Discrete Applied Mathematics 15 (1986) 315-327
North-Holland

ON BF-ORDERABLE GRAPHS*

Kurt MEHLHORN and Bernd H. SCHMIDT
Fachbereich 10, Angewandte Mathematik und Informatik, Universität des Saarlandes, 6600 Saarbrücken, West Germany

Received 16 April 1986

We introduce BF-orderable graphs as a generalization of acyclic graphs. BF-orderable graphs permit a linear time solution to the single source shortest path problem. We give a graph-theoretic characterization of BF-orderable graphs by forbidden subgraphs.

1. Introduction

Let $N=(V,E,s,f)$ be a network, where (V,E) is a directed graph, $s\in V$ is a distinguished vertex called the source, and $f:E\to\mathbf{R}$ is a cost function on the set of edges. The single source least cost path problem is to compute for each $v\in V$ the cost of a cheapest path from s to v, i.e., to compute

$$d(v)=\begin{cases}\inf\{cost(p);\ p \text{ is a path from } s \text{ to } v\}, & \text{if } v \text{ is reachable from } s;\\ \infty, & \text{otherwise}\end{cases}$$

for all $v\in V$. The cost of a path $p=e_0e_1\cdots e_{k-1}$ is the sum of the costs of its constituent edges, i.e., $cost(p)=\sum_{i=0}^{k-1} f(e_i)$.

A classical algorithm by Bellman and Ford solves the single source least cost path problem in time $O(ne)$, where $n=|V|$ and $e=|E|$; cf. [1] or [4, Section IV.7.3]. In many applications however, e.g., in VLSI design, very large graphs arise, and an $O(ne)$-time algorithm is unacceptable slow. It is therefore interesting to know classes of graphs for which a faster solution is possible. A wellknown such class is the class of acyclic graphs for which an $O(e)$-time algorithm exists. In this paper we study the following generalization of acyclic graphs:

Definition 1. Let $G=(V,E)$ be a directed graph and let $s\in V$. The graph G is *BF-orderable* (with respect to source s) if there is an ordering of the edges of G such that the edges on any simple path starting in s occur in increasing order. More precisely, there is an injective mapping $num:E\to\{1,\dots,|E|\}$ such that for all simple paths $p=e_0e_1\cdots e_{k-1}$ starting in s, we have $num(e_i)<num(e_j)$ for $0\le i<j\le k-1$. We shall call any such mapping a BF-order for G (with respect to source s).

* This research was supported by DFG, SFB 124, VLSI-Entwurf und Parallelität.

Fig. 1 shows a BF-orderable graph which is not acyclic.

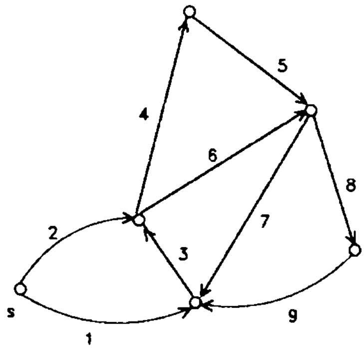

Fig. 1. A BF-orderable graph. A BF-order is indicated by labels. Note that no simple path starting in s uses the edges labelled 9 and 3 or the edges labelled 7 and 3.

Our definition is motivated by the following observation (cf. Theorem 1): In the Bellman–Ford algorithm costs are propagated along edges. The $O(ne)$ running time results from the fact that it may be necessary to propagate along a given edge as many as n times. However, if the graph is BF-orderable and edges are selected for propagation in some BF-order, then one needs to propagate only *once* along every edge.

The main contribution (Theorem 2) of this paper is a graph-theoretic characterization of BF-orderable graphs by forbidden subgraphs. We give three simple conditions which together define the class of BF-orderable graphs. The first two conditions impose restrictions on single simple cycles, and the third condition concerns the interaction between two simple cycles.

Before proving Theorem 1 we discuss previous related work. Leierson and Pinter [2] reduced the placement problem for river routing to a single source least cost path problem on a restricted class of graphs and then showed that this class of graphs admits of a linear time solution algorithm. In their algorithm they construct a BF-order *num* and then use it to compute shortest paths. In other words, all graphs arising in their problem are BF-orderable.

The compaction problem for VLSI circuits also leads to single source least cost path problems. Lengauer and Mehlhorn [3] have shown that the one-dimensional compaction of stick diagrams reduces to a least cost path problem on a graph with the property that every strongly connected component is a 'path' of antiparallel edges as shown in Fig. 2. Again, it is not difficult to come up with an appropriate BF-order.

These two examples show that BF-orderable graphs occur naturally in practice and lead to the question of characterizing this class of graphs. Note that the graph shown in Fig. 1 neither belongs to the class considered by Leierson and Pinter nor satisfies the strong condition on strongly connected components mentioned above.

