Trends in Mathematics
Information Center for Mathematical Sciences
Volume 9, Number 1, June, 2006, Pages 141–148

COMPOUND ALGORITHM FOR DECREASING OF MATRIX PROFILE SIZE

M.P.LEVIN

Abstract. An analysis of a modern methods for decreasing a sky-line matrix profile for systems of simultaneous linear algebra equations with symmetric matrices and matrices of symmetric structures based on Reverse Cuthill-McKee-Sloan algorithm are considered. For further improving of results obtained by this approach a simple sorting algorithm is proposed. Numerical examples show that a compound technic based on a sequential application of these two algorithms allows to get better results in diminution of matrix profile in real applications.

1. Introduction

Recently a lot of direct methods based on a sky-line matrix profile consideration are widely used for solution of simultaneous linear algebra equations in applications. The Reverse Cuthill-McKee's algorithm was proposed in [1] for decreasing of the profile size for sparse matrices. This algorithm consists in the enumeration of nodes of a graph associated with the considering matrix. It was essentially improved by S.W.Sloan [2] in 1986. Further J.Duff at all [3-6] developed the algorithm and implemented it in various software tools of Harwell Subroutine Library. In 1997, G.Kumfert and A.Pothen [7] investigating the large size problems proposed a new strategy for choosing of nodes from the list of eligible nodes on the second stage of Sloans algorithm. This strategy is the development of the hybrid algorithm proposed in [8]. It combines a spectral ordering and the Cuthill-McKee-Sloan's algorithms.

In this paper we consider one compound algorithm consisting of two stages. The first stage of this algorithm coincides with one modification of Cuthill-McKee-Sloan's algorithm. The second stage consists in construction a permutation by the simple sorting of the permutation vector obtained at the first stage. As our numerical practice shown this strategy can be successful in the enumeration of the large systems of simultaneous linear algebra equations with sparse symmetric matrices and non-symmetric sparse matrices of the symmetric structure.

2000 *Mathematics Subject Classification.* 15A06, 05C30, 05C50.

Key words and phrases. linear equations, enumeration of graphs and maps, graphs and matrices, sky-line matrix profile, profile decreasing, systems of simultaneous linear algebra equations, symmetric matrices, matrices of symmetric structures, reordering algorithms.

2. Background of Cuthill-McKee-Sloan's algorithm

Now let us consider a background of Cuthill-McKee-Sloan's algorithm. In this algorithm the problem of enumeration of equations in the system of simultaneous linear algebra equations reduces to the problem of enumeration of weighted graph nodes. This graph corresponds to the square matrix of the system as follows. Nodes of the graph correspond to the diagonal entries of the matrix. Here we suppose that all diagonal entries of matrix aren't equal to zero. Nodes of the considering graph are connected by ribs. The rib between two nodes with the current numbers i and j exists if following condition is satisfied

$$a_{ij} \neq 0 \bigcap a_{ji} \neq 0,$$

where a_{ij} and a_{ji} are the non-diagonal entries of the considering matrix A.

Investigating the considering problem, at first we remind some terminology from graph theory following to [9]. We call a *degree* of any node n belonging to the considering graph G the number of the nodes that are adjacent to the considering node n. Also we define the *distance* $d(n, m)$ between two nodes n and m on the graph G as the shortest path between these nodes. The *diameter length* $L_D(G)$ of graph G is defined as the greatest distance between two any nodes of graph G. As a *diameter* $D(G)$ of graph G we define a shortest path between two nodes of the graph located on the distance equal to the diameter distance L_D of the graph. A *pseudo-diameter* of the graph is defined either as a diameter or as a shortest path between two nodes on G located on the distance slightly less than the diameter length. The ends of a pseudo-diameter are usually called a *pseudo-peripheral* nodes.

2.1. **Construction of level-set structures.** To find a pseudo-diameter of the graph is possible [9] by constructing and analyzing a *level-set structures* partitioning the graph G into the level-set structures $L_k(n) = \{l_1(n), l_2(n), l_3(n), ..., l_k(n)\}$ with the root at any node n as follows

- $l_1(n) = n$,
- for any $i > 1$, $l_i(n)$ is a set of nodes adjacent to the nodes of $l_{i-1}(n)$ and not included yet into sets the $l_1(n)$, $l_2(n)$, ..., $l_{i-1}(n)$.

