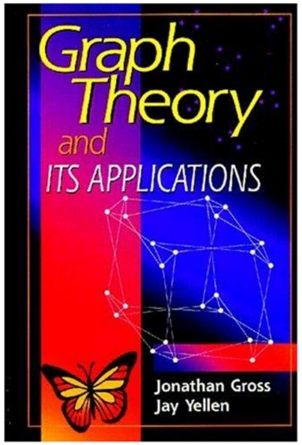
V11 Menger's theorem



Borrowing terminology from operations research consider certain *primal-dual pairs* of optimization problems that are intimately related.

Usually, one of these problems involves the maximization of some objective function, while the other is a minimization problem.

Separating set

A feasible solution to one of the problems provides a bound for the optimal value of the other problem (referred to as *weak duality*), and the optimal value of one problem is equal to the optimal value of the other (*strong duality*).

<u>Definition:</u> Let *u* and *v* be distinct vertices in a connected graph G. A vertex subset (or edge subset) *S* is *u*-*v* **separating** (or **separates** *u* and *v*), if the vertices *u* and *v* lie in different components of the deletion subgraph *G* – *S.*

$\rightarrow$ a *u*-*v* separating vertex set is a vertex-cut, and
a *u*-*v* separating edge set is an edge-cut.

When the context is clear, the term *u*-*v* **separating set** will refer either to a *u*-*v* separating vertex set or to a *u*-*v* separating edge set.

Example

For the graph *G* in the Figure below, the vertex-cut {*x,w,z*} is a *u-v* separating set of vertices of minimum size, and the edge-cut {*a,b,c,d,e*} is a *u-v* separating set of edges of minimum size.

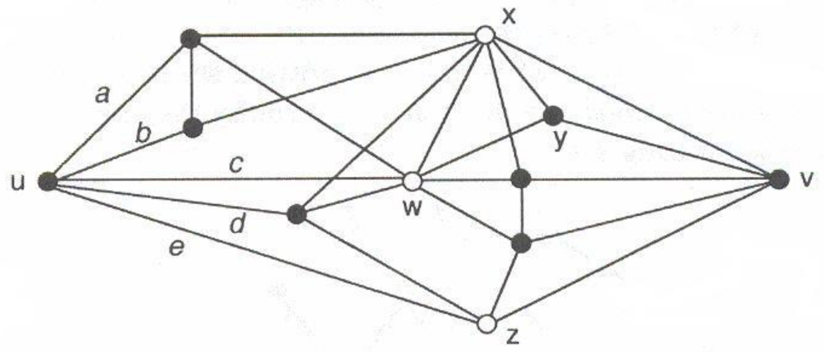


Figure 5.3.1 Vertex- and edge-cuts that are u-v separating sets.

Notice that a minimum-size *u-v* separating set of edges (vertices) need not be a minimum-size edge-cut (vertex-cut).

E.g., the set {*a,b,c,d,e*} is not a minimum-size edge-cut in *G,* because the set of edges incident on the 3-valent vertex *y* is an edge-cut of size 3.

A Primal-Dual Pair of Optimization Problems

The connectivity of a graph may be interpreted in two ways.
One interpretation is the number of vertices or edges it takes to disconnect the graph, and the other is the number of alternative paths joining any two given vertices of the graph.

Corresponding to these two perspectives are the following two optimization problems for two non-adjacent vertices *u* and *v* of a connected graph *G*.

Maximization Problem: Determine the maximum number of internally disjoint *u*-*v* paths in graph *G*.

Minimization Problem: Determine the minimum number of vertices of graph *G* needed to separate the vertices *u* and *v.*

A Primal-Dual Pair of Optimization Problems

Proposition 5.3.1: (*Weak Duality*) Let u and v be any two non-adjacent vertices of a connected graph G. Let $\mathcal{P}_{uv}$ be a collection of internally disjoint u-v paths in G, and let S_{uv} be a u-v separating set of vertices in G.

Then $|\mathcal{P}_{uv}| \leq |S_{uv}|$.

Proof: Since S_{uv} is a u-v separating set, each u-v path in $\mathcal{P}_{uv}$ must include at least one vertex of S_{uv}. Since the paths in $\mathcal{P}_{uv}$ are internally disjoint, no two paths of them can include the same vertex.

Thus, the number of internally disjoint u-v paths in G is at most $|S_{uv}|$. □

Corollary 5.3.2. Let u and v be any two non-adjacent vertices of a connected graph G. Then the maximum number of internally disjoint u-v paths in G is less than or equal to the minimum size of a u-v separating set of vertices in G.

Menger‘s theorem will show that the two quantities are in fact equal.

A Primal-Dual Pair of Optimization Problems

The following corollary follows directly from Proposition 5.3.1.

<u>Corollary 5.3.3</u>: (*Certificate of Optimality*) Let *u* and *v* be any two non-adjacent vertices of a connected graph *G.*

Suppose that $\mathcal{P}_{uv}$ is a collection of internally disjoint *u*-*v* paths in *G,*

and that S_{uv} is a *u*-*v* separating set of vertices in G, such that $|\mathcal{P}_{uv}| = |S_{uv}|$.

Then $\mathcal{P}_{uv}$ is a maximum-size collection of internally disjoint *u*-*v* paths, and S_{uv} is a minimum-size *u*-*v* separating set (i.e. *S* has the smallest size of all *u*-*v* separating sets).

Vertex- and Edge-Connectivity

<u>Example</u>: In the graph G below, the vertex sequences $\langle u,x,y,t,v \rangle$, $\langle u,z,v \rangle$, and $\langle u,r,s,v \rangle$ represent a collection $\mathcal{P}$ of three internally disjoint u-v paths in G, and the set $S = \{y,s,z\}$ is a u-v separating set of size 3.

Therefore, by Corollary 5.3.3, $\mathcal{P}$ is a maximum-size collection of internally disjoint u-v paths, and S is a minimum-size u-v separating set.