Fig. 2. A strongly connected component.

This paper is organized as follows. In Section 2 we show that the Bellmann–Ford algorithm runs in linear time on BF-orderable graphs provided that a BF-order is available. In Sections 3 and 4 we give two characterizations on BF-orderable graphs by forbidden subgraphs. In Section 5 we show that we can decide in time $O(|E|^2)$ whether a graph is BF-orderable and also compute a BF-order if it is. We conjecture that a much faster algorithm exists.

The objection might be raised at this point that the results are worthless in practice because an efficient recognition algorithm for BF-orderable graphs is missing. There are two answers to this objection. Firstly, computing a BF-order is preprocessing and pays off if the same path problem has to be solved for many cost functions. Secondly, for some classes of BF-orderable graphs (e.g. the classes studied by Leierson/Pinter and Lengauer/Mehlhorn) the conditions (BF1) to (BF3) (not (BF4) though) are readily verified by a human. Thus the characterization given in this paper might lead to the discovery of more subclasses which are practically relevant and can be efficiently ordered.

2. Single source least cost path problems on BF-orderable graphs

Theorem 1. *Let $G=(V,E,s)$ be a BF-orderable source graph and let $e_1,\dots,e_{|E|}$ be a BF-order of E. Then the single source least cost path problem may be solved in linear time for any network (V,E,s,f).*

Proof. Consider the following algorithm:

$D[s]:=0$;
$D[v]:=\infty$ for all $v\in V\setminus\{s\}$;
for i **from 1 to** $|E|$ **do**
let $e_i=(v,w)$;
$D[w]:=\min\{D[w],D[v]+f(e)\}$;
comment $\infty+a=\infty$ for $a\in\mathbf{R}$ **od**

For all $v \in V$, let $d(v)$ be as defined in the introduction, and let $d'(v)$ denote the final value of $D[v]$ in the above algorithm. Finally, for $e \in E$, the phrase "when e has just been considered" is shorthand for "at that point in time when execution reaches the end of the loop body for the *num*(e)-th time".

Claim 1. *$d'(v) = d(v)$ for all $v \in V$ with $d(v) > -\infty$.*

Proof. Induction on i shows that for all $w \in V$, any value $\neq \infty$ assigned to $D[w]$ is the cost of a path from s to w. Hence $d'(w) \geq d(w)$ for all $w \in V$. If $d(v) > -\infty$, then $d(v)$ is the cost of a simple path

$$p = \left(s = v_0 \xrightarrow{e_0} v_1 \xrightarrow{e_1} \cdots \xrightarrow{e_{k-1}} v_k = v\right)$$

from s to v. Note that the edges on p are considered by the algorithm in the same order as that in which they occur on p. It is now an easy induction on j to show that when e_j has just been considered, $0 \leq j \leq k$, then $D[v_j] \leq \sum_{i=0}^{j-1} f(e_i)$. Hence $d'(v) \leq cost(p) = d(v)$, and we may conclude that $d'(v) = d(v)$. □

Claim 2. *Let $w \in V$. Then $d(w) = -\infty$ if and only if there is an edge $e = (u, v) \in E$ such that*

(1) $d'(v) > d'(u) + f(e)$ and

(2) w is reachable from v.

Proof. Assume first that $d(w) = -\infty$. Then there is a path $p_1 p_2 p_3$ from s to w such that p_2 is a cycle and $cost(p_2) < 0$. Let $p_2 = (v_0 \xrightarrow{e_0} v_1 \xrightarrow{e_1} \cdots \xrightarrow{e_k} v_k)$ with $v_0 = v_k$. Since

$$\begin{aligned}\sum_{i=0}^{k-1} (d'(v_{i+1}) - d'(v_i) + f(e_i)) &= (d'(v_k) - d'(v_0)) - \sum_{i=0}^{k-1} f(e_i) \\ &= 0 - cost(p_2) > 0,\end{aligned}$$

there is at least one i, $0 \leq i \leq k-1$, such that $d'(v_{i+1}) > d'(v_i) + f(e_i)$. Furthermore, w is clearly reachable from v_{i+1}.