The principal idea to solve the considering enumeration problem consists in the enumeration of nodes in the narrowest level-set structure corresponding to the considering graph. The problem of construction such structure can be reduced to the problem of construction the longest level-set structure because it perhaps can be also a narrowest or an almost narrowest. Analyzing the constructed level-set structures we can mention that all nodes those are located at the same level $l_i(n)$ are situated at the same distance $(i - 1)$ from the root node n. The number of levels k in the structure $L_k(n)$ with any root node n of minimal degree is a *depth* of the level-set structure and the biggest number of the nodes at any level is a *width* of the level-set structure. For construction the level-set structure, at first, we need to define the root node. As it is mentioned in [9] the best way to define the root node is to choose it as one of the ends of graph diameter. But on such way the problem of the graph diameter evaluation must be solved before. Unfortunately, very often the problem of evaluation of the graph diameter is an expensive computational problem and it needs a lot of time for solution. The complexity of this problem is almost same as the complexity of the considering problem. Therefore in [9], it was proposed to take one of the pseudo-peripheral nodes as the root node. Also in this paper authors proposed to take the node with minimal degree as the pseudo-peripheral node. Analyzing these propositions we can mention that the problem of the root node choosing usually has not a unique solution because the real complex graphs can have a few of pseudo-peripheral nodes or nodes with minimal degree. Therefore in the ideal case, we must consider all level-set structures starting from various pseudo-peripheral nodes or nodes with minimal degree. Thus if we have not a strong time restrictions for construction of the level-set structures then we can analyze all nodes with minimal degree as the root node and choose one of them that gives the best result.

2.2. **Enumeration of nodes.** Then the level-set structure is chosen we can consider the next problem how to enumerate its nodes. In Cuthill-McKee's algorithm [1] it was proposed to provide enumeration of nodes step by step from high level to low level by sorting nodes located at the same level in increasing order of their power. Here we can note that the power of nodes at the same level is evaluated only with respect to the nodes not enumerated yet. In the more perfect Sloan's algorithm [2], the sorting of nodes at the same level is provided in increasing order of the linear combination of the distance between the current node i and the end node e of the level-set structure and the power of current node $p(i)$ taken with corresponding weights α and β as follows

$$r(i,e) = \alpha d(i,e) + \beta p(i) \ .$$

Here we can mention that the end node e in the second algorithm is evaluated by original Cuthill-McKee's algorithm. The coefficients α and β can be chosen from numerical experiments or following recommendations proposed in [7].

Thus the improved Sloan's algorithm consists of three stages. On the first stage the original Sloan's algorithm works and in result it allows us to find the end node e. On the second stage we again apply the original Sloan's algorithm to construct the level-set structure with start node e. In result we can evaluate the distances between the end node e and any node i for all nodes of the considering structure. On the third stage we again apply the Sloan's algorithm but only with modified enumeration of nodes with respect to the minimum of function $r(i,e)$ on each level of the level-set structure.

Solutions obtained by improved Sloan's algorithm also can be improved by taken solutions in reverse order according to reverse Cuthill-McKee's algorithm [1]. We call this modification as reverse improved Cuthill-McKee-Sloan's algorithm.

3. Permutation algorithm for improving Cuthill-McKee-Sloan's solution

In this section we consider various implementations of a permutation algorithm that allow us to essentially improve Cuthill-McKee-Sloan's solutions.

Let us evaluate the following vector V with elements

$$v_j = \frac{1}{2}U_j + \frac{1}{2}D_j, \quad j = 1, 2, 3, ..., n;$$

$$U_j = \min_{a_{ij} \neq 0; \quad i=1,2,3,...,n} \{i\},$$

$$D_j = \max_{a_{ij} \neq 0; \quad i=1,2,3,...,n} \{i\}.$$

Here a_{ij}, $(i, j = 1, 2, 3, ..., n)$ are entries of the considering matrix A.

Now basing on the analysis of vector V, we construct a permutation decreasing the sky-line profile of the considering matrix. For this purpose we construct the series of permutations transforming matrix a_{ij} into the matrix with vector V with non-decreasing components. This can be done by various iteration algorithms. Further we consider two of these algorithms.

Algorithm 1:

- **Step 1.** Let us consider components of vector V. If they follow in non-decreasing order then go to **Step 5**.
- **Step 2.** Evaluate permutation of vector V components in non-decreasing order.
- **Step 3.** Apply this permutation to matrix a_{ij} and reevaluate vector V for it again.
- **Step 4.** Go to **Step 1**.
- **Step 5.** Take the final permutation in reverse order.
- **Step 6.** The end.

