行政院國家科學委員會專題研究計畫成果報告

分散式稀疏矩陣 QR 分解之研究
The Study of Reordering Problem for Sparse QR Factorization

計畫編號：NSC 87-2213-E-002-005
執行期限：86 **年** 08 **月** 01 **日至** 88 **年** 07 **月** 31 **日**
主持人：陳俊良　　國立台灣大學資訊工程學系
clchen@csie.ntu.edu.tw

一、中文摘要

本研究探討的是稀疏矩陣 QR **多鋒面分解的問題。我們提出了一種方法來計算分解各鋒面矩陣時的運算個數。這些數字可以被視為相對應消去樹節點的加權。當我們要發展好的稀疏矩陣** QR **分解程式時，這些資料十分的有用。比如，利用此一資料，**QR **分解的平行執行時間大約可縮短** 10%。

關鍵詞：數值線性代數、稀疏矩陣、QR **分解、多鋒面方法、最小平方和問題**

Abstract

This research considers the sparse multifrontal QR factorization. An efficient method to evaluate numbers of multiplicative operations in factorizing each frontal matrix is proposed. These numbers can be treated as the node costs of the corresponding elimination tree. This knowledge is very useful to improve performance of sparse QR factorization. For example, experiments conducted so far show that about 10\% of the parallel execution time can be reduced.

Keywords: numerical linear algebra, sparse matrix, QR factorizatiion, multifrontal method, least-squares problem

二、計畫緣由與目的

For a matrix A, there is an orthogonal matrix Q such that

$$A = QR$$

where R is upper triangular. This is the QR factorization of A, and the matrix R will be called the R-factor of A. QR factorization is a very useful process to solve many numerical linear algebra problems, e.g., the least-squares problem. In real application, the matrix A is usually large and sparse.

Usually, we use elimination tree to represent the process of sparse QR factorization. A node represents a task. The workload of each task is different. For example, Figure 1(d) is an elimination tree with different costs. The information about node costs is very useful. For example, consider Figure 1(d) and assume that there are 2 processors. The best task allocation strategy is to assign tasks 1, 2, 3 and 4 to one processor and to assign task 5 to the other.

The goal of this research is to design an efficient method to evaluate node costs of the elimination tree associated with a sparse multifrontal QR factorization.

三、結果與討論

(1) Background

Considering the matrix A shown in Figure 1(a), the corresponding A^TA, the fill-in graph and the elimination tree are shown in Figures 1(b), 1(c) and 1(d), respectively. The frontal matrices, the associated Householder transformations and the update matrices are drawn in Figure 2. Here, $H(m, n)$ represents a Householder transformation for an m by n dense matrix. It can be proved that an $H(m, n)$ needs $(2mn + m + n)$ multiplicative operations when $m > 2$ and $n > 1$, and none operation otherwise. We can treat this number as the cost of an $H(m, n)$, denoted as $CH(m, n)$. According to Figure 2, we can calculate all of the node costs. For example, the cost of node 1 is: $CH(3, 4) + CH(2, 3) = 31 + 17 = 48$. These costs are shown in Figure 1(d).

From the above example, we understand that the key-points to evaluate node costs are the dimensions of all small dense matrices which Householder transformations are applied to. One possible way to collect these data is the profile of the numerical factorization, but the time complexity will be the same as that of numerical factorization. The following definitions, lemmas and theorems are required to establish an efficient cost evaluation method.

(2) Cost Evaluation

Consider a matrix A with m rows and n columns, $m \geqq n$.

Definition 1: For matrix A, let l_j^0 be the number of rows whose leading nonzeros are in column j.

Let m_j and n_j be the numbers of rows and columns of F_j, respectively. And let $v_{p(j,1)}$, $v_{p(j,2)}$, …, $v_{p(j,n_j)}$ be the nodes in $\{v_j\} \cup Madj(v_j)$, where $1 \leqq p(j,1) < p(j,2) < \ldots < p(j,n_j) \leqq$ n.

Definition 2: For a frontal matrix F_j, let $m_j^{p(j,i)}$ be the number of rows whose leading nonzeros are in columns $p(j,1), p(j,2), \ldots, p(j,i)$.

Theorem 1: The Cost to factorize frontal matrix F_j *is*

$$CF_j = \sum_{i=1}^{n_j} CH(m_j^{p(j,i)} - i + 1, n_j - i + 1)$$

Definition 3: For an update matrix U_j, let $l_j^{p(j,i)}$ be the number of rows whose leading nonzeros are in columns $p(j,i)$, for $2 \leqq i \leqq n_j$.

