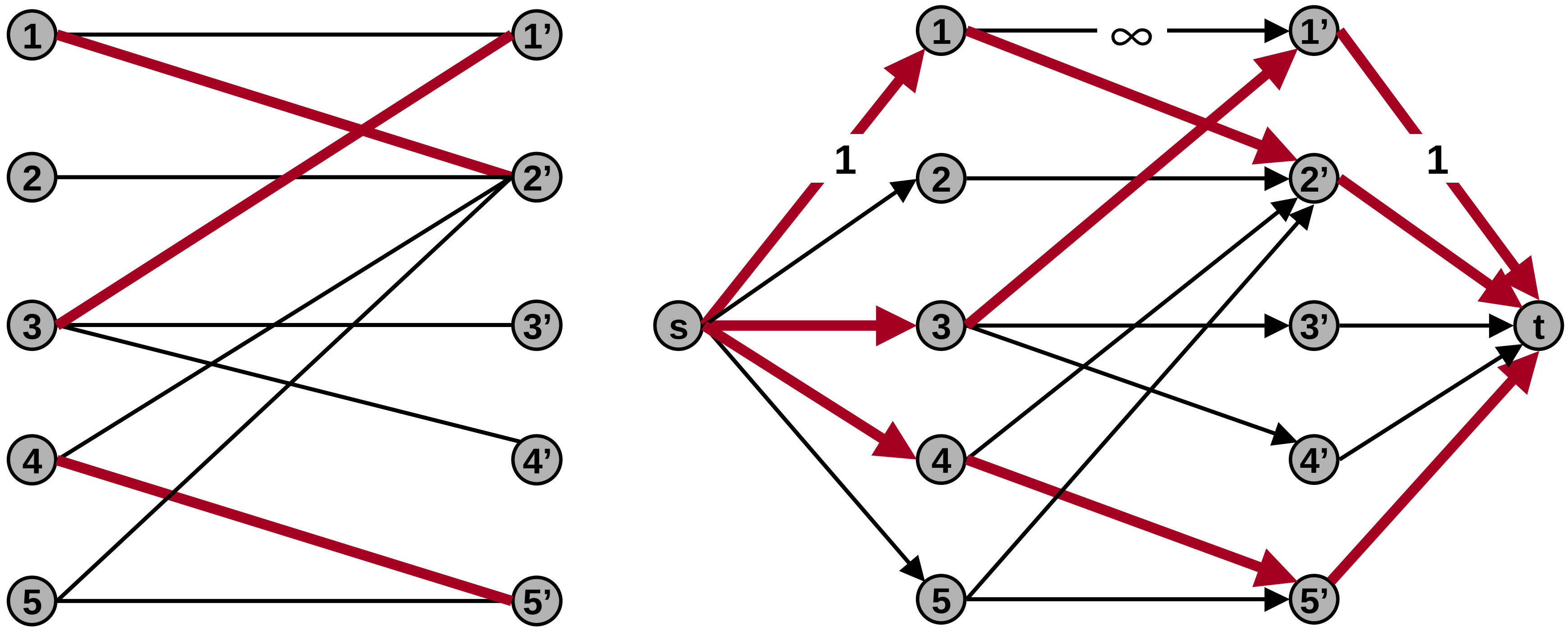
Bipartite Matching: Proof of Correctness