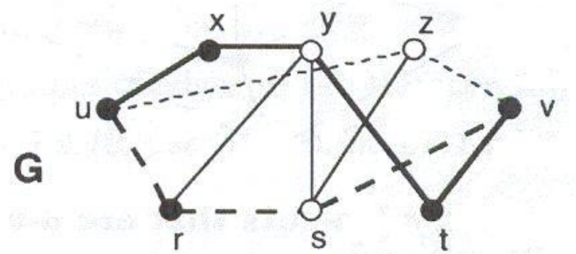


Figure 5.3.2

The next theorem proved by K. Menger in 1927 establishes a *strong duality* between the two optimization problems introduced earlier.

The proof given here is an example of a traditional style proof in graph theory. The theorem can also be proven e.g. based on the theory of network flows.

strict paths

Definition Let *W* be a set of vertices in a graph *G* and *x* another vertex not in *W*. A **strict *x*-*W* path** is a path joining *x* to a vertex in *W* and containing no other vertex of *W*. A **strict *W*-*x* path** is the reverse of a strict *x*-*W* path (i.e. its sequence of vertices and edges is in reverse order).

Example: Corresponding to the *u*-*v* separating set $W = \{y,s,z\}$ in the graph below, the vertex sequences $\langle u,x,y \rangle$, $\langle u,r,y \rangle$, $\langle u,r,s \rangle$, and $\langle u,z \rangle$ represent the four strict *u*-*W* paths, and the three strict *W*-*v* paths are given by $\langle z,v \rangle$, $\langle y,t,v \rangle$, and $\langle s,v \rangle$.

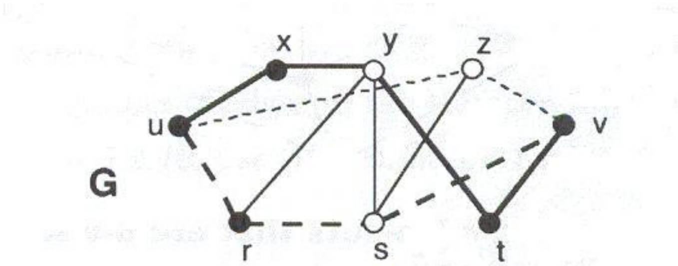
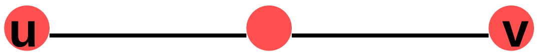


Figure 5.3.2

Menger's Theorem

Theorem 5.3.4 [Menger, 1927] Let *u* and *v* be distinct, non-adjacent vertices in a connected graph *G*.
Then the maximum number of internally disjoint *u*-*v* paths in *G* equals the minimum number of vertices needed to separate *u* and *v*.

Proof: The proof uses induction on the number of edges.
The smallest graph that satisfies the premises of the theorem is the path graph from *u* to *v* of length 2, and the theorem is trivially true for this graph.



Assume that the theorem is true for all connected graphs having fewer than *m* edges, e.g. for some $m \geq 3$.
Now suppose that *G* is a connected graph with *m* edges, and let *k* be the minimum number of vertices needed to separate the vertices *u* and *v*.
By Corollary 5.3.2, it suffices to show that there exist *k* internally disjoint *u*-*v* paths in *G*.
Since this is clearly true if $k = 1$ (since *G* is connected), assume $k \geq 2$.

Proof of Menger's Theorem

Assertion 5.3.4a If G contains a u-v path of length 2, then G contains k internally disjoint u-v paths.

Proof of 5.3.4a: Suppose that $\mathcal{P} = \langle u, e_1, x, e_2, v \rangle$ is a path in G of length 2.
Let W be a smallest u-v separating set for the vertex-deletion subgraph $G - x$.
Since $W \cup \{x\}$ is a u-v separating set for G, the minimality of k implies that $|W| \geq k - 1$. By the induction hypothesis, there are at least $k - 1$ internally disjoint $u - v$ paths in $G - x$. Path $\mathcal{P}$ is internally disjoint from any of these, and, hence, there are k internally disjoint u-v paths in G. □

If there is a u-v separating set that contains a vertex adjacent to *both* vertices u and v, then Assertion 5.3.4a guarantees the existence of k internally disjoint u-v paths in G.

The argument for *distance* $(u,v) \geq 3$ is now broken into two cases, according to the kinds of u-v separating sets that exist in G.

Proof of Menger's Theorem

In Case 1 (left picture), there exists a *u*-*v* separating set *W*, where neither *u* nor *v* is adjacent to every vertex of *W* .

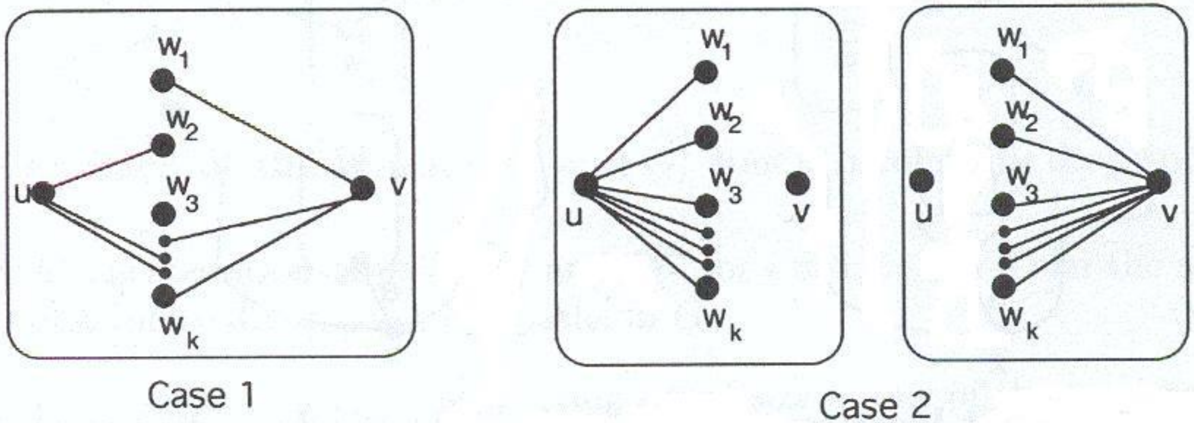


Figure 5.3.3 The two cases remaining in the proof of Menger's theorem.

