ANOTHER METHOD FOR FINDING PSEUDO-PERIPHERAL NODES

By

ILONA ARANY
Computer and Automation Institute of Hungarian Academy of Sciences
1014 Budapest, Úri u. 49.

(Received March 14, 1982)

Abstract

A method for finding pseudo-peripheral nodes is described here. Based on some previous results an important theorem is proved by which a theoretically based aspect for finding pseudo-peripheral nodes is becoming clear. The method obtained in such a way seems to be a modification for the method of Gibbs – Poole – Stockmeyer. After giving the description for the present method a short discussion is showing its efficiency; in many of tested cases it finds a pseudo-peripheral node at its first iteration step, while it requires advantageously few computer operations.

Finally, some notes are showing how efficiently the proposed method can be used, especially in bandwith/profile reduction orderings.

1. Introduction

In many application problems derived from undirected graphs, sometimes it is necessary to find peripheral (or pseudo-peripheral) node pair in the graph. Such a typical application area is when sparse linear systems are treated. Then for improving the efficient treating, different orderings [7], [8], [9], [11], [12], [14] are used for utilizing the sparsity property in a very high level. In all these ordering algorithms their main starting step is to find a pseudo-peripheral node pair of the graph. There are two well-known methods by N. Gibbs, W. Poole and P. Stockmeyer [12] and by A. George and J. Liu [10], [11] for finding pseudo-peripheral nodes and both are widely used and efficient algorithms. However, it is known that these methods have been assumed to be heuristic algorithms till [3] has given a theoretical foundation for them. So they are theoretically well-based, reliable and efficient methods.

In this paper we describe a new aspect for finding pseudo-peripheral nodes, which seems to be more efficient than either the GPS method [12] or George's method [10], [12]. First, the most important properties of the pseudo-peripheral nodes are analysed. Then based on some previous results recalled from [2] an important theorem is proved, by which a theoretically based aspect is shown for finding pseudo-peripheral nodes, by which the present method is described. A short discussion is presented for pointing to its efficiency, especially, to its arithmetic requirement.

Finally, some notes are showing its efficient usage within the bandwidth/profile reduction orderings.

2. Some related definitions and notations

We use the notation $G=(X, E)$ for undirected, connected graph; X is its node set, E is its edge set.

For describing the distance between two nodes $x, y \in X$ we use the notation $d(x, y)$.

The eccentricity for arbitrary node $x \in X$ is denoted by $l(x)$.

For level structure [1] rooted at $x \in X$ we use the notation

$$RLS(x)=\{L_0(x), L_1(x), \ldots, L_{l(x)}(x)\}$$

and the last level in it is called its eccentricity level, denoted by $L_{ec}(x)$. The nodes belonging to $L_{ec}(x)$ are the eccentricity nodes of the root vertex x.

Recall from [2] the reversible set term as follows.

Definition 2.1. For arbitrary pair of nodes $x, y \in X$ their reversible set is defined as

$$M(x, y)=\{M_0(x, y), M_1(x, y), \ldots, M_s(x, y)\}$$

where $s=d(x, y)$

$$M_j(x, y)=\{z \mid z \in L_j(x) \cap L_i(x);\ i+j=d(x, y)\},$$
$$(j=0, 1, \ldots, s=d(x, y)).$$

This term is playing an important role, when the distance function is described in level structure background [2]. The following theorem is true.

Theorem 2.1. Let us consider $RLS(x)$ for an arbitrary node $x \in X$. Let $y, z \in X \backslash \{x\}$ be arbitrary nodes for which

$$y \in L_i(x) \quad 0<i \leq l(x),$$
$$z \in L_j(x) \quad 0<j \leq l(x)$$

are assumed. Then

$$i+j-2k_x(y, z) \leq d(y, z)$$

relation is true, where

$$k_x(y, z)=d(x, M(y, z)). \ \square$$

(For proof see [2].)

Remark 2.1. We have to note, that as it is shown in [2], $d(y, z)$ can be written in the following form:

$$d(y, z)=i+j-w_x(y, z),$$

where $w_x(y, z) \geq 0$ and $w_x(y, z) \leq 2k_x(y, z)$ is also true.