Claim. **Matching in G of cardinality k induces flow in G' of value k.**

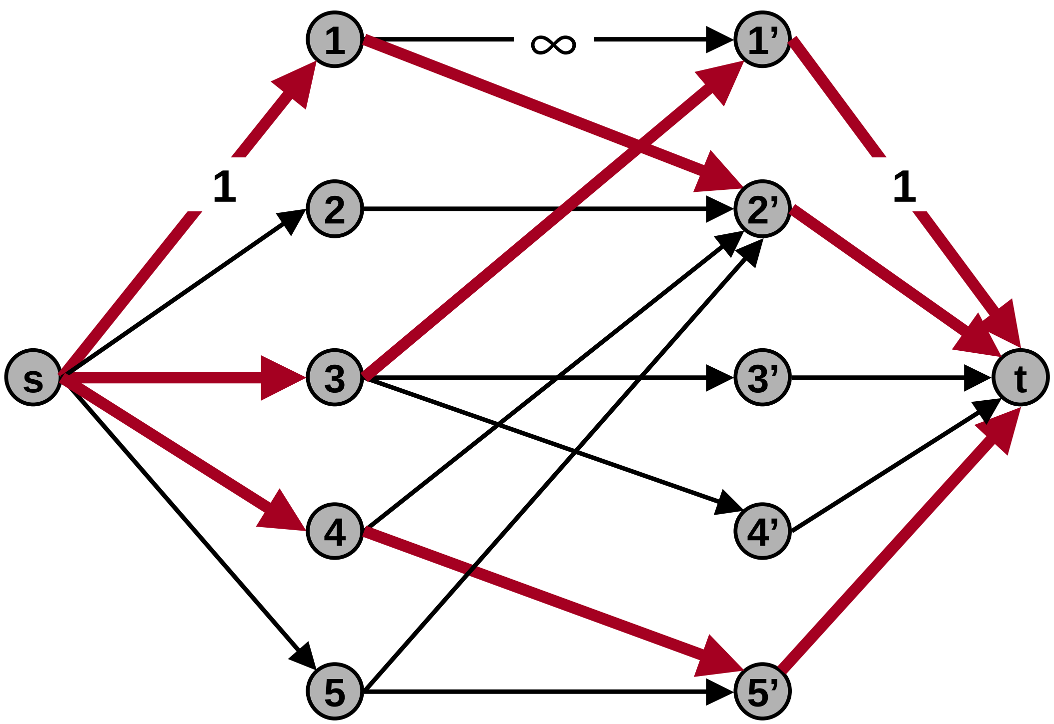
- Given matching M = { 1 - 2', 3 - 1', 4 - 5' } of cardinality 3.
- Consider flow that sends 1 unit along each of 3 paths:
  s - 1 - 2' - t,   s - 3 - 1' - t,   s - 4 - 5' - t.
- f is a flow, and has cardinality 3.

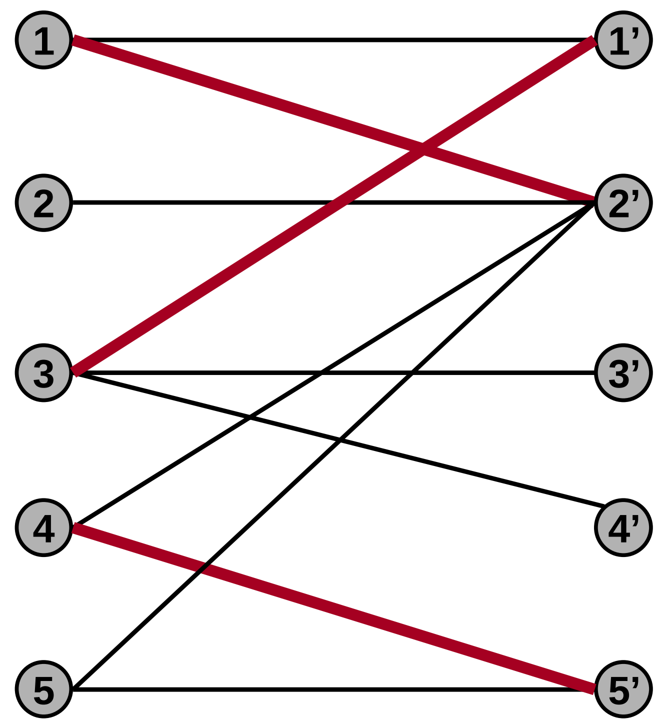


Bipartite Matching: Proof of Correctness

Claim. Flow f of value k in G' induces matching of cardinality k in G.

- By integrality theorem, there exists {0, 1}-valued flow f of value k.
- Consider M = set of edges from L to R with f(e) = 1.
  - each node in L and R participates in at most one edge in M
  - |M| = k: consider cut ($L \cup s$, $R \cup t$)





Matrix Rounding

Feasible matrix rounding.

- Given a p x q matrix $D = \{d_{i\,j}\}$ of real numbers.
- Row i sum = a_i, column j sum b_j.
- Round each d_{ij}, a_i, b_j up or down to integer so that sum of rounded elements in each row (column) equal row (column) sum.
- Original application: publishing US Census data.

Theorem: for any matrix, there exists a feasible rounding.

