Finite Domain Cuts for Minimum Bandwidth

J. N. Hooker
Carnegie Mellon University, USA

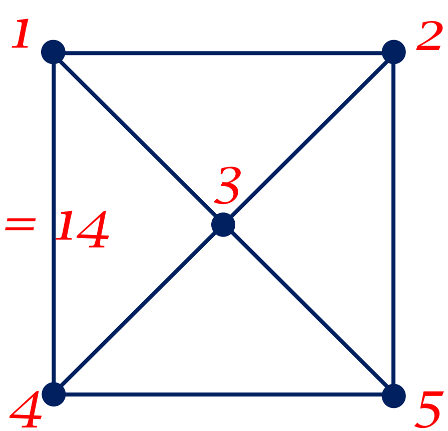
Joint work with
Alexandre Friere, Cid de Souza
State University of Campinas, Brazil

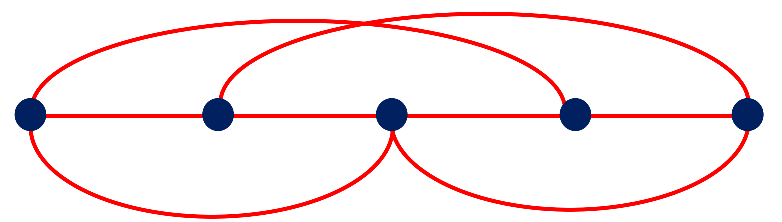
INFORMS 2013

Two Related Problems

- They differ only in the objective function.
- Minimum Linear Arrangement problem:
  - Label vertices of a graph to minimize **sum** of edge lengths.
  - Application: minimize wiring on a linear circuit board

Min edge sum = 14

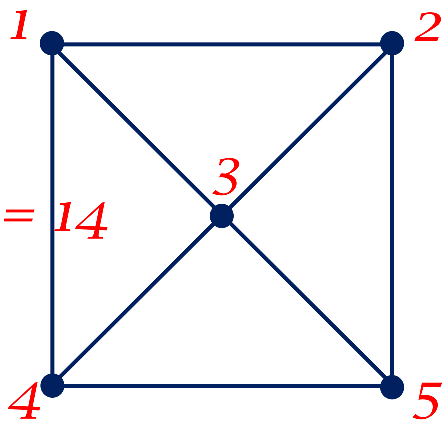




Two Related Problems

- They differ only in the objective function.

- Minimum Linear Arrangement problem:
  - Label vertices of a graph to minimize **sum** of edge lengths.
  - Application: minimize wiring on a linear circuit board



Min edge sum = 14

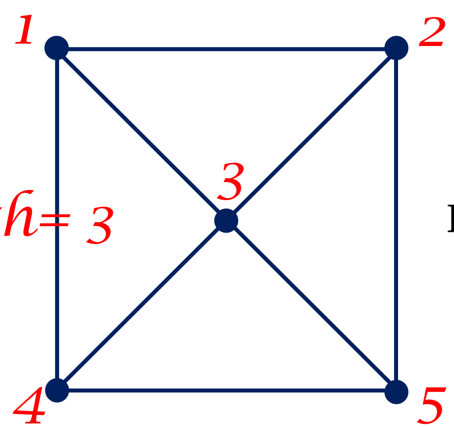
$$\min \left\{ \sum_{(i,j)\in E} u_{ij} \;\middle|\; \begin{array}{c} u_{ij} = \left|\pi_i - \pi_j\right| \\ \pi \text{ is perm. of } 1,\ldots,n \end{array} \right\}$$

Two Related Problems

- They differ only in the objective function.

- Minimum Bandwidth problem:
  - Label vertices of a graph to minimize **maximum** edge length.
  - Application: sparse matrix calculation
  - Application: ordering variables to minimize backtracking

Min bandwidth = 3



$$\min \left\{ \max_{(i,j)\in E} \{u_{ij}\} \;\middle|\; \begin{array}{c} u_{ij} = |\pi_i - \pi_j| \\ \pi \text{ is perm. of } 1,\ldots,n \end{array} \right\}$$

Finite-Domain Cuts

- Difficult to find good bounds, polyhedral or otherwise.
  - 0-1 models have weak relaxations.

- Idea: Use finite domain cuts
  - Formulate problem with **finite domain** variables rather than 0-1 variables.
  - Find valid inequalities.
  - This was useful for **graph coloring.**
    - Bergman and Hooker, CPAIOR 2012.
    - Let x_i = color assigned to vertex *i.*
    - Obtained tighter bounds in less time.

Finite-Domain Cuts

- For linear arrangement and bandwidth, use edge length variables u_{ij} only.
  - Unclear how to give problem a complete inequality formulation.
  - But we can derive valid cuts.

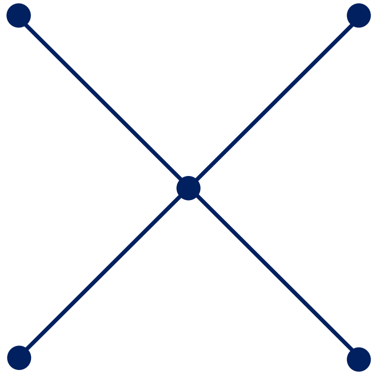
$$\min\left\{ \sum_{(i,j)\in E} u_{ij} \;\middle|\; \text{valid cuts in } u_{ij} \right\}$$

$$\min\left\{ w = \max_{(i,j)\in E}\left\{u_{ij}\right\} \;\middle|\; \text{valid cuts in } w, u_{ij} \right\}$$

Min Linear Arrangement

- This idea already applied to MLA problem.
  - Substantial improvement over existing bounds.
    - Caprara, Letchford, Salazar-González, *IJOC* 2011.
  - Polytope hard to analyze.
    - CLS studied **dominant** of polytope instead.
    - Facets found for **stars**, etc.

