Hypergraph-Partitioning-Based Sparse Matrix Ordering

Ümit V. Çatalyürek and Cevdet Aykanat

Department of Biomedical Informatics
The Ohio State University

Department of Computer Engineering
Bilkent University

Fill Reducing Ordering

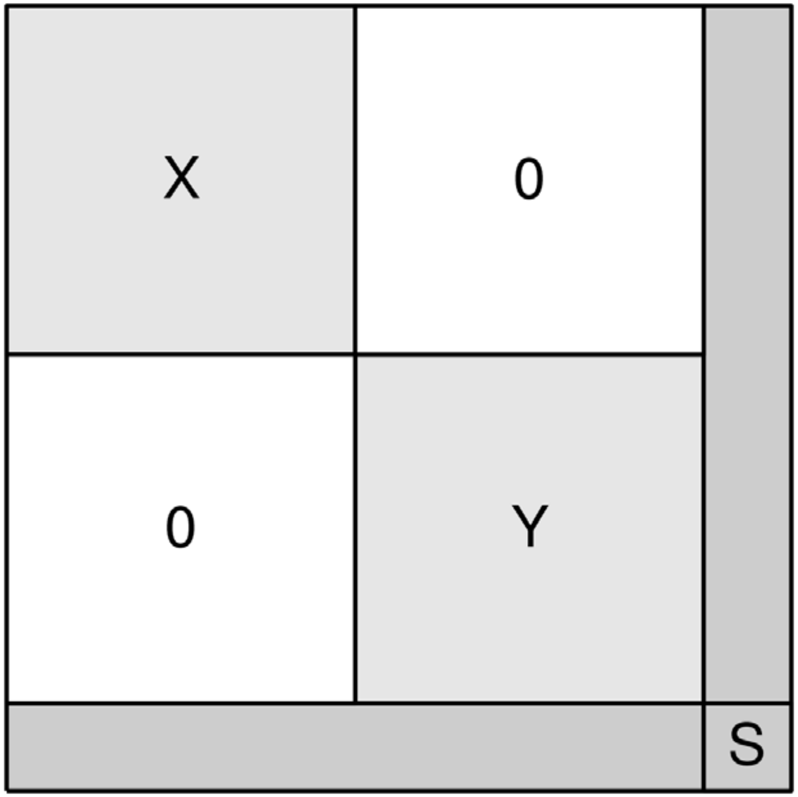
- Direct solution of Zx=b requires reordering of Z
  - Fill
  - Operation count
- Widely used methods
  - Minimum Degree (MD) and variants MMD, AMD, AMF
  - Nested Dissection (ND)
- ND uses GPVS (Graph Partitioning with Vertex Separator)
- Contributions
  - Show that GPVS can be solved through Hypergraph Partitioning
  - Propose Recursive Hypergraph bipartitioning methods for nested dissection ordering

Outline

- Preliminaries
  - Nested Dissection
  - Graph Partitioning by Vertex Separator
    - Flaw of GPVS in Multilevel Graph Partitioning
  - Hypergraph Partitioning (HP)
  - Net Intersection Graph (NIG)
- Solving GPVS via HP
  - Graph theoretical view
  - Matrix theoretical view
- Ordering for LP-type applications
  - Hypergraph Reduction
    - Node removal & Sparsening
- Generalization (2-clique model)
- Vertex Compression for ND
- Results
- Conclusion

Nested Dissection

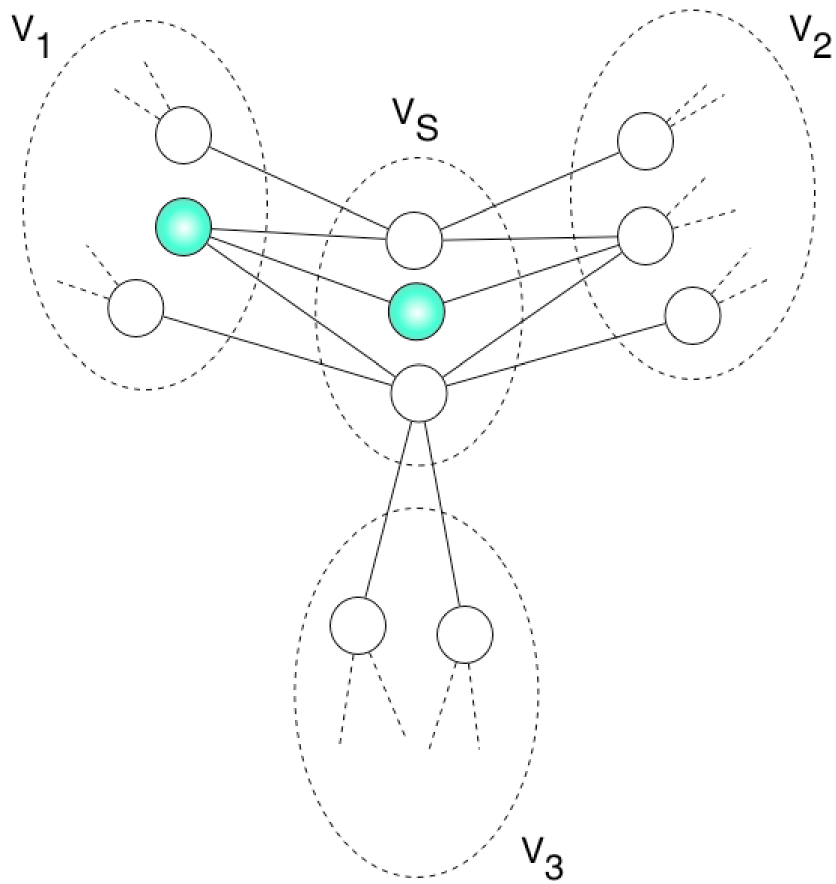
- If S is ordered after X and Y
  - No fill in off-diagonal
- Order X and Y by
  - Recursively via ND
  - Use another ordering technique, e.g. MMD
- S must be small
  - Requires good GPVS