3.14	6.8	7.3	17.24
9.6	2.4	0.7	12.7
3.6	1.2	6.5	11.3
16.34	10.4	14.5	

Original Data

3	7	7	17
10	2	1	13
3	1	7	11
16	10	15	

Possible Rounding

Matrix Rounding

Feasible matrix rounding.

- Given a p x q matrix $D = \{d_{i\,j}\}$ of real numbers.
- Row i sum = a_i, column j sum b_j.
- Round each d_{ij}, a_i, b_j up or down to integer so that sum of rounded elements in each row (column) equal row (column) sum.
- Original application: publishing US Census data.

Theorem: for any matrix, there exists a feasible rounding.

- Note: "threshold rounding" doesn't work.

0.35	0.35	0.35	1.05
0.55	0.55	0.55	1.65
0.9	0.9	0.9	

Original Data

0	0	1	1
1	1	0	2
1	1	1	

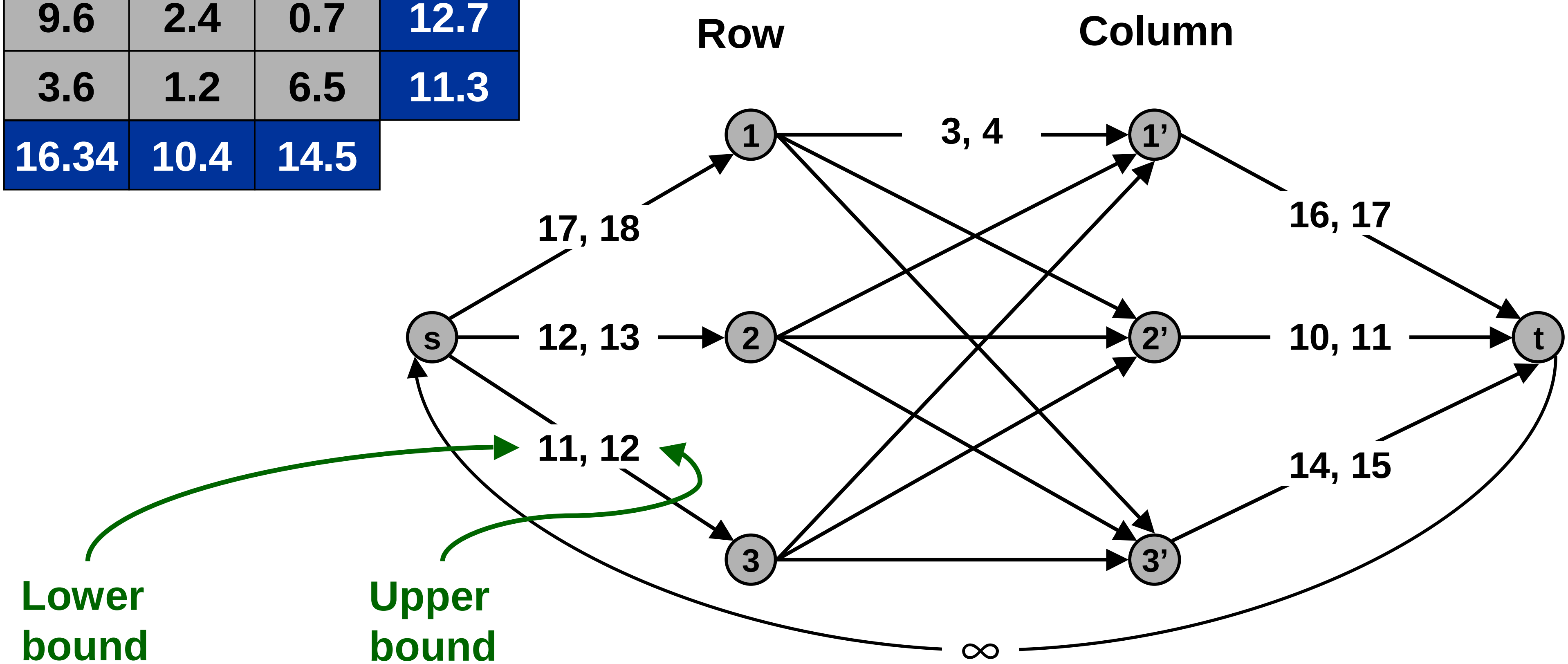
Possible Rounding

Matrix Rounding

Max flow formulation.

- Original data provides circulation (all demands 0).
- Integrality theorem $\Rightarrow$ there always exists feasible rounding!

