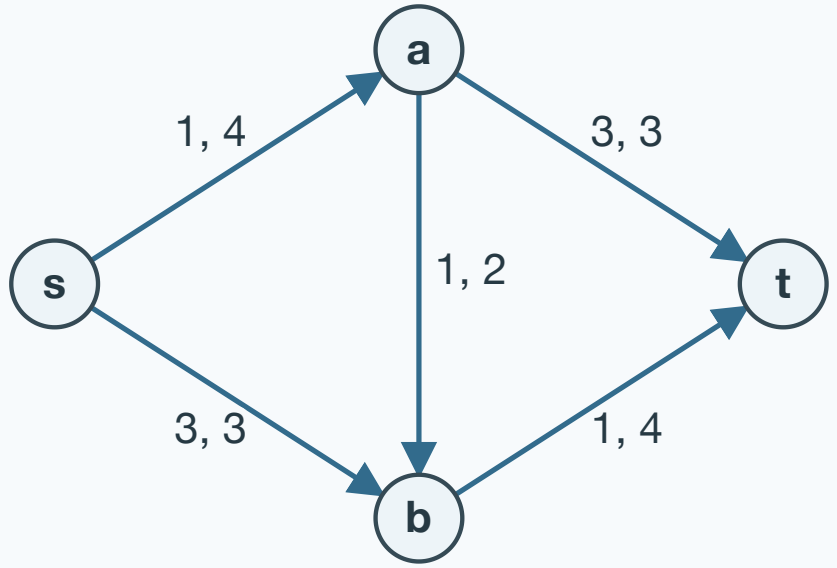
Arrow direction matters. An arc from a to b does not provide an original arc from b to a.

Deliver 5 tonnes of rice at minimum cost

Warehouse s must deliver **exactly 5 tonnes of rice** to supermarket t during one delivery period. Depots a, b receive and forward shipments.

Choose x_{ij}, the tonnes sent on route $i \to j$. Fractional tonnes are allowed.

Route	Cost ($/tonne)	Capacity (tonnes)
s → a	1	4
s → b	3	3
a → b	1	2
a → t	3	3
b → t	1	4



s: warehouse; a, b: transfer depots; t: supermarket. Arc labels: cost ($/tonne), capacity (tonnes).

Both depots start and end empty. They do not produce, consume, or lose rice. We minimize shipping cost for the fixed 5-tonne delivery.

Why must the depots conserve flow?

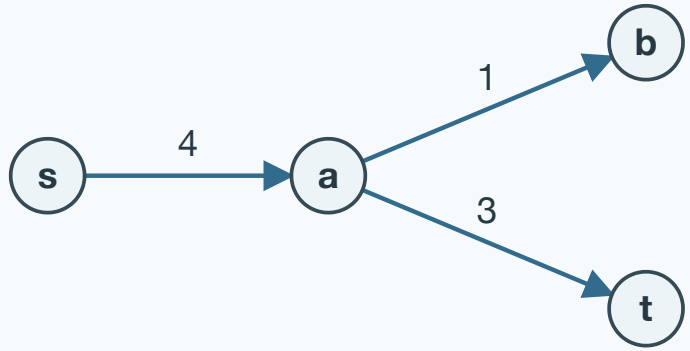
Inventory is rice stored at a depot. With no production, consumption, or loss:

$$\text{starting inventory} + \text{incoming} = \text{outgoing} + \text{ending inventory}.$$

With empty depots at the start and end, **incoming = outgoing**: **flow conservation**.

At a, suppose 4 tonnes arrive; 1 goes to b, so 3 must go to t:

$$4 = 1 + 3, \qquad x_{sa} = x_{ab} + x_{at}.$$



Focus on depot a. Labels are tonnes shipped.

One sign convention: outgoing – incoming = net supply.

$$s: \quad 5 - 0 = 5, \qquad t: \quad 0 - 5 = -5.$$

At a, b, the difference is 0. The -5 records demand, not negative flow.

The complete LP for delivering 5 tonnes

Choose x_{ij}, tonnes shipped on each route. Warehouse s sends 5 tonnes; supermarket t receives 5. Depots a, b start and end empty.

$$
\begin{aligned}
\min \quad & x_{sa} + 3x_{sb} + x_{ab} + 3x_{at} + x_{bt} \\
\text{at } s: \quad & x_{sa} + x_{sb} = 5, \\
\text{at } a: \quad & x_{ab} + x_{at} - x_{sa} = 0, \\
\text{at } b: \quad & x_{bt} - x_{sb} - x_{ab} = 0, \\
\text{at } t: \quad & -x_{at} - x_{bt} = -5, \\
& 0 \le x_{sa} \le 4, \quad 0 \le x_{sb} \le 3, \\
& 0 \le x_{ab} \le 2, \quad 0 \le x_{at} \le 3, \\
& 0 \le x_{bt} \le 4.
\end{aligned}
$$

Each objective term is unit cost ($/tonne) × tonnes shipped. The bounds enforce the route capacities; the objective measures total dollars.

Check a feasible shipment and compute its cost

$$(x_{sa}, x_{sb}, x_{ab}, x_{at}, x_{bt}) = (4, 1, 1, 3, 2).$$

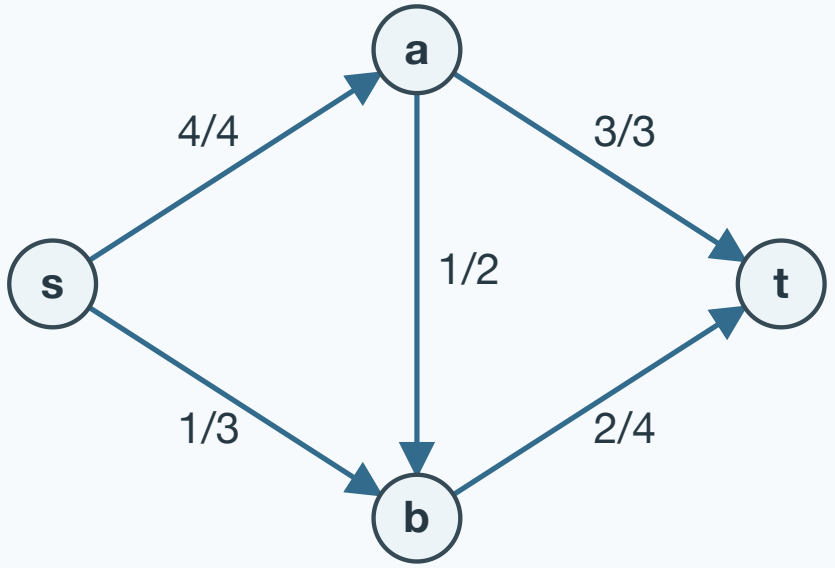
Sent from s: $4 + 1 = 5$ tonnes.

Received at t: $3 + 2 = 5$ tonnes.

At a: $4 = 1 + 3$. At b: $1 + 1 = 2$.

$$\text{cost} = 4 + 3(1) + 1 + 3(3) + 2 = 19 \text{ dollars}.$$

> Every flow respects its capacity. This verifies feasibility and cost; it does not yet prove minimum cost.



Arc labels now show flow / capacity, not cost.

The general minimum-cost flow LP

For every arc $(i, j) \in E$, specify unit cost c_{ij}, capacity u_{ij}, and flow variable x_{ij}. Node i has supply b_i; demand is recorded as $b_i < 0$.

$$\begin{aligned} \min \quad & \sum_{(i,j)\in E} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i \quad (i \in V), \\ & 0 \le x_{ij} \le u_{ij} \quad ((i,j) \in E). \end{aligned}$$

> For the shipment: $b = (5, 0, 0, -5)$. Every arc leaves one node and enters one node, so adding the balance equations gives $\sum_i b_i = 0$.