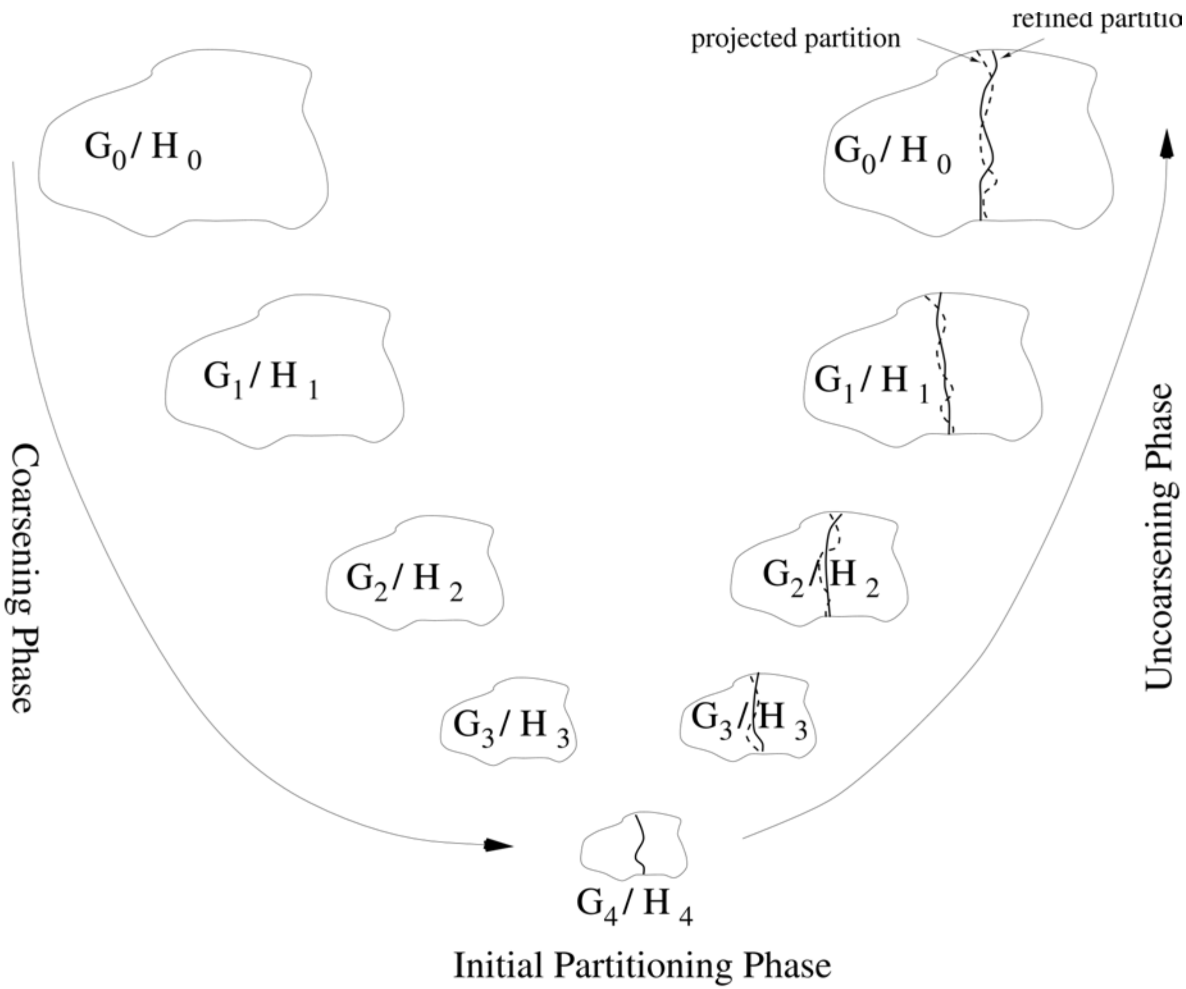
Graph Partitioning by Vertex Separator

- Graph G = (V, E) : set of vertices V and set of edges E
  - every edge $e_{ij} \in$ E connects a pair of distinct vertices $v_i, v_j \in$ V
- K-way graph partition by vertex separator: $\Pi_{GPVS} = \{V_1, V_2, \ldots, V_K ; V_S\}$
  - V_k's are nonempty and pairwise disjoint subsets of V
  - removal of separator V_S gives K disconnected parts; $V_1, V_2, \ldots, V_K$

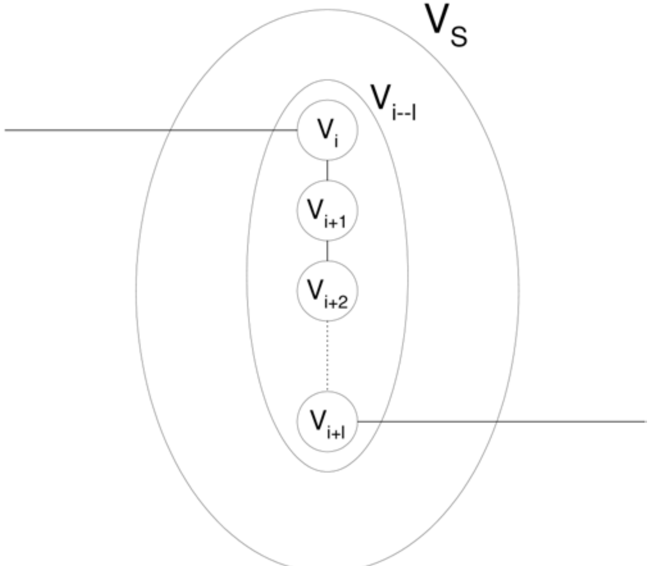
    i.e., $Adj(V_k) = V_S$ for each k
- a separator is
  - ***narrow***: if no subset of it forms a sepator
  - ***wide***: otherwise
- cost of a partition: $cutsize(\Pi) = |V_S|$