In Case 2 (right picture), no such separating set exists.

Thus, in every *u*-*v* separating set for Case 2, either every vertex is adjacent to *u* or every vertex is adjacent to *v*.

Proof of Menger's Theorem

Case 1: There exists a u-v separating set $W = \{w_1, w_2, \ldots, w_k\}$ of vertices in G of minimum size k, such that neither u nor v is adjacent to every vertex in W.

Let G_u be the subgraph induced on the union of the edge-sets of all strict u-W paths in G, and let G_v be the subgraph induced on the union of edge-sets of all strict W-v paths (see Fig. below).

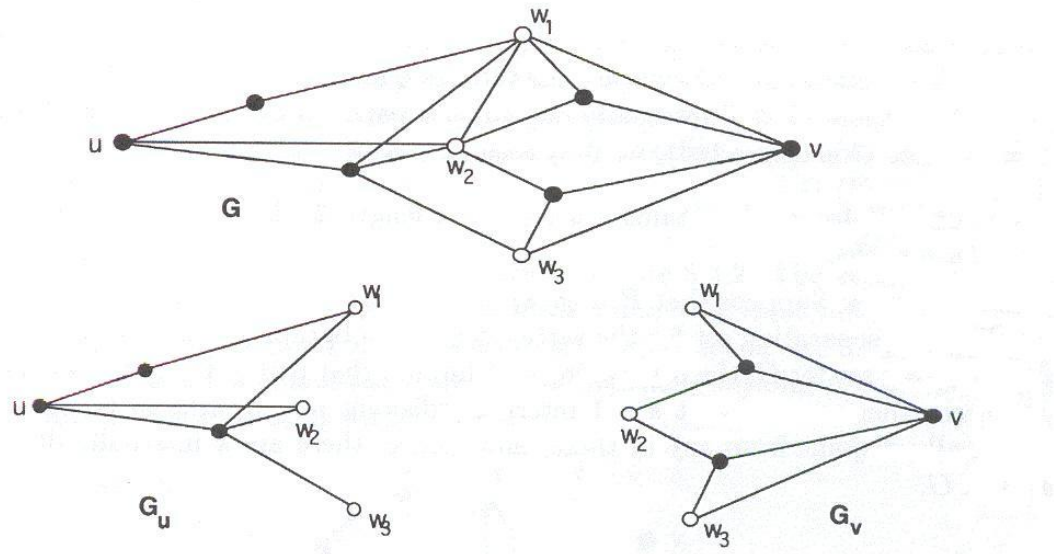


Figure 5.3.4 An example illustrating the subgraphs G_u and G_v.

Proof of Menger's Theorem

<u>Assertion 5.3.4b</u>: Both of the subgraphs G_u and G_v have more than k edges.

<u>Proof</u> of 5.3.4b: For each $w_i \in W$, there is a u-v path P_{wi} in G on which w_i is the only vertex of W (otherwise, $W - \{w_i\}$ would still be a u-v separating set, contradicting the minimality of W).

The u-w_i subpath of P_{wi} is a strict u-W path that ends at w_i.

Thus, the final edge of this strict u-W path is different for each w_i.

Hence, G_u has at least k edges.

The only way G_u could have exactly k edges would be if each of these strict u-W paths consisted of a single edge joining u and w_i, $i = 1, ..., k$.

But this is ruled out by the condition for Case 1. Therefore, G_u has more than k edges. A similar argument shows that G_v also has more than k edges. □

Proof of Menger's Theorem

<u>Assertion 5.3.4c</u>: The subgraphs G_u and G_v have no edges in common.

<u>Proof</u> of 5.3.4c: By way of contradiction, suppose that the subgraphs G_u and G_v have an edge *e* in common. By the definitions of G_u and G_v, edge *e* is an edge of both a strict *u*-*W* path and a strict *W*-*v* path.

A **strict *x-W* path** is a path joining *x* to a vertex in *W* and containing no other vertex of *W*.

A **strict *W-x* path** is the reverse of a strict *x*-*W* path (i.e. its sequence of vertices and edges is in reverse order).

Hence, at least one of the endpoints of *e*, say *x*, is not a vertex in the *u*-*v* separating set *W* (see Fig. below). This implies the existence of a *u*-*v* path in *G*-*W*, which contradicts the definition of *W*. □

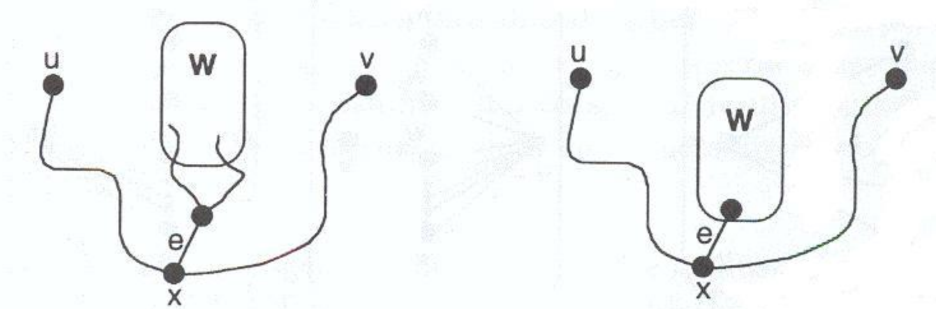


Figure 5.3.5 At least one of the endpoints of edge e lies outside W.