Balanced total supply is necessary, but disconnected routes or insufficient capacities can still make the model infeasible.

These balance equations are also a matrix equation

Order nodes as (s, a, b, t) and arcs as (sa, sb, ab, at, bt). Let A be the **node–arc incidence matrix**: each column has $+1$ at its starting node and -1 at its ending node.

$$\underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{pmatrix}}_{A} \begin{pmatrix} x_{sa} \\ x_{sb} \\ x_{ab} \\ x_{at} \\ x_{bt} \end{pmatrix} = \begin{pmatrix} 5 \\ 0 \\ 0 \\ -5 \end{pmatrix}.$$

> The compact model is $\min c^\top x$ subject to $Ax = b$, $0 \le x \le u$. Multiplying the third row reproduces node b's equation.

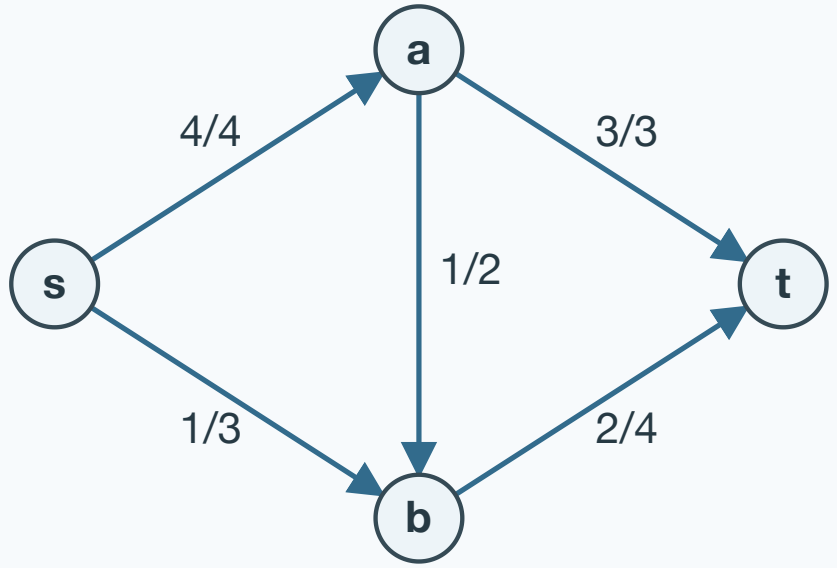
The balance rows sum to zero; one is redundant when the others and total supply balance hold.

Pause: change one arc and preserve conservation

Current shipment (tonnes): $(x_{sa}, x_{sb}, x_{ab}, x_{at}, x_{bt}) = (4, 1, 1, 3, 2)$. Keep $x_{sa} = 4$ and $x_{sb} = 1$ fixed.

> Increase x_{ab} from 1 to 2. Find x_{at} and x_{bt} to conserve flow at both depots. Check capacities.

Route	Cost ($/tonne)
s → a	1
s → b	3
a → b	1
a → t	3
b → t	1



Current flow / capacity, in tonnes. These labels are not shipping costs.

Use these unchanged shipping costs to compute the updated shipment's total cost (2 minutes):

$$x_{sa} + 3x_{sb} + x_{ab} + 3x_{at} + x_{bt}.$$

Answer: reroute one tonne through the middle arc

$$x_{ab} = 2, \qquad x_{at} = 4 - 2 = 2, \qquad x_{bt} = 1 + 2 = 3.$$

Arc order	sa	sb	ab	at	bt
New flow	4	1	2	2	3
Capacity	4	3	2	3	4

$$\text{new cost} = 4 + 3(1) + 2 + 3(2) + 3 = 18.$$

One tonne previously used route a → t, costing $3. It now uses a → b → t, costing $1+$1. The delivery stays feasible and saves $1.

This rerouting intuition will reappear when we cancel flow in a residual graph.

Transportation: supplies at origins, demands at destinations

Warehouse A has 3 units; B has 2. Store 1 needs 2; store 2 needs 3. Variable x_{ij} is the shipment from warehouse i to store j.

Unit cost	Store 1	Store 2	Supply
Warehouse A	2	5	3
Warehouse B	4	1	2
Demand	2	3	Total 5

> Every connection goes from a warehouse to a store. Each warehouse equation accounts for its supply; each store equation accounts for its demand.

For example: $x_{A1} + x_{A2} = 3$ and $x_{A1} + x_{B1} = 2$.

Pause: write and check the transportation LP

Supplies: A = 3, B = 2. Demands: store 1 = 2, store 2 = 3. Unit costs are shown below.

Unit cost	Store 1	Store 2
A	2	5
B	4	1

> Write the objective, all four balance equations, and variable signs. Then find one feasible shipping plan and compute its cost.

Take about 3 minutes. You do not need to run simplex.

Answer: a complete transportation model

$$
\begin{aligned}
\min \quad & 2x_{A1} + 5x_{A2} + 4x_{B1} + x_{B2} \\
\text{s.t.} \quad & x_{A1} + x_{A2} = 3, \quad x_{B1} + x_{B2} = 2, \\
& x_{A1} + x_{B1} = 2, \quad x_{A2} + x_{B2} = 3, \\
& x_{A1}, x_{A2}, x_{B1}, x_{B2} \geq 0.
\end{aligned}
$$

One feasible plan is $(x_{A1}, x_{A2}, x_{B1}, x_{B2}) = (2, 1, 0, 2)$, with cost $2(2) + 5(1) + 4(0) + 1(2) = 11$.

It is also optimal; verify the lower bound in the appendix. With every supply and demand equal to 1, this model becomes the **assignment problem**: match one unit from each origin to one destination.

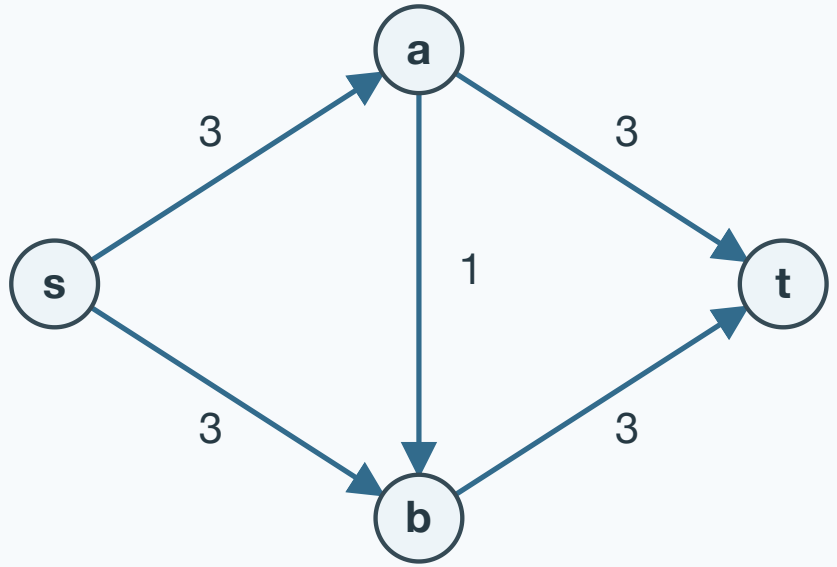
We will study a dedicated assignment algorithm in a later lecture.

Two shipments share one network

A **commodity** is a shipment type whose origin and destination we track separately. Use x_{ij}^k for commodity k's flow on arc (i, j).

Commodity	Required delivery
1	4 units from s to t
2	1 unit from a to b

Every arc costs 1 per unit of either commodity. Capacities are shared by their combined flows.