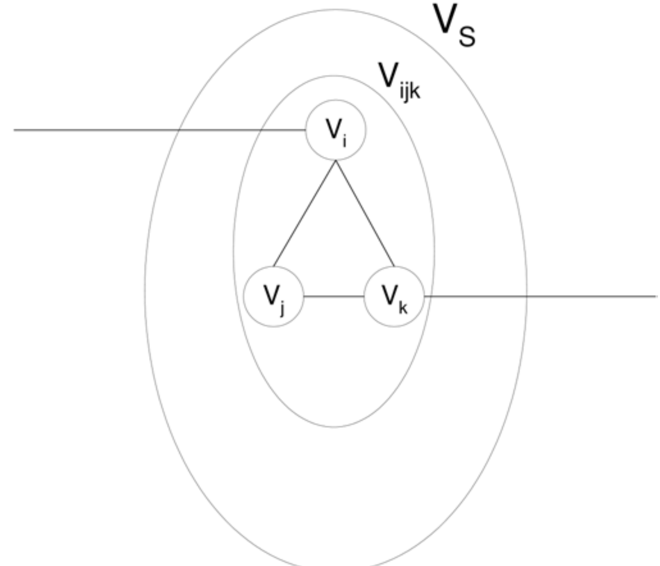


Multilevel Graph/Hypergraph Partitioning



Flaw of Multilevel Framework in GPVS



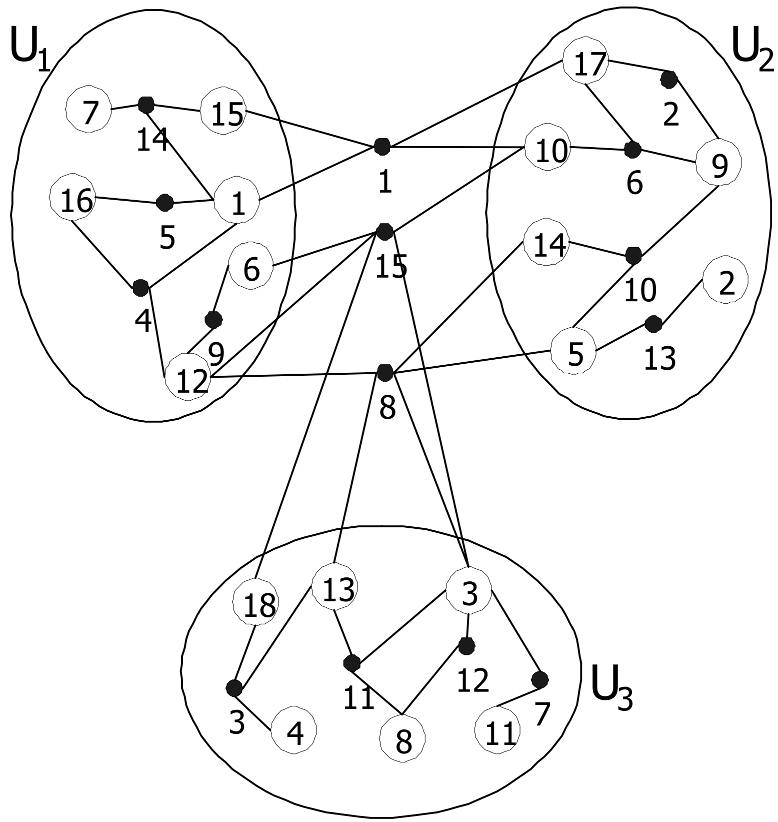


- Multilevel GPES: Edge cut for a coarse graph is an edge cut for original graph with the same cutsize
- Multilevel GPVS: separator for a coarse graph is not a truly minimal vertex cover for original graph
- Jurgen Schulze proposed coarsening quotient graphs by eliminating a set of independent variables

Hypergraph Partitioning

- Hypergraph $H = (U, N)$: a set of nodes (vertices) U and a set of nets N
  - nets (hyperedges) connect two or more vertices
    - every net $n_j \in N$ is a subset of vertices, i.e., $n_j \subseteq U$
    - nodes in a net are called its pins
  - graph is a special instance of hypergraph

- K-way hypergraph partition:
  $\Pi = \{U_1, U_2, \ldots, U_K\}$
  - a net that has at least one pin in a part is said to *connect* that part
  - a net n_j is said to be
    - ***cut*** *(external) if it connects more than one part*
    - ***uncut*** *(internal) if it connects exactly one part*
  - net-cut metric: each cut net contributes one (its weight) to the cutsize

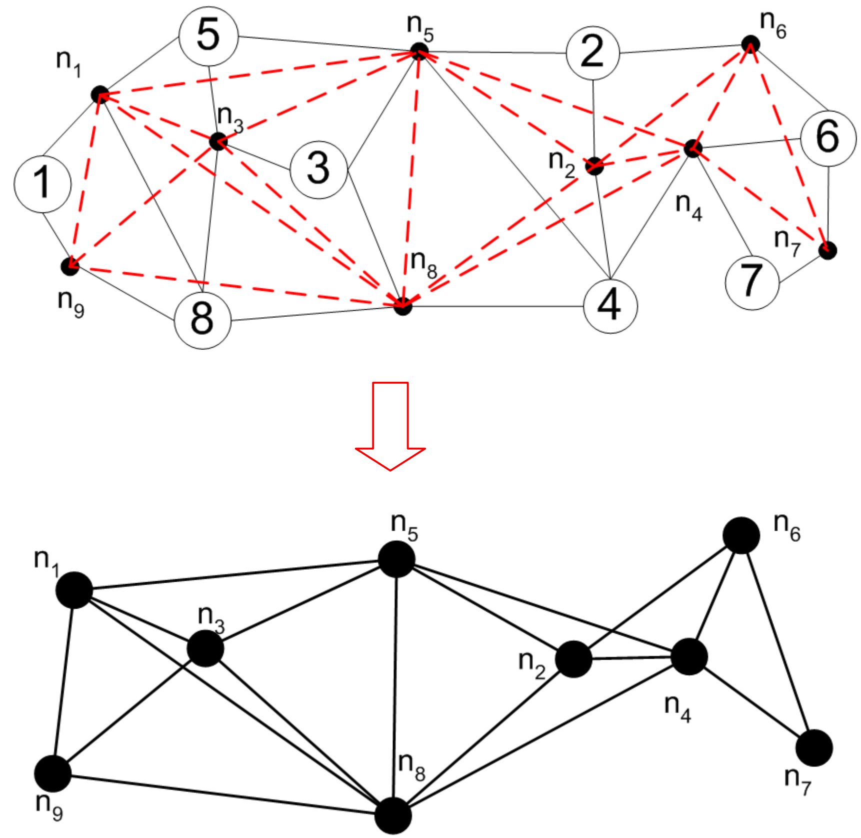


- cut nets: $N_E = \{n_1, n_8, n_{15}\}$
- Assuming unit net weights:

$$\text{cutsize}(\Pi) = |N_E| = 3$$

Net Intersection Graph (NIG) representation G of a Hypergraph H

- One vertex v_i in G for each net n_i of H
- There is an edge between two vertices of G iff they share at least one pin in H
  - i.e., $e_{i,j} \in E$ iff $Pins(n_i) \cap Pins(n_j) \neq \emptyset$