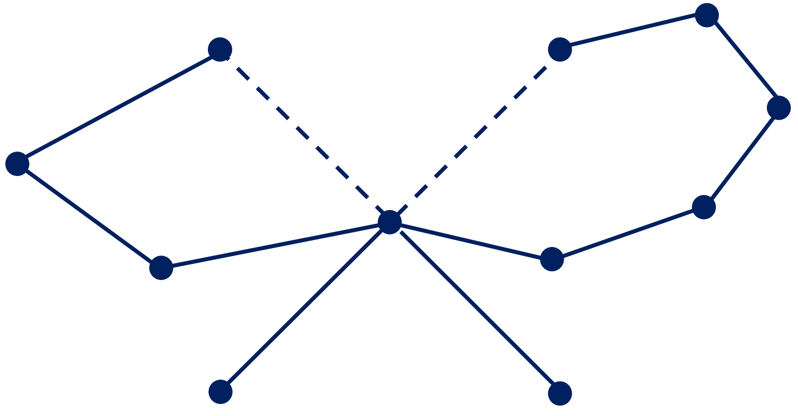
$k = 5$:



$$\sum_{(i,j)} u_{ij} \geq \left\lfloor \frac{k^2}{4} \right\rfloor$$

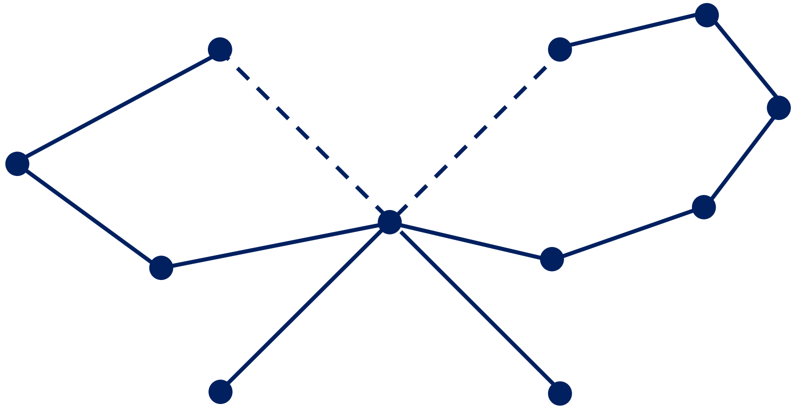
Min Linear Arrangement

- This idea has already been applied to MLA problem.
  - CLS generated cuts for stars in **complete graph**.
    - Filled in **missing edges** with **paths** in original graph.
    - Resulting cuts remain valid (not facet defining).
    - Separation algorithm found the paths.
    - Stars did most of the work.



Min Linear Arrangement

- This idea has already been applied to MLA problem.
  - CLS generated cuts for stars in **complete graph**.
    - Filled in **missing edges** with **paths** in original graph.
    - Resulting cuts remain valid (not facet defining).
    - Separation algorithm found the paths.
    - Stars did most of the work.
  - So far, our cuts do not improve on stars + separation.



Min Bandwidth

- Missing edges trick does not work for min bandwidth.
  - Resulting cuts are not valid.

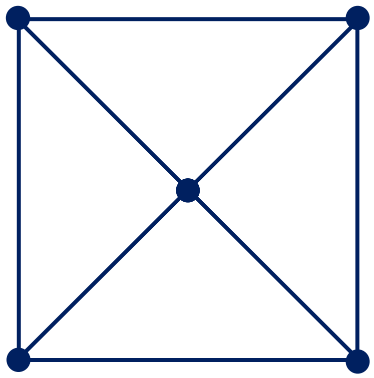
- We experiment with alternate approaches to finding cuts.
  - Focus on validity, not facets.
  - Identify cuts for large structures that fill much of the graph.
  - Modify cuts to restore validity after modifying structures.
  - Use counting arguments.

Known Bounds

- Density bound is best-known bound.
    - Chvátal (1970)
  - NP-hard to compute.
  - Not a polyhedral cut.

$$w \geq \max_{S \subseteq V} \left\lceil \frac{|S|-1}{d(S)} \right\rceil$$

diameter of S



$$w \geq \left\lceil \frac{5-1}{2} \right\rceil = 2$$

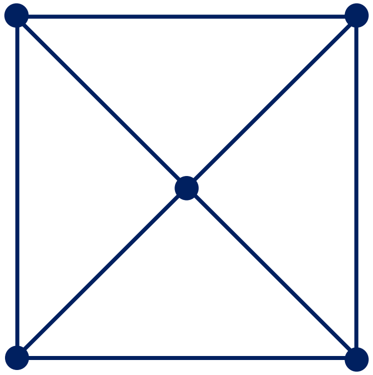
bandwidth = 3

Known Bounds

- Alternate bound (call it CS bound)
    - Caprara, Salazar-González, *IJOC* 2005.
  - P-time to compute.
  - But not a polyhedral cut.

$$w \geq \min_{v \in V} \max_{S : v \in S} \left\lceil \frac{|S| - 1}{d(v, S)} \right\rceil$$

max distance from v to vertex in S



$$w \geq \min\left\{ \left\lceil \frac{4-1}{1} \right\rceil, \left\lceil \frac{5-1}{1} \right\rceil \right\} = 3$$

bandwidth = 3

Our Proof Strategy for Bounds

- Find an upper bound on edge sum in a structured subgraph, as a function of bandwidth *w*:

$$\sum_{(i,j)} u_{ij} \leq U(w)$$

  - Then solve for bound on *w*: $w \geq U^{-1}\left(\sum_{(i,j)} u_{ij}\right)$

- Combine with lower bounds on edge sum:

$$\sum_{(i,j)} u_{ij} \geq L(w)$$

  - and with lower bounds due to CLS.

