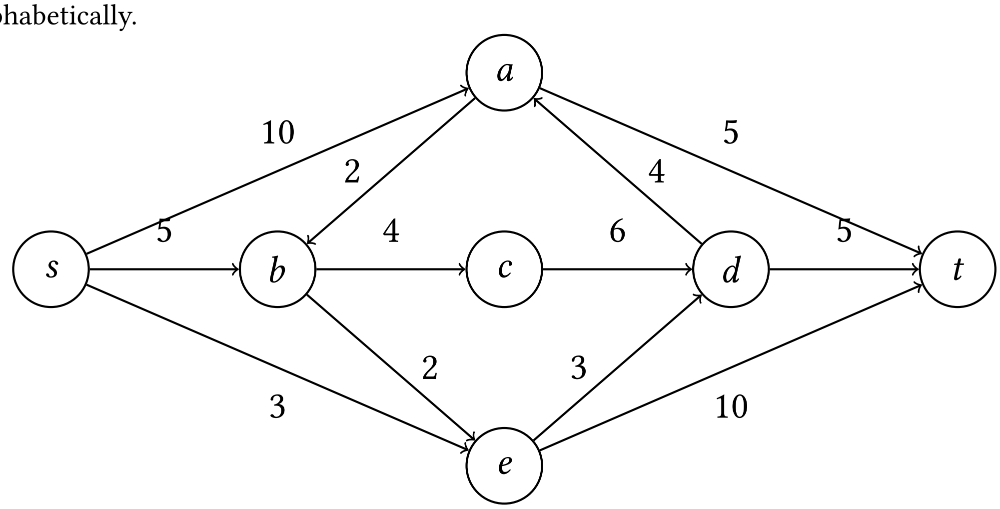
CSE 421 Winter 2026: Section 7

February 19, 2026

> **Instructions:** This section worksheet is designed to assist you in working through some example problems and developing your basics. You are encouraged to collaborate on the problems in this section worksheet as well as use the course's generative AI.

1 Mechanical calculations: Ford-Fulkerson flow

The following is a directed $s-t$ graph with capacities listed. Use the Ford-Fulkerson algorithm to calculate the *max flow* for the graph. Assume Ford-Fulkerson chooses augmenting paths by BFS with ties broken alphabetically.



2 Expressing problems as flows and cuts

This week, we are going to do something a little different. We are going to introduce a few problems that have all the flavors of a flow or cut problem, but don't exactly fit the standard setup. In this worksheet, take a look at the problem and figure out how to manipulate the problem into a flow network $G' = (V', E', c, s, t)$ by adding edges, vertices, sources, and sinks. Remember that to be a flow network, each edge needs a capacity, and there is exactly one identified source and one identified sink.

After having identified the new flow network G', argue how a max flow or min cut on the new flow network can be used to find a solution to your original problem. Like many other examples in this course, the key will be identifying a one-to-one correspondence (i.e., bijection) between solutions to the original problem and a max flow or min cut in the flow network.

2.1 Multiple sources and sinks

A city runs a daily logistics network for distributing emergency supplies. The transportation network is modeled as a directed graph $G = (V, E)$ with $|V| = n, |E| = m$. For each directed road $(u, v) \in E$, there is a known *integer* capacity $c(u, v) \geq 0$ representing the maximum number of supply crates that can be moved along that road in one day. Some vertices are *supply depots* (sources) and some are *relief sites* (sinks):

- A subset $S \subseteq V$ are depots. Each depot $s \in S$ can supply up to $b_s \in \mathbb{N}$ crates per day.
- A subset $T \subseteq V \setminus S$ are relief sites. Each site $t \in T$ can accept up to $d_t \in \mathbb{N}$ crates per day.
- Every vertex in $V \setminus (S \cup T)$ is a transfer hub where crates can be re-routed, but neither created nor destroyed.

A *shipping plan* assigns a nonnegative flow value $f(u, v)$ to each directed edge $(u, v) \in E$.

1. **Capacity constraints.** For every $(u, v) \in E$, $0 \leq f(u, v) \leq c(u, v)$.
2. **Flow conservation at transfer hubs.** For every hub $v \in V \setminus (S \cup T)$, $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)$.
3. **Supply limits at depots.** For every depot $s \in S$, $\sum_{(s,w) \in E} f(s, w) - \sum_{(u,s) \in E} f(u, s) \leq b_s$.
4. **Acceptance limits at relief sites.** For every relief site $t \in T$, $\sum_{(u,t) \in E} f(u, t) - \sum_{(t,w) \in E} f(t, w) \leq d_t$.

