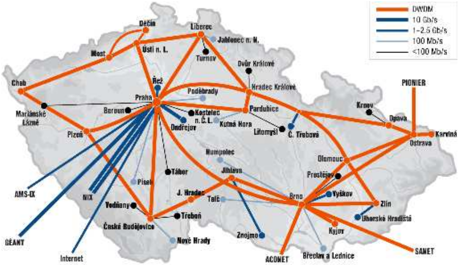
4 Graph Cuts and Network Flows

Yet another area of rich applications of graphs (digraphs) deals with so called *flow networks* and "commodity flows" in them.



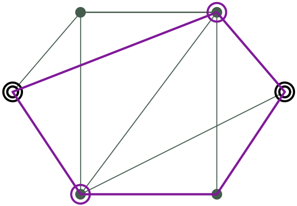
Brief outline of this lecture

- More on graph connectivity – vertex and edge cuts in graphs.
- Flow networks, admissible flows and network cuts, circulations.
- Finding a maximum flow via augmenting of residual paths.
- Extensions, and applications in connectivity, SDR and vision.

4.1 Connectivity and Cuts

Recall from Lecture 2:

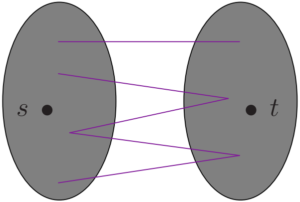
- A graph G is *edge-k-connected*, $k > 1$, if G stays connected even after removal of any subset of $\leq k-1$ edges. □
- A graph G is *(vertex-)k-connected*, $k > 1$, if G stays connected even after removal of any subset of $\leq k-1$ vertices.
  Specially, the complete graph K_n is vertex-$(n-1)$-connected by the definition.□
- Menger's theorem: A graph G is k-connected iff there exist $\geq k$ internally disjoint paths between any pair of vertices (the paths may share only their ends).

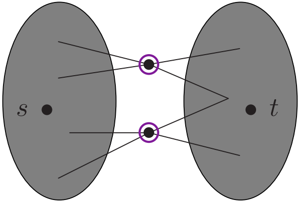


Small so-called *cuts* are the obstacles to high connectivity in graphs.

Edge / vertex cuts

An *s–t path* in a graph G is a path in G with the ends $s, t \in V(G)$.





Definition: Let G be a graph and $s, t \in V(G)$. An edge set $F \subseteq E(G)$ is an *s–t edge cut* in G if the subgraph $G - F$ (deleting of the edges F from G) has no s–t path. □

In other words, if s and t belong to different conn. components of the subgraph $G - F$.
□

Similarly, a vertex set $X \subseteq V(G)$ is a *s–t vertex cut* in G if the subgraph $G - X$ (deleting of the vertices X from G) has no s–t path. □

An s–t cut is called *minimal* if no proper subset of it is an s–t cut again. □

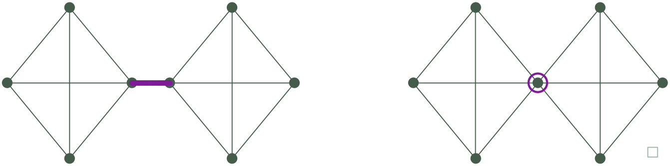
If s–t are implic. known, or irrelevant, then we shortly say an *edge / vertex cut* in G.

vertex / edge cut = hranový / vrcholový řez
minimal cut = minimální řez

Small cuts and blocks

Definition: A *bridge* in a graph is a minimal edge cut consisting of one edge.

A *cutvertex* in a graph is a minimal vertex cut consisting of one vertex.



