Fast Implementation of the Minimum Local Fill Ordering Heuristic

Esmond G. Ng and Barry W. Peyton

Abstract

It is well known that sparse matrix factorizations suffer fill. That is, some of the zero entries in a sparse matrix will become nonzero during factorization. To reduce factorization time and storage, it is important to arrange the computation so that the amount of fill is kept small. It is also well known that the amount of fill is often influenced greatly by how the rows and columns of the sparse matrix are permuted (or ordered). We focus on the Cholesky factorization of a sparse symmetric positive definite matrix, in which case the fill depends solely on the sparsity of the given matrix and on the choice of the permutation (or ordering).

We use the following notation. Let A be an $n \times n$ symmetric positive definite matrix and P be an $n \times n$ permutation matrix. Let m be the number of nonzero entries in the lower triangle of A. We denote the factorization of PAP^T by LL^T, where L is lower triangular. We use L_{*k} to denote the k-th column of L. The number of nonzero entries in a vector v is represented by $\mathrm{nnz}(v)$.

It is well known that finding an ordering to minimize fill in sparse Cholesky factorization is an NP-complete problem [7]. Thus, we rely on heuristics. Two greedy, bottom-up heuristics are the minimum degree (MD) algorithm and the minimum local fill (MF) algorithm. The MD algorithm, introduced by Tinney and Walker [5], is probably the best-known and most widely-used greedy heuristic. It reduces fill by finding a permutation P so that $\mathrm{nnz}(L_{*k})$ is minimized locally at step k of the factorization. A great deal of work has been done to reduce the runtime of the MD algorithm. The MF algorithm, also introduced by Tinney and Walker [5], is not as well known or widely used as MD. In the MF algorithm, the permutation is chosen so that the number of zero entries in the reduced matrix that become nonzero is as small as possible at each step of the factorization.

There are a number of reasons why the MF algorithm has not been as popular as the MD algorithm. The metric $\mathrm{nnz}(L_{*k})$ is easy and inexpensive to compute. By contrast, the metric required by the MF algorithm is more difficult and expensive to compute. Consequently, the general experience has been that the MF algorithm requires far greater time than the MD algorithm. For example, Rothberg and Eisenstat [4] report that "while many of the enhancements described above for minimum degree are applicable to minimum local fill (particularly supernodes), runtimes are still prohibitive". Also, Ng and Raghavan [3] report that their implementation of MF was on average slower than MD by "two orders of magnitude".

Another reason for the lack of popularity of the MF algorithm is the belief that MF orderings are often just marginally better than MD orderings [1]. It has been shown, however, that MF orderings are often considerably better than MD orderings. For example, in an early version of [3], Ng and Raghavan reported that their MF orderings, on average, resulted in 9% less fill and 21% fewer operations than their MD orderings. On a different set of test matrices, Rothberg and Eisenstat [4] similarly reported that their MF orderings, on average, resulted in 16% less fill and 31% fewer operations than their MD orderings. Consequently, a truly efficient way to compute MF orderings would prove valuable, and that is the focus of our talk.

The reason for the high runtimes of standard implementations of MF is that whenever a column's fill count may have changed, it is set to zero and recomputed from scratch. In [6], Wing and Huang described an elegant way to *update* the deficiencies rather than recomputing them from scratch. Their updating scheme was mentioned a few times in the circuit simulation literature, but it apparently was not widely used and it certainly was not adopted by the sparse matrix community.

We will describe in this talk our recent work on the Wing-Huang updating scheme. In particular, we will show that the worst-case time complexity of the MF algorithm with Wing-Huang updates is the *same* as that of the MD algorithm, namely $O(n^2m)$. We will also demonstrate that techniques for reducing the runtime of the MD algorithm, such as mass elimination and indistinguishable nodes, are equally applicable in the efficient implementation of the MF algorithm with Wing-Huang updates. It is particularly important that we can adapt the Wing-Huang updating technique so that it can be used efficiently when quotient graphs are used to represent the elimination graphs.

Results from our preliminary implementation of the MF algorithm with Wing-Huang updates are encouraging. Over a collection of 48 sparse matrices from the Florida Sparse Matrix Collection, our MF algorithm with Wing-Huang updates is just 4.6 times more expensive than the minimum degree (MMD) algorithm with multiple eliminations [2] on average. Our MF orderings, on the average, produce 17% less fill and require 31% fewer operations than the MMD algorithm. On one large test matrix (3dtube), MF produces 29% less fill and requires 55% fewer operations.

In the future, we hope to look into ways to further reduce runtimes for our implementation of the MF algorithm using Wing-Huang updating.

References

[1] Iain S. Duff, Albert M. Erisman, and John K. Reid. *Direct Methods for Sparse Matrices.* Oxford University Press, New York, NY, 2nd edition, 1989.

[2] Joseph W.H. Liu. Modification of the minimum-degree algorithm by multiple elimination. *ACM Trans. Math. Software*, 11(2):141–153, 1985.

[3] Esmond G. Ng and Padma Raghavan. Performance of greedy ordering heuristics for sparse Cholesky factorization. *SIAM J. Matrix Anal. Appl.*, 20(4):902–914, 1999.

[4] Edward Rothberg and Stanley C. Eisenstat. Node selection strategies for bottom-up sparse matrix orderings. *SIAM J. Matrix Anal. Appl.*, 19(3):682–695, 1998.

[5] W.F. Tinney and J.W. Walker. Direct solution of sparse network equations by optimally ordered triangular factorization. In *Proc. IEEE*, volume 55, pages 1801–1809, 1967.

[6] O. Wing and J. Huang. SCAP - a sparse matrix circuit analysis program. In *Proceedings - IEEE International Symposium on Circuits and Systems*, pages 213–215, 1975.

[7] Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2(1):77–79, 1981.