Objective. The total number of crates delivered is the total *net inflow* into the relief sites:

$$\text{Delivered}(f) = \sum_{t \in T} \left(\sum_{(u,t) \in E} f(u, t) - \sum_{(t,w) \in E} f(t, w) \right).$$

Reduce computing a feasible shipping plan f that maximizes $\text{Delivered}(f)$ to a flow problem.

2.2 Vertex capacities

A concert venue has a directed walkway network modeled as a graph $G = (V, E)$. There is a single entrance $s \in V$ and a single stage area $t \in V$. Each directed walkway $(u, v) \in E$ has an integer capacity $c(u, v) \geq 0$ (the maximum number of people per minute that can traverse that walkway). Assume no walkways go to s and no walkways lead away from t.

In addition, each intermediate checkpoint $v \in V \setminus \{s, t\}$ has an integer *vertex capacity* $p(v) \geq 0$, meaning at most $p(v)$ people per minute are allowed to pass through checkpoint v (due to ticket scanning or crowd control).

A *routing plan* assigns a nonnegative flow $f(u, v)$ to every edge $(u, v) \in E$.

1. **Edge capacities:** For all $(u, v) \in E$, $0 \leq f(u, v) \leq c(u, v)$.
2. **Flow conservation:** For all $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,w)\in E} f(v, w)$.
3. **Vertex capacities:** For all $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) \leq p(v)$.

Objective. The total number of people reaching the stage per minute is the flow value

$$|f| = \sum_{(s,w)\in E} f(s, w).$$

Reduce this problem to a "standard" flow problem.