Path Cuts

- Path on *k* vertices has max edge sum

$$\sum_{(i,j)} u_{ij} \le U(w) = (k-1)w$$

- So we have the cut $w \ge U^{-1}\left(\sum_{(i,j)} u_{ij}\right) = \frac{1}{k-1}\sum_{(i,j)} u_{ij}$
  - There are many long paths, but the cut is very weak.

$$w \ge \tfrac{1}{5}\sum_{(i,j)} u_{ij}$$

For w = 5: *0* *5* *10* *15*
20 *25*

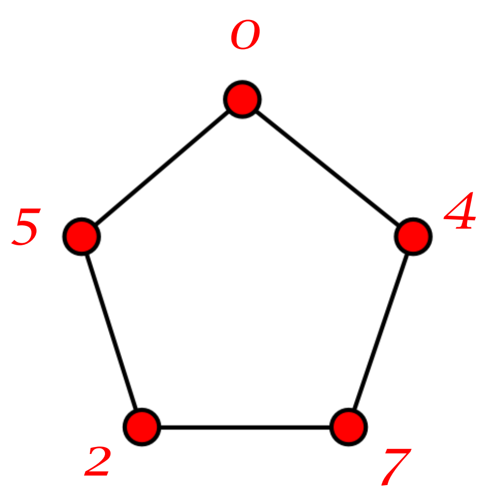
Cycle Cuts

- *k*-cycle has max edge sum

$$\sum_{(i,j)} u_{ij} \leq kw - 2 \quad \text{if } k \text{ is even}$$

$$\sum_{(i,j)} u_{ij} \leq (k-1)w \quad \text{if } k \text{ is odd}$$

For $w = 5$:



$$\sum_{(i,j)} u_{ij} \leq 4w$$

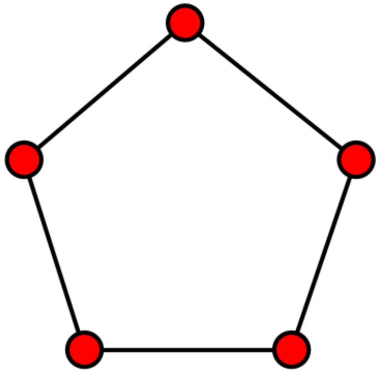
Cycle Cuts

- This yields the (sharp) bound

$$w \geq \frac{1}{k}\sum_{(i,j)} u_{ij} + \frac{2}{k} \quad \text{if } k \text{ is even}$$

$$w \geq \frac{1}{k-1}\sum_{(i,j)} u_{ij} \quad \text{if } k \text{ is odd}$$

  - Slightly stronger than for path



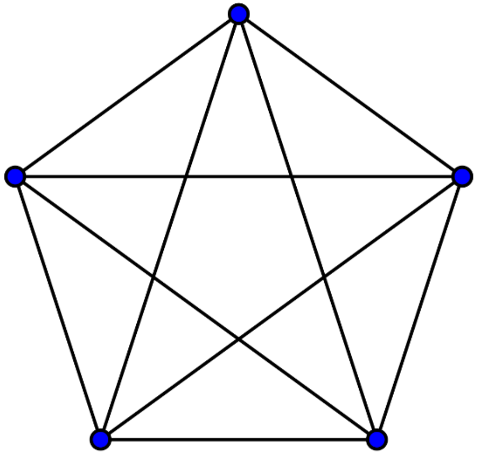
$$w \geq \tfrac{1}{4}\sum_{(i,j)} u_{ij}$$

Clique Cuts

- Clique of size *k* has max edge sum

$$\sum_{(i,j)} u_{ij} \le \frac{k^2}{4} w - \frac{k}{12}(k-1)(k-2) \qquad \text{if } k \text{ is even}$$

$$\sum_{(i,j)} u_{ij} \le \frac{k^2-1}{4} w - \frac{k+1}{12}(k-1)(k-3) \qquad \text{if } k \text{ is odd}$$



$$\sum_{(i,j)} u_{ij} \le 6w - 4$$

Clique Cuts

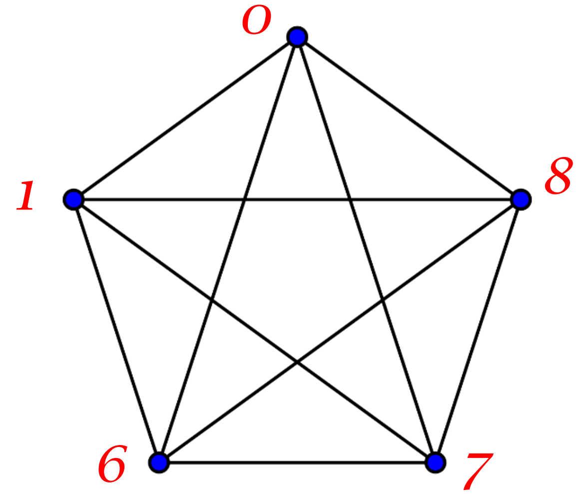
- Clique of size *k* has max edge sum

$$\sum_{(i,j)} u_{ij} \le \frac{k^2}{4} w - \frac{k}{12}(k-1)(k-2) \qquad \text{if } k \text{ is even}$$

$$\sum_{(i,j)} u_{ij} \le \frac{k^2-1}{4} w - \frac{k+1}{12}(k-1)(k-3) \qquad \text{if } k \text{ is odd}$$

  – Achieve this max by using *k*/2 smallest and *k*/2 largest vertex labels in {0, …, *w*} for *k* even, similarly for *k* odd.