Proof of Menger's Theorem

We now define two auxiliary graphs G_u^* and G_v^*:

G_u^* is obtained from G by replacing the subgraph G_v with a new vertex v^* and drawing an edge from each vertex in W to v^*, and

G_v^* is obtained by replacing G_u with a new vertex u^* and drawing an edge from u^* to each vertex in W (see Fig. below).

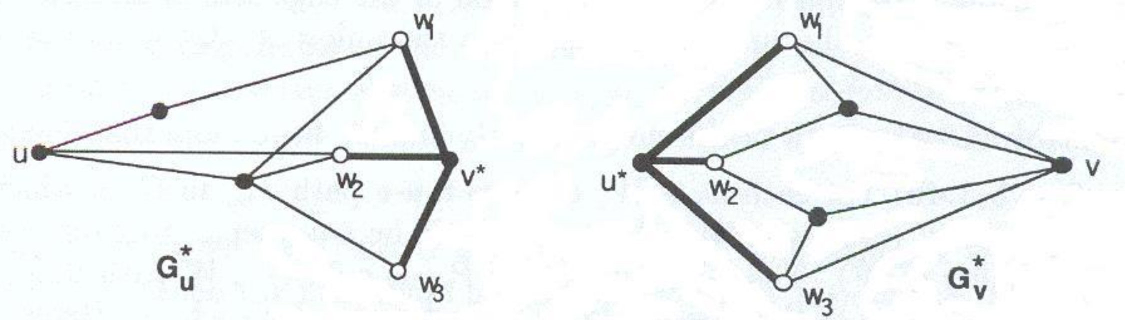


Figure 5.3.6 Illustration for the construction of graphs G_u^* and G_v^*.

Proof of Menger's Theorem

Assertion 5.3.4d: Both of the auxiliary graphs G_u^* and G_v^* have fewer edges than G.

Proof of 5.3.4d: The following chain of inequalities shows that graph G_u^* has fewer edges than G.

$$\begin{aligned} |E_G| &\geq |E_{G_u \cup G_v}| && \text{since } G_u \cup G_v \text{ is a subgraph of } G \\ &\underset{5.3.4c}{=} |E_{G_u}| + |E_{G_v}| \\ &\underset{5.3.4b}{>} |E_{G_u}| + k \\ &= |E_{G_u^*}| && \text{by the construction of } G_u^* \end{aligned}$$

A similar argument shows that G_v^* also has fewer edges than G. ☐

Proof of Menger's Theorem

By the construction of graphs G_u^* and G_v^*, every u-v^* separating set in graph G_u^* and every u^*-v separating set in graph G_v^* is a u-v separating set in graph G. Hence, the set W is a smallest u-v^* separating set in G_u^* and a smallest u^*-v separating set in G_v^*.

Since G_u^* and G_v^* have fewer edges than G, the induction hypothesis implies the existence of two collections, $\mathcal{P}_u^*$ and $\mathcal{P}_v^*$ of k internally disjoint u-v^* paths in G_u^* and k internally disjoint u^*-v paths in G_v^*, respectively (see Fig.).

For each w_i, one of the paths in $\mathcal{P}_u^*$ consists of a u-w_i path P_i' in G plus the new edge from w_i to v^*, and one of the paths in $\mathcal{P}_v^*$ consists of the new edge from u^* to w_i followed by a w_i-v path P_i'' in G.

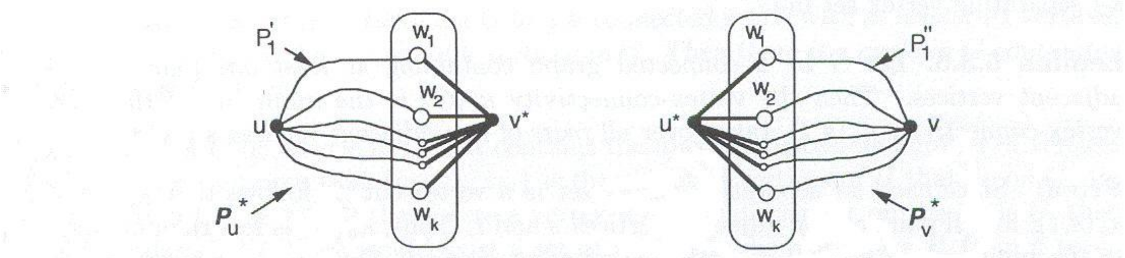


Figure 5.3.7 Each of the graphs G_u^* and G_v^* has k internally disjoint paths.

Let P_i be the concatenation of paths P_i' and P_i'', for $i = 1, ..., k$. Then the set $\{P_i\}$ is a collection of k internally disjoint u-v paths in G. ☐ (Case 1)

Proof of Menger's Theorem

Case 2: Suppose that for each *u*-*v* separating set of size *k*, one of the vertices *u* or *v* is adjacent to all the vertices in that separating set.

will not be proven in lecture = not be subject of test 3/final exam.

Let $P = \langle u,e_1,x_1,e_2,x_2, ..., v\rangle$ be a shortest *u*-*v* path in *G.*

By Assertion 5.3.4a, we can assume that *P* has length at least 3 and that vertex x_1 is not adjacent to vertex *v.*

By Proposition 5.1.3, the edge-deletion subgraph $G - e_2$ is connected.

Let *S* be a smallest *u*-*v* separating set in subgraph $G - e_2$ (see Fig.).

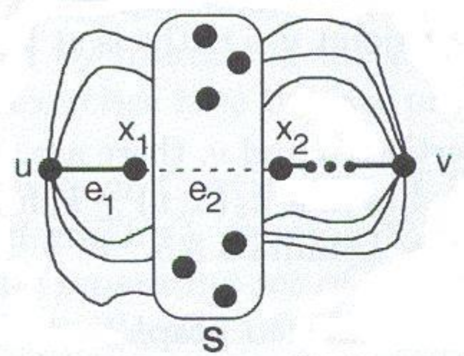


Figure 5.3.8 Completing Case 2 of Menger's theorem.