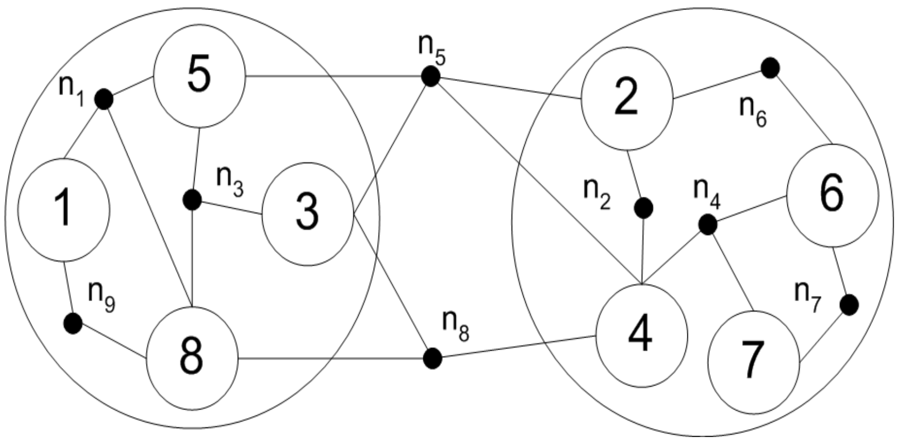


- Note: NIG G of H ≡ Clique Net Graph of dual of H
- i.e., each node u_i of H induces a clique on NIG vertices that correspond to $Nets(u_i)$

Solving GPVS through HP

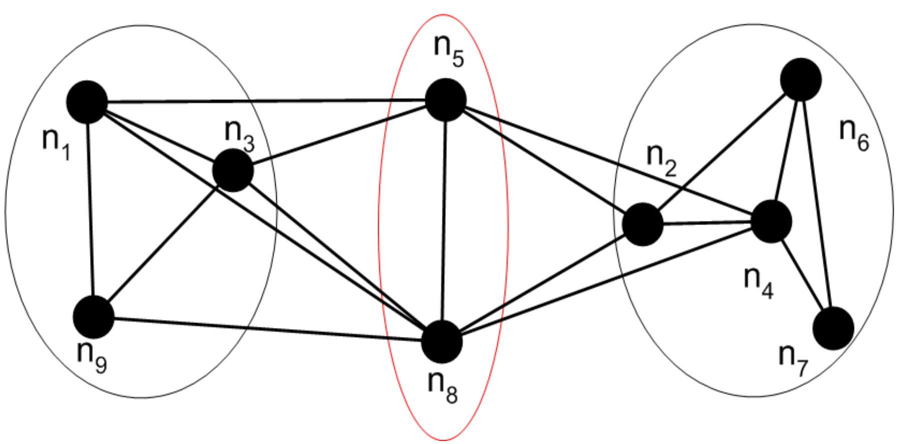
- Consider a 2-way vertex partition $\Pi_{HP} = \{U_1, U_2\}$ of H
- Decode Π_{HP} as 3-way net partition $\Pi_{HP} = \{N_1, N_2; N_S\}$ on H
  - N_1 and N_2 correspond to internal nets of U_1 and U_2
  - N_S corresponds to external nets
- Π_{HP} induces a 2-way GPVS Π_{GPVS} on NIG G where
  - $\Pi_{GPVS} = \{V_1, V_2 ; V_S\}$ where $V_1 \equiv N_1$, $V_2 \equiv N_2$, $V_S \equiv N_S$
- Consider an internal net n_i of part U_1: we have either
  - $Adj_H(n_i) \subseteq N_1$ or $Adj_H(n_i) \subseteq N_1 \cup N_S$ in H
- So we have either $Adj_G(v_i) \subseteq N_1$ or $Adj_H(v_i) \subseteq N_1 \cup N_S$ in G
- $\Rightarrow Adj_G(V_1) \subseteq V_S$ and $Adj_G(V_2) \subseteq V_S$
- $\Rightarrow V_S$ is a valid separator for NIG G