Fact: A connected graph G is not 2-connected iff G has a cutvertex or $G \simeq K_1, K_2$. □

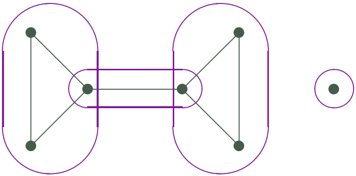
A graph G on $|V(G)| > k$ vertices is not k-connected iff G has a vertex cut of size $k-1$ or less. (Note that a disconnected graph has a cut of size 0.) □

Definition: A *block* in a graph G is a maximal (by inclusion) 2-connected subgraph of G. Moreover, a vertex with no edges and an edge not contained in any larger block of G are also called *blocks* of G. □

Proposition 4.1. *A subgraph $H \subseteq G$ is a block if, and only if, H is maximal (among subgraphs of G) with the property that H contains no cutvertex (of H itself).*

bridge = most, cutvertex = artikulace, disconnected graph = nesouvislý graf

An excersise: the structure of graph blocks



Proposition 4.2. *Let G be a graph and $B_1, \ldots, B_k$ be the blocks of G. If $B_i \cap B_j \neq \emptyset$, then $B_i \cap B_j = \{c\}$ where c is a cutvertex of G.* $\square$

Proof by means of contradiction: Let $B_i \cap B_j \supseteq \{c, d\}$ where $c \neq d$. Since B_j is connected by the definition, there exists a path $P \subseteq B_j$ with the ends c, d. $\square$

Let $B^+ := B_i \cup P \subseteq G$. Since B_i is 2-connected, so is B^+ (Thm. 2.13, "adding an ear P"). However, this contradicts the definition of a block; B_i already was a maximal 2-connected subgraph of G. $\square$

The same argument proves the last part, that $c \in B_i \cap B_j$ is a cutvertex, too. Assume not, then for some neighbours u, v of c such that $u \in V(B_i)$ and $v \in V(B_j)$, there exists a u–v path $Q \subseteq G$ and $B^+ := B_i \cup (Q + vc)$ contradicts maximality of B_i. $\square$ $\square$

Corollary 4.3. *The bipartite "incidence graph" between the blocks and the cutvertices of a graph G is a forest (it contains no cycles).*

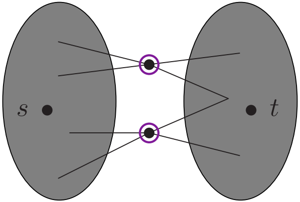
... by contradiction = důkaz sporem

Menger's theorem reformulated

Recall. . .

- Menger's theorem: A graph G is k-connected iff there exist $\geq k$ internally disjoint paths between any pair of vertices (the paths may share only their ends).

Theorem 4.4. *Let G be a graph and $s, t \in V(G)$. There exist k internally disjoint s–t paths in G if, and only if, there is no s–t vertex cut in G of size less than k.* □



A sketch: the "only if" direction is obvious, and for the "if" direction, □

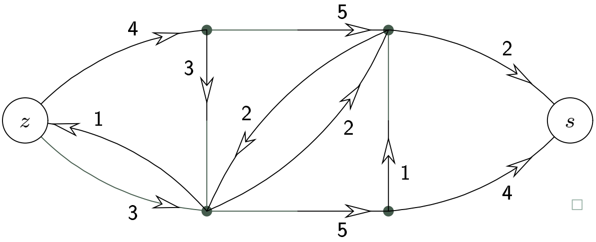
we will generalize the whole setting to "weighted connectivity" next. . .

4.2 Digraphs as Flow Networks

Flow networks present a convenient "weighted generalization" of the concept of graph connectivity and cuts, with long history of research and many practical applications. □

Definition 4.5. ***A flow network*** is a quadruple $\bar{G} = (G, z, s, w)$ such that

- G is a digraph (it is important to have directed edges),
- the vertices $z \in V(G)$, $s \in V(G)$ are the *source* and the *sink*, respectively,
- and $w : E(G) \to \boldsymbol{R}^+ \cup \{\infty\}$ is a positive weighting of the arcs (edges) of G, these weights are called *edge capacities*.



