On the Power of BFS to Determine a Graphs Diameter
(Extended Abstract)

Derek G. Corneil[1], Feodor F. Dragan[2], and Ekkehard Köhler[3]

[1] Dept. of Computer Science, University of Toronto, Toronto, Ontario, Canada, dgc@cs.toronto.edu
[2] Dept. of Computer Science, Kent State University, Kent, Ohio, U.S.A., dragan@cs.kent.edu
[3] Fachbereich Mathematik, Technische Universität Berlin, Berlin, Germany, ekoehler@math.TU-Berlin.DE

Abstract. Recently considerable effort has been spent on showing that Lexicographic Breadth First Search (LBFS) can be used to determine a tight bound on the diameter of graphs from various restricted classes. In this paper, we show that in some cases, the full power of LBFS is not required and that other variations of Breadth First Search (BFS) suffice. The restricted graph classes that are amenable to this approach all have a small constant upper bound on the maximum sized cycle that may appear as an induced subgraph. We show that on graphs that have no induced cycle of size greater than k, BFS finds an estimate of the diameter that is no worse than $\mathrm{diam}(G) - \lfloor k/2 \rfloor - 2$.

1 Introduction

Recently considerable attention has been given to the problem of developing fast and simple algorithms for various classical graph problems. The motivation for such algorithms stems from our need to solve these problems on very large input graphs, thus the algorithms must be not only fast, but also easily implementable.

Determining the diameter of a graph is a fundamental and seemingly quite time consuming operation. For arbitrary graphs (with n vertices and m edges), the current fastest algorithm runs in $O(nm)$ time which is too slow to be practical for very large graphs. This naive algorithm examines each vertex in turn and performs a Breadth First Search (BFS) starting at the particular vertex. Such a sweep starting at vertex x immediately determines $\mathrm{ecc}(x)$, the eccentricity of vertex x. Recall that the *eccentricity* of vertex x, $\mathrm{ecc}(x) = \max_{y \in V} \mathrm{d}(x, y)$, where $\mathrm{d}(x, y)$ denotes the distance between x and y; the *diameter* of G equals the maximum eccentricity of any vertex in V. It is clear that this algorithm actually computes the entire distance matrix; clearly knowing the distance matrix immediately yields the diameter of the graph.

For dense graphs, the best result known is by Seidel [12], who showed that the all pairs shortest path problem (and hence the diameter problem) can be solved

S. Rajsbaum (Ed.): LATIN 2002, LNCS 2286, pp. 209–223, 2002.

in $O(M(n) \log n)$ time where $M(n)$ denotes the time complexity for fast matrix multiplication involving small integers only. The current best matrix multiplication algorithm is due to Coppersmith and Winograd [2] and has $O(n^{2.376})$ time bound. Unfortunately, fast matrix multiplication algorithms are far from being practical and suffer from large hidden constants in the running time bound.

Note that no efficient algorithm for the diameter problem in general graphs, avoiding the computation of the whole distance matrix, has been designed. Thus, the question whether for a graph the diameter can be computed easier than the whole distance matrix remains still open.

Clearly, performing a BFS starting at a vertex of maximum eccentricity easily produces the graph's diameter. Thus one way to approximate the diameter of a graph is to find a vertex of high eccentricity; this is the approach taken in this paper. This is not, however, the only approach. For example, Aingworth et al [1] obtain a ratio 2/3 approximation to the diameter in time $O(m\sqrt{nlogn}+n^2logn)$. Note that a ratio of 1/2 can easily be achieved by choosing an arbitrary vertex (the eccentricity of any vertex is at least one half the diameter of the graph) and performing a BFS starting at this vertex. It follows also from results in [1,6] (see also paper [13] which surveys recent results related to the computation of exact and approximate distances in graphs) that the diameter problem in unweighted, undirected graphs can be solved in $\tilde{O}(\min\{n^{3/2}m^{1/2}, n^{7/3}\})$ time with an additive error of at most 2 without matrix multiplication. Here, $\tilde{O}(f)$ means $O(f \text{ polylog}(n))$. The motivation behind the work of Aingworth et al is to find a fast, easily implementable algorithm (they avoid using matrix multiplication), a motivation that we share.

Our approach is to examine the naive algorithm of choosing a vertex, performing some version of BFS from this vertex and then showing a nontrivial bound on the eccentricity of the last vertex visited in this search. In fact, this algorithm is one of the "classical" algorithms in graph theory; if one restricts one's attention to trees, then this algorithm produces a vertex of maximum eccentricity (see e.g. [9]). This approach has already received considerable attention. (In the following we let v denote the vertex that appears last in a particular search; the definition of the various searches and families of graphs will be presented in the next section.) For example, Dragan et al [8] have shown that if LBFS is used for chordal graphs, then $\text{ecc}(v) \geq \text{diam}(G)-1$ whereas for interval graphs $\text{ecc}(v) = \text{diam}(G)$. It is clear from the work of Corneil et al [4], that by using LBFS on AT-free graphs, one has $\text{ecc}(v) \geq \text{diam}(G) - 1$. Dragan [7], again using LBFS, has shown that $\text{ecc}(v) \geq \text{diam}(G) - 2$ for HH-free graphs, $\text{ecc}(v) \geq \text{diam}(G) - 1$ for HHD-free graphs and $\text{ecc}(v) = \text{diam}(G)$ for graphs that are both HHD-free and AT-free.

It is interesting to note that Corneil et al [4] have looked at double sweep LBFSs (i.e. start an LBFS from a vertex that is last in a previous arbitrarily chosen LBFS) on chordal and AT-free graphs. They have provided a forbidden subgraph structure on graphs where $\text{ecc}(v) = \text{diam}(G) - 1$. They also presented both chordal and AT-free graphs where for no c, is the c-sweep LBFS algorithm guaranteed to find a vertex of maximum eccentricity. Furthermore, they showed that for any c, there is a graph G where $\mathrm{ecc}(v) \leq \mathrm{diam}(G) - c$, where v is the vertex visited last in a 2-sweep LBFS. This graph G, however, has a large induced cycle whose size depends on c.