Solving GPVS through HP



Minimizing net cut in H

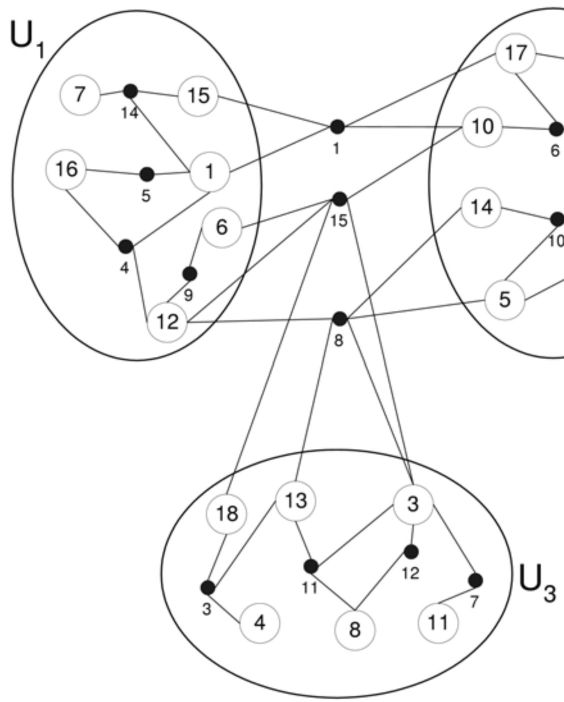
Minimizing separator size in G



Balancing internal-nets in H

Balancing vertices in G

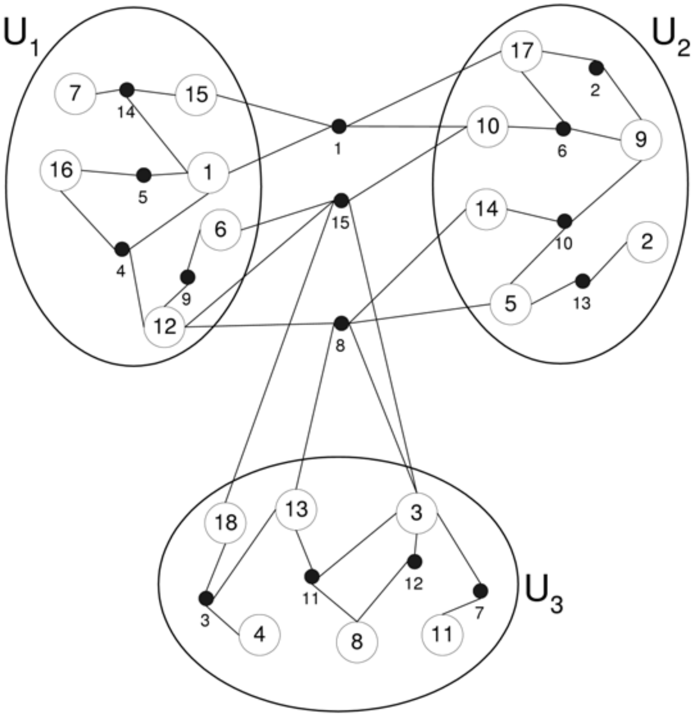
Matrix Theoretical View
Row-Net Hypergraph Representation H_A of a Matrix A

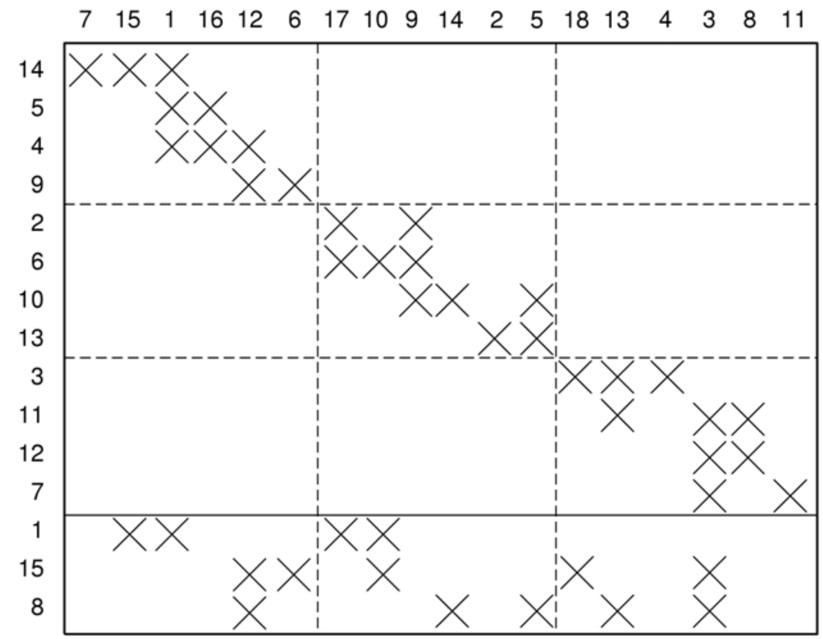


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	X									X					X		X	
2									X								X	
3				X									X					X
4	X											X				X		
5	X															X		
6									X	X							X	
7			X								X							
8			X		X							X	X	X				
9						X						X						
10					X				X					X				
11			X					X					X					
12			X					X										
13		X			X													
14	X						X								X			
15			X			X				X		X						X

- One net n_i for each row i and one vertex v_j for each column j
- net n_i contains vertices corresponding to cols that have a nonzero in row i,

Matrix Theoretical View
Π_{HP} induces a Singly Bordered (SB) form on A



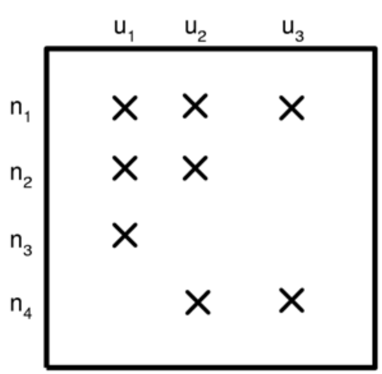


- Columns associated with vertices in U_{k+1} are ordered after vertices in U_k
- Rows associated with internal nets of U_{k+1} are ordered after internal nets of U_k
- Rows associated with cut nets are ordered last as the border
- Minimizing the net cut in H_A corresponds to minimizing the border size in A

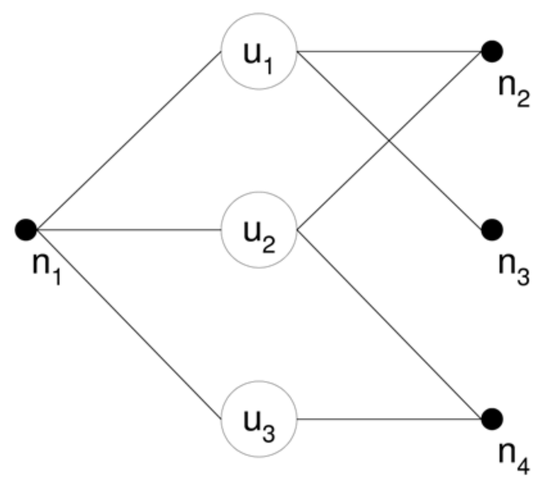
Matrix Theoretical View: Solving GPVS through HP

- H_A is the row-net hypergraph representation of a matrix A
- NIG G of H_A is the standard graph representation of $Z = AA^T$
- So, an SB form A_{SB} of A induces a Doubly Bordered (DB) form Z_{DB} of Z

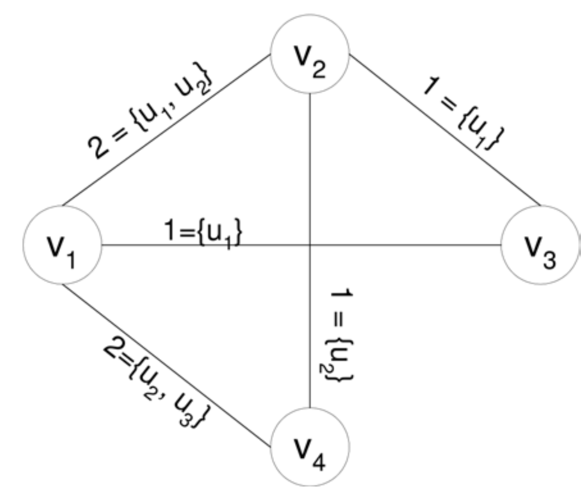
  $\Rightarrow \Pi_{HP}$ on H_A of matrix A induces a DB form on matrix Z
- Minimizing net cut in H_A corresponds to minimizing border size in Z_{DB}