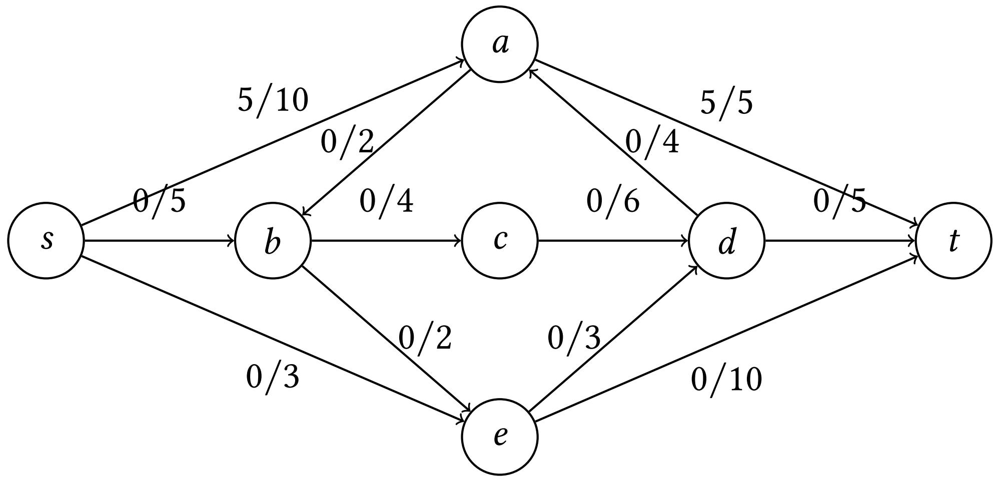
2.3 Disconnecting a graph

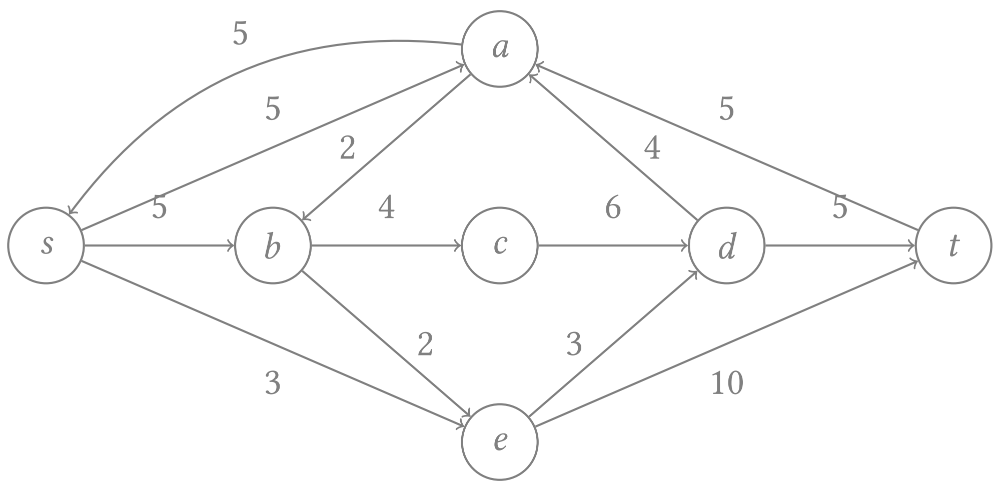
You are given a directed graph $G = (V, E)$ and two distinct vertices $s, t \in V$. Each vertex $v \in V \setminus \{s, t\}$ has an integer shutdown cost $w(v) \geq 0$. If you choose a set of vertices $X \subseteq V \setminus \{s, t\}$ to shut down, then all vertices in X (and all incident edges) are removed from the graph.
Task. Find a minimum-cost set $X \subseteq V \setminus \{s, t\}$ such that in the remaining graph there is *no directed path* from s to t.

3 Solutions

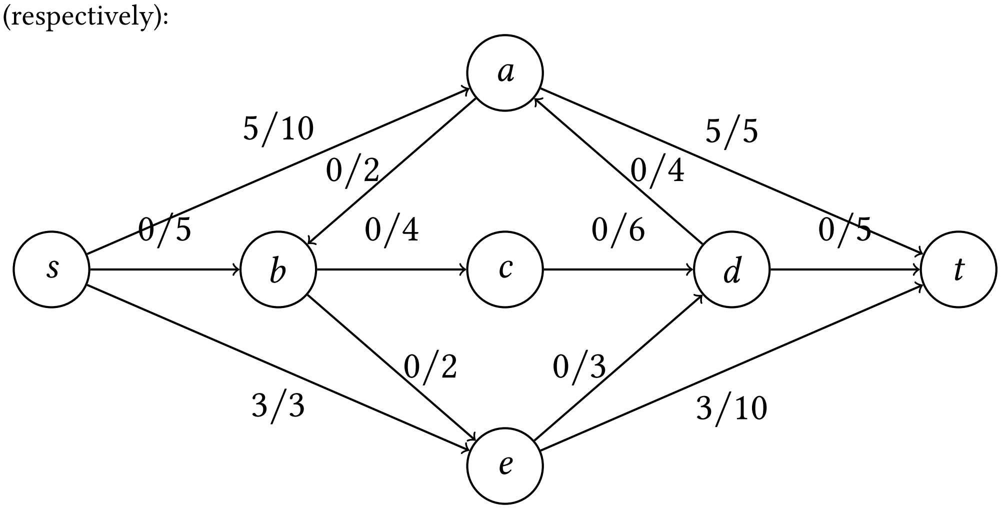
3.1 Mechanical calculations: Ford-Fulkerson Flow

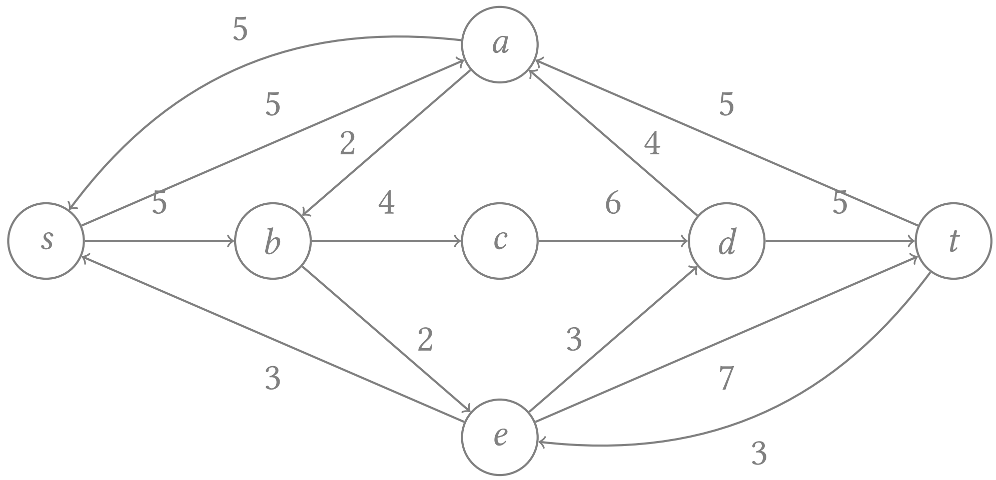
The first augmenting path will be $s \to a \to t$ of flow 5. The resulting flow and residual graph will be (respectively):