□

Remark: In a real world, more than one source or sink may exist in a flow network, but that is not a problem — we simply create a single artificial "supersource" and draw arcs from it to all the real sources (even with source capacities), and the same with a "supersink".

flow network = síť, source = zdroj, sink = stok, capacity = kapacita

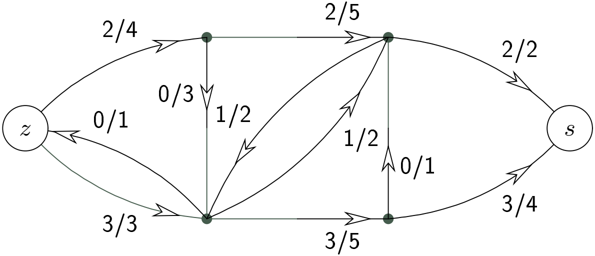
Notation: For simplicity, we shall write $e \to v$ to mean that an arc e "points to" (has its head in) the vertex v, and $e \leftarrow v$ analogously for e "leaving" (having tail in) v. □

Definition 4.6. ***A network flow,*** in a flow network $\bar{G} = (G, z, s, w)$, is an assignment $f : E(G) \to \boldsymbol{R}_0^+$ satisfying (we say f is *admissible*)

- $\forall e \in E(G) : \; 0 \le f(e) \le w(e),$ (capacity constraints)
- $\forall v \in V(G), v \ne z, s : \; \sum_{e \to v} f(e) = \sum_{e \leftarrow v} f(e).$ (flow conservation) □

The *value (size)* of a flow f is the quantity $\|f\| = \sum_{e \leftarrow z} f(e) - \sum_{e \to z} f(e)$. □

Notation: The flow value F and the capacity C of an arc in a picture of a network will be shortly denoted by F/C, respectively.



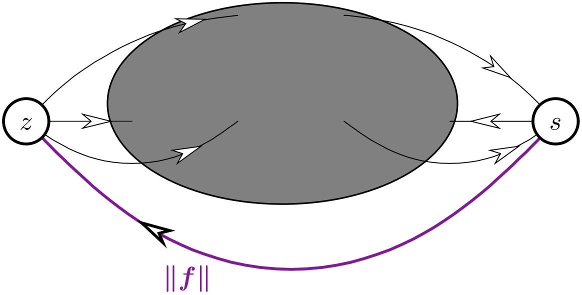
network flow = tok v síti
admissible = přípustný, capacity constraints = kapacitní omezení, flow conservation = zachování toku

Circulations

Definition: A flow in a flow network $\bar{G} = (G, z, s, w)$ satisfying the flow conservation constraint at all the vertices including the source and sink z, s is called a *circulation*.

The source and the sink hence become irrelevant for circulations. □

Fact: There is a one-to-one correspondence between a *flow* f in a flow network $\bar{G}$ and the following *circulation* in the enhanced network $\bar{G} + (s, z)$:



- simply add the "reverse" arc (s, z) assigned *flow value* $\|f\|$ and capacity $+\infty$.

circulation = cirkulace (v síti)

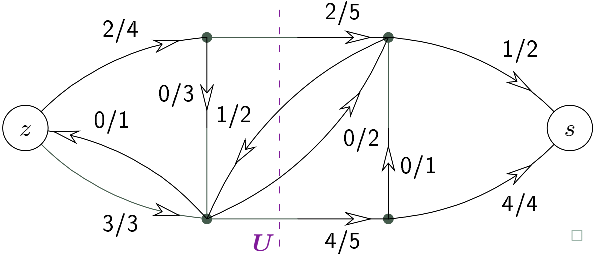
Boundary flow property

Notation: For $\bar{G}$ and $U \subseteq V(G)$, let $e \to U$ mean that an arc e points from $V(G) \setminus U$ towards U, and $e \leftarrow U$ mean that an arc e points from U towards $V(G) \setminus U$. □

Definition: Let $\bar{G}$ be a flow network and $U \subseteq V(G)$. The *boundary surplus* of a flow f on U is

$$\sum_{e \leftarrow U} f(e) - \sum_{e \to U} f(e) \,. \square$$

Surplus on U is 5 in the picture:



Fact: The boundary surplus of f on $\{z\}$ is exactly the flow size $\|f\|$ by the definition.