Theorem 2: For a node v_j,

$$m_j^{p(j,i)} = \begin{cases} l_0^j + h_j^{p(j,1)} & \text{for } i = 1 \\ m_j^{p(j,i-1)} + h_j^{p(j,i)} & \text{for } 2 \le i \le n_j \end{cases}.$$

Theorem 3: For a node v_j,

$$l_j^{p(j,i)} = \min(m_j^{p(j,i)}, i) - \min(m_j^{p(j,i-1)}, i-1).$$

(3) Evaluation Algorithm

Combining all theorems, Algorithm 2 can evaluate costs of factorizing each frontal matrix, i.e., costs of each node.

(4) Application

Under multiprocessor environment, processor allocation and task scheduling are important issues to achieve high performance. It seems that the knowledge of node costs is useful for QR factorization. Therefore, we do some experiments.

First, we modify Kan's processor allocation and task scheduling algorithm [2] so that it is suitable for QR factorization. Then, we apply the algorithm to several test matrices that are selected from Harwell-Boeing collection, and estimate the parallel execution time. The process contains two parts. In the first part, the knowledge of node costs is not supported; that is, we assume that the costs of all nodes are equivalent. In the second part, the knowledge of node costs is supported. Figure 3 is our experimental result. In Figure 3, the estimated execution time is normalized. We assume that the normalized execution time is 1, if there are infinite number of processors and the system is communication-free. Experiments conducted so far show that about 10\% of the the execution time can be reduced.

(5) Conclusion

We have proposed an efficient method to evaluate node costs of the elimination tree associated to a sparse multifrontal QR factorization. Knowledge of node costs is very useful for studying many related problems, e.g., column reordering, processor allocation and task scheduling of sparse multifrontal QR factorization.

四、計畫成果自評

最後之研究內容與原計畫大致相符。部

分結果已發表於International Conference on Parallel and Distributed Processing Techniques and Applications [1]。

五、參考文獻

[1] D. M. Jiang and C. L.Chen, 1998/07, "Efficient Cost Evaluation for Sparse Multifrontal QR Factorization," *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, Nevada, USA, pp. 1567-1574.

[2] T. T. Kan and C. L. Chen, 1998/12, "Processor Allocation and Task Scheduling for Parallel Sparse Cholesky Factorization," *Proceedings of the Second International Conference on Parallel and Distributed Computing and Networks*, Brisbane, Australia, 200-205.

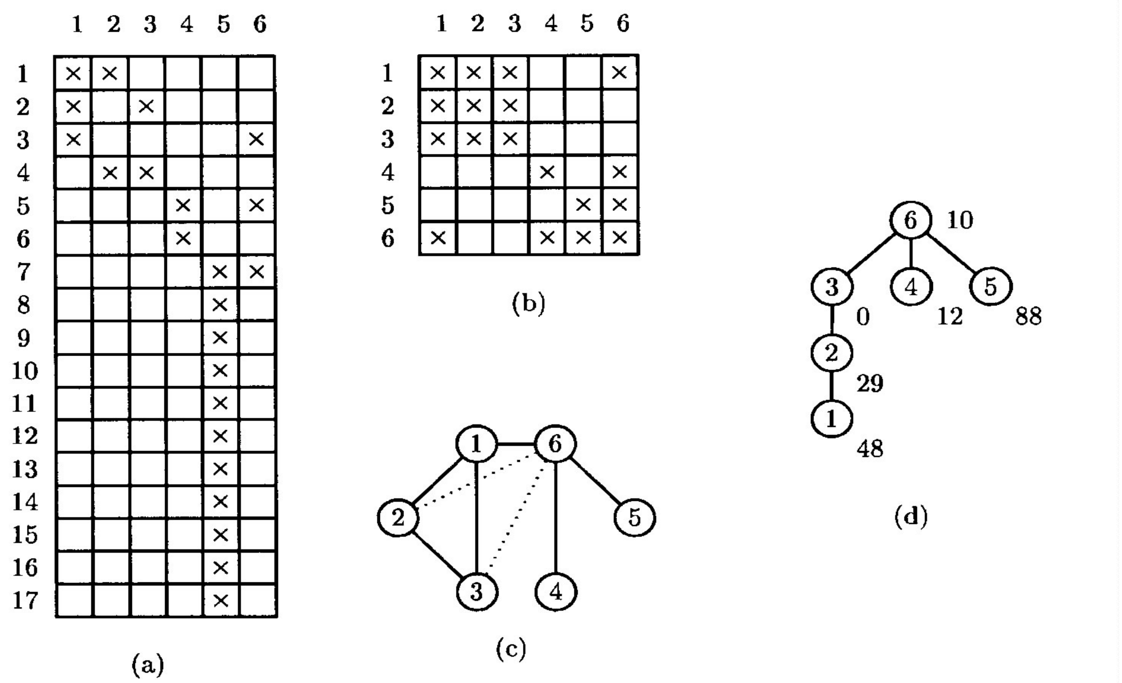

Figure 1.

