A Partitioning Algorithm for Block-Diagonal Matrices with Overlap

Guy Antoine Atenekeng Kahou* Laura Grigori † Masha Sosonkina‡

June 15, 2006

Abstract

We present a graph partitioning algorithm that aims at partitioning a sparse matrix into a block-diagonal form, such that any two consecutive blocks overlap. We denote this form of the matrix as the overlapped block-diagonal matrix. The partitioned matrix is suitable for applying the explicit formulation of Multiplicative Schwarz preconditioner (EFMS) described in [3].

The graph partitioning algorithm partitions the graph of the input matrix into K partitions, such that every partition Ω_i has at most two neighbors Ω_{i-1} and Ω_{i+1}. First, an ordering algorithm, such as the reverse Cuthill-McKee algorithm, that reduces the matrix profile is performed. An initial overlapped block-diagonal partition is obtained from the profile of the matrix. An iterative strategy is then used to further refine the partitioning by allowing nodes to be transfered between neighboring partitions. Experiments are performed on matrices arising from real-world applications to show the feasibility and usefulness of this approach.

Problem Consider the sparse matrix $A \in \mathbb{R}^{n\times n}$. The pattern of A is the set $\mathbb{P} = \{(k,l) \mid a_{kl} \neq 0\}$ which is the set of edges of the graph $G = (W, \mathbb{P})$, where W is the set of nodes and P is a set of edges. A two-neighboring graph partitioning of graph G into K partitions is defined by the sets of node $W_i \subset W,\ \ i = 1, ..., K$ such that

$$|i-j| > 1 \Longrightarrow \nexists (k,l) \in W_i \times W_j,\ a_{kl} \neq 0 \vee a_{lk} \neq 0. \tag{1}$$

This work is motivated by the recent results obtained for the explicit formulation of Multiplicative Schwarz preconditioner (EFMS) [3]. It is shown in [3] that to apply EFMS, the graph of the input matrix has to be partitioned into K partitions, such that every partition Ω_i has at most two neighbors Ω_{i-1} and Ω_{i+1}. Remark that the additional constraint introduced in Equation 1 cannot be obtained by using traditional partitioning algorithms as the ones implemented in METIS [2] or PaToH [4].

Description The partitioning algorithm is divided into three steps. In the first step, an ordering algorithm that aims at reducing the profile of the matrix (e.g., the reverse Cuthill-McKee algorithm) is used. During this step, a tree structure T is also built. This tree structure can be obtained from a breadth-first traversal of the graph of A. The number of levels of the tree T influences the quality of the partitioning: the bigger the number of levels, the better the partitioning.

Hence, the two-neighboring graph partitioning problem (Equation 1) may be seen as the partitioning of a chain. Using this observation and the work presented in [1], in the second step we obtain an initial partitioning of the graph of A into K partitions. Note that, if the number of levels in the tree is lower than the number of partitions K, then partitioning into K sub-graphs is not possible.

*Department of Computer Science, University of Yaounde 1, PO Box 812 Yaounde, Cameroon, E-mail: gaatenek@irisa.fr.

†INRIA, FUTUR, E-mail: lgrigori@inria.fr.

‡Ames Laboratory, Iowa State University, Ames, IA 50011 USA, E-mail: masha@scl.ameslab.gov.

In the third step, we perform a refinement of the partitions. The refinement consists of allowing certain nodes to move from one partition to its neighbors, when several conditions are satisfied. Let v be a node of the graph G. We define $cost(v)$ to be the cost of moving v from the current partition to the neighboring one, say, Ω_{i+1}. In particular, $cost(v)$ equals to the difference between the degree of v and the sum of neighbors of v that are also the neighbors of the vertices in Ω_{i+1}. If $cost(v) < 0$, then we move node v to a neighboring partition. The iterative process stops when there is no node to move or if a preset maximum number of iterations was reached.

Experimental results Several experimental results were performed on real world matrices. After defining a comparison criterion with traditional tools, such as METIS or PaToH, we illustrate the effectiveness of our algorithm.

References

[1] A.Pinar and C.Aykanat. Fast optimal load balancing algorithms for 1d partitioning. *Elsevier JPDC.*, 64:974–996, 2004.

[2] G.Karypis and V.Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Elsevier JPDC.*, 48(1):96–126, 1998.

[3] K.Atenekeng, E.Kamgnia, and B.Philippe. An explicit formulation of the multiplicative schwarz preconditionner. *INRIA.*, 14, 2005.

[4] U.atalyrek and C.Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7):974–996, July 1999.