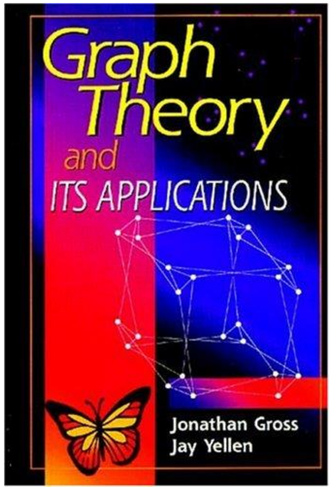
Proof of Menger's Theorem

Then S is a u-v separating set in the vertex-deletion subgraph $G - x_1$.
Thus, $S \cup \{x_1\}$ is a u-v separating set in G, which implies that $|S| \geq k - 1$, by the minimality of k. On the other hand, the minimality of
$|S|$ in $G - e_2$ implies that $|S| \leq k$, since every u-v separating set in G is also a u-v separating set in $G - e_2$.

If $|S| = k$, then, by the induction hypothesis, there are k internally disjoint u-v paths in $G - e_2$ and, hence, in G.
If $|S| = k - 1$, then $x_i \notin S$, $i = 1,2$ (otherwise $S - \{x_i\}$ would be a u-v separating set in $G - e_2$, contradicting the minimality of k).
Thus, the sets $S \cup \{x_1\}$ and $S \cup \{x_2\}$ are both of size k and both u-v separating sets of G. The condition for Case 2 and the fact that vertex x_1 is not adjacent to v imply that every vertex in S is adjacent to vertex u.
Hence, no vertex in S is adjacent to v (lest there be a u-v path of length 2).
But then the condition of Case applied to $S \cup \{x_2\}$ implies that vertex x_2 is adjacent to vertex u, which contradicts the minimality of path P and completes the proof. □

V11 – second half

This part follows closely chapter 12.1 in the book on the right
on „Flows and Cuts in Networks and
Chapter 12.2 on "Solving the Maximum-Flow Problem"



Flow in Networks can mean
- flow of oil or water in pipelines, electricity
- phone calls, emails, traffic networks ...

Equivalences between
max-flow min-cut theorem of Ford and Fulkerson
& the connectivity theorems of Menger
$\rightarrow$ led to the development of efficient algorithms for a number of practical problems to solve scheduling and assignment problems.

Single Source – Single Sink Capacitated Networks

Definition: A **single source – single sink network** is a connected digraph that has a distinguished vertex called the **source** with nonzero outdegree and a distinguished vertex called the **sink** with nonzero indegree.

Such a network with source *s* and sink *t* is often referred to as a ***s-t* network**.

Definition: A **capacitated network** is a connected digraph such that each arc *e* is assigned a nonnegative weight ***cap(e),*** called the **capacity** of arc *e.*

Notation: Let *v* be a vertex in a digraph *N*. Then ***Out(v)*** denotes the set of all arcs that are directed **from** vertex *v*. That is,

$$Out(v) = \{e \in E_N \mid tail(e) = v\}$$

Correspondingly, ***In(v)*** denotes the set of arcs that are directed **to** vertex *v:*

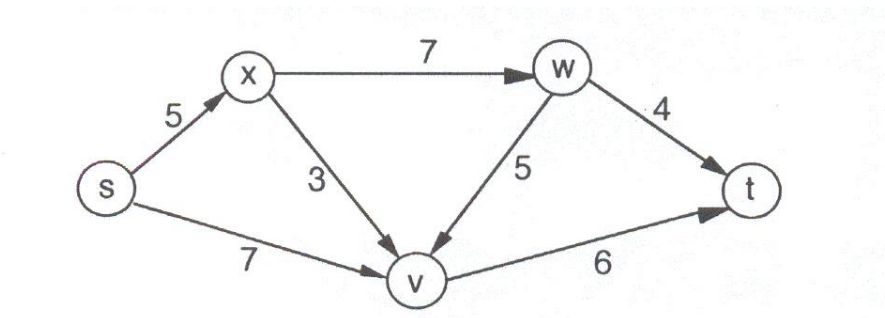
$$In(v) = \{e \in E_N \mid head(e) = v\}$$

Single Source – Single Sink Capacitated Networks

<u>Notation</u>: For any two vertex subsets *X* and *Y* of a digraph *N*, let $\langle \boldsymbol{X,Y} \rangle$ denote the set of arcs in *N* that are directed **from** a vertex in *X* **to** a vertex in *Y.*

$$\langle X,Y \rangle = \left\{ e \in E_N \middle| tail(e) \in X \text{ and } head(e) \in Y \right\}$$

<u>Example:</u> The figure shows a 5-vertex capacitated *s*-*t*-network.
If *X* = *{x,v}* and *Y* = *{w,t}*, then the elements of arc set $\langle X,Y \rangle$ are the arc directed from vertex *x* to vertex *w* and the arc directed from vertex *v* to sink *t.*



A 5-vertex capacitated network with source *s* and sink *t.*

The only element in arc set $\langle Y,X \rangle$ is the arc directed from vertex *w* to vertex *v.*

Feasible Flows

<u>Definition</u>: Let *N* be a capacitated *s*-*t*-network.

A **feasible flow** *f* in *N* is a function $f{:}E_N \to R^+$ that assigns a nonnegative real number

1. (**capacity constraints**) $f(e) \le cap(e)$, for every arc *e* in network *N.*
2. (**conservation constraints**) $$\sum_{e \in In(v)} f(e) = \sum_{e \in Out(v)} f(e)$$

for every vertex *v* in network *N*, other than source *s* and sink *t.*

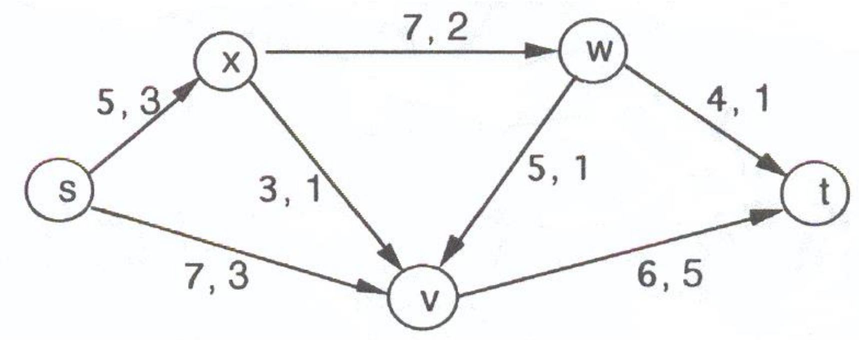
Property 2 above is called the **conservation-of-flow** condition.