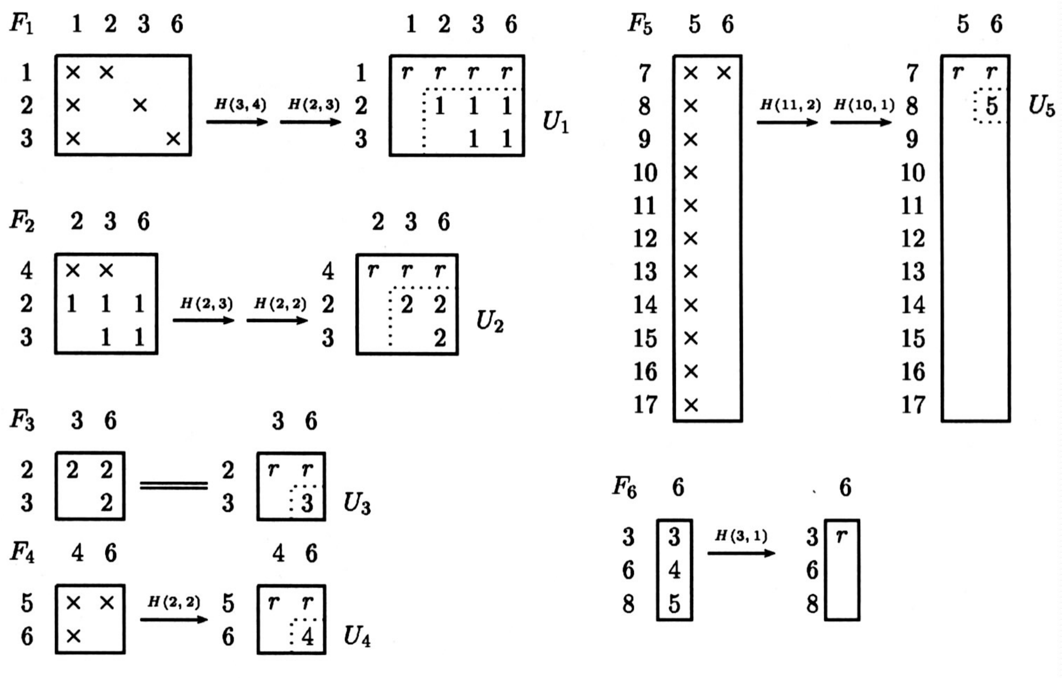

Figure 2.

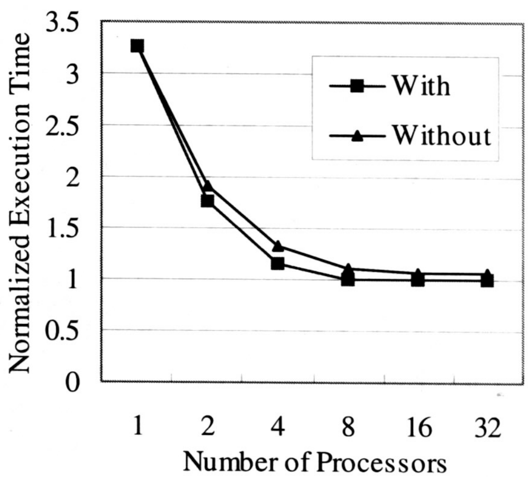

Figure 3.

Algorithm 2: *Cost Evaluation*

Input: $Madj(v_j)$ and l_0^j, for $1 \leq j \leq n$.
Output: CF_j, the cost to factorize frontal matrix F_j, for $1 \leq j \leq n$.
Step1: **for** node v_j, $1 \leq j \leq n$, **do**
  for each $v_k \in \{v_j\} \cup Madj(v_j)$ **do**
    $h_j^k = 0$
Step2: **for** node v_j, $1 \leq j \leq n$, **do**
Step2.1: **for each** $v_k \in \{v_j\} \cup Madj(v_j)$ **do**
    Calculate m_j^k by Theorem 2
Step2.2: Calculate CF_j by Theorem 1
Step2.3: **for each** $v_k \in Madj(v_j)$ **do**
    Calculate l_j^k by Theorem 3
Step2.4: **for each** $v_k \in Madj(v_j)$ **do**
    $h_{p(j,2)}^k = h_{p(j,2)}^k + l_j^k$