These results motivate a number of interesting questions:

- Is it an inherent property of LBFS to end in a vertex of high eccentricity for the various restricted graph families mentioned above? What happens if we use other variants of BFS?
- Why do AT-free and chordal graphs, two families with very disparate structure, exhibit such similar behaviour with respect to the efficacy of LBFS to find vertices of high eccentricity?
- Although LBFS "fails" to find vertices of high eccentricity for graphs in general, all known examples that exhibit such failure have large induced cycles. If we bound the size of the largest induced cycle, can we get a bound on the eccentricity of the vertex that appears last in an LBFS?
- If the previous question is answered in the affirmative, is the full power of LBFS needed? What happens if we just use BFS?

This paper addresses these questions. In the next section, we present the various forms of BFS and define the graph theoretic terminology used throughout the paper. In Section 3 we examine the behaviour of the different versions of BFS on various restricted graph families. We establish some new bounds and show, by example, that all stated bounds on $\mathrm{ecc}(v)$ are tight. In Section 4, we examine families of graphs where the size of the largest induced cycle is bounded and show that BFS does succeed in getting vertices of high, with respect to k, eccentricity.

2 Notation and Definitions

First we formalize the notion of BFS and then discuss various variations of it. We caution the reader that there is some confusion in the literature between BFS and what we call LL, defined below. In defining the various versions of BFS, we are only concerned with identifying the last vertex visited by the search; straightforward modifications produce the list of vertices in the order that they are visited by the search. It should be noted that none of the orderings are unique; instead, each search identifies one of the possible end-vertices.

Algorithm BFS: Breadth First Search
Input: graph $G(V, E)$ and vertex u
Output: vertex v, the last vertex visited by a BFS starting at u

Initialize queue Q to be $\{u\}$ and mark u as "visited".
while $Q \neq \emptyset$ **do**
  Let v be the first vertex of Q and remove it from Q.
  Each unvisited neighbour of v is added to the end of Q and marked as "visited".

Note that the above algorithm can easily be modified to obtain the "layers" of V with respect to u. In particular, for each $0 \leq i \leq \mathrm{ecc}(u)$, the *$i$-th layer of V with respect to u*, denoted $L_i(u)$, $= \{v : \mathrm{d}(u, v) = i\}$. This motivates the next algorithm, LL.

Algorithm LL: Last Layer
Input: graph $G(V, E)$ and vertex u
Output: vertex v, a vertex in the last layer of u

Run BFS to get the layering of V with respect to u.
Choose v to be an arbitrary vertex in the last layer.

Clearly any vertex returned by BFS can also be returned by LL; the converse is not true as shown by the graph in Figure 1.

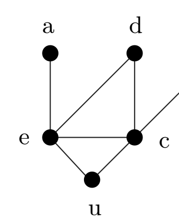

Fig. 1. No BFS starting at u can return vertex d.

Now we modify this algorithm to obtain a vertex in the last layer that has minimum degree with respect to the vertices in the previous layer.

Algorithm LL+: Last Layer, Minimum Degree
Input: graph $G(V, E)$ and vertex u
Output: vertex v, a vertex in the last layer of u, that has minimum degree with respect to the vertices in the previous layer

Run BFS to get the layering of V with respect to u.
Choose v to be an arbitrary vertex in the last layer that has minimum degree with respect to the vertices in the previous layer.

Finally we introduce Lexicographic Breadth First Search (LBFS). This search paradigm was discovered by Rose et al [11] and was shown to yield a simple linear time algorithm for the recognition of chordal graphs. In light of the great deal of work currently being done on LBFS, it is somewhat surprising that interest in LBFS lay dormant for quite a while after [11] appeared.

Algorithm LBFS: Lexicographic Breadth First Search
Input: graph $G(V, E)$ and vertex u
Output: vertex v, the last vertex visited by an LBFS starting at u

Assign label $\emptyset$ to each vertex in V.
for $i = n$ **downto** 1 **do**
  Pick an unmarked vertex v with the largest (with respect to lexicographic order) label.
  Mark v "visited".
  For each unmarked neighbour y of v, add i to the label of y.

If vertex e is removed from the graph in Figure 1, we have a graph where a vertex, namely d, can be visited last by a BFS from u but by no LBFS from u.

We now turn to the definitions of the various graph families introduced in the previous section. A graph is *chordal* if it has no induced cycle of size greater than 3. An *interval* graph is the intersection graph of intervals of a line. Lekkerkerker and Boland [10] defined an *asteroidal triple* to be a triple of vertices such that between any two there is a path that avoids the neighbourhood of the third and showed that a graph is an interval graph iff it is both chordal and asteroidal triple-free (AT-free). A *claw* is the complete bipartite graph $K_{1,3}$, a *hole* is an induced cycle of length greater than 4, a *house* is a 4-cycle with a triangle added to one of the edges of the C_4 and a *domino* is a pair of C_4s sharing an edge. A graph is *HH-free* if it contains no induced houses or holes and is *HHD-free* if it contains no induced houses, holes or dominos.

Finally, in order to capture the notion of "small" induced cycles, we define a graph to be *k-chordal* if it has no induced cycles of size greater than k. Note that chordal graphs are precisely the 3-chordal graphs and AT-free graphs are 5-chordal. We define the *disk* of radius r centered at u to be the set of vertices of distance at most r to u, i.e., $D_r(u) = \{v \in V : \mathrm{d}(u, v) \leq r\} = \bigcup_{i=0}^{r} L_i(u)$.

3 Restricted Families of Graphs

We now see how the four search algorithms mentioned in the previous section behave on the following families of graphs: chordal, AT-free, {AT,claw}-free, interval, and hole-free. The results are summarized in the following table. In this table the references refer to the paper where the lower bound was established; a [*] indicates that the result is new. A figure reference refers to the appropriate figure where it is shown that the lower bound is tight. In each of the figures the vertex pair a,b forms a diametral pair, i.e. $\mathrm{d}(a, b) = \mathrm{diam}(G)$. Below each figure a BFS, LBFS, LL, or LL+ ordering is given that achieves the corresponding bounds; vertex u is always the start-vertex and v the end-vertex of the appropriate search; different BFS-layers are separated by a |.