For $w = 8$:



$$\sum_{(i,j)} u_{ij} \le 6w - 4$$

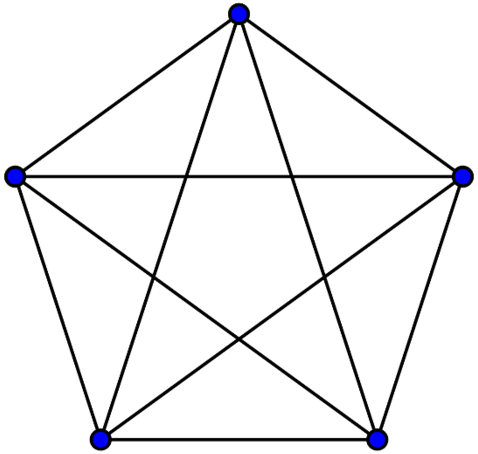
Clique Cuts

- This yields the (sharp) bound:

$$w \geq \frac{4}{k^2} \sum_{(i,j)} u_{ij} + \frac{1}{3k}(k-1)(k-2) \quad \text{if } k \text{ is even}$$

$$w \geq \frac{4}{(k^2-1)} \sum_{(i,j)} u_{ij} + \frac{1}{3}(k-3) \quad \text{if } k \text{ is odd}$$

  - Strong cut, but hard to find large cliques



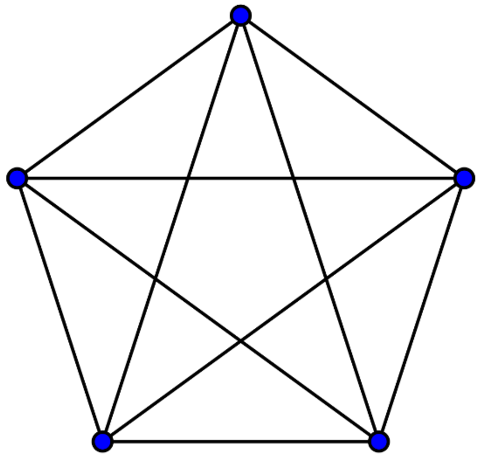
$$w \geq \tfrac{1}{6} \sum_{(i,j)} u_{ij} + \tfrac{2}{3}$$

Clique Cuts

- We also have the (sharp) lower bound on edge sum

$$\sum_{(i,j)} u_{ij} \geq \frac{k}{6}(k+1)(k-1) = \binom{k}{3}$$

  - Combined with LB on *w*, this yields $w \geq k-1$ which is sharp for a clique and same as density and CS bound.



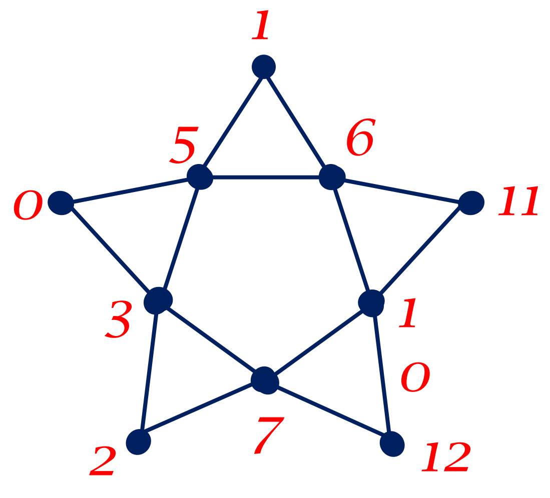
$$w \geq 4$$

Clique Cycles

- Cycles of cliques can provide good bounds for large structures.
- Cycle of *k* 3-cliques has max edge sum

$$\sum_{(i,j)} u_{ij} \leq 2kw$$

For $w = 5$:



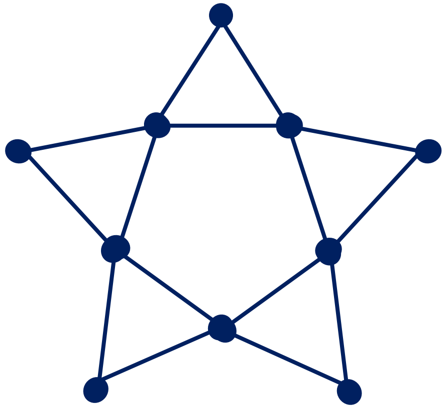
$$\sum_{(i,j)} u_{ij} \leq 10w$$

Clique Cycles

- This yields the (sharp) bound

$$w \geq \frac{1}{2k} \sum_{(i,j)} u_{ij}$$

  - For odd and even *k*



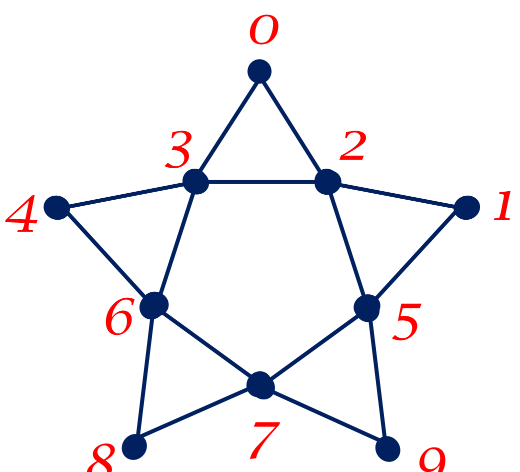
$$w \geq \frac{1}{10} \sum_{(i,j)} u_{ij}$$