Labels are shared capacities. The arc a → b can carry only 1 unit in total.

Two individually feasible plans can conflict

Required deliveries remain 4 units from s to t and 1 unit from a to b.

Arc	sa	sb	ab	at	bt
Commodity 1	3	1	1	2	2
Commodity 2	0	0	1	0	0
Combined flow	3	1	2	2	2
Shared capacity	3	3	1	3	3

On arc (a, b), the combined flow is $x^1_{ab} + x^2_{ab} = 1 + 1 = 2 > 1$. Giving each commodity its own copy of the capacity would model a different network.

Conservation must hold separately for each commodity; capacity must hold for their sum.

Keep separate balances and one shared capacity

Use the node order (s, a, b, t). Commodity supplies are $b^1 = (4, 0, 0, -4)$, $b^2 = (0, 1, -1, 0)$. Capacities in arc order (sa, sb, ab, at, bt) are $(3, 3, 1, 3, 3)$.

$$\min \sum_{(i,j)\in E} (x_{ij}^1 + x_{ij}^2), \qquad x_{ij}^1 + x_{ij}^2 \le u_{ij}, \quad x^1, x^2 \ge 0.$$

$$\sum_{j:(i,j)\in E} x_{ij}^k - \sum_{j:(j,i)\in E} x_{ji}^k = b_i^k \qquad (i \in V,\ k = 1, 2).$$

At node a: $x_{ab}^1 + x_{at}^1 - x_{sa}^1 = 0$, but $x_{ab}^2 + x_{at}^2 - x_{sa}^2 = 1$. On arc (a, b): $x_{ab}^1 + x_{ab}^2 \le 1$.

The shared-capacity inequalities connect the two sets of otherwise separate balance equations.

A routing that respects the shared capacity

Arc	sa	sb	ab	at	bt
Commodity 1	2	2	0	2	2
Commodity 2	0	0	1	0	0
Combined flow	2	2	1	2	2
Shared capacity	3	3	1	3	3

Commodity 1 uses s → a → t and s → b → t, two units each. Commodity 2 alone uses a → b.

$$\text{total cost} = 2 + 2 + 1 + 2 + 2 = 9.$$

Each commodity keeps its own origin and destination; all arc totals respect the shared capacities.

Appendix: why 9 is the minimum cost for these data.

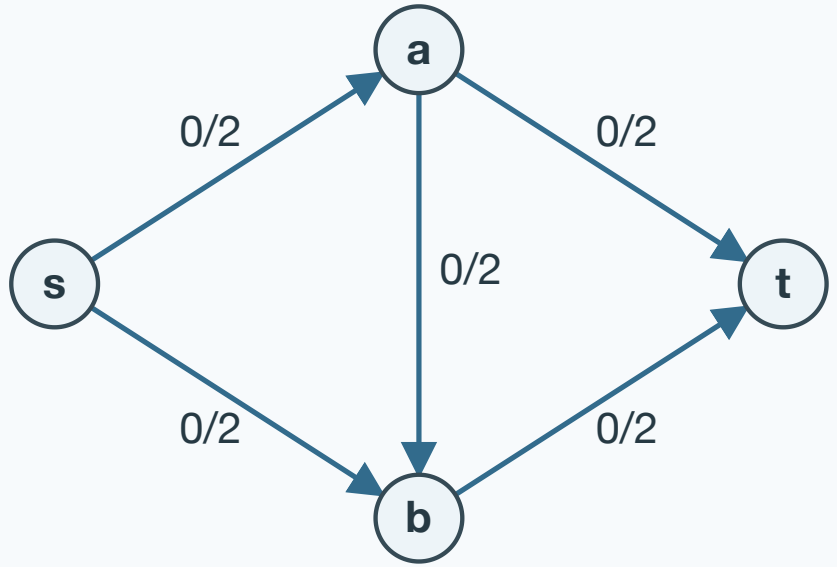
Maximum flow: how much can this network deliver?

Arcs $(s,a), (s,b), (a,b), (a,t), (b,t)$ each have capacity 2.

The **source** s supplies flow; the **sink** t receives it. Nodes a, b conserve flow.

> Choose the largest delivered amount F, rather than prescribing it. Here no original arc enters s or leaves t, so $F = x_{sa} + x_{sb} = x_{at} + x_{bt}$.

Costs no longer matter; capacities determine the maximum.



Each arc label is flow / capacity. Every original arc is directed as shown.

Write the maximum-flow LP

Arcs $(s,a),(s,b),(a,b),(a,t),(b,t)$ each have capacity 2.

$$\begin{aligned} \max \quad & F = x_{sa} + x_{sb} \\ \text{s.t.} \quad & x_{sa} = x_{ab} + x_{at}, \\ & x_{sb} + x_{ab} = x_{bt}, \\ & 0 \le x_{sa}, x_{sb}, x_{ab}, x_{at}, x_{bt} \le 2. \end{aligned}$$

> Adding the two middle-node equations gives $x_{sa} + x_{sb} = x_{at} + x_{bt}$: the amount leaving the source reaches the sink.

For a general network that also has arcs entering the source, define F as **net** source outflow: outgoing minus incoming.

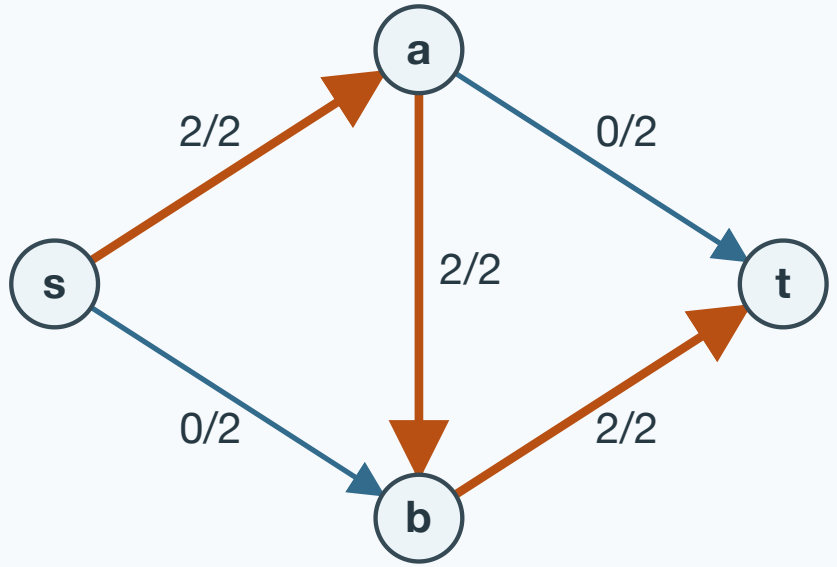
Start with zero flow and send 2 units along a path

Arcs $(s,a),(s,b),(a,b),(a,t),(b,t)$ each have capacity 2.

Choose the directed path $s \to a \to b \to t$. Its smallest available capacity is $\min(2,2,2) = 2$.

$$(x_{sa}, x_{sb}, x_{ab}, x_{at}, x_{bt}) = (2, 0, 2, 0, 2).$$

Increase all three path flows by 2. At each middle node, one incoming and one outgoing flow increase equally. The flow value becomes $F = 2$.



Each arc label is flow / capacity. Every original arc is directed as shown.