GRAPH CLASS	LL	LL+	BFS	LBFS
chordal graphs	$\geq D-2$ [3] Fig. 4	$\geq D-2$ [3] Fig. 5	$\geq D-1$ [*] Fig. 2	$\geq D-1$ [8] Fig.6
AT-free graphs	$\geq D-2$ [*] Fig. 3	$\geq D-1$ [*] Fig. 7	$\geq D-2$ [*] Fig. 3	$\geq D-1$ [4] Fig. 7
{AT,claw}-free graphs	$\geq D-1$ [*] Fig. 2	$= D$ [*]	$\geq D-1$ [*] Fig. 2	$= D$ [*]
interval graphs	$\geq D-1$ [*] Fig. 2	$= D$ [*]	$\geq D-1$ [*] Fig. 2	$= D$ [8]
hole-free graphs	$\geq D-2$ [*] Fig. 8	$\geq D-2$ [*] Fig. 8	$\geq D-2$ [*] Fig. 8	$\geq D-2$ [*] Fig. 8

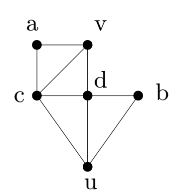

Fig. 2. BFS: u|bcd|av

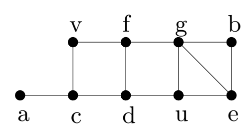

Fig. 3. BFS: u|dge|cfb|av

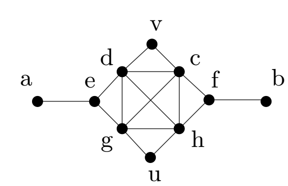

Fig. 4. LL: u|gh|edcf|abv

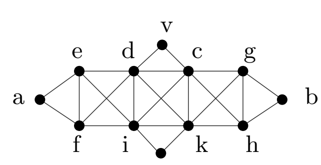

Fig. 5. LL+: u|ik|fedcgh|abv

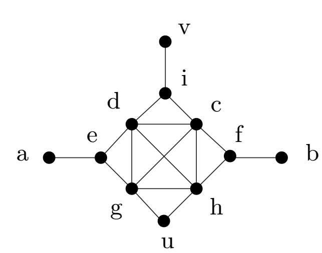

Fig. 6. LBFS: u|gh|edcf|aib|v

To illustrate the types of techniques that are used to establish lower bounds in the table, we now show that $\text{ecc}(v) \geq \text{diam}(G) - 1$ for chordal graphs when BFS is used. This result subsumes the result shown in [8] that this lower bound holds when LBFS is used. The journal version of the paper will contain proofs for all new results mentioned in the table.

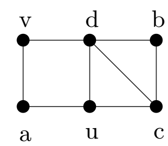

Fig. 7. LBFS: u|cda|bv

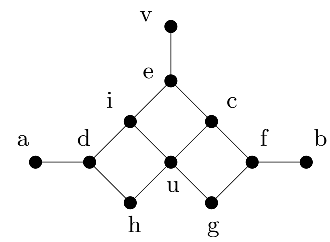

Fig. 8. LBFS: u|ghic|dfe|abv

First we comment on the BFS algorithm. In particular, we may regard BFS as having produced a numbering from n to 1 in decreasing order of the vertices in V where vertex u is numbered n. As a vertex is placed on the queue, it is given the next available number. The last vertex visited, v, is given the number 1. Thus BFS may be seen to generate a rooted tree T with vertex u as the root. A vertex y is the *father* in T of exactly those neighbours in G which are inserted into the queue when y is removed.

An ordering $\sigma = [v_1, v_2, \ldots, v_n]$ of the vertex set of a graph G generated by a BFS will be called a *BFS–ordering* of G. Let $\sigma(y)$ be the number assigned to a vertex y in this BFS–ordering σ. Denote also by $f(y)$ the father of a vertex y with respect to σ. The following properties of a BFS–ordering will be used in what follows. Since all layers of V considered here are with respect to u, we will frequently use notation L_i instead of $L_i(u)$.

(P1) If $y \in L_q$ $(q > 0)$ then $f(y) \in L_{q-1}$ and $f(y)$ is the vertex from $N(y) \bigcap L_{q-1}$ with the largest number in σ.
(P2) If $x \in L_i$, $y \in L_j$ and $i < j$, then $\sigma(x) > \sigma(y)$.
(P3) If $x, y \in L_j$ and $\sigma(x) > \sigma(y)$, then either $\sigma(f(x)) > \sigma(f(y))$ or $f(x) = f(y)$.
(P4) If $x, y, z \in L_j$, $\sigma(x) > \sigma(y) > \sigma(z)$ and $f(x)z \in E$, then $f(x) = f(y) = f(z)$ (in particular, $f(x)y \in E$).

In what follows, by $P(x, y)$ we will denote a path connecting vertices x and y. Proof of the following lemma is omitted.

Lemma 1. *If vertices a and b of a disk $D_r(u)$ of a chordal graph are connected by a path $P(a, b)$ outside of $D_r(u)$ (i.e., $P(a, b) \bigcap D_r(u) = \{a, b\}$), then a and b must be adjacent.*

Let σ be a BFS–ordering of a chordal graph G started at a vertex u. Let also $P(a, b) = (a = x_1, x_2, \ldots, x_{k-1}, x_k = b)$ be a shortest path of G connecting vertices a and b. We say that $P(a, b)$ is a *rightmost shortest path* if the sum $\sigma(x_1) + \sigma(x_2) + \ldots + \sigma(x_{k-1}) + \sigma(x_k)$ is the largest among all shortest paths connecting a and b.