We aim to extend this finding to other suitable U's.

boundary surplus = přebytek na hranici

Lemma 4.7. *Let $\bar{G}$ be a flow network and $\emptyset \neq U \subseteq V(G)$. If f is a circulation in $\bar{G}$, then the boundary surplus of f on U is always 0.*

Proof: We use induction on $|U|$. For $U = \{x\}$, the claim is just the flow conservation constraint at x. □

Consider now arbitrary $U \subseteq V(G)$ where $|U| \geq 2$ and $y \in U$. By the induction assumption, the claim holds for $U' = U \setminus \{y\}$. Now we compute

$$\sum_{e \leftarrow U} f(e) = \left[\sum_{e \leftarrow U'} f(e) - \sum_{e \leftarrow U' \,\wedge\, e \rightarrow y} f(e)\right] + \left[\sum_{e \leftarrow y} f(e) - \sum_{e \leftarrow y \,\wedge\, e \rightarrow U'} f(e)\right] \square$$

and similarly

$$\sum_{e \rightarrow U} f(e) = \left[\sum_{e \rightarrow U'} f(e) - \sum_{e \rightarrow U' \,\wedge\, e \leftarrow y} f(e)\right] + \left[\sum_{e \rightarrow y} f(e) - \sum_{e \rightarrow y \,\wedge\, e \leftarrow U'} f(e)\right] \square.$$

Therefore, the new surplus is

$$\sum_{e \leftarrow U} f(e) - \sum_{e \rightarrow U} f(e) = \left[\sum_{e \leftarrow U'} f(e) - \sum_{e \rightarrow U'} f(e)\right] + \left[\sum_{e \leftarrow y} f(e) - \sum_{e \rightarrow y} f(e)\right] = 0 + 0.$$

□

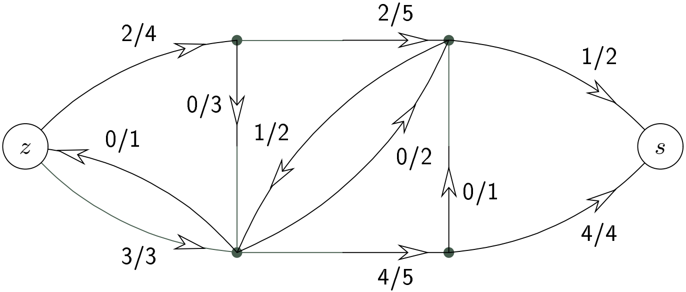
Where to measure flow size

Lemma 4.7. *Let $\bar{G}$ be a flow network and $\emptyset \neq U \subseteq V(G)$. If f is a circulation in $\bar{G}$, then the boundary surplus of f on U is always 0.*

Corollary 4.8. *Let $\bar{G} = (G, z, s, w)$ be a flow network and f an admissible flow in $\bar{G}$. For any $U \subseteq V(G)$ such that $z \in U \not\ni s$ (U "separating" z from s), the boundary surplus of f on U is always the same, equal to $\|f\|$.* □

Proof: Turn f into a circulation f' in $G + sz$ by letting $f' \equiv f$ on G and $f'(sz) = \|f\|$, and apply Lemma 4.7.

Then, for any such U, it is $sz \to U$ and so $f'(sz)$ is accounted for in the boundary surplus of f' on U, which is 0. The surplus of f on U hence equals $f'(sz) = \|f\|$. □



4.3 The Max-flow Min-cut Theorem

Edge cuts in graphs have a natural weighted generalization into flow networks...