It is easy to see that

$$d(y, z) = i+j$$

is true if and only if $x \in M(y, z)$ is satisfied.

3. Some previous results

First, we recall from [2] the pseudo-peripheral node term defined as follows.

Definition 3.1. Let $x \in X$ be an arbitrary node. Let $y \in L_{ec}(x)$ be one of its eccentricity node for which

$$l(y) = \max_{z \in L_{ec}(x)} l(z)$$

is satisfied. The node x is a pseudo-peripheral node, if $l(x) = l(y)$ is true.

Nodes x and y form a pseudo-peripheral node pair and they are corresponding to each other. The common value $l(x) = l(x)$ is the corresponding pseudo-diameter, with starting point x and endpoint y.

As it is shown in [3], nodes obtained by using GPS method [12] are satisfying this definition. On the other hand, this term serves as the most important basis in the foundation of a theoretical background for both the GPS method [12] and for its different versions [10].

The following statements are also true.

Theorem 3.1. If $x \in X$ is pseudo-peripheral node, then for every $y \in L_{ec}(x)$

$$l(x) = l(y)$$

relation is held. □

For proof see [2].

We have to note, however, that is may happen as it is shown in [2], [3], that although x is pseudo-peripheral node, then for a node $z \in L_{ec}(x)$ there exists $u \in L_{ec}(z)$ for which

$$l(u) > l(z) = l(x)$$

is obtained. Consequently, the pseudo-peripheral node term is non-symmetric. Its symmetric version was also introduced in [2] as quasi-peripheral node term.

Remark 3.1. If $x \in X$ is a pseudo-peripheral node, then for its each eccentricity node $y \in L_{ec}(x)$ x and y form pseudo-peripheral node pair with pseudo-diameter $l(x)$, and they are corresponding to each other.

Finally, recall from [2] an important theorem as follows.

Theorem 3.2. Let $x \in X$ be an arbitrary node. Let $y \in X$ be such an eccentricity node of x, for which $y \in L_{ec}(x)$; $l(y) = \max_{q \in L_{ec}(x)} l(q)$. Let $z \in X$ be an arbitrary eccentricity node of y, so $z \in L_{ec}(y)$.

If

$$x \in M(y, z) \tag{3.1}$$

is true, then z is a pseudo-peripheral node; z and y are pseudo-peripheral pair corresponding to each other. □

4. New aspects

It is clear from theorem 3.2 that if we want to find a pseudo-peripheral node, then such a node x has to be found for which (3.1) is satisfied. In this section we consider what kind of nodes are satisfying (3.1). The following theorem is true.

Theorem 4.1. Let $x \in X$ be an arbitrary node from which let us generate $RLS(x)$. Let $s \in L_{ec}(x)$ be a node of largest eccentricity in $L_{ec}(x)$. Then for every $t \in L_{ec}(s)$

$$w_x(t, s) = \min_{y \in L_{ec}(x)} w_x(t, y)$$

is true, where

$$0 \leq w_x(t, s) \leq 2d(x, M(t, s)) . \quad \Box$$

Proof. By the assumption let $s \in L_{ec}(x)$ be a node of largest eccentricity in $L_{ec}(x)$, that is

$$l(s) = \max_{z \in L_{ec}(x)} l(z) . \tag{4.1}$$

Let $t \in L_{ec}(s)$ be an arbitrary eccentricity node for s. Then for every $y \in L_{ec}(x)$

$$d(t, s) \geq d(t, y) \tag{4.2}$$

is satisfied, since from (4.1) $l(s) \geq l(y)$ follows. Now, let us use Theorem 2.1 for describing the distances in (4.2), then

$$d(t, s) = d(t, x) + d(x, s) - w_x(t, s) , \tag{4.3}$$

$$d(t, y) = d(t, x) + d(x, y) - w_x(t, y) \tag{4.4}$$

are obtained. Since, it is assumed that $s, y \in L_{ec}(x)$ so in (4.3) and (4.4)

$$d(x, s) = l(x); \quad d(x, y) = l(x)$$

are satisfied respectively. So the only difference in their forms is in their w_x-values. It can be seen easily that all the distances between any node of $L_{ec}(x)$ and $t \in L_{ec}(s)$ can also be written by the same two distances ($d(t, x)$ and $l(x)$), and the corresponding w_x-member. Consequently, from (4.2)