The first routing decision need not be permanent

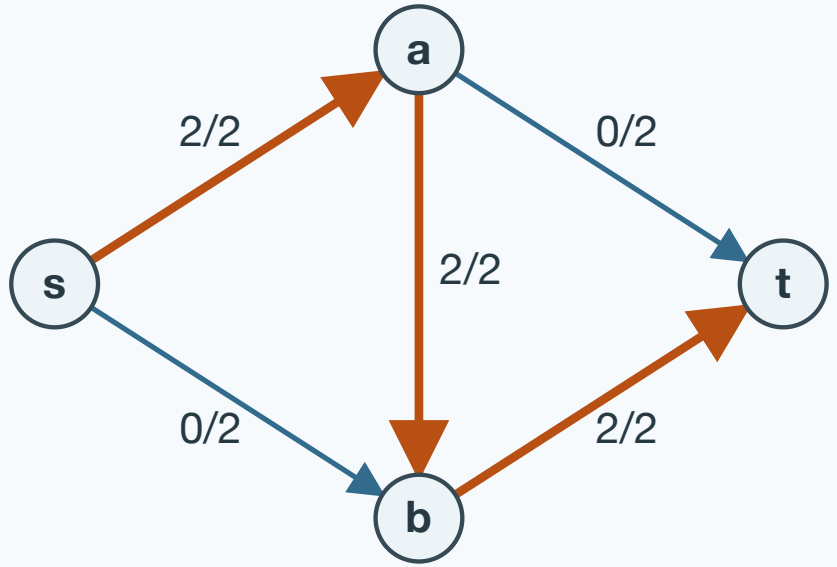
Arcs $(s,a),(s,b),(a,b),(a,t),(b,t)$ each have capacity 2.

Current flow is $(2,0,2,0,2)$, with $F=2$.

The original path through a cannot accept more because s → a is full. The original path through b cannot accept more because b → t is full.

> But s → b and a → t are both unused. We can use them if we undo the 2 units previously sent from a to b.

We need a systematic way to represent both increases and cancellations.



Each arc label is flow / capacity. Every original arc is directed as shown.

Residual arcs describe allowed changes to a flow

For an original arc (i, j) with capacity u_{ij} and current flow x_{ij}:

Allowed change	Residual arc	Residual capacity
Increase x_{ij}	$i \to j$	$u_{ij} - x_{ij}$
Decrease x_{ij}	$j \to i$	x_{ij}

> The **residual graph** contains these allowed changes with positive residual capacity. A reverse residual arc cancels existing flow; it does not create a new physical connection.

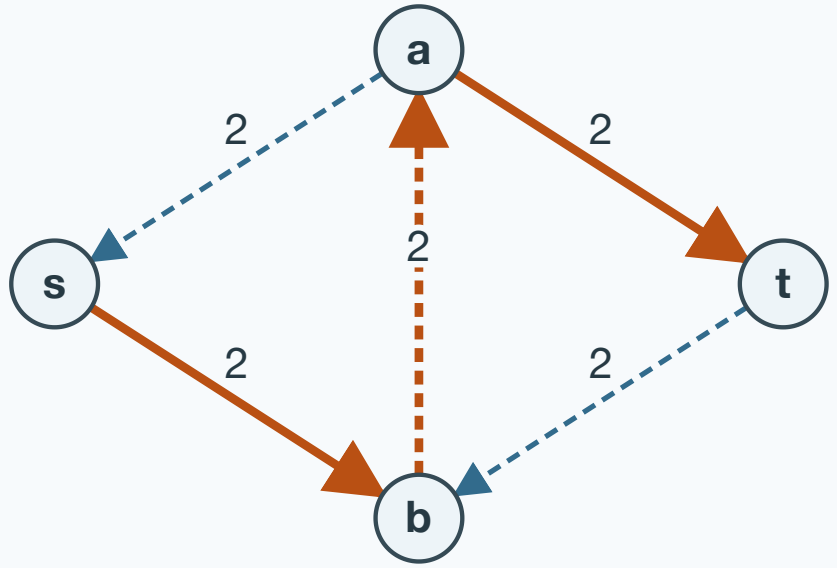
Example: $x_{ab} = 2$, $u_{ab} = 2$. We cannot increase it, but the residual arc $b \to a$ lets us decrease it by as much as 2.

Read the residual graph after the first path

Original flow: $(2, 0, 2, 0, 2)$; every original capacity is 2.

Residual arc	Action on original flow
s → b	Increase sb
b → a	Decrease ab
a → t	Increase at

An **augmenting path** is a source-to-sink path in the residual graph. Increasing the net delivered flow along it may decrease some original arc flows.



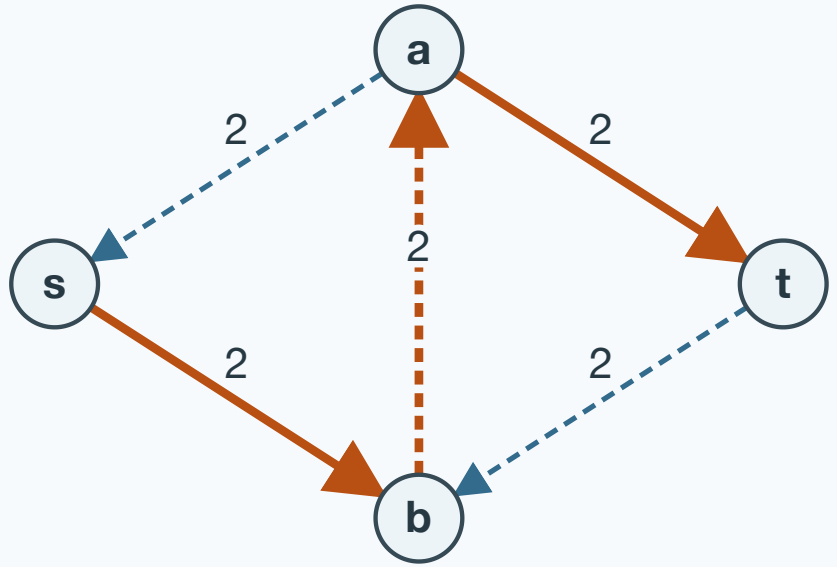
Residual capacities are shown. Dashed arcs decrease existing flow; solid arcs increase it. Orange marks the next path.

Pause: use the reverse arc to improve the flow

Start from $(x_{sa}, x_{sb}, x_{ab}, x_{at}, x_{bt}) = (2, 0, 2, 0, 2)$. All original capacities are 2.

> Use the residual path $s \to b \to a \to t$. What is the maximum step? Write the five new original arc flows. Check both middle-node balances and the new flow value.

Take about 5 minutes. The middle path step decreases the original flow on a → b.



Residual capacities are shown. Dashed arcs decrease existing flow; solid arcs increase it. Orange marks the next path.

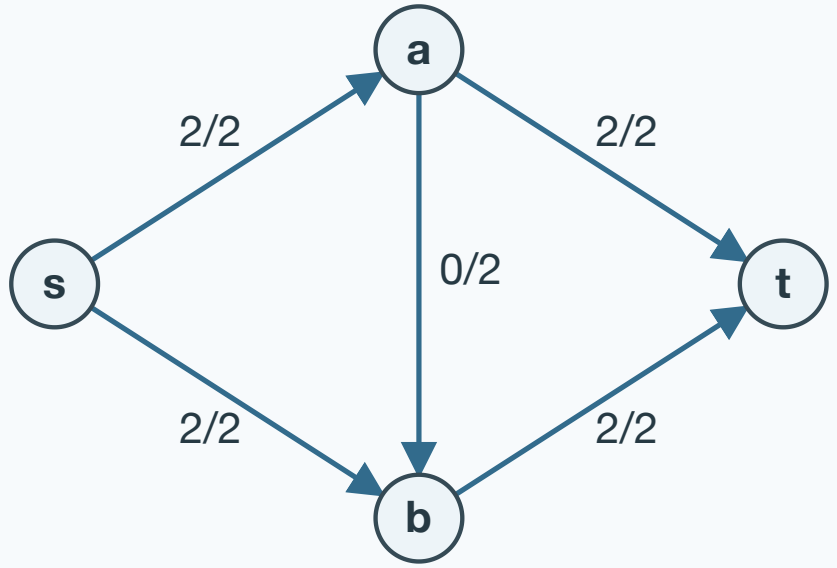
Answer: cancel a → b while delivering 2 more units

$$\delta = \min(2, 2, 2) = 2.$$

$$\begin{aligned} x_{sb} &= 0 + 2 = 2, \\ x_{ab} &= 2 - 2 = 0, \\ x_{at} &= 0 + 2 = 2. \end{aligned}$$

Keep $x_{sa} = x_{bt} = 2$. At a: $2 = 0 + 2$. At b: $2 + 0 = 2$.