3.14	6.8	7.3	17.24
9.6	2.4	0.7	12.7
3.6	1.2	6.5	11.3
16.34	10.4	14.5	



Project Selection

Projects with prerequisites.

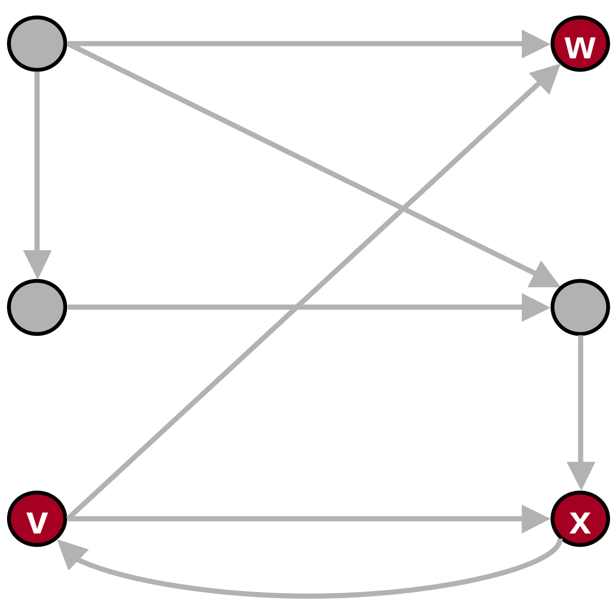
- **Set P of possible projects. Project v has associated revenue p_v.**
  - **some projects generate money: create interactive e-commerce interface, redesign cs web page**
  - **others cost money: upgrade computers, get site license for encryption software**
- **Set of prerequisites E. If (v, w) $\in$ E, can't do project v and unless also do project w.**
  - **can't start on e-commerce opportunity unless you've got encryption software**
- **A subset of projects A $\subseteq$ P is feasible if the prerequisite of every project in A also belongs to A.**
  - **for each v $\in$ P, and (v, w) $\in$ E, we have w $\in$ P**

Project selection (max weight closure) problem: choose a feasible subset of projects to maximize revenue.

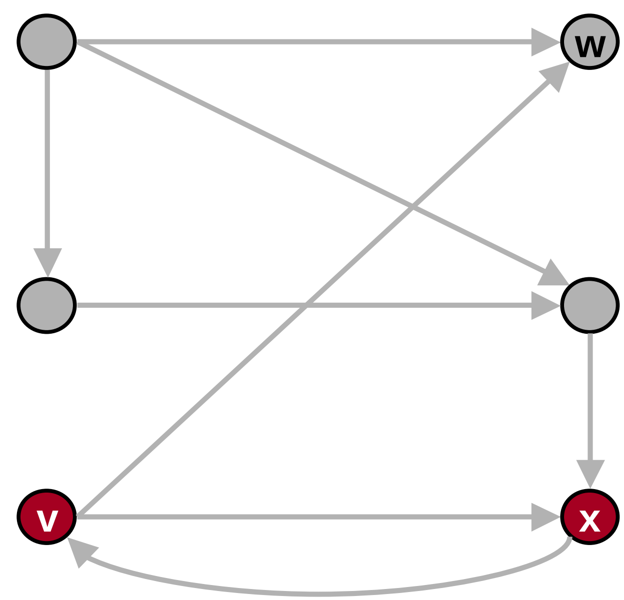
Project Selection

Prerequisite graph.

- Include an arc from v to w if can't do v without also doing w.
- {v, w, x} is feasible subset of projects.
- {v, x} is infeasible subset of projects.