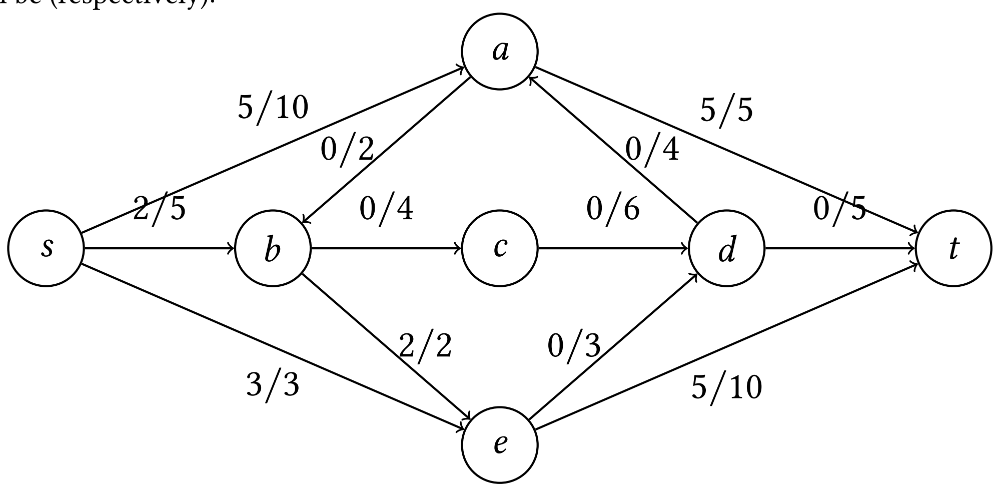


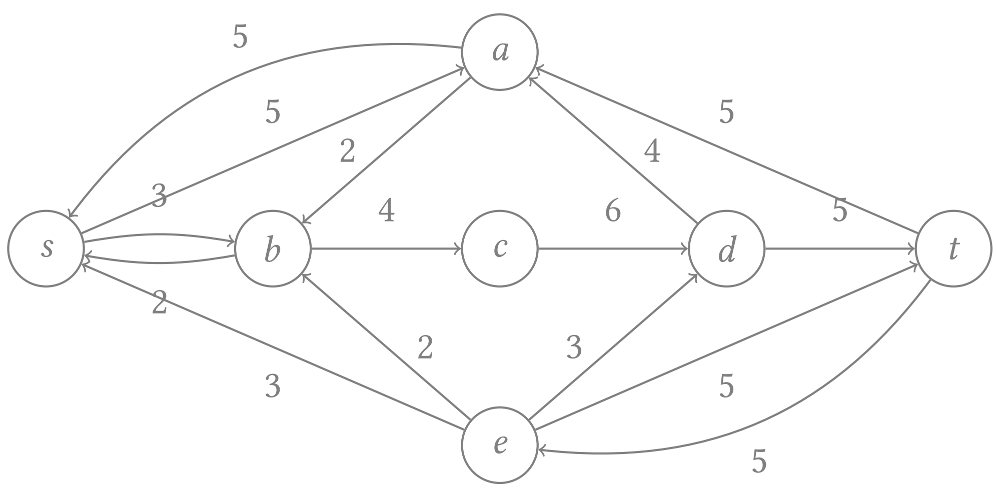
The second augmenting path will be $s \to e \to t$ of flow 3. The resulting flow and residual graph will be (respectively):



