# Hypergraph-Partitioning-Based Sparse Matrix Ordering

Umit Catalyurek† and Cevdet Aykanat‡

‡ Department of Biomedical Informatics, The Ohio State University
‡ Department of Computer Engineering, Bilkent University

**Introduction**

In this work we propose novel sparse matrix ordering approaches based on hypergraph partitioning. The significance of hypergraph-partitioning-based (HP-based) ordering is three-fold. First, almost all of the successful nested dissection [6] tools [7, 9, 10] are based on multilevel graph partitioning tools [7, 8, 10] with some extra initial partitioning and refinement strategies specific to the solution of the Graph Partitioning by Vertex Separator (GPVS) problem. However, GPVS-based multilevel ordering has a flaw as will be discussed in the next section. Second, direct solutions of the systems in the form of $ADA^Tx = b$ requires factorization of $ADA^T$, where $A$ is a sparse and possibly rectangular matrix, and $D$ is a diagonal matrix. Here we present an approach for formulating GPVS problem as a Hypergraph Partitioning (HP) problem. Using that formulation coupled with hypergraphs ability to model unsymmetric matrices [4, 5], we propose a new method for finding a fill-reducing ordering of $ADA^T$ by finding a nested dissection of unsymmetric and possibly rectangular matrix $A$. Third, we generalize the proposed method to order any symmetric matrices.

**Flaw of Multilevel GPVS-Based Orderings**

The multilevel approach has recently proved to be successful in both graph and hypergraph partitioning problems. A multilevel partitioning tool consist of three phases; coarsening, initial partitioning and uncoarsening. In the coarsening phase, various vertex clustering heuristics are used at each level, starting from the original graph/hypergraph, to reduce the original partitioning problem down to a series of smaller partitioning problems. Here, clustering corresponds to coalescing highly interacting nodes to the supernodes of the next level. In the second phase, a partition is obtained on the coarsest graph/hypergraph using various heuristics. In the third phase, the partition found in the second phase is successively projected back towards the original problem by refining the projected partitions on the intermediate level uncoarser graph/hypergraphs using various heuristics. In both graph partitioning by edge separator and hypergraph partitioning problems, we have the nice property that a partition of the coarse graph/hypergraph with a valid and narrow edge/hyperedge separator induces a valid and narrow separator of equal cutsize on the original graph/hypergraph. Unfortunately, vertex clustering methods used in multilevel GPVS algorithms and tools lack this important property.

**Describing GPVS Problem as a HP Problem**

Consider a hypergraph $\mathcal{H} = (\mathcal{U}, \mathcal{N})$ and its net-intersection graph (NIG) [1] $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A 2-way vertex partition $\Pi_{HP} = \{\mathcal{U}_1, \mathcal{U}_2\}$ of $\mathcal{H}$ can be decoded as 3-way net partitioning $\Pi_{HP} = \{\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_S\}$ of $\mathcal{H}$ as follows. $\mathcal{N}_1$ and $\mathcal{N}_2$ correspond to the internal nets of part $\mathcal{U}_1$ and $\mathcal{U}_2$, respectively. $\mathcal{N}_S$ corresponds to the external nets. Here, we consider net-partition $\Pi_{HP} = \{\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_S\}$ of $\mathcal{H}$ as inducing a GPVS $\Pi_{GPVS} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_S\}$ on its NIG representation $\mathcal{G}$, where $\mathcal{V}_1 \equiv \mathcal{N}_1$, $\mathcal{V}_2 \equiv \mathcal{N}_2$, $\mathcal{V}_S \equiv \mathcal{N}_S$. Let $Adj_{\mathcal{H}}(n_i)$ denote the set of nets that share vertices with net $n_i$. Consider an internal net $n_i$ of part $\mathcal{U}_1$,i.e., $n_i \in \mathcal{U}_1$. It is clear that we have either $Adj_{\mathcal{H}}(n_i) \subseteq \mathcal{N}_1$ or $Adj_{\mathcal{H}}(n_i) \subseteq \mathcal{N}_1 \cup \mathcal{N}_S$. Recall that NIG $\mathcal{G}$ contains a vertex $v_i$ for each net $n_i$ of $\mathcal{H}$. So we have either $Adj_{\mathcal{G}}(v_i) \subseteq \mathcal{V}_1$ or $Adj_{\mathcal{G}}(v_i) \subseteq \mathcal{V}_1 \cup \mathcal{V}_S$ in NIG $\mathcal{G}$. In other words, $Adj_{\mathcal{G}}(v_i) \cap \mathcal{V}_2 = \emptyset$. In the respective $\Pi_{GPVS}$, this corresponds to $Adj_{\mathcal{G}}(\mathcal{V}_1) \cap V_2 = Adj_{\mathcal{G}}(V_2) \cap \mathcal{V}_1 = \emptyset$ which in turn corresponds to $Adj_{\mathcal{G}}(V_1) \subseteq \mathcal{V}_S$ and $Adj_{\mathcal{G}}(V_2) \subseteq \mathcal{V}_S$. Thus, $\mathcal{V}_S$ of $\Pi_{GPVS}$ constitutes a valid separator of size $|\mathcal{V}_S| = |\mathcal{N}_S|$. Recall that in the GPVS problem, balancing is defined on the vertex counts of parts $\mathcal{V}_1$ and $\mathcal{V}_2$. Hence, the GPVS problem on NIG $\mathcal{G}$ can be described as an HP problem according to the net-cut metric with balancing on the internal nets of parts $\mathcal{U}_1$ and $\mathcal{U}_2$.