Clique Cycles

- We also have the (sharp) lower bound on edge sum

$$\sum_{(i,j)} u_{ij} \geq 8(k-1)$$

  - Combined with LB on *w*, this yields $w \geq \left\lceil 4 - \frac{4}{k} \right\rceil$
  - For *k* = 5, this yields *w* ≥ 4 (sharp), stronger than density bound 3, same as CS bound.

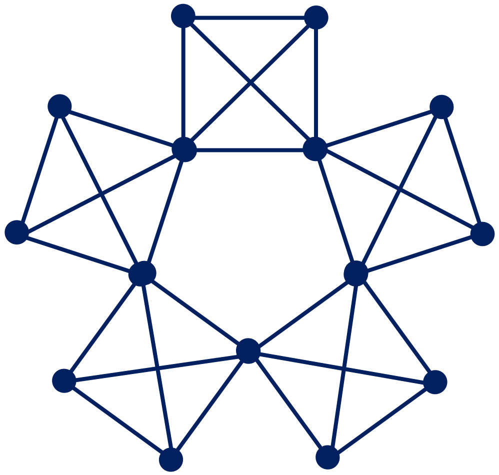


$$\sum_{(i,j)} u_{ij} \geq 32$$

Clique Cycles

- Conjecture: Cycle of *k* 4-cliques has max edge sum

$$\sum_{(i,j)} u_{ij} \leq 4kw - 2k + 1$$

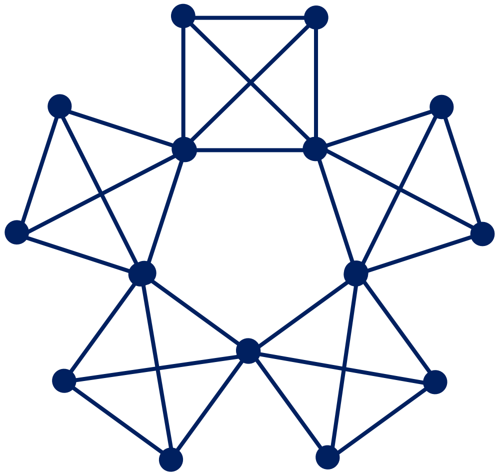


$$\sum_{(i,j)} u_{ij} \leq 20w - 9$$

Clique Cycles

- This yields the bound

$$w \geq \frac{1}{4k} \sum_{(i,j)} u_{ij} + \frac{2k-1}{4k}$$



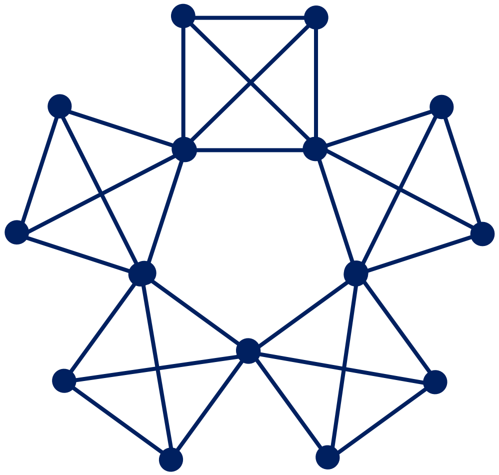
$$w \geq \tfrac{1}{20} \sum_{(i,j)} u_{ij} + \tfrac{9}{20}$$

Clique Cycles

- We also have the (sharp) lower bound on edge sum

$$\sum_{(i,j)} u_{ij} \geq 16k - 15$$

  - Combined with LB on *w*, this yields $w \geq \left\lceil 4.5 - \frac{1}{4k} \right\rceil$
  - For $k = 5$, this yields $w \geq 5$ (sharp). Density and CS bound are also 5.

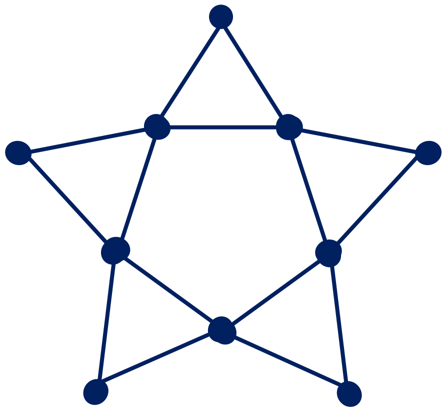


$$\sum_{(i,j)} u_{ij} \geq 65$$

Cuts when Edges Are Missing

- A certificate of a sharp bound on edge sum can yield a sharp bound when edges are removed.
  - Start with a sharp upper bound *U*(*w*) on edge sum**.**

For $w = 5$:

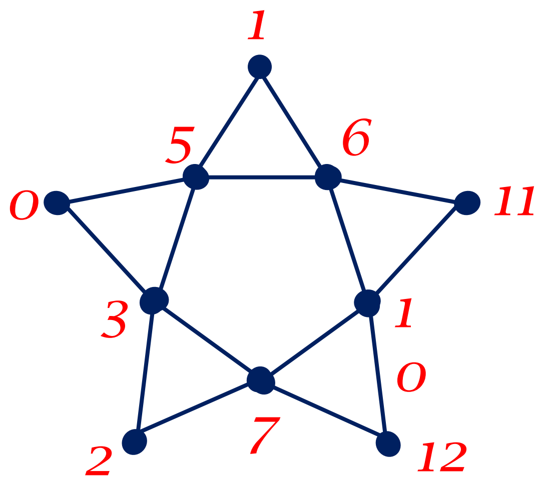


$$\sum_{(i,j)} u_{ij} \le 10w$$

Cuts when Edges Are Missing

- A certificate of a sharp bound on edge sum can yield a sharp bound when edges are removed.
  - Start with a sharp upper bound $U(w)$ on edge sum.
  - Let v be a vertex labeling that achieves edge sum $U(w)$.

For $w = 5$:

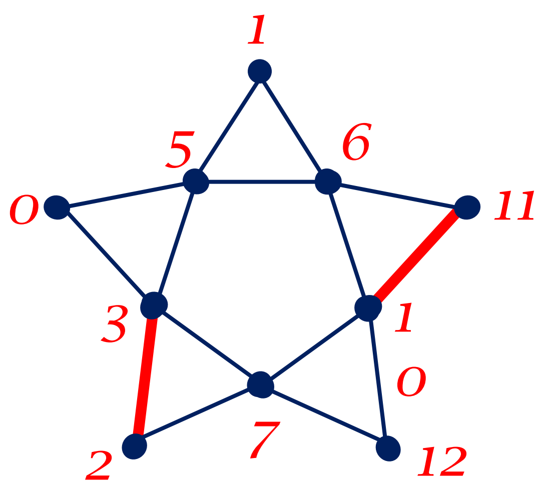


$$\sum_{(i,j)} u_{ij} \leq 10w$$

Cuts when Edges Are Missing

- A certificate of a sharp bound on edge sum can yield a sharp bound when edges are removed.
  - Start with a sharp upper bound $U(w)$ on edge sum.
  - Let v be a vertex labeling that achieves edge sum $U(w)$.
  - Remove m edges (i, j) for which $|v_i - v_j| = 1$.

For $w = 5$:

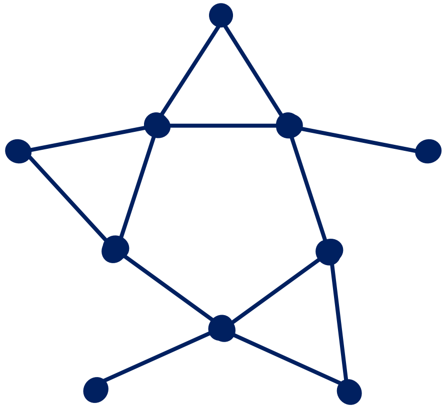


$$\sum_{(i,j)} u_{ij} \leq 10w$$

Cuts when Edges Are Missing

- A certificate of sharp bound on edge sum can yield a sharp bound when edges are removed.
  - Start with a sharp upper bound $U(w)$ on edge sum.
  - Let v be a vertex labeling that achieves edge sum $U(w)$.
  - Remove m edges (i, j) for which $|v_i - v_j| = 1$.
  - Now $U(w) - m$ is a sharp upper bound on edge sum.

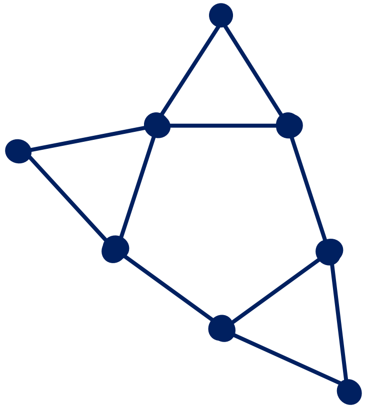
For $w = 5$:



$$\sum_{(i,j)} u_{ij} \leq 10w - 2$$

Cuts when Edges Are Missing

- Can get a valid (nonsharp) bound after removing any *m* edges
  - $U(w) - m$ is an upper bound on edge sum.



$$\sum_{(i,j)} u_{ij} \le 10w - 4$$

Counting Arguments

- Take any graph with 12 vertices.
  - And any subgraph with *m* edges.
  - For any bandwidth *w*, compute maximum edge sum.
  - Sum the first *m* edge labels starting with *w*.

Edge Label	**Max No. Edges**	**Possible edges**
11	1	(12,1)
10	2	(12,2) (11,1)
9	3	(12,3) (11,2) (10,1)
8	4	(12,4) (11,3) (10,2) (9,1)
7	5	(12,5) (11,4) (10,3) (9,2) (8,1)
6	6	(12,6) (11,5) (10,4) (9,3) (8,2) (7,1)
etc.		

Counting Arguments

- For example, assuming subgraph has 7 edges
  - For bandwidth $z = 11$
  - Max edge sum is $1 \cdot 11 + 2 \cdot 10 + 3 \cdot 9 + 1 \cdot 8 = 66$

Edge Label	Max No. Edges	Possible edges
11	1	**(12,1)**
10	2	**(12,2) (11,1)**
9	3	**(12,3) (11,2) (10,1)**
8	4	**(12,4)** (11,3) (10,2) (9,1)
7	5	(12,5) (11,4) (10,3) (9,2) (8,1)
6	6	(12,6) (11,5) (10,4) (9,3) (8,2) (7,1)
etc.		