The new flow is $(2, 2, 0, 2, 2)$, with $F = 4$. Canceling one internal flow has increased the total delivery.



Each arc label is flow / capacity. Every original arc is directed as shown.

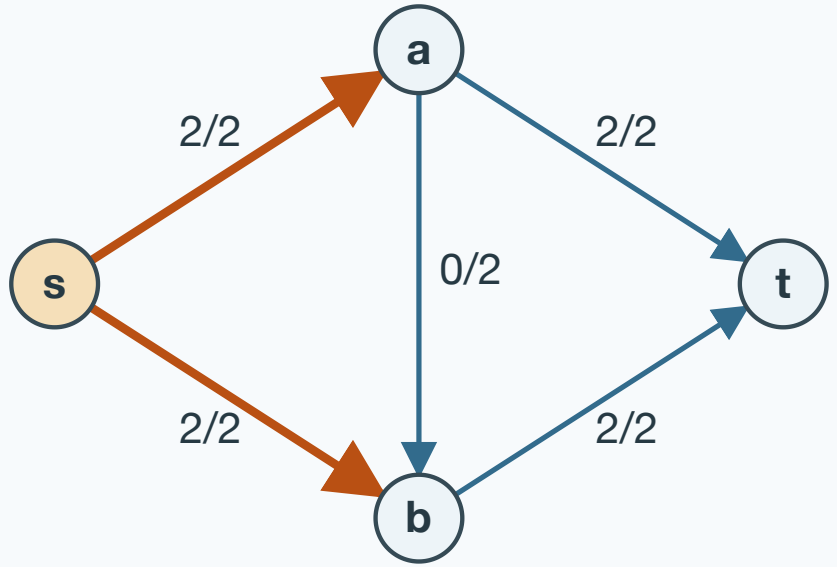
Why 4 is optimal: the source can send at most 4

Arcs $(s,a),(s,b),(a,b),(a,t),(b,t)$ each have capacity 2.

The feasible flow $(2,2,0,2,2)$ delivers 4.

$$F = x_{sa} + x_{sb} \le 2 + 2 = 4.$$

The upper bound equals the feasible flow value, so $F^* = 4$. The two source arcs form a **cut**: every source-to-sink route must cross from $S = \{s\}$ to $T = \{a,b,t\}$.



The orange cut arcs have total capacity 4; both are full.

Every cut gives an upper bound on the flow

An s-t **cut** partitions the nodes into S, T, with $s \in S$, $t \in T$. Its capacity sums only arcs directed from S to T:

$$u(S,T) = \sum_{i \in S,\ j \in T : (i,j) \in E} u_{ij}.$$

> Add the conservation equations inside S. Internal flows cancel. The delivered amount equals outgoing crossing flow minus incoming crossing flow.

$$F = \sum_{S \to T} x_{ij} - \sum_{T \to S} x_{ij} \le \sum_{S \to T} u_{ij} = u(S,T).$$

Therefore a feasible flow and a cut with equal values certify optimality, just as matching primal and dual bounds do.

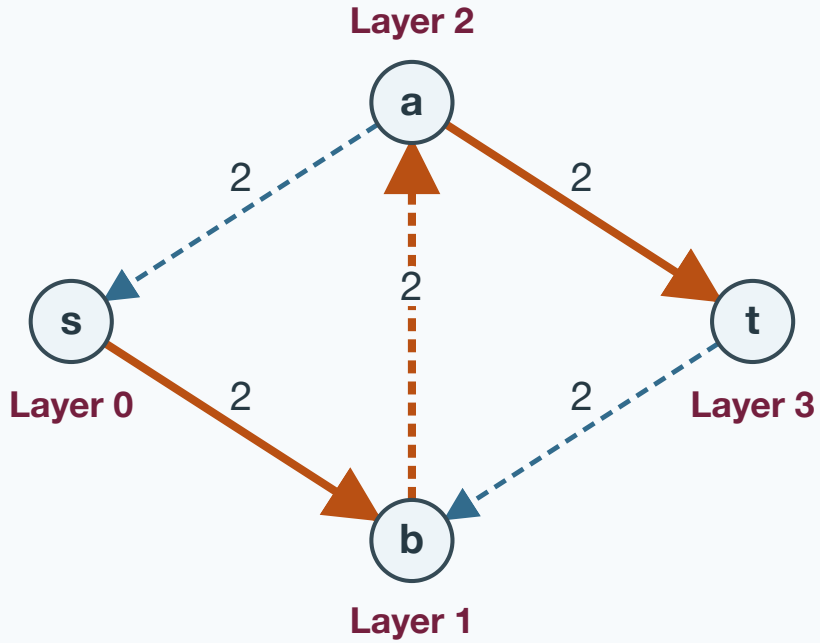
Find an augmenting path with breadth-first search

Revisit the earlier flow: $(x_{sa}, x_{sb}, x_{ab}, x_{at}, x_{bt}) = (2, 0, 2, 0, 2)$, $F = 2$. All original capacities are 2.

Breadth-first search (BFS) explores by **layers**: nodes reachable using 0 residual arcs, then 1 arc, then 2, and so on.

Layer	New node	Record how it was reached
0	s	Start here
1	b	s → b
2	a	b → a
3	t	a → t

Use positive-capacity arcs. Mark each node on first discovery; at a, ignore a → s because s is already marked.



Residual capacities. Dashed b → a decreases original a → b flow; layers count arcs from s.

Trace the records backward from t. The path is $s \to b \to a \to t$.

Ford–Fulkerson: search, augment, repeat

Start with zero flow when all lower bounds are zero. Repeat the following operations:

Step	Operation
1	Construct the residual graph, including reverse arcs with positive capacity.
2	Search from s, marking reached nodes and recording the arc that first reaches each node.
3	If t is reached, reconstruct its path and take the smallest residual capacity as the step.
4	Add the step on original forward arcs; subtract it on original arcs traversed in reverse.

> **If the search cannot reach t:** stop. The flow is maximum; the reached nodes form the source side of a minimum cut.

Our first path $s \to a \to b \to t$ was an arbitrary choice. BFS from zero would find $s \to a \to t$ or $s \to b \to t$.

For our calculations, use BFS to find each augmenting path. It chooses a path with the fewest residual arcs.

No augmenting path: a cut proves optimality

Let S be the nodes reachable from s in the residual graph. If $t \notin S$, set $T = V \setminus S$.

Every original $S \to T$ arc is full: unused forward capacity would make a node in T reachable.

Every original $T \to S$ arc has zero flow: a reverse residual arc would otherwise reach T.

$$F = \sum_{S \to T} u_{ij} = u(S, T).$$

Max-flow/min-cut theorem: in a finite directed network with nonnegative finite capacities, the maximum feasible flow value equals the minimum cut capacity.