$$d(t, x) + l(x) - w_x(t, s) \geq d(t, x) + l(x) - w_x(t, y)$$

follows, from which

$$w_x(t, y) \geq w_x(t, s) \tag{4.5}$$

is obtained for every $y \in L_{ec}(x)$. That is, distance formed by the above two distances takes its maximal value if the corresponding w_x-value is minimum. Consequently,

$$w_x(t, s) = \min_{y \in L_{ec}(x)} w_x(t, y) \tag{4.6}$$

is really satisfied. $\Box$

Remark 4.1. In the proof $t \in L_{ec}(s)$ was chosen arbitrarily. Instead, if we choose $z \in L_{ec}(s)$ for which

$$d(z, x) = \min_{t \in L_{ec}(s)} d(t, x) \tag{4.7}$$

is true, then

$$w_x(z, s) \leq w_x(t, s) \tag{4.8}$$

is satisfied.

Proof. It is known, that for a given node $p \in X$ $l(p)$ is fix value. So $l(s) = d(t, s)$ is also well-defined value, and

$$d(t, s) = l(x) + H(s) \leq 2l(x)$$

where $0 \leq H(s) \leq l(x)$. That is

$$d(t, x) + l(x) - w_x(t, s) = l(x) + H(s)$$

$$d(t, x) - w_x(t, s) = H(s) \tag{4.9}$$

is followed, where $H(s)$ is fixed by s.

It is clear from (4.9), that $d(t, x) \geq H(s)$. On the other hand, if we choose $z \in L_{ec}(s)$ for which (4.7) is satisfied, then the corresponding w_x-value is decreased, that is

$$w_x(z, s) = \min_{t \in L_{ec}(s)} w_x(t, s) \tag{4.10}$$

is true, where for every $w_x(t, s)$ (4.6) is satisfied by the above theorem. Consequently, (4.8) is really satisfied. □

Conclusion 4.1. If in (4.10) $w_x(z, s) = 0$ is true, then

$$d(z, s) = d(z, x) + d(x, s)$$

is followed. Here by the sense of remark 2.1 $k_x(z, s) = 0$ necessarily follows, consequently

$$d(x, M(z, s)) = 0$$

has to be satisfied.

In this case we are given the nodes

$$x \in X, \quad s \in L_{ec}(x), \quad z \in L_{ec}(s),$$

where $l(s) = \max_{y \in L_{ec}(x)} l(y)$, while $x \in M(z, s)$ is true.

That is, all the conditions of theorem 3.2 are satisfied, consequently, z is pseudo-peripheral node, and z and s are forming pseudo-peripheral pair, and they are corresponding to each other.

We have to note, in addition, that in most of the tested cases $w_x(z, s) = 0$ is obtained. Consequently, in most cases for arbitrary node $x \in X$ its eccentricity node of largest eccentricity is to be the endnode corresponding to $z \in L_{ec}(s)$ (defined by 4.6)) pseudo-peripheral node.

The above discussion serves as a theoretical foundation for our present method.

5. An example

For illustrating the above result let us see an example shown in Figure 1. Let us consider node x (in Figure 1) as the node chosen arbitrarily.