Counting Arguments

- For example, assuming subgraph has 7 edges
  - For bandwidth *z* = 10
  - Max edge sum is 2·10 + 3·9 + 2·8 = 63

Edge Label	Max No. Edges	Possible edges
11	1	(12,1)
→ **10**	2	**(12,2) (11,1)**
9	3	**(12,3) (11,2) (10,1)**
8	4	**(12,4) (11,3)** (10,2) (9,1)
7	5	(12,5) (11,4) (10,3) (9,2) (8,1)
6	6	(12,6) (11,5) (10,4) (9,3) (8,2) (7,1)
etc.		

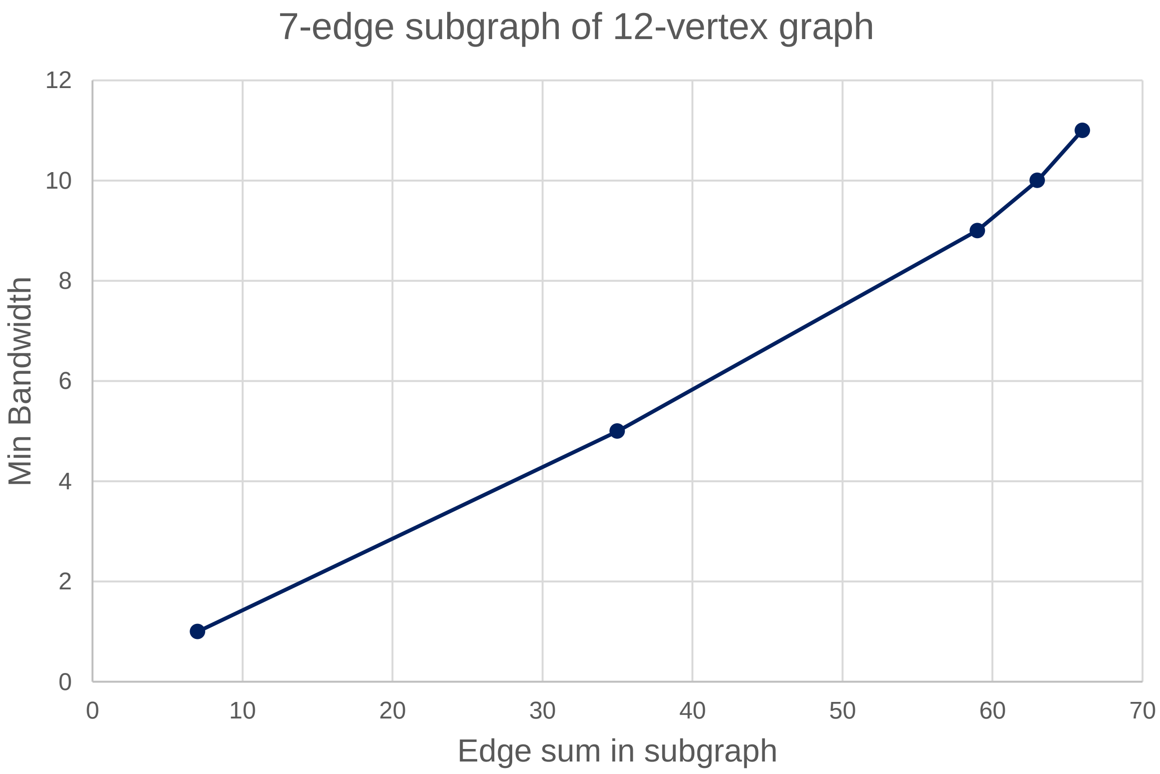
Counting Arguments

- For example, assuming subgraph has 7 edges
  - For bandwidth $z = 9$
  - Max edge sum is $3 \cdot 9 + 4 \cdot 8 = 59$

Edge Label	Max No. Edges	Possible edges
11	1	(12,1)
10	2	(12,2) (11,1)
9	3	**(12,3) (11,2) (10,1)**
8	4	**(12,4) (11,3) (10,2) (9,1)**
7	5	(12,5) (11,4) (10,3) (9,2) (8,1)
6	6	(12,6) (11,5) (10,4) (9,3) (8,2) (7,1)
etc.		