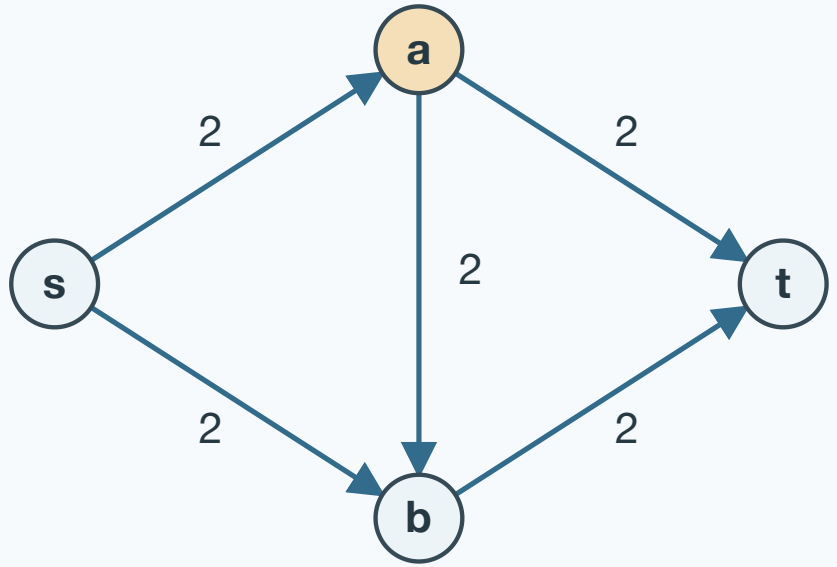
What if a processing node also has a capacity?

Arcs $(s,a),(s,b),(a,b),(a,t),(b,t)$ each have capacity 2.

Add a new condition: node a can process at most 1 unit. All material entering a must leave a.

$$x_{sa} \le 1, \qquad x_{ab} + x_{at} = x_{sa}.$$

A node capacity limits the combined flow through the node, not each outgoing arc separately. We can represent that combined limit using one new arc.



Original arc capacities remain 2. Node a now has an additional processing capacity 1.

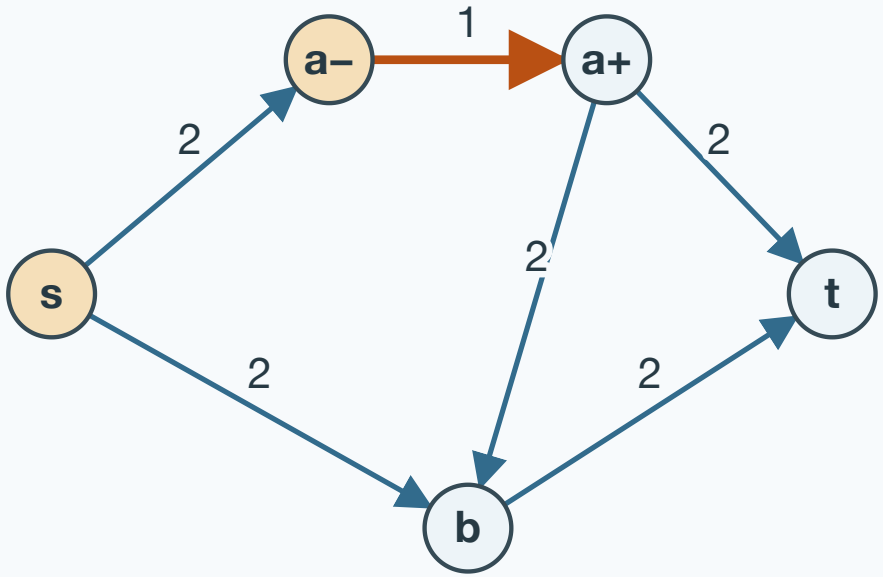
Replace the limited node by an entrance and an exit

Replace a by entrance a^- and exit a^+.

Redirect incoming arcs to a^-, outgoing arcs from a^+, and add $a^- \to a^+$ with capacity 1.

> Conservation forces all flow through the processing node to cross this one new arc. Its capacity imposes the combined node limit.

This transformation is called **node splitting**.



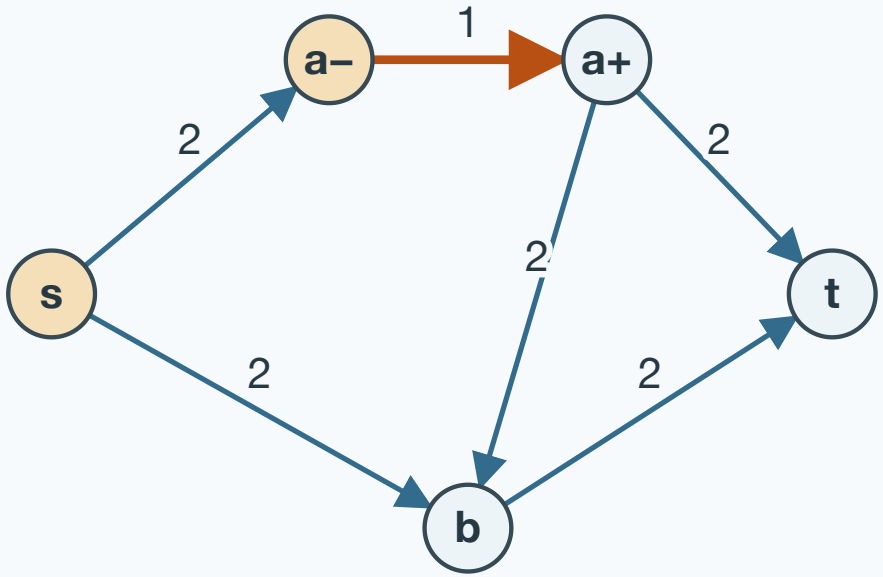
The new arc a– → a+ has capacity 1. All original capacities remain 2. Shading marks the cut side S.

Pause: find a flow and a matching cut after splitting

Every old arc has capacity 2. The new arc $a^- \to a^+$ has capacity 1.

> Find a feasible source-to-sink flow. Then use the shaded set $S = \{s, a^-\}$ to bound its value. List every directed arc in the split network crossing from S to its complement.

Take about 4 minutes. The answers should certify the same value.



The new arc a– → a+ has capacity 1. All original capacities remain 2. Shading marks the cut side S.

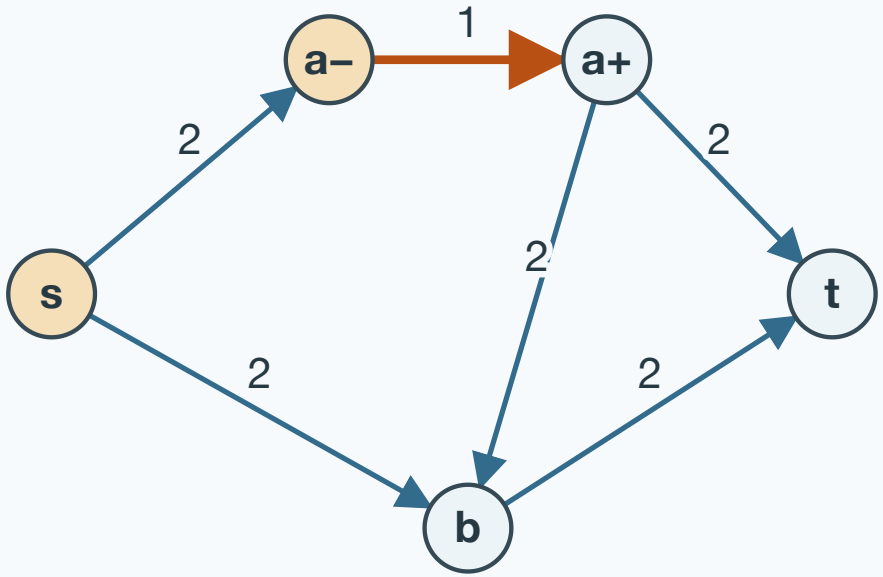
Answer: the node limit reduces the maximum from 4 to 3

Send 1 unit on $s \to a^- \to a^+ \to t$ and 2 units on $s \to b \to t$. The feasible flow value is 3.