Definition 4.9. ***A cut*** in a flow network $\bar{G} = (G, z, s, w)$
is a subset of edges (arcs) $C \subset E(G)$ such that there is no $z \to s$ directed path in G completely avoiding C (i.e., s is not reachable from z in $G - C$). □

The *capacity (size)* of a cut C is the sum of the capacities of arcs in C, i.e.,

$$\|C\| = \sum_{e \in C} w(e).\square$$

The most important result in this area is the following:

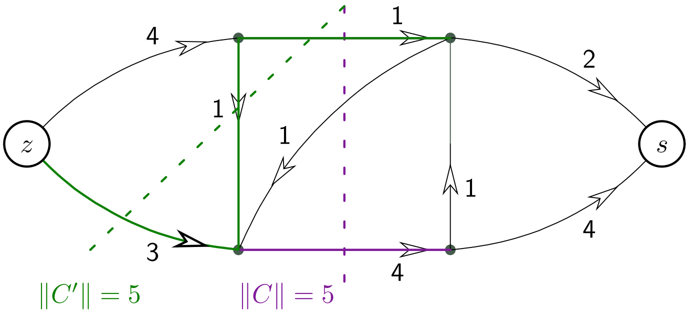
Theorem 4.10. (Ford–Fulkerson) *In any flow network $\bar{G}$, there exists an admissible flow of size $r \in \boldsymbol{R}^+$ if, and only if, there is no cut in $\bar{G}$ of capacity less than r.* □

Maximum possible flow size = minimum cut capacity.

An example

What is a cut of the least capacity in this example?

Is it unique?



We can see two cuts C and C' of the same size 5, and no smaller one...

On the flow–cut duality

Theorem 4.10 (repeated). In any flow network $\bar{G}$, there exists an admissible flow of size $r \in \boldsymbol{R}^+$ if, and only if, there is no cut in $\bar{G}$ of capacity less than r.

How to read this theorem? □

- If one looks for a certificate that a flow is maximum possible, then it always suffices to exhibit a cut of the same value. □
- Likewise, to certify minimality of a cut, one exhibits a flow of the same value. □
- Nice properties of this kind are commonly called *good characterizations* (of a problem)—one can certify optimality of a solution by giving an obvious obstacle.□

  Such as, in our case, the following:

Proposition 4.11. *In any $\bar{G}$, for any adm. flow f and any cut C, it holds $\|f\| \leq \|C\|$.*□

Proof: Let U be the set of vertices of G reachable from z in $G - C$, where $s \notin U$. Let $F_U = \{e : e \leftarrow U\}$. Then $F_U \subseteq C$ by the definition of U. By Corollary 4.8,

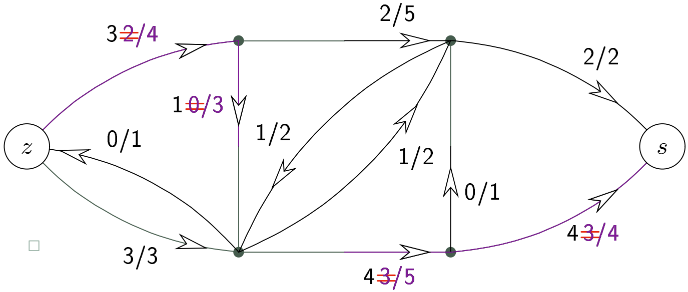
$$\|f\| = \sum_{e \in F_U} f(e) - \sum_{e \to U}(e) \;\leq\; \sum_{e \in F_U} w(e) \leq \|C\|.$$

□

good characterizations = dobrá charakterizace, obstacle = překážka

4.4 Finding the Maximum Flow

A question: is the following flow maximum possible?



Now it is, of size 6, and we have got a cut of capacity 6 as well. □

Fact: There exist quite simple and fast algorithms to find the maximum flow (and the minimum cut at the same time) in a given flow network.

– These simple algorithms are based on an idea to saturate "residual z–s paths".

Problem formulation

Problem 4.12. ***The Max–Flow problem***
Given a flow network $\bar{G} = (G, z, s, w)$, the task is to find a flow f in $\bar{G}$ from z to s such that the flow size $\|f\|$ is maximized (among all admissible flows in $\bar{G}$). □

Where can one find the max-flow problem?