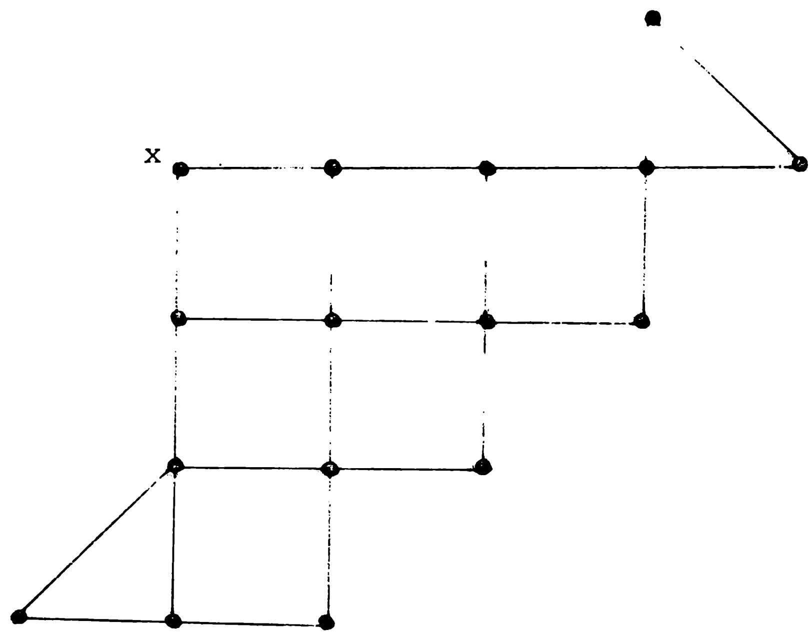

Figure 1.

Then generating $RLS(x)$ its levels are shown in Figure 2, from which

$$L_{ec}(x) = \{y_1, y_2, y_3, y_4, y_5\},$$
$$l(x) = 4$$

are obtained.

The related eccentricities are as follows

$$l(y_1) = l(y_2) = 7$$
$$l(y_3) = 5$$
$$l(y_4) = 4$$
$$l(y_5) = 6$$

from which

$$l(y_1) = l(y_2) = \max_{1 \le i \le 5} l(y_i)$$

is obtained.

Let us generate $RLS(y_1)$ and $RLS(y_2)$, then

$$L_{ec}(y_1) = \{z_1, z_2\}, \quad L_{ec}(y_2) = \{z_1, z_2\}$$

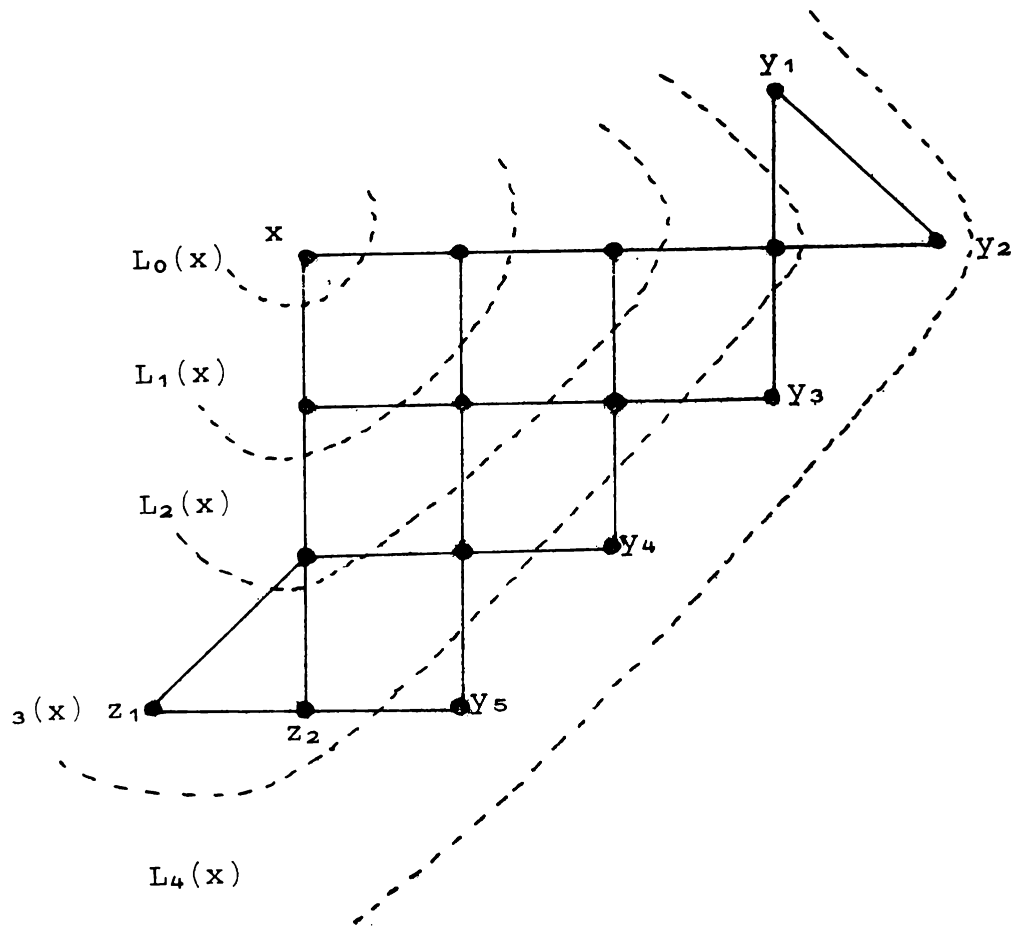

Figure 2.