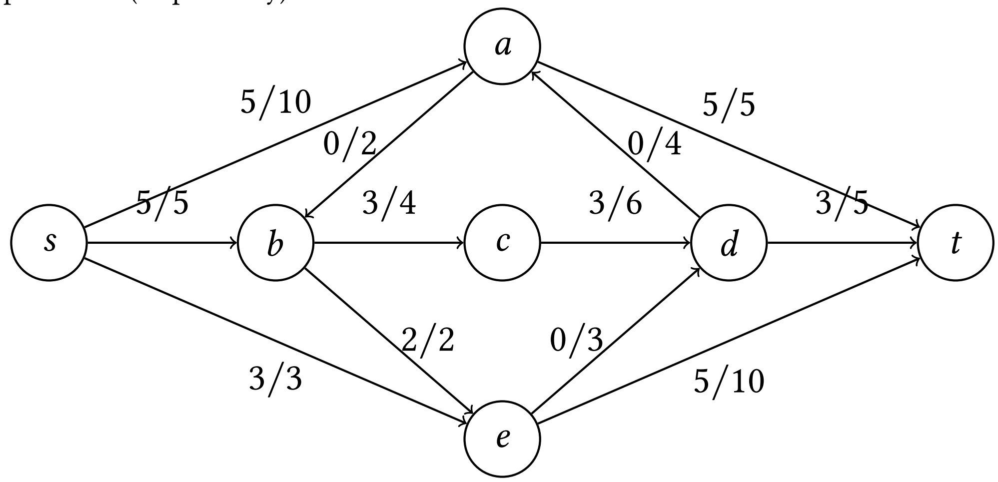


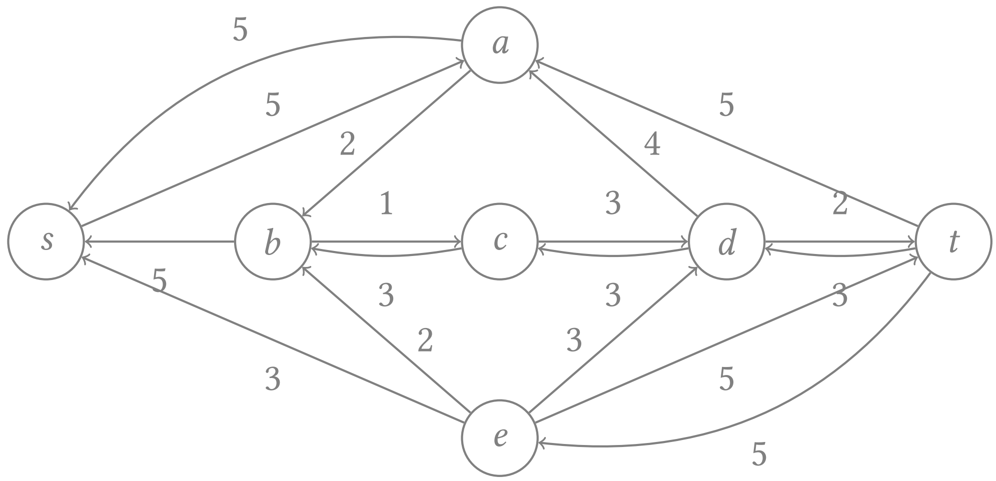
The third augmenting path will be $s \to b \to e \to t$ of flow 2. The resulting flow and residual graph will be (respectively):