E.g. for an oil pipeline, the total flow of oil going into any juncture (vertex) in the pipeline must equal the total flow leaving that juncture.

<u>Notation</u>: to distinguish visually between the flow and the capacity of an arc, we adopt the convention in drawings that when both numbers appear, the capacity will always be in bold and to the left of the flow.

Feasible Flows

Example: The figure shows a feasible flow for the previous network.
Notice that the total amount of flow leaving source *s* equals 6, which is also the net flow entering sink *t.*



Definition: The **value of flow** *f* in a capacitated network *N*, denoted with ***val(f)****,* is the net flow leaving the source *s*, that is

$$val(f) = \sum_{e \in Out(s)} f(e) - \sum_{e \in In(s)} f(e)$$

Definition: The **maximum flow *f**** in a capacitated network *N* is a flow in *N* having the maximum value, i.e. $val(f) \leq val(f^*)$, for every flow *f* in *N.*

Cuts in *s-t* Networks

By definition, any nonzero flow must use at least one of the arcs in *Out(s).*
In other words, if all of the arcs in *Out(s)* were deleted from network *N*,
then no flow could get from source *s* to sink *t.*

This is a special case of the following definition, which combines the concepts of **partition-cut** and **s-t separating set.**

From V11

Definition: Let G be a graph, and let X_1 and X_2 form a partition of V_G.
The set of all edges of G having one endpoint in X_1 and the other endpoint in X_2 is called a **partition-cut** of G and is denoted $\langle X_1,X_2\rangle$.

From V12

Definition: Let u and v be distinct vertices in a connected graph G.
A vertex subset (or edge subset) S is u-v **separating** (or **separates** u and v), if the vertices u and v lie in different components of the deletion subgraph $G – S$.

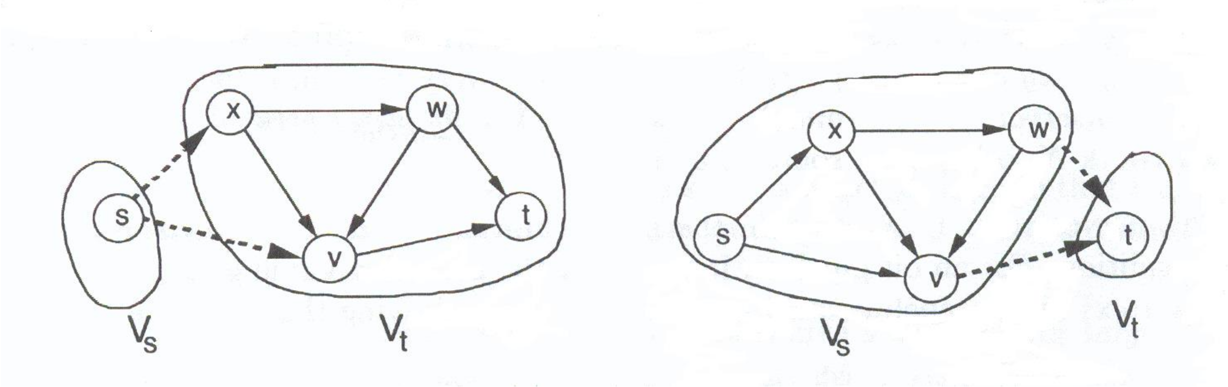
Cuts in *s-t* Networks

Definition: Let N be an s-t network, and let V_s and V_t form a partition of V_G such that source $s \in V_s$ and sink $t \in V_t$.
Then the set of all arcs that are directed from a vertex in set V_s to a vertex in set V_t is called an ***s-t* cut** of network N and is denoted $\langle V_s, V_t \rangle$.

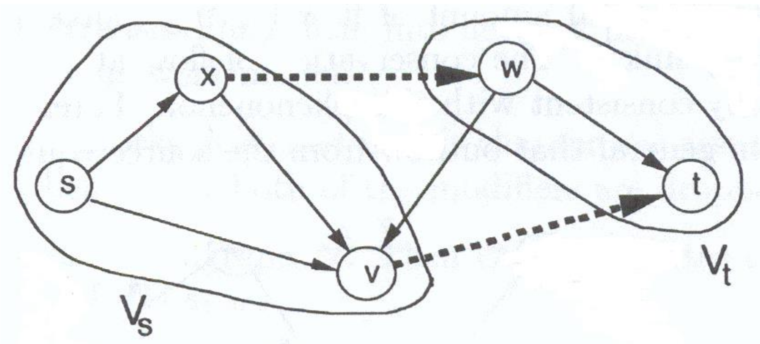
Remark: The arc set ***Out(s)*** for an s-t network N is the s-t cut $\langle \{s\}, V_N - \{s\} \rangle$, and ***In(t)*** is the s-t cut $\langle V_N - \{t\}, \{t\} \rangle$.

Cuts in *s-t* Networks

Example. The figure portrays the arc sets *Out(s)* and *In(t)* as *s-t* cuts, where *Out(s)* = ⟨ {*s*}, {*x,v,w,t*} ⟩ and *In(t)* = ⟨ {*s,x,v,w*},{*t*} ⟩.