- in transportation (or distribution) networks of goods, electricity, etc.
- in pipe networks (water, gas, oil, sewerage, etc.) □
- in IP packet routing (real-time transmission of large data), □
- in various "matching" or "representatives" problems, □
- in computer vision – as image segmentation (the min-cut).

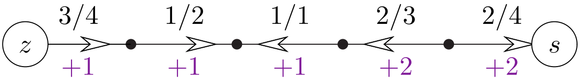
Residual and Augmenting Paths

Definition: Consider a flow network $\bar{G}$ and an admissible flow f in it.

A *residual z–x path* (in $\bar{G}$ w.r.t. f and the source z) is an undirected path in G from the source z to any vertex x, i.e. a sequence of adjacent edges $e_1, e_2, \ldots, e_m$; □

- such that $f(e_i) < w(e_i)$ if e_i is directed away from z,
- and $f(e_i) > 0$ if e_i is directed towards z ("backwards").

residual capacities:



> 0

The quantity $w(e_i) - f(e_i)$, or $f(e_i)$, resp., is called the *residual capacity* of edge e_i.□

The background idea is as follows.

- A residual z–s path has strictly positive residual capacity $\varepsilon > 0$, and so one can "push" an additional ε amount of flow from the source z to the sink s. □
- However, what does "pushing flow against an arc" mean??? □
  Actually, we stop a (bit of) "returning" flow, and send new amount of flow instead of it. (Imagine stoppig a crowd of returning people and thus making more room.)

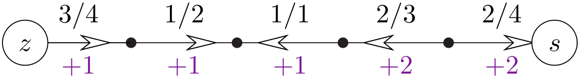
residual path = nenasycená (zbytková) cesta, residual capacity = zbytková kapacita

Method 4.13. ***Maximizing flow via residual paths.***

The overall idea is really simple; □

one should repeatedly augment (meaning to enlarge) the current flow by adding to it along existing residual paths. . .

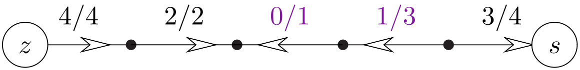
How to augment a residual path:

residual capacities: 

> 0

□

min. residual capacity $r = 1 > 0$ $\rightsquigarrow$ augmenting the current flow by $+1$:

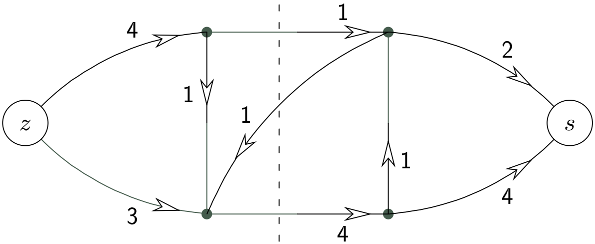


□

Fact: Augmenting (enlarging) an admissible flow by the minimal residual capacity of a residual z–s path results in an admissible flow, again.

augmentation = zvětšení

A simple Residual Path Algorithm



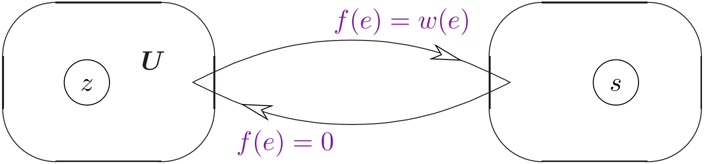
Algorithm 4.14. ***Ford–Fulkerson's for network flows.***

`input` $\leftarrow$ *a flow network* $\bar{G} = (G, z, s, w)$;
flow $f \equiv 0$;
`repeat {`
    Search (BFS) the graph G *to find the set* U *of those vertices*
        reachable from the source z *along residual paths*;
    `if (` $s \in U$ `) {`
        `P =` *any residual* z–s *path in* $\bar{G}$ *(this* P *then called an augmenting path)*;
        Augment ("enlarge") f *by the minimal residual capacity along* P;
    `}`
`until (` $s \notin U$ `)`;
`output` $\rightarrow$ *a maximum flow* f *in* $\bar{G}$;
`output` $\rightarrow$ *a minimum cut in* $\bar{G}$ *from* U *to* $V(G) - U$.