Lemma 2. *Let x and y be two arbitrary vertices of a chordal graph G. Every rightmost shortest path $P(x,y)$ between x and y can be decomposed (see Figure 9) into three shortest subpaths $P_x = (x = x_1, x_2, \ldots, x_l)$ (called* the vertical $x-$subpath*), $P_y = (y = y_1, y_2, \ldots, y_k)$ (called* the vertical $y-$subpath*) and $P(x_l, y_k)$ (called* the horizontal subpath*) such that*

1. *$P(x_l, y_k) \subseteq L_j$ for some $j \in \{0, 1, \ldots, \mathrm{ecc}(u)\}$, and $\mathrm{d}(x_l, y_k) \leq 2$ (i.e., x_l and y_k either coincide or are adjacent or have a common neighbour in $G[L_j] \cap P(x_l, y_k)$);*
2. *$x_{l-i} \in L_{j+i}$ for $0 \leq i \leq l-1$;*
3. *$y_{k-i} \in L_{j+i}$ for $0 \leq i \leq k-1$.*

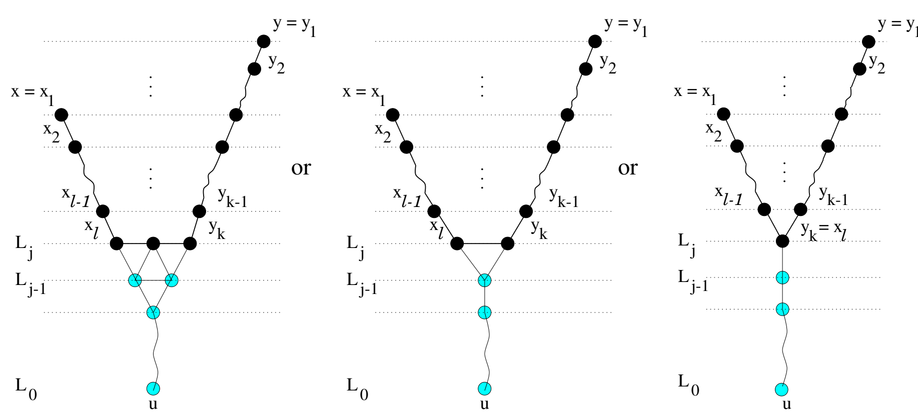

Fig. 9. The structure of rightmost shortest path in chordal graphs.

Proof. First we prove that $|P(x,y) \bigcap L_i| \leq 3$ for any $i = 1, 2, \ldots, \mathrm{ecc}(s)$. Assume that the intersection of $P(x,y)$ and a layer L_q, for some index q, contains at least four vertices. Let a, b, c, d be the first four vertices of $P(x,y) \bigcap L_q$ on the way from x to y. We claim that $ab, bc, cd \in E$.

If $ab \notin E$ then, by Lemma 1, subpath $P(a,b)$ of the path $P(x,y)$ is completely contained in disk $D_q(s)$. In particular, the neighbor b' of b on $P(a,b)$ belongs to $D_{q-1}(s)$. Using the same arguments, we conclude that $bc \in E$ or subpath $P(b,c)$ of the path $P(x,y)$ is contained in $D_q(s)$. If $bc \in E$ then a neighbor v of c in L_{q-1} must be adjacent with b' (by Lemma 1). Since $\sigma(v) > \sigma(b)$, we get a contradiction to $P(x,y)$ is a rightmost path (we can replace vertex b of $P(x,y)$ with v and get a shortest path between x and y with larger sum). If $bc \notin E$ then the neighbor b'' of b on $P(b,c)$ is also contained in $D_{q-1}(s)$. By Lemma 1, vertices b' and b'' must be adjacent, but this is impossible since $P(x,y)$ is a shortest path.

Thus, vertices a and b have to be adjacent. If vertices b and c are not adjacent, then the neighbor b'' of b on $P(b, c)$ belongs to $D_{q-1}(s)$ and, by Lemma 1, it must be adjacent to any neighbor u of a in L_{q-1}. Since $\sigma(u) > \sigma(b)$ holds, again a contradiction to $P(x, y)$ is a rightmost path arises. Consequently, vertices b and c are adjacent, too. Completely in the same way one can show that c and d have to be adjacent. Note also that the adjacency of a with b and b with c is proved without using the existence of the vertex d.

We have now $ab, bc, cd \in E$ and the induced path (a, b, c, d) is a rightmost shortest path (as a subpath of the rightmost shortest path $P(x, y)$). Consider neighbors a' and d' in L_{q-1} of a and d, respectively. Since $\sigma(a')+ \sigma(d') > \sigma(b)+ \sigma(c)$ and the path (a, b, c, d) is rightmost and shortest, vertices a' and d' can neither coincide nor be adjacent. But then we get a path (a', a, b, c, d, d') connecting two non-adjacent vertices of $D_{q-1}(s)$ outside of the disk. A contradiction to Lemma 1 obtained shows that $|P(x, y) \bigcap L_i| \leq 3$ holds for any $i = 1, 2, \ldots, \mathrm{ecc}(s)$.

Now let $|P(x, y) \bigcap L_q| = 3$ and let a, b, c be the vertices of $P(x, y) \bigcap L_q$ on the way from x to y. It follows from the discussion above that $ab, bc \in E$ and both the neighbor a' of a on subpath $P(x, a)$ and the neighbor c' of c on subpath $P(c, y)$ (if they exist) belong to the layer L_{q+1}. If for example $a' \in L_{q-1}$, then, by Lemma 1, a' must be adjacent to any neighbor v of b in L_{q-1}. Again, since $\sigma(v) > \sigma(a)$, a contradiction to $P(x, y)$ is a rightmost path arises.

Furthermore, if for some index q, $P(x, y) \bigcap L_q = \{a, b\}$, and $P(x, y) = (x, \ldots, a', a, b, b', \ldots, y)$, then both a' and b' belong to the layer L_{q+1}.