Algorithm 2:

- **Step 1.** Let us consider components of vector V. If they follow in non-decreasing order then go to **Step 6**.
- **Step 2.** Find two first components of vector V following in non-decreasing order.
- **Step 3.** Define the permutation changing positions of only two these components in vector V.
- **Step 4.** Apply this permutation to matrix a_{ij} and reevaluate vector V for it again.
- **Step 5.** Go to **Step 1**.
- **Step 6.** Take the final permutation in reverse order.
- **Step 7.** The end.

As our numerical experience shows both these algorithms allows to decrease the matrix sky-line profile. But both of them are sufficiently expensive from the computational point view. Therefore using the results of the reverse improved Cuthill-McKee-Sloan's solutions as an initial data for these algorithms allows to essentially decrease the volume of computations. Also we can mention that **Algorithm 1** converges more rapidly than **Algorithm 2**, but **Algorithm 2** allows to get the more perfect solutions. Also we can evaluate the more perfect solutions by using the following formula for the evaluation of permutation vector V with elements

$$v_j = \gamma\ U_j + \delta\ D_j, \quad j = 1, 2, 3, ..., n;$$

where $\gamma + \delta = 1$. The coefficients γ and δ can be fitted from the numerical experiments.

4. Numerical examples

Now let us shortly describe our experience in application of the algorithm described below to the enumeration of equations for the systems of linear algebra equations. In this section we consider some numerical examples for small and huge matrices.

4.1. **Small matrix.** At first we applied proposed technique to the enumeration of small systems consisting of 13 equations with the symmetric structure and compared our results with results obtained by the spectral technique described in [9].

The structure of the considering system is described by the following a semi-definite Laplace matrix

$$A = \left\{ \begin{array}{ccccccccccccc}
\mathbf{2} & \mathbf{-1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{-1} & \mathbf{3} & \mathbf{0} & \mathbf{-1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{-1} & \mathbf{0} & \mathbf{3} & \mathbf{0} & \mathbf{0} & \mathbf{-1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{-1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{3} & \mathbf{0} & \mathbf{0} & \mathbf{-1} & \mathbf{-1} & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{3} & \mathbf{0} & \mathbf{0} & \mathbf{-1} & \mathbf{-1} & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{3} & \mathbf{-1} & \mathbf{-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{1}
\end{array} \right\}.$$

The sky-line profile size for this matrix is equal to 77. Here the elements of the matrix that are in the sky-line profile are shown by the bold font. Applying to this matrix the reverse improved Cuthill-McKee-Sloan's algorithm we obtain the following enumeration vector $n_1 = \{8, 9, 5, 4, 2, 1, 3, 6, 7, 10, 11, 12, 13\}^T$, which produces the following matrix

$$A_1 = \left\{ \begin{array}{ccccccccccccc}
\mathbf{1} & 0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{-1} & \mathbf{-1} & \mathbf{3} & 0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{0} & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{-1} & \mathbf{-1} & \mathbf{3} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{2} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{3} & \mathbf{-1} & \mathbf{-1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & \mathbf{0} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{3} & \mathbf{-1} & \mathbf{-1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & \mathbf{0} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{3} & \mathbf{-1} & \mathbf{-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{1}
\end{array} \right\}.$$

with the sky-line profile size equal to 45. Taking vector n_1 in reverse order we produce the matrix with the sky-line profile size equal to 44. This matrix is shown here

$$A_2 = \left\{ \begin{array}{ccccccccccccc}
\mathbf{1} & 0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{0} & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{-1} & \mathbf{-1} & \mathbf{3} & 0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{0} & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{-1} & \mathbf{-1} & \mathbf{3} & 0 & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{-1} & \mathbf{3} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{2} & \mathbf{-1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{3} & \mathbf{-1} & \mathbf{-1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & \mathbf{0} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{3} & \mathbf{-1} & \mathbf{-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{1} & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & \mathbf{0} & \mathbf{1}
\end{array} \right\}.$$

Now let us evaluate vector V with weights $\gamma = \delta = \frac{1}{2}$ for matrix A_2. In result we have

$$V_2 = \{4, 5, 6, 9, 10, 13, 13, 16, 19, 19, 22, 23, 24\}^T .$$

Because all elements of vector V_2 are following in increase order we can not improve the obtained result by applying permutation algorithms 1 or 2. This means that vector V_2 is a trivial solution of Algorithms 1 or 2.