Proof of Algorithm 4.14:

For any flow f and any cut C in $\bar{G}$, it holds $\|f\| \leq \|C\|$. If the algorithm stops with a flow f in $\bar{G}$ and a cut C such that $\|C\| = \|f\|$, then it is clear that f is a maximum flow in $\bar{G}$. (We have, however, not proved yet that the algorithm stops!) □

So to prove that whenever the algorithm stops with f, C, then $\|f\| = \|C\|$, we use the following schematic picture (in which s does not belong to the reachable set U):



□

Since no further vertex than U is reachable along residual paths, every arc e leaving U has full flow $f(e) = w(e)$, and every arc e entering U has zero flow $f(e) = 0$. Therefore;

$$\sum_{e \leftarrow U} f(e) - \sum_{e \rightarrow U} f(e) = \sum_{e \leftarrow U} f(e) = \sum_{e \in C} w(e) = \|C\| \ .$$

Finally, by Corollary 4.8, we have $\|f\| = \sum_{e \leftarrow U} f(e) - \sum_{e \rightarrow U} f(e) = \|C\|$, finishing the proof. □

Basic Consequences

Algorithm 4.14 and its proof provides several interesting mathematical findings:

- Together with Prop. 4.11, it nearly(!) *proves* Theorem 4.10 (flow-cut duality); □
  - what is missing, is a proof that the algorithm terminates, □
  - and the latter is not obvious at all for arbitrary real capacities—there do exist non-terminating (and even non-convergent) real examples. □
- If all the capacities are *integers*, then every step of Algorithm 4.14 deals with integral residua and flows; □
  - consequently, the algorithm must terminate under any circumstances,
  - and the resulting flow will be integral as well. □
- For instance, if the edge capacities are all set to 1, one obtains the following.

  For a graph and vertices s, t, there exist k edge-disjoint s–t paths iff there is no s–t edge cut of size less than k. □

Compare this to Menger's theorem, and to Section 4.5...

More precise formulations

…of the previous findings…

Theorem 4.15. (Edmonds–Karp)
If Algorithm 4.14 always chooses an augmenting path among residual paths of the least length (measured by the number of edges), e.g., by the BFS, then the algorithm is guaranteed to terminate after $O(|V(G)| \cdot |E(G)|)$ iterations.

Consequently, Theorem 4.10 is proved. □

Proposition 4.16. *If the edge capacities in a flow network $\bar{G}$ are integral, then there exists a maximum flow which is integral, too.*

Algorithm 4.14 outputs such an integral flow in a finite number of steps.

4.5 Further Improvements and Extensions

More officient flow algorithms

- ***Edmonds–Karp*** (Theorem 4.15):

  BFS is used to search for an augmenting path, runtime $O(|V(G)| \cdot |E(G)|^2)$. □

- ***Dinitz***:
  - BFS is used to find all the shortest residual paths in $\bar{G}$, creating a "layered" residual network.
  - The layered network is then completely saturated in one run.

  Only $O(|V(G)|)$ iterations of the main cycle, runtime $O(|V(G)|^2 \cdot |E(G)|)$. □

- ***MPM "Three Indians"***:

  Similar to [Dinitz], but a layered network is saturated faster, runtime $O(|V(G)|^3)$. □

- An advance note on a special "*planar*" case:

  A minimum cut can be found as a shortest path in the dual graph (see Lecture 8).

Networks with lower capacities

In a flow network with *lower capacities*, in addition to the weight function w, there is another weight function $\ell : E(G) \to \boldsymbol{R}_0^+$ giving the lower edge capacities.