Summarizing all these we conclude that, while moving from x to y along the path $P(x, y)$, we can have only one horizontal edge or only one pair of consecutive horizontal edges. Here by horizontal edge we mean an edge with both end-vertices from the same layer. All other vertices of the path $P(x, y)$ belong to higher layers.

Having the structure of a rightmost shortest path established, we can now prove the main result for chordal graphs. In presenting a rightmost shortest path, we use "/"s to differentiate the appropriate subpaths.

Theorem 1. *Let v be the vertex of a chordal graph G last visited by a BFS. Then* $\mathrm{ecc}(v) \geq \mathrm{diam}(G) - 1$.

Proof. Let x, y be a pair of vertices such that $\mathrm{d}(x, y) = \mathrm{diam}(G)$, and consider two rightmost shortest paths

$$P(x, v) = (x = x_1, x_2, \ldots, x_{l-1}/x_l, \ldots, v'_h/v'_{h-1}, \ldots, v'_2, v'_1 = v)$$

and

$$P(y, v) = (y = y_1, y_2, \ldots, y_{k-1}/y_k, \ldots, v''_g/v''_{g-1}, \ldots, v''_2, v''_1 = v)$$

connecting vertex v with x and y, respectively (see Figure 10). By Lemma 2, each of these paths consists of two (perhaps of length 0) vertical subpaths and one horizontal path of length not greater than 2. Assume, without loss of generality, that $h \leq g$ and let $x_l, \ldots, v'_h \in L_q$. Since $v \in L_{\mathrm{ecc}(u)}$ we also have $l \leq h$ and $k \leq g$.

By Lemma 1, vertices v'_h, v''_h in L_q either coincide or are adjacent. Note that, if $\mathrm{d}(x_l, v''_h) \le 1$, then $\mathrm{d}(x, y) \le \mathrm{d}(x, x_l) + 1 + \mathrm{d}(v''_h, y) \le \mathrm{d}(v, v''_h) + 1 + \mathrm{d}(v''_h, y) = \mathrm{d}(v, y) + 1 \le \mathrm{ecc}(v) + 1$. That is, $\mathrm{ecc}(v) \ge \mathrm{d}(x, y) - 1 = \mathrm{diam}(G) - 1$, and we are done. Hence, we may assume that $\mathrm{d}(x_l, v''_h) \ge 2$ and, therefore, $x_l \ne v'_h$.

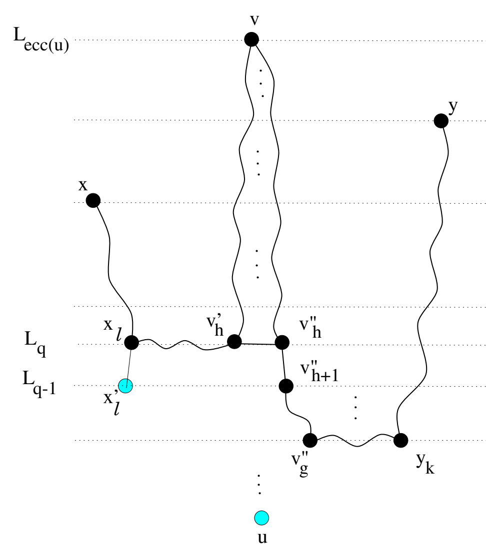

Fig. 10. Rightmost shortest paths $P(x, v)$ and $P(y, v)$.

We distinguish between two cases. The first one is simple. Only for the second case we will need to use the special properties of a BFS–ordering.

Case $g > h$**.**
In this case there exists a vertex v''_{h+1} in the intersection $P(y, v) \bigcap L_{q-1}$. Consider also a neighbour x'_l of x_l in L_{q-1}. Since vertices x'_l and v''_{h+1} are connected by path $(x'_l, x_l, \ldots, v'_h, v''_h, v''_{h+1})$ outside of the disk $D_{q-1}(u)$, by Lemma 1, they are adjacent if they do not coincide. Hence, $\mathrm{d}(x_l, v''_{h+1}) \le 2$ and, therefore, $\mathrm{d}(x, y) \le \mathrm{d}(x, x_l) + 2 + \mathrm{d}(v''_{h+1}, y) \le \mathrm{d}(v, v''_h) + 2 + \mathrm{d}(v''_{h+1}, y) = \mathrm{d}(v, y) + 1 \le \mathrm{ecc}(v) + 1$, i.e., again $\mathrm{ecc}(v) \ge \mathrm{d}(x, y) - 1 = \mathrm{diam}(G) - 1$.

Case $g = h$**.**
From the discussion above (now, since $g = h$ we have a symmetry), we may assume that $\mathrm{d}(y_k, v'_h) \ge 2$ and $y_k \ne v''_h$. Consider neighbours x'_l and y'_k in L_{q-1} of vertices x_l and y_k, respectively (see Figure 11(a)). By Lemma 1, they are adjacent if do not coincide, i.e., $\mathrm{d}(x_l, y_k) \le 3$. Now, if at least one of the equalities $\mathrm{d}(x_l, v'_h) = 2$, $\mathrm{d}(y_k, v''_h) = 2$ holds, then we are done. Indeed, if for example $\mathrm{d}(x_l, v'_h) = 2$, then $\mathrm{d}(x, v) = \mathrm{d}(x, x_l) + 2 + \mathrm{d}(v'_h, v)$ and, therefore, $\mathrm{d}(x, y) \le \mathrm{d}(x, x_l) + \mathrm{d}(x_l, y_k) + \mathrm{d}(y_k, y) \le \mathrm{d}(x, x_l) + 3 + \mathrm{d}(v'_h, v) = \mathrm{d}(x, v) + 1 \le \mathrm{ecc}(v) + 1$.