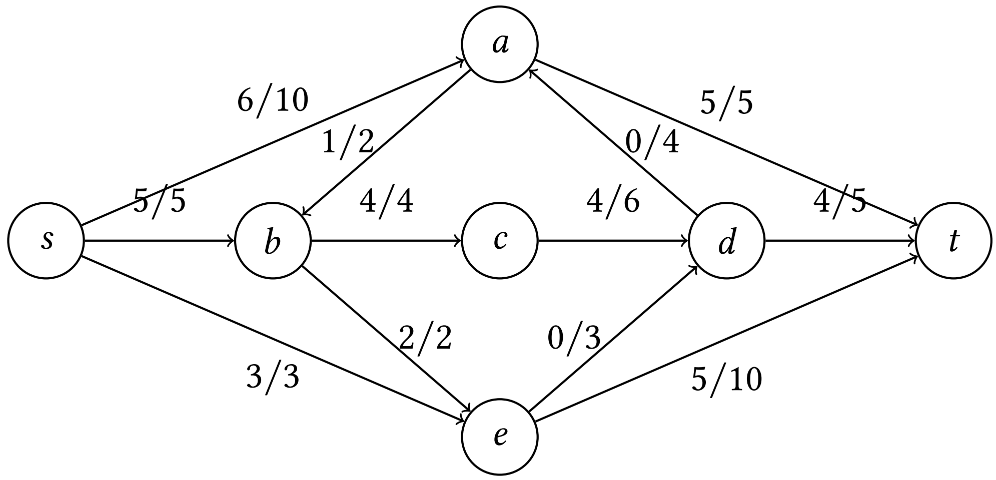


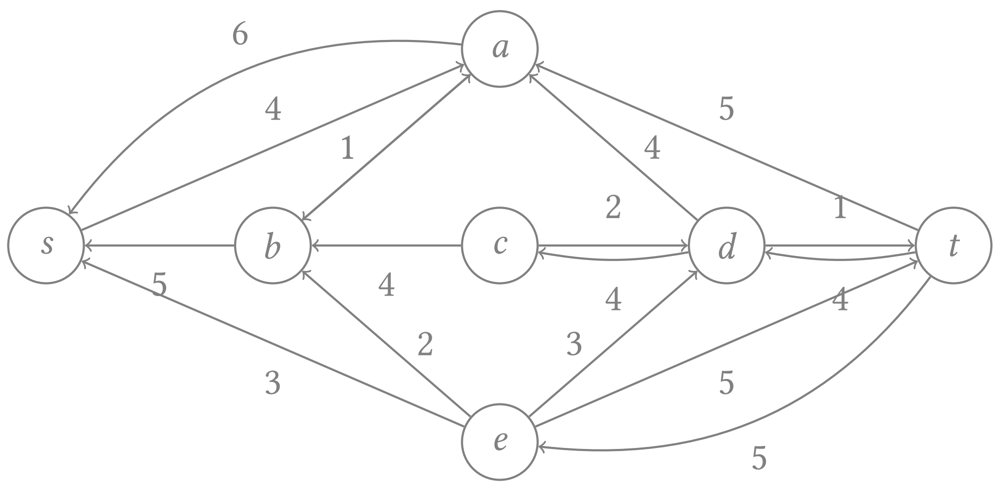
The fourth augmenting path will be $s \to b \to c \to\to d \to t$ of flow 3. The resulting flow and residual graph will be (respectively):





The fifth augmenting path will be $s \to a \to b \to c \to\to d \to t$ of flow 1. The resulting flow and residual graph will be (respectively):





There are now no more paths $s \rightsquigarrow t$ in the residual graph. Therefore, there are no additional augmenting paths, and the max flow is achieved.

We can also see this by recognizing that we have identified a flow of value 14 and the cut $S = \{s, a, b\}$, $T = \{c, d, e, t\}$ has cut $C(S, T) = 14$. As we have found a flow and cut that are equal in value, we know our algorithm has found the max flow by the duality theorem.

3.2 Flow reductions

3.2.1 Multiple sources and sinks

Reduction to a standard max flow instance. Construct a new flow network

$$G' = (V', E', c', \sigma, \tau)$$

as follows. Let $V' = V \cup \{\sigma, \tau\}$. For every original edge $(u, v) \in E$, include (u, v) in E' with capacity $c'(u, v) = c(u, v)$. For each depot $s \in S$, add an edge (σ, s) with capacity $c'(\sigma, s) = b_s$. For each relief site $t \in T$, add an edge (t, τ) with capacity $c'(t, \tau) = d_t$. Run a max flow algorithm on (G', σ, τ) and output the flow values on the original edges as the shipping plan.

Correctness. We show a one-to-one correspondence between feasible shipping plans f in G and feasible σ–τ flows F in G' with equal objective value.

Map shipping plan → flow. Given a feasible shipping plan f in G, define a flow F in G' by setting $F(u, v) = f(u, v)$ for every $(u, v) \in E$. For each depot $s \in S$, define its net outflow in f as

$$\Delta(s) = \sum_{(s,w)\in E} f(s, w) - \sum_{(u,s)\in E} f(u, s).$$

Set $F(\sigma, s) = \Delta(s)$. This is feasible because $\Delta(s) \leq b_s$ by the depot constraint, and also $\Delta(s) \geq 0$ because there are no flow conservation requirements forcing negative net outflow at a depot (if $\Delta(s) < 0$ then s would be a net sink, which can be removed by reducing incoming flow without decreasing delivery). For each relief site $t \in T$, define its net inflow in f as