Matrix A

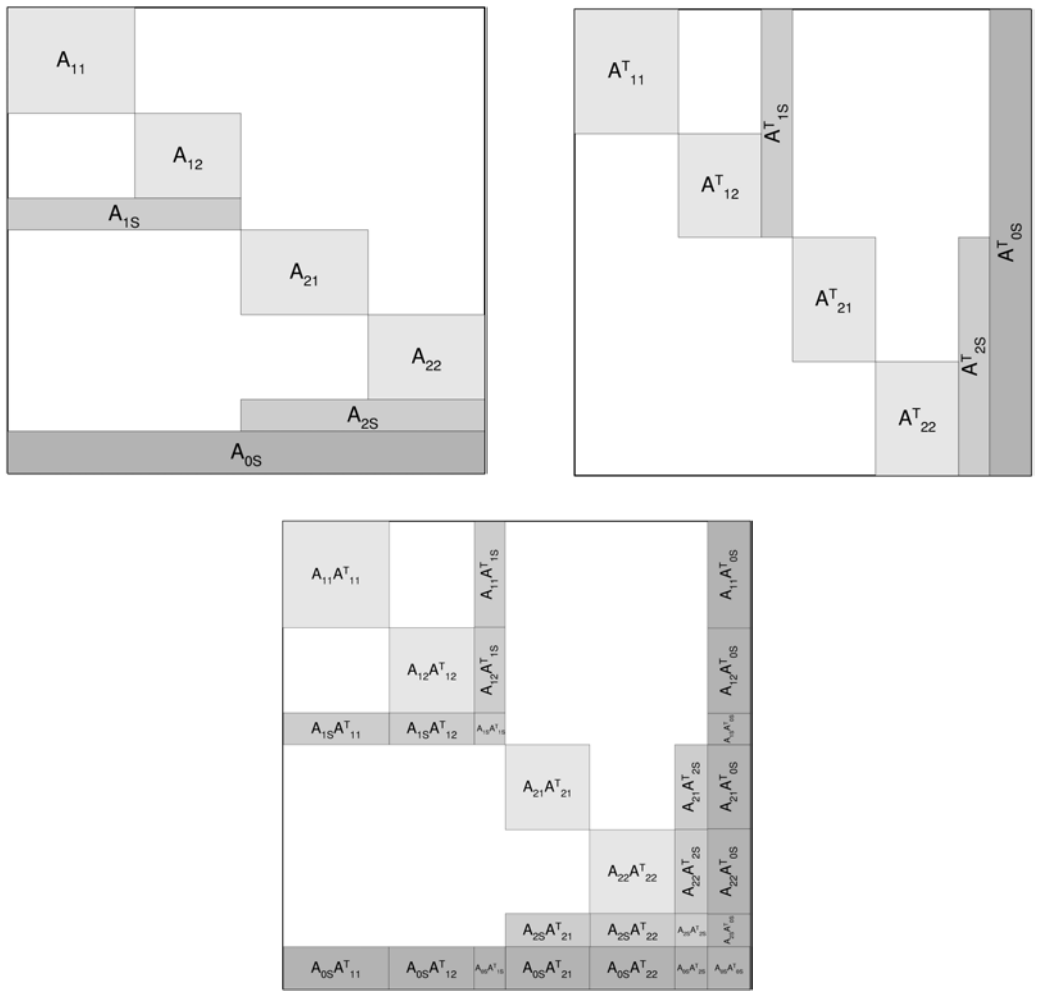


Row-net hypergraph H_A of A



NIG of H_A

Matrix-Theoretical View of the Relation Between HP and GPVS



Ordering Coefficient Matrices in LP-type Applications

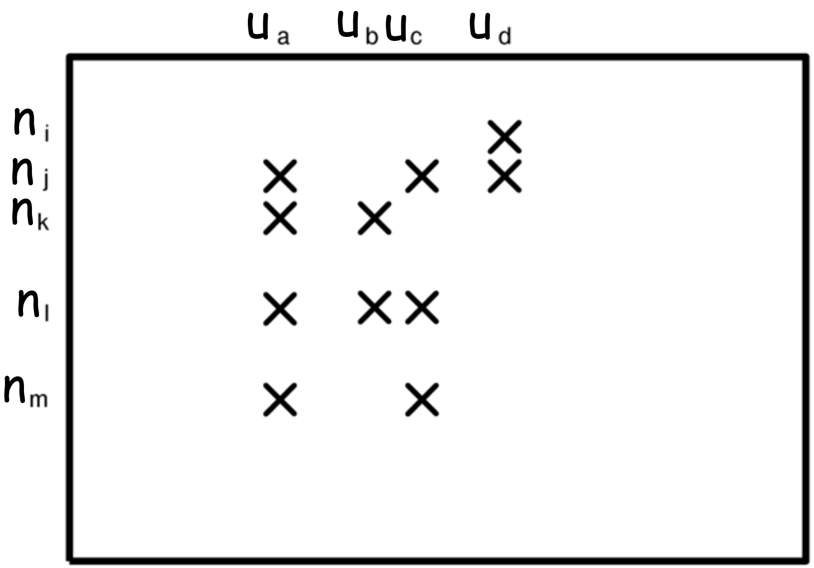
- Given a hypergraph H its NIG G is well defined
- But there is no unique reverse construction

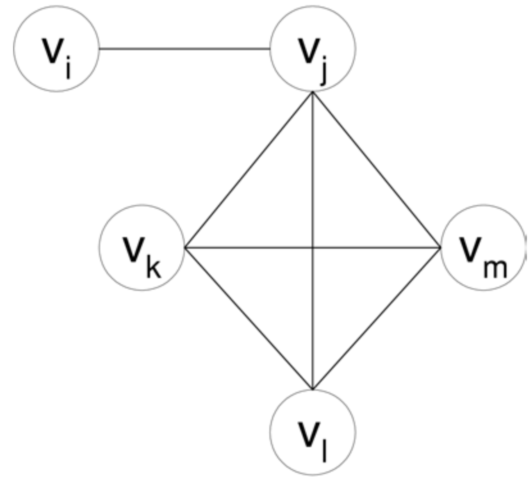
- Interior Point type solvers: solve Zx=b, where $Z = ADA^T$