For $S = \{s, a^-\}$, the outgoing cut arcs are $s \to b$ (capacity 2) and $a^- \to a^+$ (capacity 1).

$$F \le 2 + 1 = 3.$$

A feasible flow of 3 and a cut of capacity 3 prove the new maximum is 3.



The new arc a– → a+ has capacity 1. All original capacities remain 2. Shading marks the cut side S.

What you should be able to do on a small network

Model: write node balances and arc bounds. For multiple commodities, use separate balances and shared capacity sums.

Carry out maximum flow by hand: build the residual graph, find a path, take its smallest residual capacity, update the original flows including cancellations, and repeat.

Prove the final value: identify a cut whose capacity equals your feasible flow value.

Unless a search rule is specified, any valid augmenting path is acceptable. Different intermediate flows are allowed. Follow BFS on a small graph; programming and running-time proofs are optional study.

Next: shortest routes and minimum-cost connecting trees.

Appendix: proofs, extra examples, and algorithm details.

Appendix: proofs and additional examples

- Why balance equations are dependent
- Transportation and multicommodity bounds
- Maximum flow as minimum-cost circulation
- Cut bound and no-path optimality proofs
- Termination, slow path choices, and the BFS bound
- A larger complete maximum-flow example

Why one balance equation is redundant

Every column of the incidence matrix A has one $+1$ and one -1. Therefore $\mathbf{1}^\top A = 0$.

> If all but one balance equation hold and $\sum_i b_i = 0$, adding those equations forces the remaining equation.

If the network is connected when arc directions are ignored, $\operatorname{rank}(A) = |V| - 1$. To see why, $A^\top p = 0$ says $p_i = p_j$ along every arc, so connectivity makes every p_i equal.

For a disconnected network, balance is required separately in each connected component; deleting only one equation does not remove every dependency.

A one-variable check proves the transportation optimum

Supplies are (3,2), demands are (2,3), and costs by warehouse are (2,5) and (4,1). Set $q = x_{A1}$.

$$x_{A1} = q, \quad x_{A2} = 3 - q, \quad x_{B1} = 2 - q, \quad x_{B2} = q.$$

All flows are nonnegative exactly when $0 \le q \le 2$. The objective is

$$2q + 5(3 - q) + 4(2 - q) + q = 23 - 6q.$$

> The cost decreases as q increases, so $q = 2$ is optimal, with value 11.

An assignment LP has all supplies and demands equal to 1. With integer supplies and demands, a transportation LP with a finite optimum has an integer optimal solution; not every optimal solution must be integer.

Why the shared-capacity example has minimum cost 9

Commodity 1 sends 4 units from s to t; commodity 2 sends 1 unit from a to b. Every arc costs 1 per unit.

The five arcs are sa, sb, ab, at, bt. The network is directed and has no cycles.

> Each unit of commodity 1 must use at least two arcs, so its cost is at least $2(4) = 8$. Commodity 2 must use at least one arc, so its cost is at least 1.

$$\text{total cost} \geq 8 + 1 = 9.$$

Flows $x^1 = (2, 2, 0, 2, 2)$, $x^2 = (0, 0, 1, 0, 0)$ respect the shared capacities $(3, 3, 1, 3, 3)$ and attain 9, proving optimality.

Maximum flow is also a minimum-cost circulation

A **circulation** satisfies conservation at every node: all supplies are zero. Add an arc (t, s) to the original maximum-flow network.

Arc type	Cost	Capacity
Original arc	0	Original capacity
New t → s arc	-1	No upper bound needed

$$\min \quad -x_{ts} \quad \text{s.t. conservation at every node and original capacity bounds.}$$

> Conservation forces x_{ts} to equal the net original flow from s to t. Minimizing $-x_{ts}$ therefore maximizes the delivered amount.

Original finite capacities bound the possible delivery even though the new return arc has no explicit upper bound.

Proof: every cut bounds every feasible flow

Choose $s \in S, t \notin S$. Sum outgoing-minus-incoming balances over nodes in S. Intermediate nodes contribute zero and the source contributes F.

$$F = \sum_{S \to T} x_{ij} - \sum_{T \to S} x_{ij}.$$

Arcs with both endpoints in S occur once positively and once negatively, so they cancel.

$$F \le \sum_{S \to T} x_{ij} \le \sum_{S \to T} u_{ij}.$$

> The first inequality uses nonnegative flow on arcs entering S. The second uses the capacity of every arc leaving S. Thus no feasible flow exceeds any cut capacity.

Proof: no augmenting path implies optimality

Let S be the nodes reachable from s in the residual graph. If no augmenting path exists, $t \notin S$.

For original $i \in S, j \notin S$, residual forward capacity $u_{ij} - x_{ij}$ must be zero; otherwise j is reachable. Thus $x_{ij} = u_{ij}$.

For original $i \notin S, j \in S$, reverse residual capacity x_{ij} must be zero; otherwise the residual arc $j \to i$ reaches i.

$$F = \sum_{S \to T} u_{ij} - 0 = u(S, T).$$

> The cut bound is attained. The current flow is maximum and this cut is minimum.

This also proves the max-flow/min-cut theorem once a maximum flow is considered: an augmenting path at a maximum would contradict maximality.

Why integer capacities guarantee termination

Assume every capacity is a nonnegative integer. Start from zero flow and always take the full smallest residual capacity along the selected path.

> Initially flows and residual capacities are integers. The minimum positive capacity on a path is an integer, so updating by this amount preserves integrality. Every augmentation increases the flow value by at least 1.

$$\text{number of augmentations} \le F^* \le \sum_{j:(s,j)\in E} u_{sj}.$$

Here F^* is the maximum flow value. Thus there can be only finitely many augmentations. When they stop, no residual path remains and the cut argument proves optimality.

This bound depends on numerical capacity values. The next page shows that a small graph with large capacities can actually require many augmentations.

With irrational capacities, arbitrary path choices may fail to terminate.

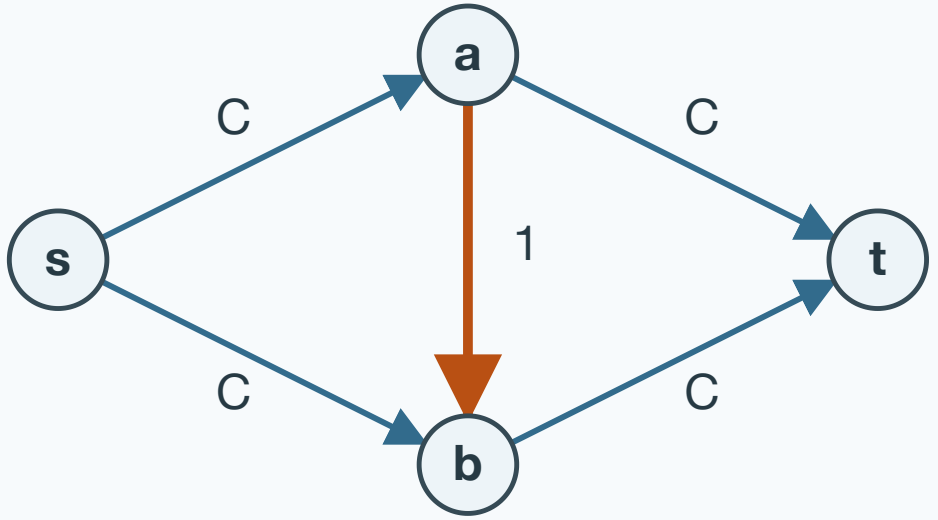
A small graph can need many augmentations

A changed-capacity example: the four outer arcs have capacity C, an integer with $C \geq 2$; the middle arc $a \to b$ has capacity 1. Start from zero flow.