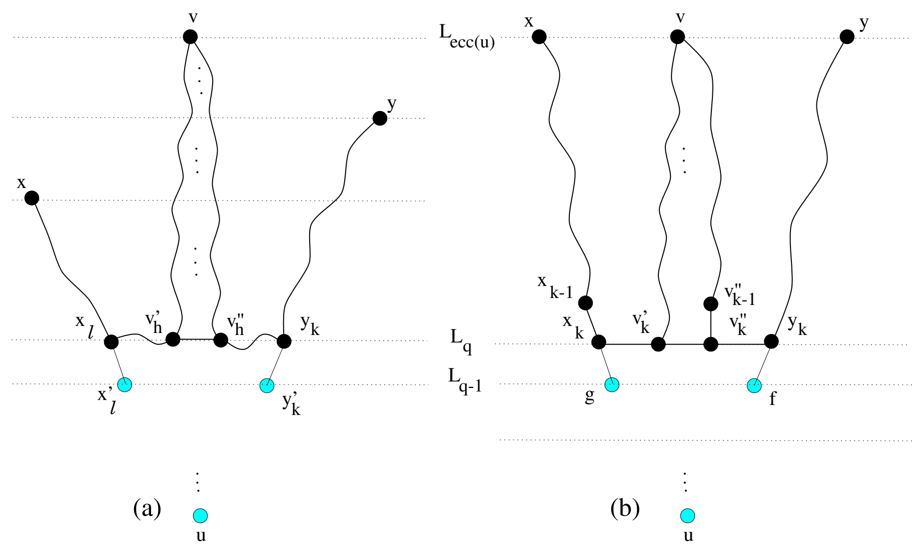

Fig. 11. (a) Horizontal subpaths of $P(x, v)$ and $P(y, v)$ are in the same layer, (b) Paths $P(x, v)$ and $P(y, v)$ have similar shape and the same length $2k - 1$.

So, we may assume that $x_l v'_h, y_k v''_h \in E$. Moreover, since $\mathrm{d}(y_k, v'_h) \geq 2$, vertices v'_h and v''_h cannot coincide, i.e., they are adjacent. If $l < h$ or $k < h$ or $\mathrm{d}(x_l, y_k) < 3$, again we will get $\mathrm{d}(x, y) \leq \mathrm{ecc}(v) + 1$ by comparing distances $\mathrm{d}(v, y) = h + k - 1$, $\mathrm{d}(v, x) = h + l - 1$ with $\mathrm{d}(x, y) \leq l - 1 + \mathrm{d}(x_l, y_k) + k - 1 \leq l + 1 + k$. Thus, we arrive at a situation when $l = k = h$, $x_l v'_h, y_k v''_h, v'_h v''_h \in E$, and $\mathrm{d}(x_l, y_k) = 3$, $\mathrm{d}(v, x) = \mathrm{d}(v, y) = 2k - 1$ (see Figure 11(b)). We may also assume that $\mathrm{d}(x, y) = 2k + 1$, since otherwise, $\mathrm{d}(x, y) \leq 2k = \mathrm{d}(v, y) + 1 \leq \mathrm{ecc}(v) + 1$ and we are done. We show that this final configuration (with $\mathrm{d}(x, y) = 2k + 1$) is impossible because of the properties of BFS–orderings.

Assume, without loss of generality, that $\sigma(y_k) > \sigma(x_k)$ and consider the fathers $f = f(y_k)$, $g = f(x_k)$ of y_k and x_k, respectively. Since $\mathrm{d}(x_k, y_k) = 3$, we have $f \neq g$ and $f x_k, g y_k \notin E$. By Lemma 1 and property (P3) of BFS–ordering, vertices f and g are adjacent and $\sigma(f) > \sigma(g)$. Chordal graphs cannot contain an induced cycle of length greater than 3. Therefore, in the cycle formed by $g, x_k, v'_k, v''_k, y_k, f$, at least chords $g v'_k$ and $f v''_k$ must be present. Since for the father $f(v''_k)$ of v''_k we have $\sigma(f(v''_k)) \geq \sigma(f) > \sigma(g)$, inequality $\sigma(v''_k) > \sigma(x_k)$ must hold (here we used properties (P1) and (P3) of BFS–orderings). We will need the inequality $\sigma(v''_k) > \sigma(x_k)$ later to get our final contradiction.

Now consider vertices x_{k-1} and v''_{k-1}. We claim that $\sigma(v''_{k-1}) < \sigma(x_{k-1})$. Assume that this is not the case, and let j ($j \in \{1, 2, \ldots, k-2\}$) be the largest index such that $\sigma(v''_j) < \sigma(x_j)$ (recall that $\sigma(v''_1) = \sigma(v) = 1 < \sigma(x) = \sigma(x_1)$). Then $\sigma(v''_{j+1}) > \sigma(x_{j+1})$ holds, and since $j \leq k-2$ and $\mathrm{d}(v, x) = 2k-1$, we obtain $\mathrm{d}(x_j, v''_j) \geq 5$ (because of $2k - 1 = \mathrm{d}(v, x) \leq \mathrm{d}(v, v''_j) + \mathrm{d}(v''_j, x_j) + \mathrm{d}(x_j, x) =$

$2(j-1)+\mathrm{d}(v''_j, x_j) \leq 2(k-3)+\mathrm{d}(v''_j, x_j) = 2k-6+\mathrm{d}(v''_j, x_j))$. Consider the father $t = f(x_j)$ of x_j. From the distance requirement and properties of BFS–orderings we conclude that $t \neq v''_{j+1}$, $t \neq x_{j+1}$ and $\sigma(t) > \sigma(v''_{j+1}) > \sigma(x_{j+1})$. Moreover, vertex t has to be adjacent to x_{j+1} (by Lemma 1), but cannot be adjacent to x_{j+2} (since $\sigma(t) > \sigma(x_{j+1})$ and the path $P(x, v)$ is rightmost). Consider now the father $z = f(t)$ of the vertex t. It is adjacent to x_{j+2}, by Lemma 1, and has to be adjacent to x_{j+1}, to avoid an induced cycle $(z, t, x_{j+1}, x_{j+2}, z)$ of length 4. Applying property (P4) to $\sigma(t) > \sigma(v''_{j+1}) > \sigma(x_{j+1})$ and $zx_{j+1} \in E$, we get $zv''_{j+1} \in E$ which is impossible since $\mathrm{d}(x_j, v''_j) \geq 5$. This contradiction shows that, indeed, the inequality $\sigma(v''_{k-1}) < \sigma(x_{k-1})$ must hold.