- So, given standard graph representation G_Z of matrix Z
  - H_A : row-net hypergraph representation of matrix A,
  - where NIG of H_A is G_Z
- Recursive bisection on hypergraph H_A

  $\Rightarrow$ nested dissection on $Z = ADA^T$
- Simplifications in H_A / A
  - Node / column removal
  - Sparsening thru pin / nonzero removal

Hypergraph Reduction via Node Removal





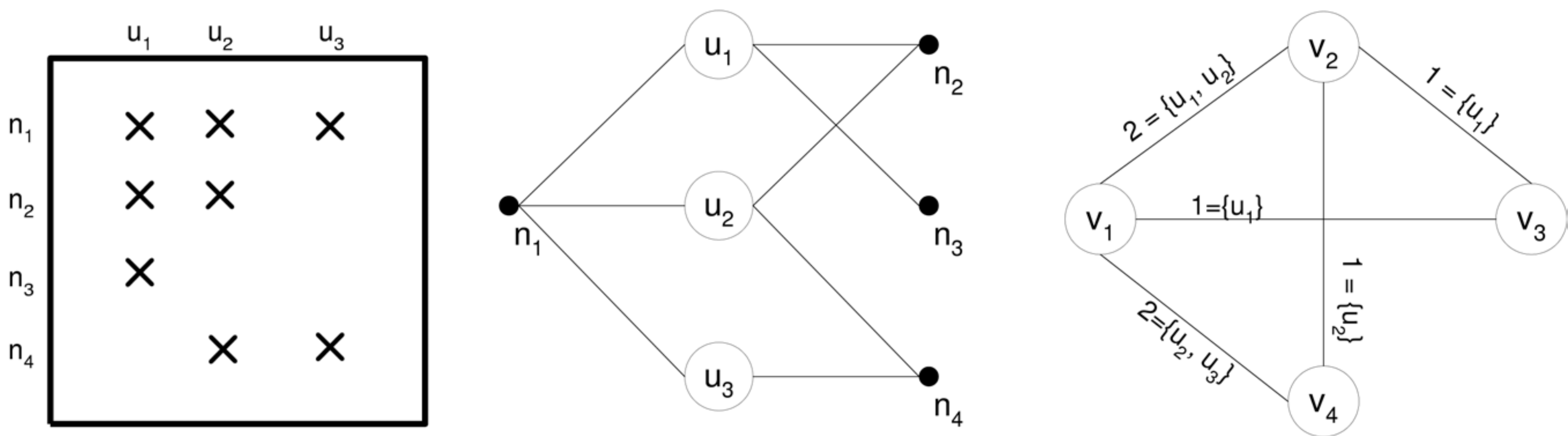
- each node u_x of H_A (column of A) induces a clique on NIG vertices that correspond to Nets (u_x)
- So, if $u_x \subseteq u_y$ we can remove node u_x
- In the above example, we can remove nodes (columns) u_b and u_c
- approximately 2% of the nodes/cols of H_A /A are removed on average

Vertex Removal Algorithm

initialize $delete[u_i] \leftarrow$ FALSE for $u_i \in \mathcal{U}$
for node $u_i \in \mathcal{U}$ in non-increasing degree order
    if $delete[u_i] =$ FALSE then
        for each $n_j \in nets[u_i]$ do
            if $deg_{\mathcal{G}}(v_j) = deg_{\mathcal{H}}(u_i) - 1$ then
                for each $u_k \in pins[n_j]$ do
                    if $u_k \neq u_i$ and $delete[u_k] =$ FALSE then
                        $delete[u_k] \leftarrow$ TRUE
delete all nodes u_i of $\mathcal{H}$ with $delete[u_i] =$ TRUE

Sparsening of H_A/A thru pin/nonzero removal

- Two vertices of NIG G_{AA^T} are adjacent if the respective nets share pins in H_A
- If they share more than one pin, only one of them suffices for our purpose
- pin (n_i, u) can be deleted if $W_{ij} > 1$ for each net $n_j \in$ Nets (u) - $\{n_i\}$
  (W_{ij} = number of common pins of nets n_i and n_j)



- pin (n_1,u_2) can be deleted since both $W_{12} > 1$ and $W_{14} > 1$

- 20% of the pins/nonzeroes of H_A/A are deleted on average

Sparsening Algorithm

initialize $W[j] \leftarrow 0$ for $i = 1, \ldots, |\mathcal{N}|$
for each net $n_i \in \mathcal{N}$ do
    for each node $u \in pins[n_i]$ do
        for each $n_j \in nets[u]$ do
            $W[j] \leftarrow W[j] + 1$
    for each node $u \in pins[n_i]$ do
        flag $\leftarrow$ TRUE
        for each $n_j \in nets[u]$ do
            if $n_j \neq n_i$ and $W[j] = 1$ then
                flag $\leftarrow$ FALSE
                break
        if flag = TRUE then
            $nets[u] \leftarrow nets[u] - \{n_i\}$
            $pins[n_j] \leftarrow pins[n_j] - \{u\}$
            for each $n_j \in nets[u]$ do
                $W[j] \leftarrow W[j] - 1$
    for each node $u \in pins[n_i]$ do
        for each $n_j \in nets[u]$ do
            $W[j] \leftarrow 0$