Choose these paths repeatedly	Effect
s → a → b → t	Add 1; fill a → b.
s → b → a → t	Add 1; empty a → b.

After r pairs, flows in order (sa, sb, ab, at, bt) are $(r, r, 0, r, r)$. Repeat until $r = C$: $2C$ **augmentations**, final value $2C$.

BFS instead uses $s \to a \to t$ and $s \to b \to t$, sending C each time: **2 augmentations**.



Capacity labels: C on each outer arc, 1 on the middle arc.

If $C = 2^k$, each large capacity takes $k + 1$ binary digits to write, but this bad sequence needs 2^{k+1} augmentations. That is exponential in the input length.

Why BFS gives a polynomial bound

Let $d(v)$ be the fewest residual arcs from s to v; use ∞ for an unreachable node. Write $n = |V|$, $m = |E|$ for the original graph.

1. Distances cannot decrease after a BFS augmentation. Every old residual arc $u \to v$ satisfies $d(v) \le d(u) + 1$. A newly created reverse arc points backward along the chosen shortest path, so it also satisfies this inequality. Summing along any new source-to-node path proves its length is at least the old distance.

2. An arc can limit only $O(n)$ augmentations. If $u \to v$ becomes full on a BFS path, then $d(v) = d(u) + 1$. Before that residual arc can become full again, a later path must use $v \to u$. At that time $d_{\text{later}}(u) = d_{\text{later}}(v) + 1 \ge d_{\text{earlier}}(u) + 2$. Reachable distances are at most $n - 1$.

Each augmentation fills at least one residual arc. There are at most $2m$ directed residual arcs, so there are $O(nm)$ augmentations. Each search/update takes $O(n + m)$: the usual bound is $O(nm^2)$ after ignoring isolated nodes.

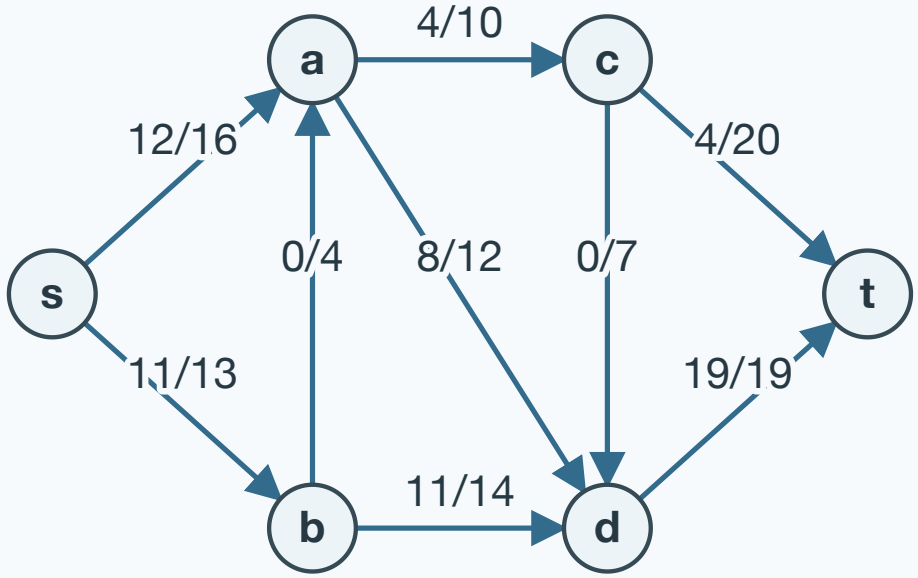
A larger feasible flow: check before claiming optimality

Every arc is directed. Labels show flow / capacity; the displayed feasible value is 23.

> The unused forward path $s \to a \to c \to t$ still permits 4 more units:

$$\min(16-12, 10-4, 20-4) = 4.$$

Thus value 23 is not optimal for this network.



Arc labels are flow / capacity. All arrows are original arcs; this is not a residual graph.

Two more augmentations deliver 29

Arc order	sa	sb	ba	ac	ad	bd	cd	ct	dt
Capacity	16	13	4	10	12	14	7	20	19
Initial flow	12	11	0	4	8	11	0	4	19
After s → a → c → t	16	11	0	8	8	11	0	8	19
After s → b → a → c → t	16	13	2	10	8	11	0	10	19

> First step: 4. Second step: $\min(13-11, 4-0, 10-8, 20-8) = 2$. The final source outflow is $16 + 13 = 29$.

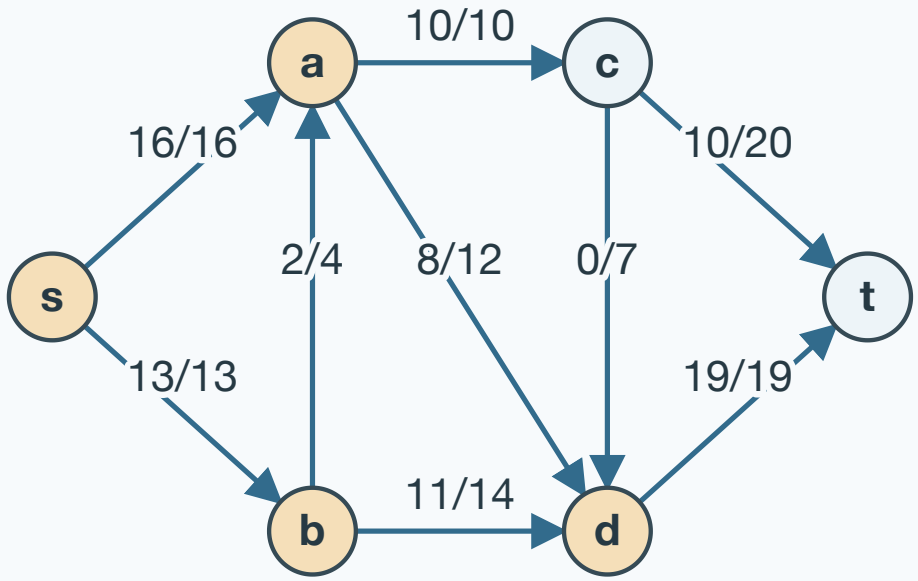
Each row preserves conservation at a, b, c, and d. Every arc respects its original capacity.

A cut certifies the larger network optimum

Take $S = \{s, a, b, d\}, T = \{c, t\}$.

> The outgoing cut arcs are $a \to c$ and $d \to t$, with total capacity $10 + 19 = 29$. The reverse crossing arc $c \to d$ has zero flow.

The feasible flow value 29 equals the cut capacity, proving optimality.



Arc labels are flow / capacity. All arrows are original arcs; this is not a residual graph.

Why node splitting preserves the processing limit

For a node v with no supply or demand and processing limit U_v, send all incoming arcs to v^-, all outgoing arcs from v^+, and join them by an arc of capacity U_v.

$$\sum_{i:(i,v)\in E} x_{iv} = x_{v^-v^+} = \sum_{j:(v,j)\in E} x_{vj}, \qquad 0 \le x_{v^-v^+} \le U_v.$$

> Every feasible original flow extends to a split flow by assigning the total throughput to the new arc. Conversely, conservation at the two split nodes gives the original balance and processing limit.

Node limits on sources or sinks require a clear convention for whether produced or consumed flow counts toward processing. The present construction applies to intermediate nodes with zero supply.