So, we have $\sigma(v''_{k-1}) < \sigma(x_{k-1})$ and $\sigma(v''_k) > \sigma(x_k)$. We repeat our arguments from the previous paragraph considering index $k-1$ instead of j. (The only difference is that now we do not have the vertex $x_{j+2} = x_{k+1}$ on the path $P(x, v)$.) Again, consider the father $t = f(x_{k-1})$ of x_{k-1}. Clearly, $t \neq v'_k$. Since $\mathrm{d}(x_k, y_k) = 3$, vertex t is not adjacent to y_k. Furthermore t does not coincide with v''_k since Lemma 1 would require $x_k v''_k \in E$. Hence, $t \neq v''_k$ and, by property (P3), $\sigma(t) > \sigma(v''_k) > \sigma(x_k)$. Vertex t must be adjacent to x_k, by Lemma 1, but cannot be adjacent to v'_k, since the path $P(v, x)$ is rightmost and $\sigma(t) > \sigma(x_k)$. To avoid an induced cycle of length 4, vertex t is not adjacent to v''_k as well.

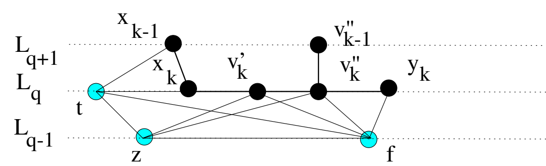

Fig. 12. Illustration to the proof of Theorem 1.

Consider also the father $z = f(t)$ of the vertex t (see Figure 12). If $zy_k \in E$ then $\mathrm{d}(x, y) \leq \mathrm{d}(y, y_k) + 1 + \mathrm{d}(z, x_{k-1}) + \mathrm{d}(x_{k-1}, x) \leq k-1+1+2+k-2 = 2k$, and a contradiction to the assumption $\mathrm{d}(x, y) = 2k+1$ arises. Therefore, z and y_k are not adjacent and hence $z \neq f$ (recall that f is the father of y_k and $fv''_k \in E$, $fx_k \notin E$). By Lemma 1, $zf \in E$. Since $\sigma(t) > \sigma(v''_k)$ and $fv''_k \in E$, by properties (P3) and (P1), we get $\sigma(z) \geq \sigma(f(v''_k)) \geq \sigma(f)$, i.e. $\sigma(z) > \sigma(f)$. Consequently, $\sigma(t) > \sigma(y_k)$. Now, vertex z cannot be adjacent to x_k, since this would apply the adjacency of z with y_k, too (by $\sigma(t) > \sigma(y_k) > \sigma(x_k)$ and property (P4)). But then, in the cycle $(z, t, x_k, v'_k, v''_k, f, z)$ only chords zv'_k, zv''_k, fv'_k, ft are possible, which are not enough to avoid an induced cycle of length greater than 3 in G. A contradiction with the chordality of G completes the proof of the theorem.

4 *k*-chordal Graphs

As mentioned in the introduction, the examples that show that LBFS fails to find vertices of high eccentricity all have large induced cycles. Furthermore, both chordal and AT-free graphs have constant bounds on the maximum size of induced cycles. Thus one would hope that for k-chordal graphs where k is a constant, some form of BFS would succeed in finding a vertex whose eccentricity is within some function of k of the diameter. In fact, we show that LL is sufficiently strong to ensure this. First, a lemma that is used in the proof.

Lemma 3. *If vertices a and b of a disk $D_r(u)$ of a k-chordal graph are connected by a path $P(a,b)$ outside of $D_r(u)$ (i.e., $P(a,b) \bigcap D_r(u) = \{a,b\}$), then* $\mathrm{d}(a,b) \leq \lfloor k/2 \rfloor$.

Proof. Assume $\mathrm{d}(a,b) > \lfloor k/2 \rfloor$, and let P be an induced subpath of $P(a,b)$ connecting vertices a and b. Consider shortest paths $P(a,u)$ and $P(b,u)$ (connecting a with u and b with u, respectively). Using vertices of these paths we can construct an induced path $Q(a,b)$ with the property that all its vertices except a and b are contained in $D_{r-1}(u)$. By our construction, the cycle C obtained by the concatenation of P and $Q(a,b)$ is induced. Since $\mathrm{d}(a,b) > \lfloor k/2 \rfloor$, both paths P and $Q(a,b)$ must be of length greater than $\lfloor k/2 \rfloor$. Therefore, the cycle C has the length at least $\lfloor k/2 \rfloor + 1 + \lfloor k/2 \rfloor + 1 > k$, that is impossible.

Theorem 2. *Let G be a k-chordal graph. Consider a LL, starting in some vertex u of G and let v be a vertex of the last BFS layer. Then* $\mathrm{ecc}(v) \geq \mathrm{diam}(G) - \lfloor k/2 \rfloor - 2$.

Proof. Let x, y be a pair of vertices such that $\mathrm{d}(x,y) = \mathrm{diam}(G)$, and consider two shortest paths $P(x,v)$ and $P(y,v)$ connecting vertex v with x and y, respectively. Let also q be the minimum index such that