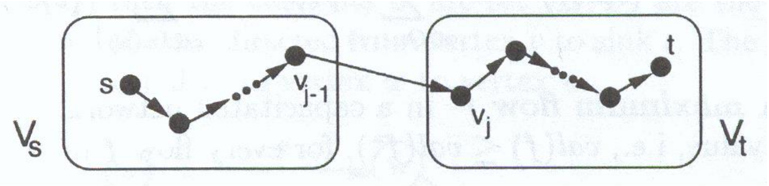
Example: a more general *s-t* cut ⟨ V_s,V_t ⟩ is shown below, where V_s = {*s,x,v*} and V_t = {*w,t*}.



Cuts in *s-t* Networks

<u>Proposition 12.1.1</u> Let $\langle V_s, V_t \rangle$ be an *s-t* cut of a network *N.*
Then every directed *s-t* path in *N* contains at least one arc in $\langle V_s, V_t \rangle$.

<u>Proof</u>. Let $P = \langle s = v_0, v_1, v_2, \ldots, v_l = t \rangle$ be the vertex sequence of a directed *s-t* path in network *N.*
Since $s \in V_s$ and $t \in V_t$, there must be a first vertex v_j on this path that is in set V_t (see figure below).
Then the arc from vertex v_{j-1} to v_j is in $\langle V_s, V_t \rangle$. □



Relationship between Flows and Cuts

Similar to viewing the set *Out(s)* of arcs directed from source *s* as the *s*-*t* cut $\langle\{s\}, V_N - \{s\}\rangle$, the set *In(s)* may be regarded as the set of „backward“ arcs relative to this cut, namely, the arc set $\langle V_N - \{s\}, \{s\}, \rangle$.

From this perspective, the definition of *val(f)* may be rewritten as

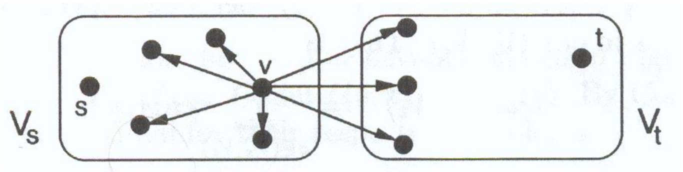
$$val(f) = \sum_{e \in \langle \{s\}, V_N - \{s\} \rangle} f(e) - \sum_{e \in \langle V_N - \{s\}, \{s\} \rangle} f(e)$$

Relationship between Flows and Cuts

Lemma 12.1.2. Let $\langle V_s, V_t \rangle$ be any *s-t* cut of an *s-t* network *N.* Then

$$\bigcup_{v \in V_s} Out(v) = \langle V_s, V_s \rangle \cup \langle V_s, V_t \rangle \quad \text{and} \quad \bigcup_{v \in V_s} In(v) = \langle V_s, V_s \rangle \cup \langle V_t, V_s \rangle$$

Proof: For any vertex $v \in V_s$, each arc directed from *v* is either in $\langle V_s, V_s \rangle$ or in $\langle V_s, V_t \rangle$. The figure illustrates for a vertex *v* the partition of *Out(v)* into a 4-element subset of $\langle V_s, V_s \rangle$ and a 3-element subset of $\langle V_s, V_t \rangle$.
Similarly, each arc directed to vertex *v* is either in $\langle V_s, V_s \rangle$ or in $\langle V_t, V_s \rangle$. □



$$\bigcup_{v \in V_s} Out(v) = \langle V_s, V_s \rangle \cup \langle V_s, V_t \rangle$$

Relationship between Flows and Cuts

Proposition 12.1.3. Let f be a flow in an s-t network N, and let $\langle V_s, V_t \rangle$ be any s-t cut of N. Then $$val(f) = \sum_{e \in \langle V_s, V_t \rangle} f(e) - \sum_{e \in \langle V_t, V_s \rangle} f(e)$$

Proof: By definition,

$$val(f) = \sum_{e \in Out(s)} f(e) - \sum_{e \in In(s)} f(e)$$

And by the conservation of flow

$$\sum_{e \in Out(v)} f(e) - \sum_{e \in In(v)} f(e) = 0 \quad \text{for every } v \in V_s \text{ other than } s. \text{ Thus}$$

$$val(f) = \sum_{v \in V_s} \left(\sum_{e \in Out(v)} f(e) - \sum_{e \in In(v)} f(e) \right) = \sum_{v \in V_s} \sum_{e \in Out(v)} f(e) - \sum_{v \in V_s} \sum_{e \in In(v)} f(e) \qquad (1)$$

By Lemma 12.1.2.

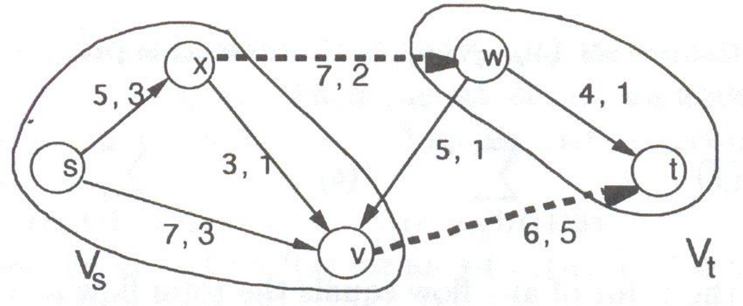
$$\sum_{v \in V_s} \sum_{e \in Out(v)} f(e) = \sum_{e \in \langle V_s, V_s \rangle} f(e) + \sum_{e \in \langle V_s, V_t \rangle} f(e) \quad \text{and}$$

$$\sum_{v \in V_s} \sum_{e \in In(v)} f(e) = \sum_{e \in \langle V_s, V_s \rangle} f(e) + \sum_{e \in \langle V_t, V_s \rangle} f(e) \qquad (2)$$

Now enter the right hand sides of (2) into (1) and obtain the desired equality. □

Example

The flow *f* and cut ⟨{*s,x,v*},{*w,t*}⟩ shown in the figure illustrate Proposition 12.1.3.