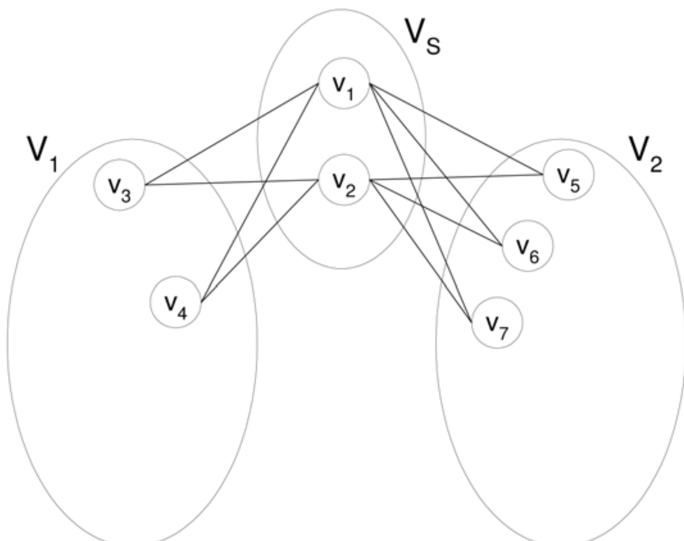
Generalization: 2-clique Model

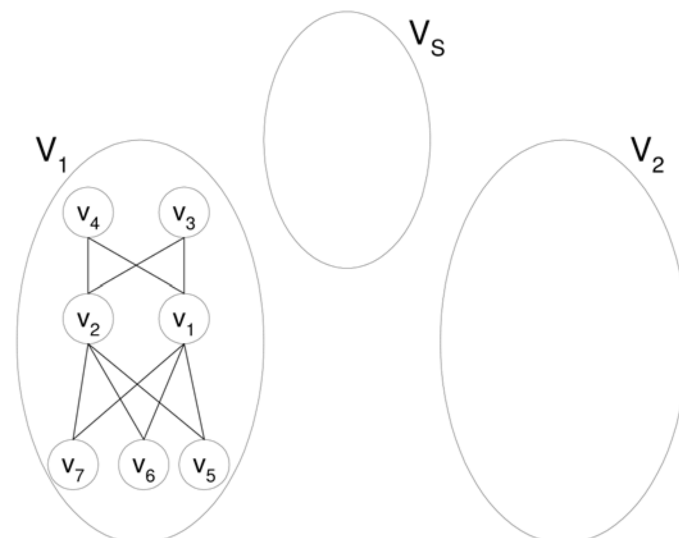
- Factorization of $Z = AA^T$, what is A $\Leftrightarrow$ find H such that its NIG G is standard graph representation of Z

- 2-clique decomposition
  - for G = (V, E), construct H = (U, N)
    - net set N: one net $n_i \in E$ for each vertex v_i in G
    - node set U: one node $u_{ij} \in U$ for each edge $e_{ij} \in E$
    - $u_{ij} \in n_i$ and $u_{ij} \in n_j$
      - Each node connects exactly two nets
  - 2-clique decomposition: A = edge-incidence matrix of Z

Vertex Compression in G_{AA^T} for ND
$\Rightarrow$ Net Compression in H_A

- *Supernode*: set of **connected** vertices with the same adjacency structure
- Observation for ND: disconnected vertices with identical adjacency structure can also be compressed
  - If any constituent vertex of a supernode belongs to V_S (V_1) in Π_{GPVS}
  - Then all other consituent vertices belong to V_S (V_1)



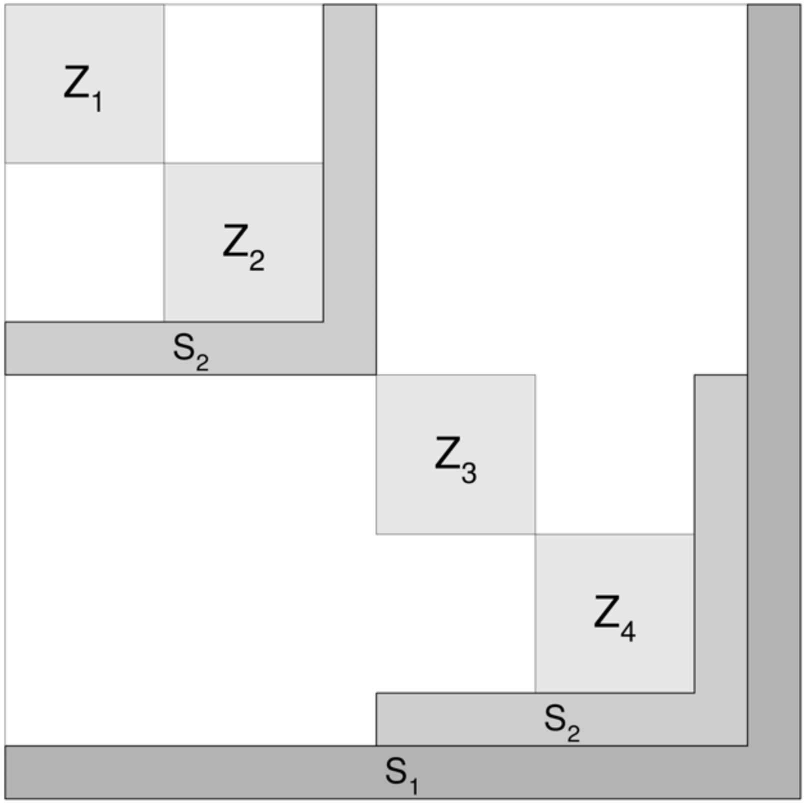


Matrix Type / *Supernodes*	**General**	**LP**
Connected	26%	5.5%
Disconnected	1%	3.6%

Ordering Schemes

- 4 schemes
  - Nested dissection-MD (ND-MD), e.g. onmetis
  - Nested dissection-CMD (ND-CMD), e.g., BEND
  - Multisection-MD (MS-MD)
  - Multisection-CMD (MS-CMD), e.g. SMOOTH

- oPaToH-ND is HP-based ND-CMD
- oPaToH-MS is HP-based MS-CMD



Performance of Ordering Methods wrt MMD

name	onmetis	SMOOTH	2-Clique oPaToH MS	2-Clique oPaToH ND	oPaToH using A MS	oPaToH using A ND
Operation counts relative to MMD						
LP	0.66	0.95	0.65	0.64	0.59	0.55
non-LP	0.79	0.73	0.75	0.78	-	-
Nonzero counts relative to MMD						
LP	0.86	1.02	0.83	0.83	0.80	0.78
non-LP	0.92	0.88	0.89	0.90	-	-
Runtimes relative to MMD						
LP	0.29	1.82	3.96	3.53	1.43	1.19
non-LP	0.95	4.71	7.00	7.02	-	-

Conclusion

- Hypergrah-partioning-based nested dissection ordering
- 17% - 43% better orderings of matrices arising from LP
- Comparable orderings of general matrices
- Finding 3- and 4-cliques for general matrices?

End