are obtained. Here $d(z_1, x) = d(z_2, x) = 3$, so both z_1 and z_2 are satisfying (4.7). Now, let us create reversible set $M(y_1, z_2)$. Then by generating $RLS(y_1)$ and $RLS(z_2)$ their levels are shown in Figure 3, signed by single and dotted lines, respectively, while nodes belonging to $M(y_1, z_2)$ are denoted by small squares. In Figure 3 the sign of x is showing that $x \in M(y_1, z_2)$ is true, in fact, that is $w_x(y_1, z_2) = 0$ is obtained here.

However, if we would choose any node in $L_{ec}(x)$ having not the largest eccentricity, say y_3, then $L_{ec}(y_3) = \{z_1, z_2\}$. As Figure 4 is showing, now $x \notin M(y_3, z_2)$ is satisfied, since

$$d(x, M(y_3, z_2)) = 1$$

is true, and $w_x(y_3, z_2) = 2$.

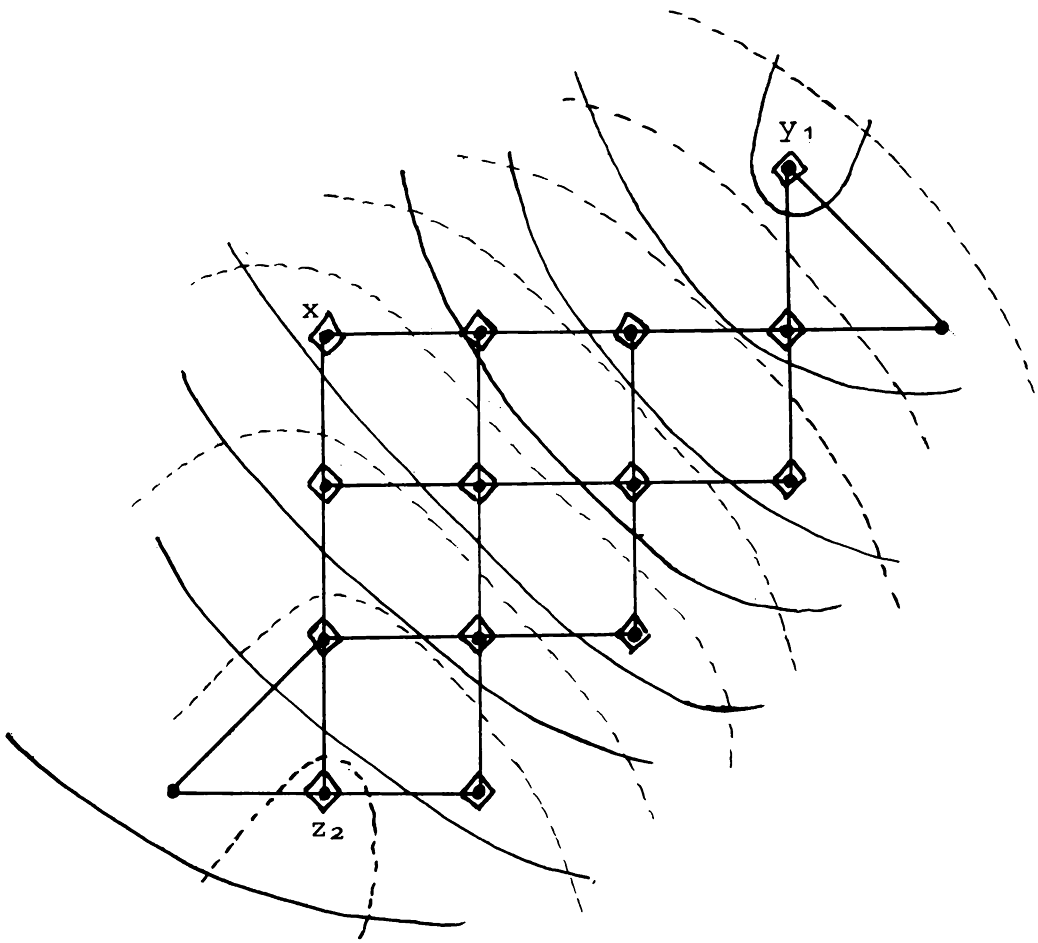

Figure 3.