Now let $e = (u, v)$ be an edge with $d'(v) > d'(u) + f(e)$. Since $d(v) > d(u) + f(e)$ is impossible, we may conclude from Claim 1 that $d(u) = -\infty$ or $d(v) = -\infty$. But then clearly $d(w) = -\infty$ for any vertex w reachable from v. This completes the proof of Claim 2. □

Armed with Claims 1 and 2, it is now easy to complete the algorithm:

> $W := 0$;
> **for** all $e = (u, v) \in E$ **do**
> **if** $D[v] > D[u] + f(e)$ **then** $W := W \cup \{v\}$ **fi**
> **od**
> $D[w] := -\infty$ for all $w \in V$ reachable from some vertex in W;

This algorithm can clearly be made to run in linear time. □

3. A graph-theoretic characterization of BF-orderable graphs

We use the following graph-theoretic notations. Let $G=(V,E)$ be a digraph and let $s\in V$. For an edge $e=(v,w)\in E$ we write $v=source(e)$ and $w=sink(e)$ and $e^{-1}=(w,v)$ for the reverse edge. A path is a sequence $e_0,e_1,\ldots,e_{k-1}$ of edges with $sink(e_i)=source(e_{i+1})$ for $0\le i<k-1$. A path is an s-path if $source(e_0)=s$. A path is a cycle if $source(e_0)=sink(e_{k-1})$ and $k\ge 3$, i.e., antiparallel edges ee^{-1} are not considered a cycle. A simple cycle is a simple path which is a cycle.

Let P be a set of edges. Then $V(P)=\{sink(e), source(e); e\in P\}$ is the set of end-points of edges in P. If $W\subsetneq V$ is a set of nodes then $w\in W$ is an *entry point* of W if there is a simple s-path ending in w and not going through any other node of W.

We are now ready to define the three characterizing properties

Definition 2. A source graph $G=(V,E,s)$ has *property BF1* if every simple cycle C has at most two entry points.

Lemma 1. *Every BF-orderable graph has property BF1.*

Proof. Assume otherwise, i.e. there is a simple cycle C with three entry points, say u, v and w. Write $C=p_1p_2p_3$ where p_1 runs from u to v, p_2 runs from v to w and p_3 runs from w to u. Let p be a simple s-path which witnesses that u is an entry point of C. Then pp_1p_2 is a simple path and hence all edges in p_1 must have smaller numbers than all edges in p_2. The same reasoning applies to p_2 and p_3 and to p_3 and p_1. Hence the graph is not BF-orderable. □

Definition 3. Let G be a graph satisfying BF1. It is said to satisfy BF2 if for every simple cycle C with two entry points the two entry points of C are neighbours on C.

Lemma 2. *Every BF-orderable graph has property BF1 and BF2.*

Proof. Similar to the proof of Lemma 1 and therefore left to the reader. □

For the third property we need an additional concept. Let C be a simple cycle in a graph satisfying BF1 and BF2. Assume that C has two entry points u, v with edge (u,v) being an edge of cycle C. Then v is called the second entry point of C and u is called the first entry point of C. If a cycle C has only one entry point then call this point also the first entry point of C.

Definition 4. Let G be a graph satisfying BF1 and BF2. It is said to satisfy BF3 if there is no pair of simple cycles C_1 and C_2 such that

(a) C_1 and C_2 have two entry points each; say u_i is the first and v_i is the second entry point of C_i, $i=1,2$.

(b) $V((C_1)-\{u_1\})\cap V(C_2)=\emptyset$ and $V(C_1)\cap(V(C_2)-\{u_2\})=\emptyset$, i.e., C_1 and C_2 are vertex disjoint except for the possibility $u_1=u_2$.

(c) There is a path p from u_1 to u_2 and the antiparallel path p^{-1} from u_2 to u_1 with $V(p)\cap((V(C_1)-\{u_1\})\cup(V(C_2)-\{u_2\}))=\emptyset$.

(d) v_1 and v_2 are entry points of the set $V(C_1)\cup V(C_2)\cup V(p)$.

Lemma 3. *If $G=(V,E)$ with source $s\in V$ is BF-orderable, then G satisfies BF1, BF2 and BF3.*

Proof. We have already shown that G satisfies BF1 and BF2. So let us assume that it does not satisfy BF3. Let $C_1, C_2, v_1, v_2, u_1, u_2$ and p be as in the definition of BF3, and let p_i be a s-path to v_i which wittnesses that v_i is an entry point of $V(C_1)\cup V(C_2)\cup V(p)$, cf. Fig. 3. Let $C_i=(u_i,v_i)q_i$ for $i=1,2$, i.e. q_i is the part of the cycle C_i from v_i to u_i. Let e_i be the first edge of q_i. Next observe that the path $p_1q_1p(u_2,v_2)e_2$ is simple and hence $num(e_1)<num(e_2)$ and that the path $p_2q_2p^{-1}(u_1,v_1)e_1$ is simple and hence $num(e_2)<num(e_1)$. Thus G is not BF-orderable. □

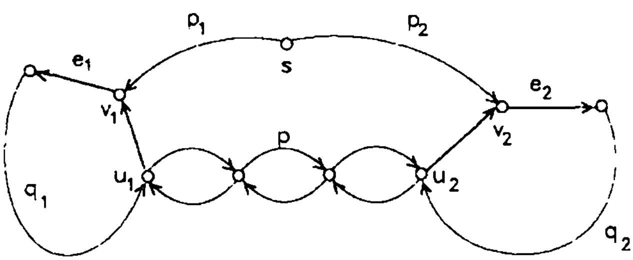

Fig. 3. A counterexample to BF3.