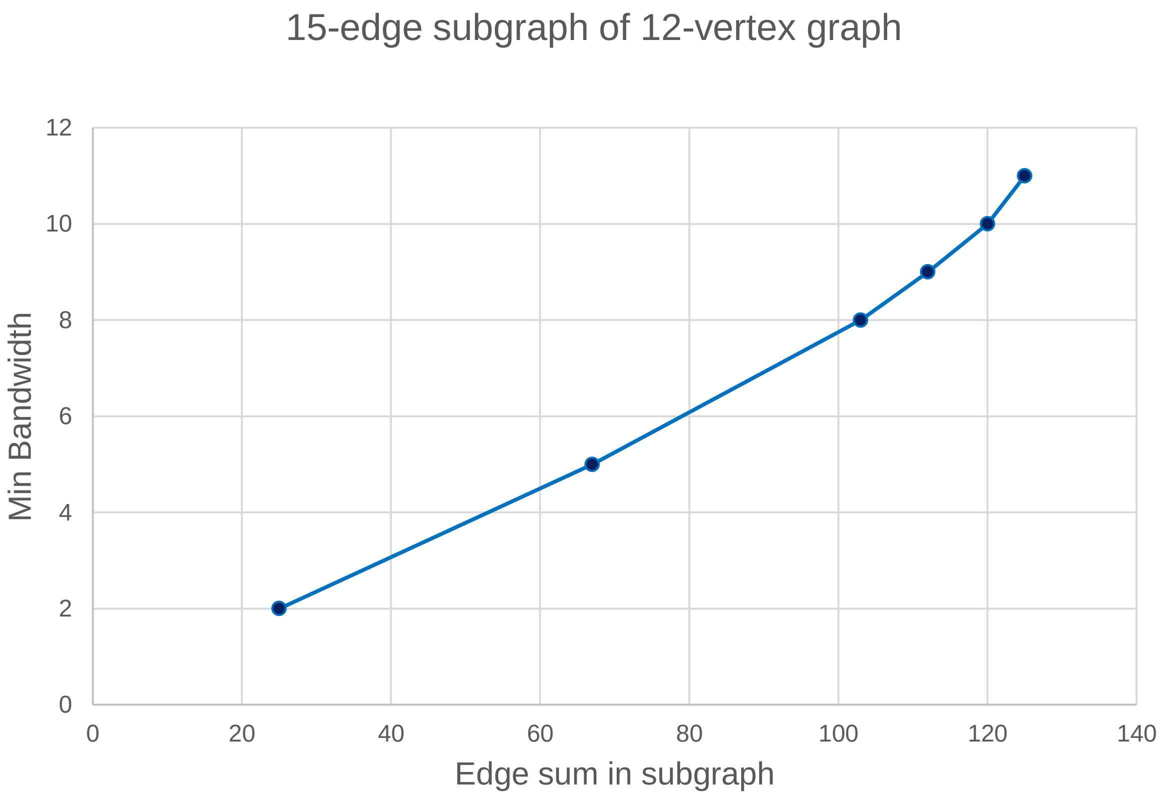
Counting Arguments

- This yields family of **valid cuts**
  - of the form $w \geq \alpha \sum_{(i,j)} u_{ij} + \beta$



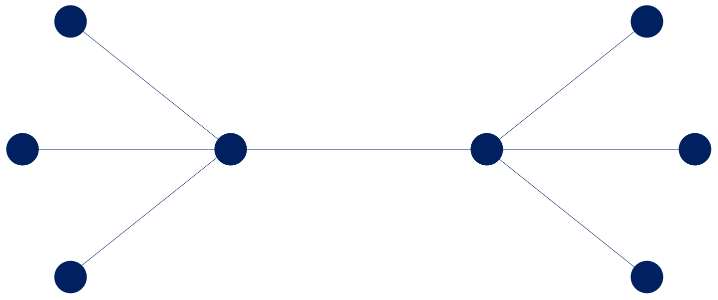
Counting Arguments

- This yields family of **valid cuts**
  - of the form $w \geq \alpha \sum_{(i,j)} u_{ij} + \beta$



Counting Arguments

- For stronger cuts, we must consider structure of subgraph.
  - We will use degree constraints.
- Example:
  - Suppose $n = 12$.
  - Consider subgraph with $m = 7$.
  - Enforce degree constraints
    - 2 vertices degree 4
    - 6 vertices degree 1



Counting Arguments

- Use greedy algorithm to get upper bound on edge sum
  - For bandwidth $z = 10$
  - UB is $1 \cdot 11 + 2 \cdot 10 + 2 \cdot 9 + 2 \cdot 8 = 65$ (not 66)
    - Max is 65

Edge Label	Max No. Edges	Possible edges
→ **11**	1	**(12,1)**
10	2	**(12,2) (11,1)**
9	3	**(12,3)** (11,2) **(10,1)**
8	4	**(12,4)** (11,3) (10,2) **(9,1)**
7	5	(12,5) (11,4) (10,3) (9,2) (8,1)
6	6	(12,6) (11,5) (10,4) (9,3) (8,2) (7,1)
etc.		

Counting Arguments

- Use greedy algorithm to get upper bound on edge sum
  - For bandwidth $z = 10$
  - Max edge sum is $2 \cdot 10 + 2 \cdot 9 + 2 \cdot 8 + 1 \cdot 7 = 61$ (not 63)
    - Max is 58

	Edge Label	Max No. Edges	Possible edges
	11	1	(12,1)
→	**10**	2	**(12,2) (11,1)**
	9	3	**(12,3)** (11,2) **(10,1)**
	8	4	**(12,4)** (11,3) (10,2) **(9,1)**
	7	5	**(12,5)** (11,4) (10,3) (9,2) (8,1)
	6	6	(12,6) (11,5) (10,4) (9,3) (8,2) (7,1)
	etc.		

Counting Arguments

- Research question:
  - For what structures does greedy algorithm yield valid upper bound?

Edge Label	Max No. Edges	Possible edges
11	1	(12,1)
10	2	**(12,2) (11,1)**
9	3	**(12,3)** (11,2) **(10,1)**
8	4	**(12,4)** (11,3) (10,2) **(9,1)**
7	5	**(12,5)** (11,4) (10,3) (9,2) (8,1)
6	6	(12,6) (11,5) (10,4) (9,3) (8,2) (7,1)
etc.		

Ongoing Research

- Explore other substructures with dense regions.
- Develop cuts based on certificates.
- Identify structures for which greedy algorithm works in counting argument.
- Find separation algorithms.
- Computational tests.