It can be checked easily, that

$$d(x, M(y_4, z_2)) = 2, \quad w_x(y_4, z_2) = 4,$$

$$d(x, M(y_5, z_2)) = 3, \quad w_x(y_5, z_2) = 6$$

are true. That is

$$w_x(y_1, z_2) = \min_{1 \le i \le 5} w_x(y_i, z_2)$$

is really satisfied.

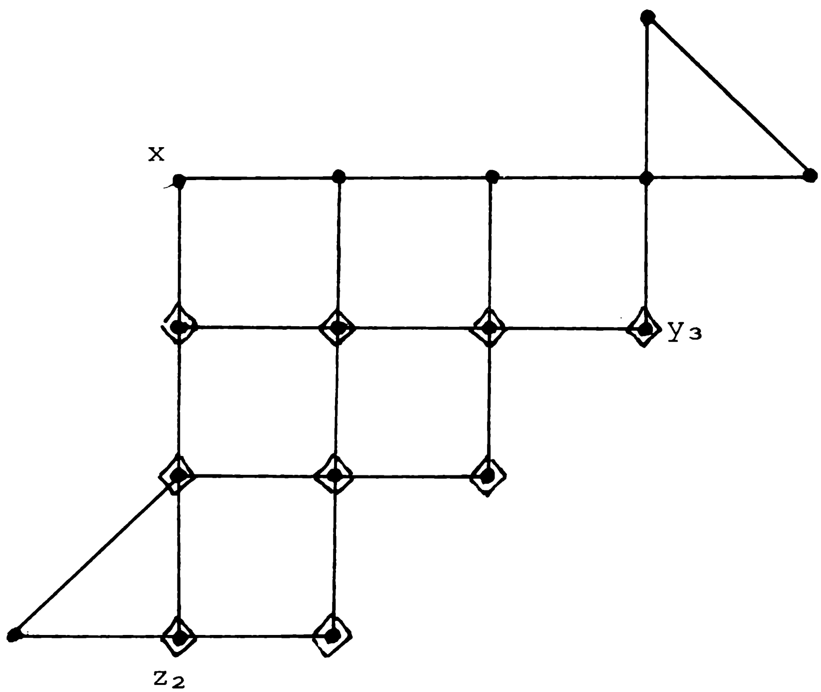

Figure 4.

6. Remarks on GPS method

As a result of the above discussion the efficiency of GPS method can be increased by the following proposals.

- The starting point selection step can be disregarded, because every node x can be used for this reason.
- Instead of turning to the next iteration step when a node of larger (than $l(x)$) eccentricity is found we had better check all nodes in $L_{ec}(x)$ by their eccentricities for looking for pseudo-peripheral end-node in $L_{ec}(x)$.

Applying the above suggestions to GPS method in many tested cases a pseudo-peripheral node is found at its first iteration step.

7. Present method

Our suggestions in the previous section are to be an immediate consequences of theorem 4.1, by which a modified version of GPS method is obtained as follows:

Let the starting point x be chosen arbitrarily.

Step 1. (Level structure generation)
Let us generate $RLS(x)$
$RLS(x) = \{L_0(x), L_1(x), \ldots, L_{ec}(x)\}$.

Step 2. (Find the node of largest eccentricity)
Check all the nodes in $L_{ec}(x)$ for finding the node of largest eccentricity (denoted by y).

Step 3. (Test for termination)
If $l(y) = l(x)$, go to step 4;
otherwise $x \leftarrow y$ and go to step 1.

Step 4. (Exit)
Nodes x and $\forall z \in L_{ec}(x)$ form pseudo-peripheral node pair corresponding to each other (x is the starting point; z is the endpoint).