Applying the spectral algorithm to considering matrix A we evaluate the following Fiddler's vector for it $F = \{-0.231, -0.659, 0.219, -0.729, -0.953, 0.242,$

$0.625, -1.055, -1.055, 0.691, 0.904, 1.0, 1.0\}^T$. Sorting the components of this vector in increasing order gives us four following permutation vectors

$$N_1 = \{8, 9, 5, 4, 2, 1, 3, 6, 7, 10, 11, 12, 13\}^T,$$
$$N_2 = \{9, 8, 5, 4, 2, 1, 3, 6, 7, 10, 11, 12, 13\}^T,$$
$$N_3 = \{8, 9, 5, 4, 2, 1, 3, 6, 7, 10, 11, 13, 12\}^T,$$
$$N_4 = \{9, 8, 5, 4, 2, 1, 3, 6, 7, 10, 11, 13, 12\}^T.$$

We see that the first permutation vector N_1 obtained by spectral algorithm is coincided with our numerical solution n_1. Also we can mention that direct application of all permutation vectors $N_i,\ (i = 1, 2, 3, 4)$ to matrix A gives matrices with the same sky-line profile size equal to 45. Also substitution in reverse order gives matrices with the sky-line profile sizes equal to 44.

4.2. **Huge matrix.** Now let us consider one more complicated example of huge matrix A. This matrix was generated by NAFET program developed at General Energy Technologies Ltd by Dr. Vitaly Maidannik. NAFET program is intended for the mathematical modeling of hydro-thermal processes in the second circuits of the nuclear power plants. The size of considering matrix was equal to 437 and the size of sky-line profile was equal to 91873. After application of the first two stages of the reverse improved Cuthill-McKee-Sloan's algorithm the size of sky-line profile was reduced to 48458. After application of the full reverse improved Cuthill-McKee-Sloan's algorithm the size of sky-line profile was reduced to 43611. After application of **Algorithm 2** with 21516 iterations the size of sky-line profile was reduced to 29101. Application of **Algorithm 1** needs only 18239 iterations and reduces size of sky-line profile to 30044. In these computations the following values of coefficients α, β, γ and δ were used

$$\alpha = 100, \quad \beta = -11, \quad \gamma = \frac{87}{160}, \quad \delta = \frac{73}{160}.$$

5. Conclusions

The presented results show that application of the compound algorithm allows to essentially decrease the sizes of sky-line matrix profile in comparison with the reverse improved Cuthill-McKee-Sloan's algorithm.

The author is grateful to Dr. Vitaly Maidannik for presented example of huge matrix and helpful discussions.

References

[1] E.Cuthill and J.McKee, *Reducing the bandwidth of sparse symmetric matrices. Proceedings 24th National Conference of the Association for Computing Machinery*, New Jersey, Brandon Press, 1969, pp. 157–172.

[2] S.W.Sloan, *An algorithm for profile and wavefront reduction of sparse matrices*, Int. J. Numer. Meth. Engng. 23 (1986), pp. 239-251.

[3] I.S.Duff, A.M.Erisman and J.K.Reid, *Direct methods for sparse matrices.* Oxford University Press, London, 1986.

[4] I.S.Duff, J.K.Reid and J.A.Scott, *The use of profile reduction algorithms with a frontal code*, Int. J. Numer. Meth. Engng. 28 (1989), pp. 2555-2568.

[5] I.S.Duff and J.K.Reid, *Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Softw. 22 (1996), 227-257.

[6] I.S.Duff and J.A.Scott, *The design of a new frontal code for solving sparse, unsymmetric systems*, ACM Trans. Math. Softw. 22 (1996), pp. 30-45.

[7] G.Kumfert and A.Pothen, *Two improved algorithms for envelope and wavefront reduction*, BIT 18 (1997), pp. 559-590.

[8] S.T.Barnard, A.Pothen and H.Simon, *A spectral algorithm for envelope reduction of sparse matrices*, Numerical Linear Algebra with Applications 2 (1995), pp. 317-334.

[9] N.E.Gibbs, W.G.Poole and P.K.Stockmeyer. *An algorithm for reducing the bandwidth and profile of a sparse matrix*, SIAM J. Numer. Anal. 13 (1976), pp. 236-250.

[10] M.P.Levin, *Substructuring technique and parallel computations*, Trends in Mathematics, 4 (2001), pp. 108–115.

Kraftway Corporation PLC, Moscow, Russia

E-mail address: levin@kraftway.ru;Mikhail_Levin@hotmail.com