Feasible

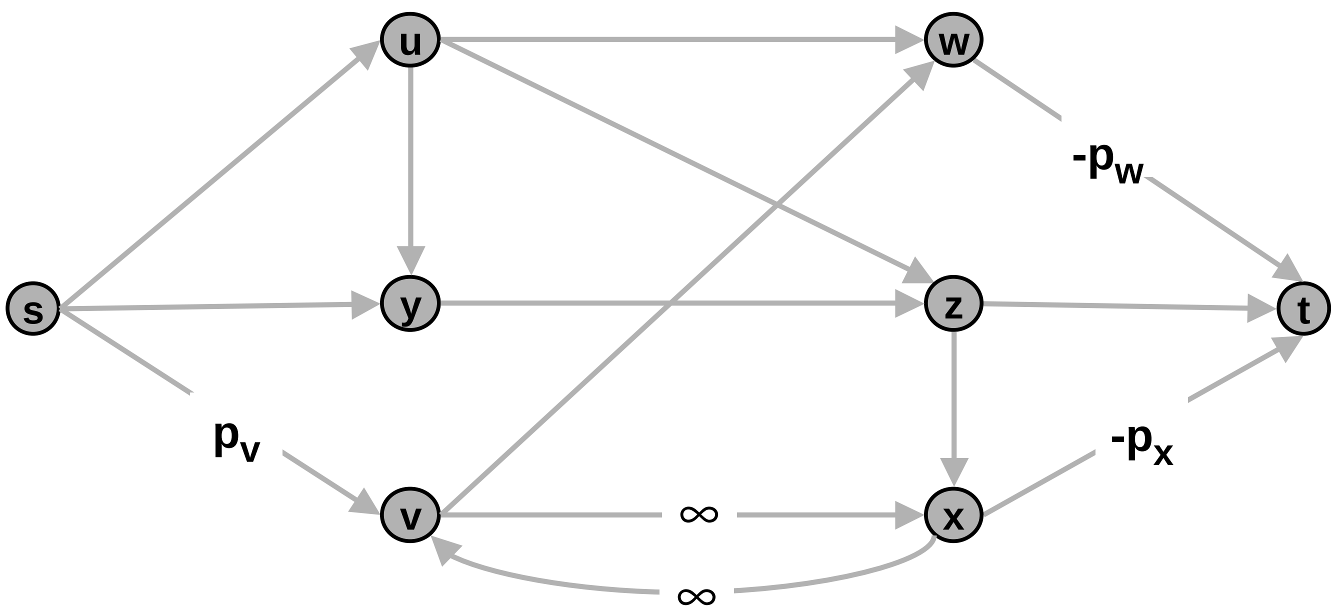


Infeasible

Project Selection

Project selection formulation.

- Assign infinite capacity to all prerequisite arcs.
- Add arc (s, v) with capacity p_v if $p_v > 0$.
- Add arc (v, t) with capacity $-p_v$ if $p_v < 0$.
- For notational convenience, define $p_s = p_t = 0$.