$$6 = val(f) = \sum_{e \in \langle \{s,x,v\},\{w,t\} \rangle} f(e) - \sum_{e \in \langle \{w,t\},\{s,x,v\} \rangle} f(e) = 7 - 1$$

The next corollary confirms something that was apparent from intuition: the net flow out of the source *s* equals the net flow into the sink *t.*

<u>Corollary 12.1.4</u> Let *f* be a flow in an *s-t* network. Then

$$val(f) = \sum_{e \in In(t)} f(e) - \sum_{e \in Out(t)} f(e)$$

<u>Proof</u>: Apply proposition 12.1.3 to the *s-t* cut *In(t)* = ⟨ V_N – {*t*}, {*t*} ⟩. □

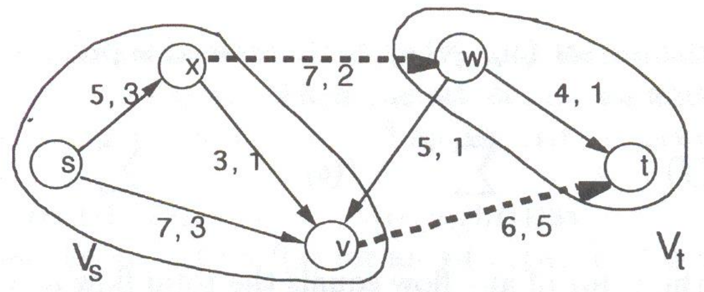
Example

Definition. The **capacity of a cut** $\langle V_s, V_t\rangle$ denoted **cap**$\langle V_s, V_t\rangle$, is the sum of the capacities of the arcs in cut $\langle V_s, V_t\rangle$. That is

$$cap\left\langle V_s, V_t \right\rangle = \sum_{e \in \langle V_s, V_t \rangle} cap(e)$$

Definition. The **minimum cut** of a network N is a cut with the minimum capacity.

Example. The capacity of the cut shown in the previous figure is 13,
And the cut $\langle$\{s,x,v,w\},\{t\}$\rangle$ with capacity 10, is the only minimum cut.



Maximum-Flow and Minimum-Cut Problems

The problems of finding the maximum flow in a capacitated network N and finding a minimum cut in N are closely related.

These two optimization problems form a *max-min* pair.

The following proposition provides an upper bound for the maximum-flow problem.

Maximum-Flow and Minimum-Cut Problems

Proposition 12.1.5 Let f be any flow in an s-t network, and let $\langle V_s, V_t \rangle$ be any s-t cut.

Then $$val(f) \leq cap\langle V_s, V_t \rangle$$

Proof:

$$\begin{aligned} val(f) &= \sum_{e \in \langle V_s, V_t \rangle} f(e) - \sum_{e \in \langle V_t, V_s \rangle} f(e) && \text{(by proposition 12.1.3)} \\ &\leq \sum_{e \in \langle V_s, V_t \rangle} cap(e) - \sum_{e \in \langle V_t, V_s \rangle} f(e) && \text{(by capacity constraints)} \\ &= cap\langle V_s, V_t \rangle - \sum_{e \in \langle V_t, V_s \rangle} f(e) && \text{(by definition of } cap\langle V_s, \mathrm{V}_t \rangle\text{)} \\ &\leq cap\langle V_s, V_t \rangle && \text{(since each } f(e) \text{ is nonnegative)} \quad \square \end{aligned}$$

Maximum-Flow and Minimum-Cut Problems

<u>Corollary 12.1.6</u> (Weak Duality) Let *f** be a maximum flow in an *s*-*t* network *N*, and let *K** be a minimum *s*-*t* cut in *N.* Then

$$val(f^*) \le cap(K^*)$$

<u>Proof:</u> This follows immediately from proposition 12.1.5.

<u>Corollary 12.1.7</u> (Certificate of Optimality) Let *f* be a flow in an *s*-*t* network *N* and *K* an *s*-*t* cut, and suppose that *val(f) = cap(K).*
Then flow *f* is a maximum flow in network *N*, and cut *K* is a minimum cut.

<u>Proof</u>: Let *f'* be any feasible flow in network *N.*
Proposition 12.1.5 and the premise give

$$val(f') \le cap(K) = val(f)$$

On the other hand, let $\langle V_s, V_t \rangle$ be any *s*-*t* cut. Proposition 12.1.5:

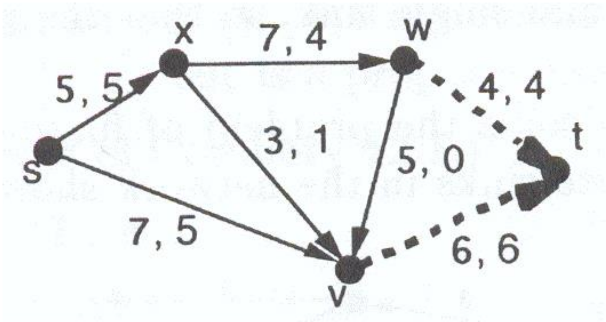
$$cap(K) = val(f) \le cap\langle V_s, V_t \rangle$$

Therefore, *K* is a minimum cut. □

Example

<u>Example</u> The flow for the example network shown in the figure has value 10, which is also the capacity of the *s*-*t* cut $\langle \{s,x,v,w\},\{t\} \rangle$.

By corollary 12.1.7, both the flow and the cut are optimal for their respective problem.



A maximum flow and minimum cut.

<u>Corollary</u> 12.1.8 Let $\langle V_s, V_t \rangle$ be an *s*-*t* cut in a network *N*, and suppose that *f* is a flow such that

$$f(e) = \begin{cases} cap(e) & \text{if } e \in \langle V_s, V_t \rangle \\ 0 & \text{if } e \in \langle V_t, V_s \rangle \end{cases}$$

Then *f* is a maximum flow in *N*, and $\langle V_s, V_t \rangle$ is a minimum cut.