A flow f is then *admissible* if $\ell(e) \leq f(e) \leq w(e)$ for every edge e of the network. □

Notice that an admissible flow may not exist in such a lower-capacitated network. □

Algorithm 4.17. ***Max-flow in a lower-capacitated network***

The solution in a network $\bar{G} = (G, z, s, w, \ell)$ is found in two stages:

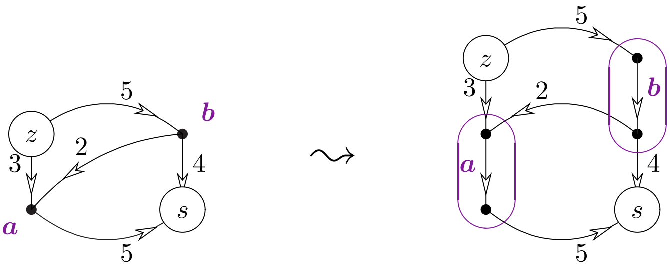
- *First, an admissible circulation r is found in $\bar{G} + sz$, respecting both the lower an upper bounds ℓ, w. This is done by finding a maximum flow in an artificial network modelling the "surplus" of lower capacities at every vertex...* □
- *Second, a maximum flow h is found in a modified network $\bar{G}' = (G, z, s, w - r)$ (Alg. 4.14); this network has only upper capacities $w(e) - r(e)$ for an edge e.* □
- *The resulting flow is the sum $f \equiv r + h$.*

Networks with vertex capacities

In a flow network with *vertex capacities* (retaining edge capacities as well), the capacity function is $w : E(G) \cup V(G) \to \boldsymbol{R}^+$.

The meaning of capacity constraints at the vertices for *admissible* flows is that the total sum of *incoming flow* to any vertex x is not more than $w(x)$.
(Differently applicable to the source or the sink, though. . .)□

Algorithm 4.18. ***Max-flow in a vertex-capacitated network***
Translate a vertex-capacitated network to an ordinary flow network via "doubling" the capacitated vertices (replacing them with new arcs between the two copies), as follows:



Then solve the ordinary network with Algorithm 4.14.

Back to Menger's theorem

Setting capacities of all vertices (except the terminals s, t) to 1, immediately proves:

Theorem 4.4. Let G be a graph and $s, t \in V(G)$. There exist k internally disjoint s–t paths in G if, and only if, there is no s–t vertex cut in G of size less than k. □

Systems of distinct representatives (SDR)

Definition: Let $M_1, M_2, \ldots, M_k$ be a collection of nonempty sets. A *system of distinct representatives (SDR)* of the set family $\{M_1, M_2, \ldots, M_k\}$ is a sequence of pairwise distinct elements $(x_1, x_2, \ldots, x_k)$ such that $x_i \in M_i$ for $i = 1, 2, \ldots, k$. □

Theorem 4.19. (Hall) *Let $\{M_1, M_2, \ldots, M_k\}$ be a family of nonempty sets. Then there exists a system of its distinct representatives if, and only if,*

$$\forall J \subset \{1, 2, \ldots, k\} : \left| \bigcup_{j \in J} M_j \right| \geq |J| \,,$$

i.e., the union of any subfamily of these sets has at least that many elements as the number of sets in it.

Necessity of Hall's condition in this theorem is obvious, and its sufficiency can be proved by an application of network flows again...

Flows in image segmentation

Yet another profitable application of flow networks lies in the computer vision area:

The basic *image segmentation* problem asks for decomposing a given image into a foreground and a background.

- Let the input consists of a pixel matrix, each pixel carrying two values of likelihood to be in the foreground and in the background. Additionally, separation penalties are assigned to the neighbouring pairs of pixels. □
- The network is constructed by introducing a source z and a sink s such that the arcs from z to the pixels have their capacities equal to the foreground likelihood, and the arcs from the pixels to s have their capacities equal to the background likelihood. Additional bidirectional arcs join the neighbouring pixel pairs.
- A minimal cut in this network then defines the separation between foreground and background parts.