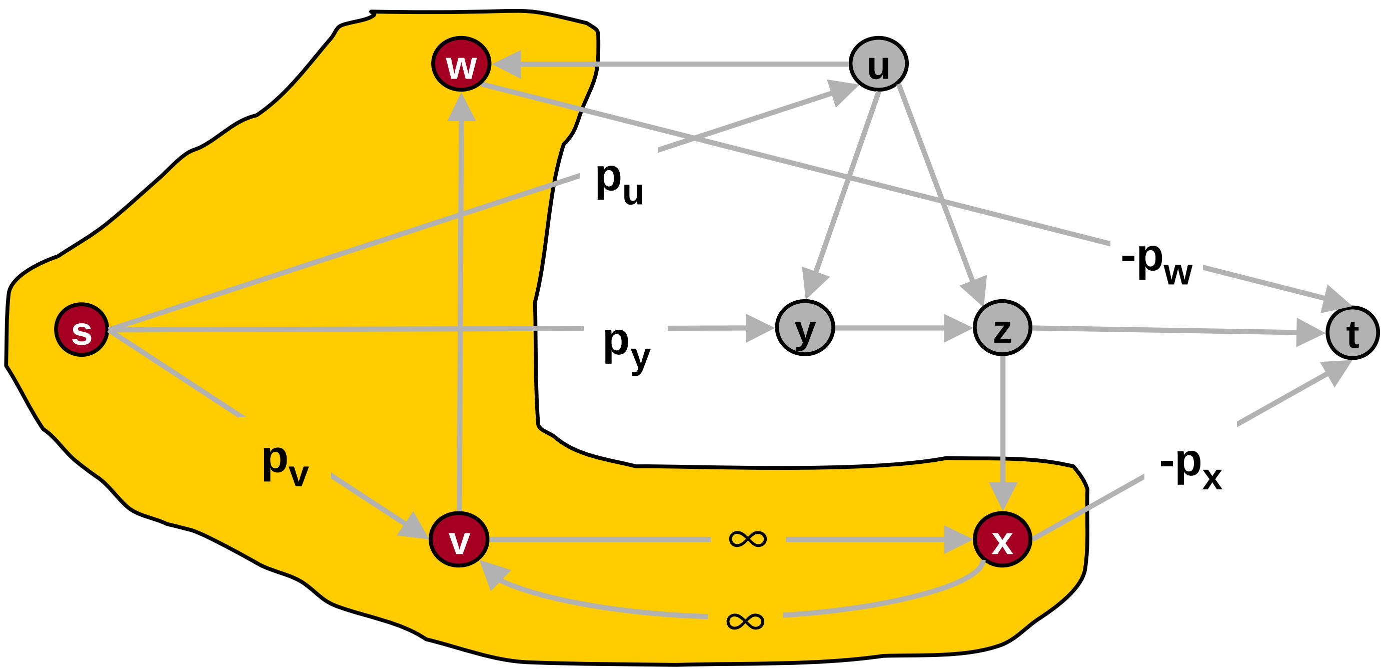
Project Selection

Claim. **(S, T) is min cut if and only if S \ {s} is optimal set of projects.**

- **Infinite capacity arcs ensure S \ {s} is feasible.**
- **Max revenue because:**

$$cap(S, T) = \sum_{v \in T:\, p_v > 0} p_v + \sum_{v \in S:\, p_v < 0} (-p_v)$$

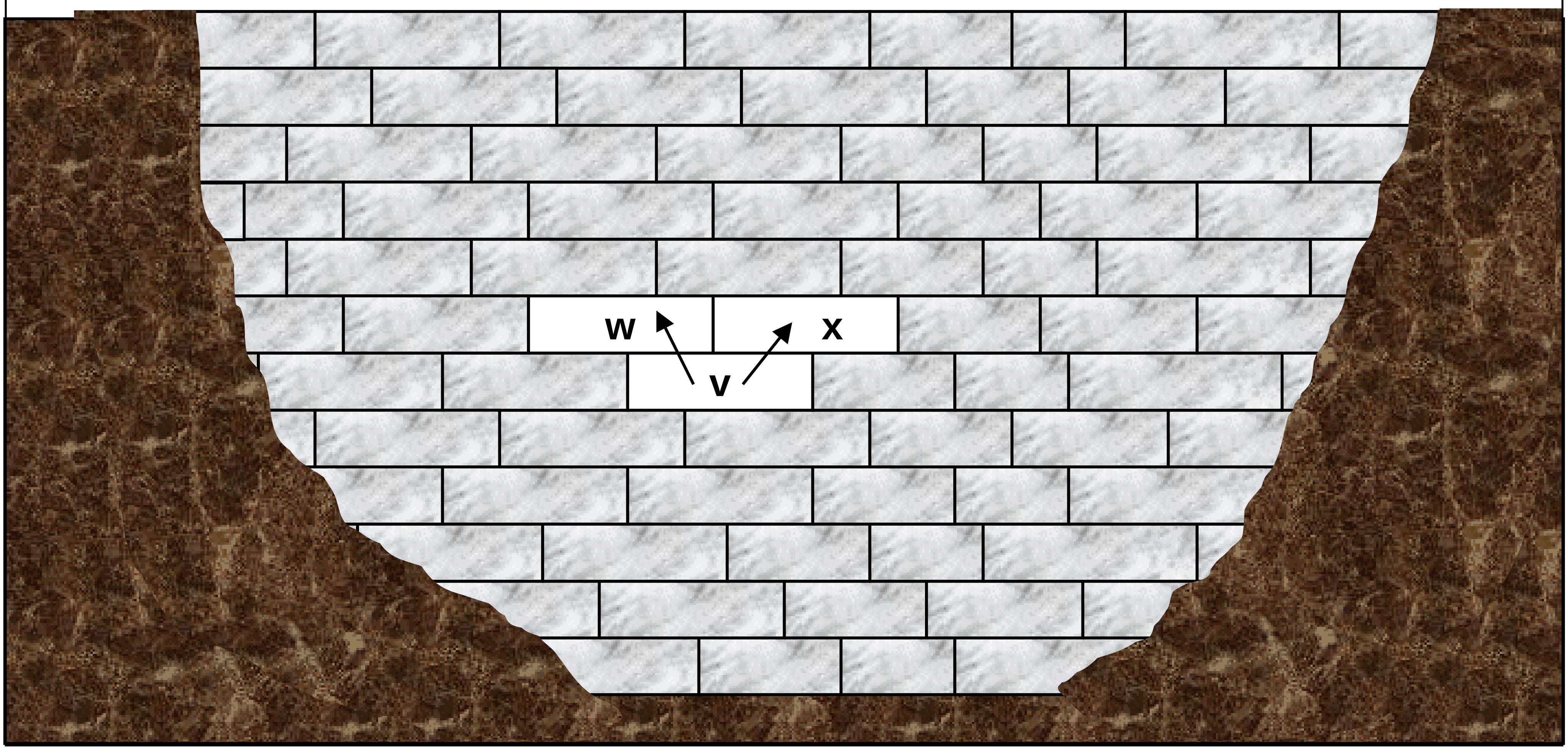
$$= \underbrace{\sum_{v:\, p_v > 0} p_v}_{\text{constant}} - \sum_{v \in S} p_v$$