$$L_q \bigcap (P(x,v) \bigcup P(y,v)) \neq \emptyset.$$

Consider a vertex $z \in L_q \bigcap (P(x,v) \bigcup P(y,v))$, and assume, without loss of generality, that z belongs to $P(y,v)$, i.e., we have $\mathrm{d}(v,z) + \mathrm{d}(z,y) = \mathrm{d}(v,y)$. Assume, for now, that $z \neq u$. Consider also a shortest path $Q(x,u)$ between x and u, vertices $x' \in L_q \bigcap Q(x,u)$, $x'' \in L_{q-1} \bigcap Q(x,u)$, and a neighbour w of z in L_{q-1}. Since vertices x'' and w belong to $D_{q-1}(u)$ and can be connected outside of $D_{q-1}(u)$, by Lemma 3, $\mathrm{d}(x'', w) \leq \lfloor k/2 \rfloor$ must hold. We have also $\mathrm{d}(x, x') \leq \mathrm{d}(v,z)$ because $v \in L_{\mathrm{ecc}(u)}$. Therefore, $\mathrm{d}(x,y) \leq \mathrm{d}(x,x') + 1 + \mathrm{d}(x'',w) + 1 + \mathrm{d}(z,y) \leq \mathrm{d}(v,z) + \lfloor k/2 \rfloor + 2 + \mathrm{d}(z,y) = \mathrm{d}(v,y) + \lfloor k/2 \rfloor + 2 \leq \mathrm{ecc}(v) + \lfloor k/2 \rfloor + 2$, i.e., $\mathrm{ecc}(v) \geq \mathrm{diam}(G) - \lfloor k/2 \rfloor - 2$. Finally, if $z = u$, then a similar argument as above establishes a tighter bound on $\mathrm{ecc}(v)$.

This result can be strengthened further for 4-chordal graphs and 5-chordal graphs. We can prove the following.

Theorem 3. *Let G be a 5-chordal graph. Consider a LL, starting in some vertex u of G and let v be a vertex of the last BFS layer. Then* $\mathrm{ecc}(v) \geq \mathrm{diam}(G) - 2$.

Again this bound on $\mathrm{ecc}(v)$ is tight. Figure 8 represents a 4-chordal graph G for which a LBFS exists such that the vertex v, last visited by this LBFS, has eccentricity equal to $\mathrm{diam}(G) - 2$.

Examples of k-chordal graphs with larger difference between diameter and the eccentricity of the vertex last visited in some variant of BFS are given in the following two figures. Figure 13 shows a 6-chordal graph. For this graph G, there is an LL, starting in vertex u, that ends in vertex v, where $\mathrm{ecc}(v) = \mathrm{diam}(G) - 3$. This shows that the bound of $\mathrm{diam}(G) - 2$ which holds for 4- and 5-chordal graphs (for LL and LL+) can not hold for 6-chordal graphs as well.

In Figure 14 each of the dashed edges stands for a path of length k. Thus this graph G is $4k$-chordal. The diameter of G is also $4k$, the eccentricity of v, the vertex last visited by some LBFS started in u, is $\mathrm{ecc}(v) = 2k + 1$. Hence the difference between the diameter and the eccentricity $\mathrm{ecc}(v)$ is $2k - 1$. This shows that, at least for the $4k$-chordal graphs, the bound on $ecc(v)$ (for LL) given in Theorem 2 is close to the best possible. It is within 3 of the bound that could be achieved by LBFS.

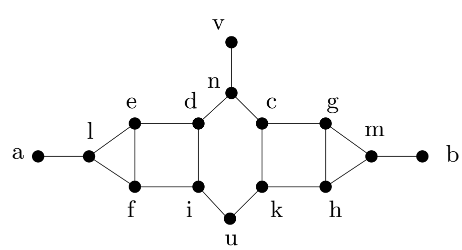

Fig. 13. LL: u|ik|fdch|lengm|abv

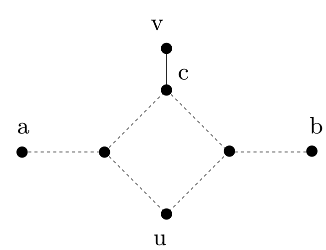

Fig. 14. LBFS: u|... |acb|v

Acknowledgements:

D.G.C. and E.K. wish to thank the Natural Science and Engineering Research Council of Canada for their financial support.

References

1. D. Aingworth, C. Chekuri, P. Indyk and R. Motwani, Fast estimation of diameter and shortest paths (without matrix multiplication), SIAM J. on Computing, 28 (1999), 1167–1181.

2. D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progression, Proceedings of the 19th ACM Symposium on Theory of Computing, 1987, 1–6.
3. V.D. Chepoi and F.F. Dragan, Linear-time algorithm for finding a central vertex of a chordal graph, "Algorithms - ESA'94 " Second Annual European Symposium, Utrecht, The Netherlands, September 1994, Springer, LNCS 855 (Jan van Leeuwen, ed.), (1994), 159–170.
4. D.G. Corneil, F.F. Dragan, M. Habib and C. Paul, Diameter determination on restricted graph families, Discrete Appl. Math., 113 (2001), 143–166.
5. D.G. Corneil, S. Olariu and L. Stewart, Linear time algorithms for dominating pairs in asteroidal triple–free graphs, SIAM J. on Computing, 28 (1999), 1284–1297.
6. D. Dor, S. Halperin, and U. Zwick, All pairs almost shortest paths, Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, 1996, 452–461.
7. F.F. Dragan, Almost diameter of a house-hole-free graph in linear time via LexBFS, Discrete Appl. Math., 95 (1999), 223–239.
8. F.F. Dragan, F. Nicolai and A. Brandstädt, LexBFS–orderings and powers of graphs, Proc. of the WG'96, LNCS 1197, (1997), 166–180.
9. G. Handler, Minimax location of a facility in an undirected tree graph, Transportation Sciences, 7 (1973), 287–293.
10. C.G. Lekkerkerker and J.C. Boland, Representation of a finite graph by a set of intervals on the real line, Fundamenta Mathematicae, 51 (1962), 45–64.
11. D. Rose, R.E. Tarjan and G. Lueker, Algorithmic aspects on vertex elimination on graphs, SIAM J. on Computing, 5 (1976), 266–283.
12. R. Seidel, On the all-pair-shortest-path problem, Proceedings of the 24th ACM Symposium on Theory of Computing, 1992, 745–749.
13. Uri Zwick, Exact and Approximate Distances in Graphs - A survey, "Algorithms - ESA'01 " 9th Annual European Symposium, Aarhus, Denmark, August 2001, Springer, LNCS 2161, (2001), 33–48.