$$\Gamma(t) = \sum_{(u,t)\in E} f(u, t) - \sum_{(t,w)\in E} f(t, w),$$

and set $F(t, \tau) = \Gamma(t)$, which is feasible since $\Gamma(t) \leq d_t$. All other edges incident to σ and τ do not exist.
We now check flow conservation in G'. Every hub $v \in V \setminus (S \cup T)$ satisfies conservation by assumption. For each depot $s \in S$, conservation in G' becomes

$$F(\sigma, s) + \sum_{(u,s)\in E} F(u, s) = \sum_{(s,w)\in E} F(s, w),$$

which holds by the definition of $F(\sigma, s) = \Delta(s)$. For each relief site $t \in T$, conservation in G' becomes

$$\sum_{(u,t)\in E} F(u, t) = \sum_{(t,w)\in E} F(t, w) + F(t, \tau),$$

which holds by the definition of $F(t, \tau) = \Gamma(t)$. Therefore F is a feasible σ–τ flow.
Finally, the value of F equals the delivered crates:

$$|F| = \sum_{(\sigma,s)\in E'} F(\sigma, s) = \sum_{s\in S} \Delta(s) = \sum_{t\in T} \Gamma(t) = \text{Delivered}(f),$$

where $\sum_{s\in S} \Delta(s) = \sum_{t\in T} \Gamma(t)$ because total net outflow over all vertices equals total net inflow over all vertices.

Map flow → shipping plan. Given a feasible σ–τ flow F in G', define a shipping plan f by restriction: $f(u,v) = F(u,v)$ for each $(u,v) \in E$. Edge capacities are preserved immediately. Every hub $v \in V \setminus (S \cup T)$ satisfies conservation because it satisfies conservation in G'. For each depot $s \in S$, conservation at s in G' implies

$$\sum_{(s,w)\in E} f(s,w) - \sum_{(u,s)\in E} f(u,s) = F(\sigma, s) \le b_s,$$

so the supply constraint holds. For each relief site $t \in T$, conservation at t in G' implies

$$\sum_{(u,t)\in E} f(u,t) - \sum_{(t,w)\in E} f(t,w) = F(t,\tau) \le d_t,$$

so the acceptance constraint holds. Also,

$$\text{Delivered}(f) = \sum_{t\in T} F(t,\tau) = |F|.$$

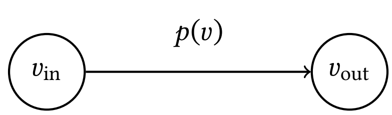
Therefore maximizing delivered crates is exactly maximizing the σ–τ flow value, so a maximum flow in G' yields an optimal shipping plan in G.

3.3 Vertex capacities

Reduction to a standard max flow instance. Construct a new flow network

$$G' = (V', E', c', s, t)$$

by splitting each vertex. For every vertex $v \in V \setminus \{s,t\}$, create two vertices v_{in} and v_{out}. Add an edge $(v_{\text{in}}, v_{\text{out}})$ with capacity $c'(v_{\text{in}}, v_{\text{out}}) = p(v)$.



Keep s and t unchanged. For each original edge $(u,v) \in E$, add an edge in G' from $u_{\text{out}} \to v_{\text{in}}$ with capacity $c'(u_{\text{out}}, v_{\text{in}}) = c(u,v)$, where we interpret $s_{\text{out}} = s$ and $t_{\text{in}} = t$. Run a standard max flow algorithm on (G', s, t).

Correctness. We show a one-to-one correspondence between feasible routing plans f in G and feasible s–t flows F in G'.

Map routing plan → flow. Given a feasible routing plan f in G, define F in G' by setting

$$F(u_{\text{out}}, v_{\text{in}}) = f(u, v) \quad \text{for each } (u, v) \in E.$$