The algorithm presented here seems to be a modification of the GPS method. It is very simple, reliable and theoretically well-based method requiring only advantageously few computer operations. In many practical cases a pseudo-peripheral node is found at its first iteration step. Then the number of level structures to be generated is $|L_{ec}(x)|+1$.

On the other hand, the amount of work required for its computer implementation is also advantageous. As it is known, A. George and J. Liu in their excellent work [11] gave some very useful and reach descriptions about the most important subroutines used by the SPARSPAK. Among others there is the subroutine ROOTLS for generating rooted level structure on connected components of a graph. Now, by using it only a few additional work is required for implementing the presented method, in addition, it can be fitted easily also into the SPARSPAK.

8. Efficient usage in bandwidth reduction

In solving the bandwidth reduction problem a level structure is needed whose length is as large as possible while its width is the possible smallest one. A. George and J. Liu solved this two aimed problem by their strategy 2 [10] and the number of nodes starting from which RLS-generation had to be tried is approximately $\left|\frac{|X|}{2}\right|$, (as the authors said in [10]). Whereas, by using the method presented here this number can be reduced to $|L_{ec}(x)|+1$. On the other hand, after having got the result node y different level structures whose lengths may be greater (see section 2) or equal to $l(y)$ can be generated, in a number $|L_{ec}(y)|$. In addition, if $p \in L_{ec}(y)$ is found for which $l(p) > l(y)$, then probably "better" level structure is obtained. Here only a little work is needed to select the best one from them.

Finally, we have to mention, that all the pseudo-peripheral node finder methods, including also the presented method, have a common problem. Namely, never is known whether the result nodes determined are exactly peripheral nodes. For solving this open problem further research is required.

REFERENCES

[1] *I. Arany, W. F. Smyth* and *L. Szóda:* An improved method for reducing the bandwidth for sparse symmetric matrices, in: Proc. of IFIP Congress 71, North-Holland, Amsterdam (1972) pp. 1246 – 1250.

[2] *I. Arany:* Distance – level structure – pseudo-peripheral nodes, in Mathematical Models and Algorithms, ed. Computing Centre of Eötvös Lorand University, Budapest (1983) pp. 5 – 32.

[3] *I. Arany:* The method of Gibbs – Poole – Stockmeyer is non-heuristic, in Annales, Sectio Computatorica, ed. Eötvös Lorand's University, Budapest 4 (1983) to appear.

[4] *W. K. Chen:* Applied Graph Theory, North-Holland Publ. Co., Amsterdam, (1971).

[5] *N. Christofides:* Graph Theory -- An Algorithmic Approach, Academic Press Inc., London (1975).

[6] *E. Cuthill* and *J. McKee:* Reducing the bandwidth of sparse symmetric matrices, Proc. 24th Nat. Conf. Assoc. Comput. Mach., ACM Publ. (1969) pp. 157 – 172.

[7] *A. George* and *J. W – H. Liu:* Algorithms for matrix partitioning and the numerical solution of finite element systems, SIAM J. Num. Anal. 15 (1978) pp. 297 – 327.

[8] *A. George* and *J. W – H. Liu:* An automatic one-way dissection algorithm for irregular finite element problems, in: Proc. 1977. Dundee Conf. on Numerical Analysis, Lecture Notes No 630, Springer Verlag (1978), pp. 76 – 8.

[9] *A. George* and *J. W – H. Liu:* An automatic nested dissection algorithm for irregular finite element problems, SIAM J. Num. Anal. 15 (1978) pp. 1053 – 1069.

[10] *A. George* and *J. W – H. Liu:* An implementation of a pseudo-peripheral node finder, ACM Trans. on Math. Software 5 (1979) pp. 286 – 295.

[11] *A. George* and *J. W – H. Liu:* Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, (1981).

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

[13] *W. F. Smyth* and *W. M. L. Benzi:* An algorithm for finding the diameter of a graph, in: Proc. of IFIP Congress, North-Holland Publ. Co., Amsterdam, (1974) pp. 500 – 503.

[14] *W. F. Smyth* and *I. Arany:* Another algorithm for reducing the bandwidth and profile of a sparse matrix, in: Proc. AFIPS 1976 NCC, AFIPS Press, Montvale, N. J., pp. 987 – 994.