Graph Models and their Efficient Implementation for Sparse Jacobian Matrix Determination ⋆

Shahadat Hossain [a,*] and Trond Steihaug [b]

[a] *Department of Mathematics and Computer Science, University of Lethbridge, Canada*

[b] *Department of Informatics, University of Bergen, Norway*

Abstract

Algorithms for solving large-scale combinatorial scientific computing problems arising in sparse or otherwise structured matrix computation are often represented by appropriate graph models and sometimes the same problem can be formulated in more than one graph models with similar asymptotic computational complexity. The relative merits of different graph models for the same problem can then be expressed in terms of factors such as generality of the model and ease of computer implementation. In this note we briefly review the contemporary graph formulations for large-scale sparse Jacobian matrix determination problem (JMDP) and suggest the pattern graph model which can be viewed as a unifying framework for the unidirectional and bidirectional approaches for JMDP. Due to the irregular memory access pattern combined with low floating point calculations relative to the volume of data movements the actual running time of sparse matrix and graph algorithms may achieve only a small fraction of the theoretical performance. We proffer the use of array-based data structures as the basic building-blocks for efficient implementation of fundamental graph algorithms on modern cache-based computer architectures. Numerical results comparing our implementation (DSJM toolkit) with ColPack [4] is given.

Key words: Sparse Matrix Data Structures, Intersection Graph, Bipartite Graph, Hypergraph, Pattern Graph

1 Introduction

We consider the problem of determining the Jacobian matrix $F'(x)$ of a mapping $F : \Re^n \mapsto \Re^m$. In this paper graphs are undirected. The colon notation of [6] is used to specify sections of a matrix. The (i, j)th entry is denoted by

⋆ This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Research Council of Norway (NFR).

* Corresponding author.

$A(i,j)$ or a_{ij}. When a matrix is displayed the nonzero entries are explicitly shown while a blank or a '0' marks a zero entry. We assume that the sparsity pattern of matrix $F'(x)$ is known a priori and that it is computationally more economical to compute the entire column of $F'(x)$ than computing individual entries. Using differences the jth column of the Jacobian matrix may be approximated as

$$\left.\frac{\partial F(x+ts)}{\partial t}\right|_{t=0} = F'(x)s \approx As = \frac{1}{\varepsilon}[F(x+\varepsilon s) - F(x)] \equiv b \tag{1}$$

with one extra function evaluation. Also Algorithmic Differentiation (AD) [7] forward mode gives $b = F'(x)s$ at a cost which is a small multiple of the cost of one function evaluation. The Jacobian matrix determination problem can be stated as below.

Problem 1 (JMDP) *Obtain vectors $s_i \in \Re^n, i = 1, \ldots, p$ and $w_j \in \Re^m, j = 1, \ldots, q$ with $p+q$ minimized such that the products ($b_k = As_k,\ j = 1, \cdots, p$ or $B = AS$) and ($c_k^T = w_k^T A,\ k = 1, \cdots, q$ or $C^T = W^T A$) determine the matrix A uniquely.*

In absence of any sparsity information, one may use the Cartesian basis vectors $e_i, i = 1, \ldots, n$ in (1) using n extra function evaluations. However, if the columns are structurally orthogonal i.e. no two columns have nonzero entries in the same row position only one extra function evaluation

$$F'_j + F'_k \approx A(:,j) + A(:,k) = \tfrac{1}{\varepsilon}[F(x + \varepsilon(e_j + e_k)) - F(x)]$$

is sufficient to read off the nonzero entries from the product $b = As$. If the columns can be partitioned into p structurally orthogonal groups then the Jacobian matrix is *directly determined* from the compressed matrix $B = AS$. Similarly, the rows can be partitioned into q structurally orthogonal groups and the Jacobian can be directly determined from $C^T = W^T A$ using the reverse mode of AD. The problem of finding minimum cardinality orthogonal column (or row) partition of matrix A can be solved as different vertex coloring problems (which in general are NP-hard and therefore are solved by heuristics) of suitable graph(s) associated with A.

2 Graph Models and Computer Implementation

With regard to the choice of the graph model for the partitioning problem we expect that the graph formulation

(1) retains exploitable matrix structures,
(2) enables efficient implementation of pertinent algorithms, and
(3) generic enough to encapsulate the combined row-and-column determination as in Problem JMDP.

The *intersection graph* of the columns of A is denoted by $G(A) = (V, E)$ where corresponding to $A(:, j), j = 1, 2, \ldots, n$, there is a vertex $v_j \in V$ and $\{v_j, v_l\} \in E$ if and only if $A(:, j)$ and $A(:, l), l \neq j$ have nonzero elements in the same row position. Then an orthogonal partition of the columns of A is equivalent to a coloring ϕ of the vertices of $G(A)$ such that $\phi(u) \neq \phi(v)$ if and only if $\{u, v\} \in E$ [1]. Unfortunately, the column intersection graph of A or A^T is unable to expose all the exploitable matrix sparsity. Alternative formulations define graphs based on column segments to allow for better utilization of available structure or sparsity [8]. However, these alternative formulations apply to either column direction or row direction but not to a combination of row and column directions – henceforth *bidirectional determination*.