For each internal vertex $v \in V \setminus \{s, t\}$, set

$$F(v_{\text{in}}, v_{\text{out}}) = \sum_{(u,v)\in E} f(u, v).$$

This satisfies the capacity constraint on $(v_{\text{in}}, v_{\text{out}})$ because the vertex constraint in G says

$$\sum_{(u,v)\in E} f(u, v) \le p(v).$$

Flow conservation at v_{in} and v_{out} holds because all inflow to v_{in} is sent across $(v_{\text{in}}, v_{\text{out}})$, and all outflow from v_{out} matches the original outflow from v; this is exactly the original conservation equation at v.
Map flow → routing plan. Given a feasible flow F in G', define a routing plan f in G by

$$f(u, v) = F(u_{\text{out}}, v_{\text{in}}).$$

Edge capacities hold since $c'(u_{\text{out}}, v_{\text{in}}) = c(u, v)$. For any internal vertex v, flow conservation in G' implies that the total inflow to v_{in} equals $F(v_{\text{in}}, v_{\text{out}})$ and also equals the total outflow from v_{out}. Therefore

$$\sum_{(u,v)\in E} f(u, v) = F(v_{\text{in}}, v_{\text{out}}) \le p(v),$$

so the vertex capacity constraint holds, and the equality of inflow and outflow gives the original conservation at v. The flow value $|f| = \sum_{(s,w)\in E} f(s, w)$ equals the s–t flow value in G' because s has no incoming edges and t has no outgoing edges.
Therefore, the maximum routing rate in G equals the maximum s–t flow value in G'.

3.4 Disconnecting a graph

Reduction to a standard min cut instance. We reduce the minimum-cost vertex shutdown problem to an s–t minimum cut problem by splitting vertices and assigning capacities so that cutting a vertex has exactly its shutdown cost.
Build a directed flow network

$$G' = (V', E', c', s, t)$$

as follows. For every vertex $v \in V \setminus \{s, t\}$, create v_{in} and v_{out} and add an edge $(v_{\text{in}}, v_{\text{out}})$ with capacity $c'(v_{\text{in}}, v_{\text{out}}) = w(v)$. Keep s and t unchanged. For every original edge $(u, v) \in E$, add an edge $(u_{\text{out}}, v_{\text{in}})$ with capacity $c'(u_{\text{out}}, v_{\text{in}}) = \infty$. Compute a minimum s–t cut (A, B) in G', and output

$$X = \{v \in V \setminus \{s,t\} : v_{\text{in}} \in A \text{ and } v_{\text{out}} \in B\}.$$

Correctness. *(1) Any vertex shutdown set gives a cut of the same cost.* Let $X \subseteq V \setminus \{s,t\}$ be any set of vertices whose deletion disconnects s from t in G. Consider the graph $G - X$ and let R be the set of vertices reachable from s in $G - X$. Define a cut (A, B) in G' by putting

$$s \in A, \quad t \in B,$$

and for each $v \in V \setminus \{s,t\}$:

$$v \in R \Rightarrow v_{\text{in}}, v_{\text{out}} \in A, \qquad v \notin R \Rightarrow v_{\text{in}}, v_{\text{out}} \in B,$$

except that for $v \in X$ we place $v_{\text{in}} \in A$ and $v_{\text{out}} \in B$. Since there is no path from s to t in $G - X$, there is no original edge in G that goes from a reachable vertex in R to a non-reachable vertex outside R without passing through a deleted vertex; in G' this implies no M-capacity edge crosses from A to B. The only edges crossing the cut are $(v_{\text{in}}, v_{\text{out}})$ for $v \in X$, so the cut capacity is $\sum_{v \in X} w(v)$.
(2) Any finite cut corresponds to a vertex shutdown set. Let (A, B) be any s–t cut in G' with finite capacity Then the cut cannot contain any edge of capacity ∞, because a single such edge would already give cut capacity equaling ∞. Therefore, every edge crossing from A to B must be of the form $(v_{\text{in}}, v_{\text{out}})$, and we can define

$$X = \{v : v_{\text{in}} \in A,\ v_{\text{out}} \in B\}.$$

We claim that deleting X disconnects s from t in G. If there were a directed path from s to t in $G - X$, then in G' there would be a directed path from s to t that uses only M-capacity edges $(u_{\text{out}}, v_{\text{in}})$ and the internal edges $(v_{\text{in}}, v_{\text{out}})$ for vertices not in X. Such a path would cross from A to B at some point, which would force an M-capacity edge to cross the cut, contradicting that the cut capacity is less than M. Thus X is a valid shutdown set.
(3) Optimality. By (1), every valid shutdown set X gives an s–t cut with capacity $\sum_{v \in X} w(v)$. By (2), every minimum cut gives a valid shutdown set with the same total cost. Therefore, a minimum s–t cut in G' produces a minimum-cost set of vertices to shut down in G.