Project Selection

Open-pit mining (studied since early 1960's).

- Blocks of earth are extracted from surface to retrieve ore.
- Each block v has net value p_v = value of ore - processing cost.
- Can't remove block v before w or x.



An Outlook: Min-Max Theorems in Combinatorial Optimization

There are a number of theorems in combinatorial optimization which state that the maximum of one quantity is always equal to the minimum of another one, so they are in a dual-type relationship. A typical, and best known, example is the Max Flow Min Cut Theorem.

These results are important, for several reasons:

- They can provide a certificate of optimality, just like we have seen with LP duality.
- They usually significantly contribute to the deeper understanding of the problem structure.
- In most cases, such theorems signal the possibility of an efficient (polynomial time) algorithm for the problem.

As an outlook, we summarize below some interesting samples of min-max theorems.

1. **Max Flow Min Cut Theorem.**
2. Theorems about the **maximum number of disjoint paths,** edge- or node-disjoint, between a source and a target node (see in the slide set "Application of Maximum Flows to

Solve Other Optimization Problems”). These are variants of what is usually referred to as **Menger’s Theorem.**

3. **Kőnig-Egerváry Theorem:** In a bipartite graph, the maximum possible number of edges in a matching is equal to the number of nodes in a minimum vertex cover. (Vertex cover: a set of nodes that cover all edges, that is, it contains at least one endpoint from every edge.)

4. **Tutte-Berge Formula:** In an arbitrary graph, the maximum possible number of edges in a matching is equal to the following minimum

$$\min_{U \subseteq V} \frac{1}{2}(|U| - \mathrm{odd}(V-U) + |V|).$$

   Here V is the entire vertex set of the graph, U is an arbitrary subset of it (which is chosen to minimize the formula), and $\mathrm{odd}(H)$ means the number of connected components in the set H that have an odd number of vertices. In the above formula we apply it to $H = V - U$.

5. **Tutte and Nash-Williams Tree Packing Theorem.** In a graph with vertex set V, let $\mathcal{P}$ denote a partition of the vertices. The *size* of the partition is denoted by $|\mathcal{P}|$, and it means the number of partition classes. Then the maximum possible number of edge-disjoint spanning trees in the graph is equal to the minimum number k, such that

every partition of V has at least $k(|\mathcal{P}| - 1)$ edges that go between different partition classes.

6. **Nash-Williams Theorem on Arboricity.** The arboricity of an undirected graph is the minimum number of edge-disjoint forests that together cover all edges of the graph. (A forest is a graph with no circuit. If it is also connected then it is a tree.) This minimum is equal to the following maximum:
$$\max \left\lceil \frac{m(S)}{n(S) - 1} \right\rceil$$
where S runs over all subgraphs of the original graph, $m(S)$ and $n(S)$ mean the number of edges and nodes, respectively, in the subgraph S.

7. **Lucchesi-Younger Theorem.** Consider a directed graph. If A is a subset of nodes that only has incoming edges, then this set of edges is called a *directed cut.* Then the minimum number of edges covering all directed cuts is equal to the maximum number of disjoint directed cuts.

8. **Dilworth's theorem.** An *antichain* in a partially ordered set is a set of elements no two of which are comparable to each other, and a *chain* is a set of elements every two of which are comparable. Then, for any partially ordered set, the maximum size of an antichain is equal to the minimum number of disjoint chains that together cover all elements.