From a matrix theoretical point of view, let $A$ be a matrix and $\mathcal{H}$ be its row-net hypergraph representation [5], the NIG $\mathcal{G}$ would be the standard graph representation of matrix $AA^T$. Hence, finding a doubly-bordered form of matrix $AA^T$ (finding GPVS on $\mathcal{G}$) is equivalent to finding a singly-bordered form of matrix $A$ (finding a net partition on $\mathcal{H}$). Although this finding looks very impressive, it is not very useful on itself. For a general GPVS problem on $\mathcal{G}$, which is equivalent to finding a doubly-bordered form of associated matrix (say $Z$) of $\mathcal{G}$, we should know a suitable decomposition of matrix $Z$ into $AA^T$, where decomposition refers to the requirement that $Z$ and $AA^T$ have the same sparsity patterns.

**Nested Disection Ordering of Matrices of the form $ADA^T$**

The solution of normal equations that arise in interior point methods for linear programming requires the factorization of coefficient matrices of the form $ADA^T$, where $A$ is a rectangular constraint matrix and $D$ is a diagonal matrix. Here, we propose a HP-based nested dissection ordering for the ordering of matrix $Z = ADA^T$. Nested dissection ordering requires finding a doubly-bordered (DB) form of the matrix. In DB form, borders correspond to separator $S$, and block-diagonals correspond to $X$ and $Y$ parts of nested dissection as mentioned earlier. Nested dissection simply orders rows/columns of $S$ after the rows/columns of $X$ and $Y$. Together with the formulation of GPVS problem as an HP problem, described in the previous section, we can construct an ordering of $Z$ by just recursively dissecting $A$. That is, in each bisection of $A$ cutnets in $\mathcal{N}_S$ correspond to separator vertices in $S$ in the nested dissection.

**Two-Clique Decomposition for General Nested Disection Ordering**

Previous discussion relies on the assumption that there exists a decomposition of $Z$ into $AA^T$ such that nonzero patterns of $Z$ and $AA^T$ are the same. However, in most of the applications this is not the case, that is, $A$ is usually unknown. Here, we propose a simple yet effective decomposition of symmetric matrices for HP-based nested dissection. Let $\mathcal{G}$ be the standard graph model representation of matrix $Z$. Our aim is to find a matrix $A$ such that $Z$ and $AA^T$ have the same sparsity patterns. In graph theoretical view, we are trying to find a hypergraph $\mathcal{H}$ such that its NIG is $\mathcal{G}$. Obviously the net set of the target hypergraph $\mathcal{H}$ is already identified by the definition of NIG. That is, there must be a net $n_i$ in hypergraph $\mathcal{H}$ corresponding to each vertex $v_i$ in $\mathcal{G}$. The node set of $\mathcal{H}$ is defined as follows. There is a node $u_{ij}$ in $\mathcal{H}$ corresponding to edge $e_{ij} \in \mathcal{E}$ with the net list $nets[u_{ij}] = \{n_i, n_j\}$. During the construction of NIG $\mathcal{G}$ from a hypergraph $\mathcal{H}$, each node of $\mathcal{H}$ induces a clique among the vertices of $\mathcal{G}$ that correspond to nets incident to that node in $\mathcal{H}$. It is clear that, with the proposed decomposition, each node of $\mathcal{H}$ induces distinct 2-cliques, therefore the proposed decomposition is referred to here as *2-clique* decomposition.

In matrix theoretical view, matrix $A$ is the edge-incidence matrix of NIG $\mathcal{G}$. That is, each row of matrix $A$ corresponds to a vertex in $\mathcal{G}$. Each column of matrix $A$ corresponds to an edge in $\mathcal{G}$ such that there are exactly two nonzeros in each column representing the two end points of the edge. Note that the hypergraph mentioned in the previous paragraph is the row-net representation of the edge-incident matrix $A$.

**Results and Conclusion**

We have implemented an HP-based nested disection ordering algorithm. The ordering algorithm leverages Ashcraft and Liu's works [3, 11]. We first do nested disections to construct decoupled block-diagonal submatrices. All of those submatrices are ordered first with constrained minimum degree and then all separators are ordered together. Our results show that, in general matrices hypergraph partitioning-based ordering produces comparable orderings with the state of the art ordering tools, whereas for ordering matrices of the form $Z = ADA^T$, it achieves 17% and 43% better orderings than onmetis [10] and SMOOTH [2] in terms of operation count.

# References


[1] C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: A survey. *VLSI Journal*, 19(1–2):1–81, 1995.

[2] C. Ashcraft and J. W. H. Liu. *SMOOTH: A software package for ordering sparse matrices*, 1996.

[3] C. Ashcraft and J. W. H. Liu. Robust ordering of sparse matrices using multisection. *SIAM Journal on Matrix Analysis and Applications*, 19(3):816–832, 1998.

[4] C. Aykanat, A. Pınar, and U. V. Çatalyürek. Permuting sparse rectangular matrices into block-diagonal form. *SIAM Journal on Scientific Computing*, 26(6):1860–1879, 2004.

[5] U. V. Çatalyürek and C. Aykanat. Hypergraph-partitioning based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7):673–693, 1999.

[6] J. A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10:345–363, 1973.

[7] A. Gupta. Watson graph partitioning package. Tech. Rep. RC 20453, IBM T. J. Watson Research Center, 1996.

[8] B. Hendrickson and R. Leland. *The Chaco user's guide, version 2.0.* Sandia National Laboratories, 1995.

[9] B. Hendrickson and E. Rothberg. Effective sparse matrix ordering: just around the BEND. In *Proc. Eighth SIAM Conf. Parallel Processing for Scientific Computing*, 1997.

[10] G. Karypis and V. Kumar. *MeTiS A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices Version 3.0.* University of Minnesota, 1998.

[11] J. W. H. Liu. On the minimum degree ordering with constraints. *SIAM J. Sci. Statist. Comput.*, 10:1136–1145, 1989.