The *bipartite graph* associated with matrix A is denoted by $G_b(A) = (V_c \cup V_r, E)$ where corresponding to $A(:, j), j = 1, 2, \ldots, n$, there is a column vertex $v_j \in V_c$ and corresponding to $A(i, :), i = 1, 2, \ldots, m$, there is a row vertex $v_i \in V_r$ and $\{v_i, v_j\} \in E$ if and only if $a_{ij} \neq 0, i = 1, 2, \ldots, m, j = 1, 2, \ldots, m$. The bipartite graph model has been proposed independently by Coleman and Verma [2] and Hossain and Steihaug [9] in connection with bidirectional determination of Jacobian matrices. In [3] the bipartite graph model has been considered for column partitioning. The zero-nonzero structure of the underlying matrix is accurately represented in its bipartite graph making it a natural logical data structure for bidirectional determination. On the other hand, for the unidirectional determination the model is "asymmetrical" in the sense that it contains extraneous information. This difficulty is manifested in [3] where the unidirectional determination posed as distance-2 coloring needed the qualification "partial" as only one set of vertices are colored.

A *hypergraph* is a graph in which edges are generalized as *hyperedges* where a hyperedge may connect more than vertices. The hypergraph model presented here is more general than the ones considered in [3]. The hypergraph $H(A) = (V, E)$ associated with the matrix A has the vertex set

$$V = \{v_i | \exists k \text{ for which } a_{ik} \neq 0, i = 1, 2, \ldots, m\} \cup \{v_j | \exists k \text{ for which } a_{kj} \neq 0, j = 1, 2, \ldots, n, \}$$

and corresponding to each row i and each column j the hyperedges $e_i \in E$ and $e_j \in E$, respectively, are defined by

$$e_i = \{v_k | a_{ik} \neq 0\} \text{ and } e_j = \{v_k | a_{kj} \neq 0\}.$$

A *lateral neighbor* of $a_{ij} \neq 0$ is a nonzero $a_{ij'} \neq 0$ in row i of A such that $j' - j$ is the smallest if $j' > j$ or such that $j - j'$ is the smallest if $j > j'$ among all such indices j' in row i. Vertical neighbors can be interpreted in an analogous way with the roles of i and j interchanged. The *sparsity-pattern graph* (or simply the pattern graph) associated with A is $G_{\mathcal{P}}(A) = (V, E)$, where

$$V = \{v_{ij} : a_{ij} \neq 0, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n\}$$

and

$$\{v_{ij}, v_{i'j'}\} \in E \text{ if } a_{ij} \text{ and } a_{i'j'} \text{ are lateral or vertical neighbors.}$$

An unknown a_{ij} is said to be *covered* (by S or W) if it can be uniquely solved in

$$\hat{S}_i^T A(i,\mathcal{J}_i)^T = B(i,:)^T \text{ or } A(\mathcal{I}_j, j)\hat{W}_j = C(:,j).$$

where $\hat{S}_i(\hat{W}_j)$ is the submatrix of $S(W)$ corresponding to the nonzero entries in row i (column j) indicated by $\mathcal{J}_i(\mathcal{I}_j)$. The matrices S and W are said to constitute a cover for A if each $a_{ij} \neq 0$ is covered. A cover is a *direct cover* if A can be determined directly from the cover. Given a mapping $\Phi : V \mapsto \mathcal{S} \cup \mathcal{W}$ where $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_p\}, \mathcal{W} = \{\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_q\}$ we define matrices $S \in \{0,1\}^{n\times p}$ and $W \in \{0,1\}^{m\times q}$,

$$S(:,k) = \sum_j e_j, \mathcal{S}_k = \Phi(v_{ij}) \text{ and } W(:,l) = \sum_i e_i, \mathcal{W}_l = \Phi(i).$$

The mapping $\Phi : V \mapsto \mathcal{S} \cup \mathcal{W}$ is said to yield a cover for A if the matrices S and W constitute a cover for A. Denote by $u_{ij} \overset{\geq 1}{\approx} u_{i'j'}$ a path of length at least 1.

Theorem 2 *Let $G_{\mathcal{P}}(A) = (V,E)$ be the pattern graph associated with A. Define the mapping $\Phi : V \mapsto \mathcal{S} \cup \mathcal{W}$ where $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_p\}, \mathcal{W} = \{\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_q\}$ such that for each $v_{ij} \in V$,*

EITHER

(1) (a) $v_{ij} \overset{\geq 1}{\approx} v_{ij'}, j \neq j'$ implies $k \neq k'$ where $\Phi(v_{ij}) = \mathcal{S}_k, \Phi(v_{ij'}) = \mathcal{S}_{k'}$ and
(b) $v_{ij} \overset{\geq 1}{\approx} v_{ij'} \overset{\geq 1}{\approx} v_{i'j'}, i \neq i', j \neq j'$ implies $k \neq k'$ where $\Phi(v_{ij}) = \mathcal{S}_k, \Phi(v_{i'j'}) = \mathcal{S}_{k'}$

OR

(2) (a) $v_{ij} \overset{\geq 1}{\approx} v_{i'j}, i \neq i'$ implies $k \neq k'$ where $\Phi(v_{ij}) = \mathcal{W}_k, \Phi(v_{i'j}) = \mathcal{W}_{k'}$ and
(b) $v_{ij} \overset{\geq 1}{\approx} v_{i'j} \overset{\geq 1}{\approx} v_{i'j'}, i \neq i', j \neq j'$ implies $k \neq k'$ where $\Phi(v_{ij}) = \mathcal{W}_k, \Phi(v_{i'j'}) = \mathcal{W}_{k'}$.

Then matrices S and W constitute a direct cover for A.

One of the strengths of the pattern graph model is that the result given above can be specialized to unidirectional and column segments determinations without changing the graph.

The sparsity pattern of matrix A can be efficiently represented using two arrays: array `colind` that stores the column indices of the nonzero entries row-by-row, and array `rowptr` that contains the index of the first nonzero element of each row of the sparse matrix stored in `colind` array. For easy access to the adjacent vertices sparsity pattern of A^T is explicitly stored using analogous arrays `rowind` and `colptr` . This storage is of order $\Theta(\max(n, m, nnz))$ which meets the design strategies for sparse linear algebra implementation [5]. Next we consider the computations on this data structure relevant to algorithms for Problem JMDP. In many graph coloring heuristics on static graphs a frequently executed operation is to find the neighbors of a given vertex v_j.

Assuming v_j a column vertex this information is obtained easily as

$$\{\{v_j, v_i\} | i = \text{ rowind(k), k = colptr(j) : colptr(j+1)-1 } \}.$$

This computation can be performed independent of the graph models discussed above. Further, the array representation ensures better cache performance and the computational cost of the operation is proportional to the size of the data accessed and the number of nonzero arithmetic operations. Clearly, it is not necessary to compute the intersection graphs explicitly as advocated in [3]. Indeed, almost all the well-known coloring heuristics for JMDP can be implemented in time proportional to $\sum_{i=1}^{m} \rho_i^2$ where ρ_i denotes the number of nonzero entries in row i of A. As an indication of the efficiency of the data structure proposed here we report in the table below the timing experiments for incidence degree order (IDO) coloring where `ot` and `ct` represent ordering and coloring times, respectively. `ColPack` implements IDO coloring with partial distance-2 coloring scheme on bipartite graph while `DSJM` implements IDO coloring using the data structure described here using column intersection graph. The times reported here do not include data structure set up times. The sparse matrices are obtained from University of Florida Sparse Matrix Collection.

Matrix	m	n	nnz	ColPack [4]		DSJM	
				ot	ct	ot	ct
lpcreb	9648	77137	260785	13.9	1.49	2.46	1
lpcred	8926	73948	246614	14.79	1.48	2.46	1.01
lpfit2d	25	10524	129042	64.19	24.94	16.32	17.2
lpken18	105127	154699	358171	15.92	0.63	1.47	0.48
lposa07	1118	25067	144812	161.76	28.34	40.84	18.35

References

[1] T. F. Coleman and J. J. Moré. Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM J. Numer. Anal.*, 20(1):187–209, 1983.

[2] T. F. Coleman and A. Verma. The efficient computation of sparse Jacobian matrices using automatic differentiation. *SIAM J. Sci. Comput.*, 19(4):1210–1233, 1998.

[3] A. H. Gebremedhin, F. Manne, and A. Pothen. What Color Is Your Jacobian? Graph Coloring For Computing Derivatives. *SIAM Review*, 47(4):629–705.

[4] A. H. Gebremedhin, A. Tarafdar, D. Nguyen, and A. Pothen. ColPack. http://www.cs.odu.edu/~dnguyen/dox/colpack/html/ (accessed May 2009)

[5] J. R. Gilbert, S. Reinhardt, and V. Shah. High-performance graph algorithms from parallel sparse matrices. In B. Kågström, E. Elmroth, J. Dongarra, and J. Wasniewski, editors, *PARA*, volume 4699 of *Lecture Notes in Computer Science*, pages 260–269. Springer, 2006.

[6] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[7] A. Griewank, Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, Penn., 2008.

[8] Shahadat Hossain and Trond Steihaug. Graph coloring in the estimation of sparse derivative matrices: Instances and applications, *Discrete Applied Mathematics*, 156(2):280–288, 2008.

[9] A. S. Hossain and T. Steihaug. Computing a sparse Jacobian matrix by rows and columns. *Optimization Methods and Software*, 10:33–48, 1998.