We are now ready for the main theorem.

Theorem 2. *If G satisfies BF1, BF2 and BF3, then G is BF-orderable.*

Proof. Let $G=(V,E)$ be a source graph. We consider the following auxiliary graph $G'=(E,K)$ with $(e_1,e_2)\in K$ iff there is a simple s-path $p=qe_1,e_2$, i.e., a simple s-path having e_1 and e_2 as its last two edges. Note that the vertex set of G' is the edge set of G.

If G' were acyclic, then G is clearly BF-orderable, since a topological order of G' is a BF-order of the edges of G. So let us assume for the sake of a contradiction that G' is not acyclic.

Let $Z=(e_0,e_1),(e_1,e_2),\ldots,(e_{k-1},e_0)$ be a simple cycle in G' and let $e_i=(v_i,v_{i+1})$ where indices are mod k. Then $p(Z)=e_0,e_1,\ldots,e_{k-1}$ is a cycle in G, cf. Fig. 4.

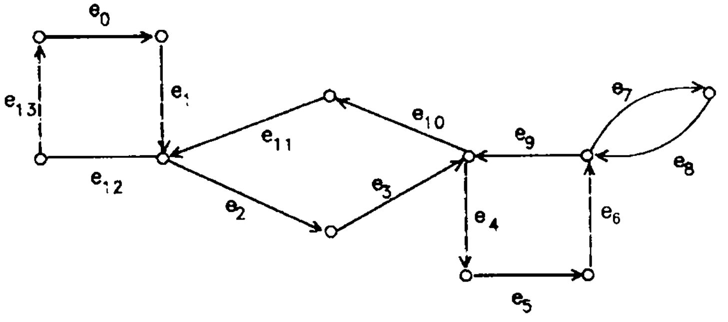

Fig. 4. A cycle Z in G'.

The cycle $p(Z)$ is not necessarily simple in G. Let $B(Z)$ be the set of double points in $p(Z)$, i.e.

$$B(Z)=\{i;\ \exists j: 0\le j\le k-1, j\ne i, v_i=v_j\}$$

and let $b(Z)=|B(Z)|$ be the number of double points in $B(Z)$. Note that double points are counted with multiplicity. Also note that k is at least three since $(e,e')\in K$ implies $sink(e)=source(e')$ and $source(e)\ne sink(e')$. From now on we consider a fixed simple cycle Z defined as follows. Z is chosen such that $b(Z)$ is minimal. Among the cycles with minimal $b(Z)$ we choose the shortest one.

Assume first that $b(Z)=0$. Then $p(Z)$ is a simple cycle in G. We show that $p(Z)$ violates either BF1 or BF2. Assume otherwise. We may assume w.l.o.g. that v_0 is the first entry point of $p(Z)$. Then the set of entry points of $p(Z)$ is contained in $\{v_0, v_1\}$ or BF1 or BF2 is violated. Let q be an s-path such that $qe_{k-1}e_0$ is simple. We can write $qe_{k-1}e_0=q'q''e_{k-1}e_0$ where q' ends in an entry point of $p(Z)$. Since the path $qe_{k-1}e_0$ is simple this entry point must be different from v_0 and v_1. Thus either BF1 or BF2 is violated.

It remains to consider the case that $b(Z)\ne 0$. We will first characterize the shape of simple cycles in G formed by edges appearing in Z.

Lemma 4. *Let $i_0, i_1, \ldots, i_{j-1}$ be such that $C=e_{i_0}, e_{i_1}, \ldots, e_{i_{j-1}}$ is a simple cycle in G with first entry point $source(e_{i_0})$. Then*

(a) $i_{l+1}=i_l+1 \pmod k$ *for $0\le l\le j-2$, i.e., the edges of C occur consecutively on $p(Z)$.*

(b) *The cycle C has two entry points and $V(C)\cap B(Z)=\{source(e_{i_0})\}$, i.e., exactly the first entry point of C is a double point of $p(Z)$.*

Proof. (a) Let $u=source(e_{i_0})$ be the first entry point of C and let q be an s-path to u which verifies this property. Then $qe_{i_0}e_{i_1}\cdots e_{i_l}$ is simple for $l<j-1$ and hence $(e_{i_{l-1}}, e_{i_l})\in K$ for $l<j-1$. Thus $i_l=i_{l-1}+1 \pmod k$ for $1\le l<j-1$ by minimality of Z. It remains to show that $i_{j-1}=i_{j-2}+1 \pmod k$. Assume otherwise. By minimality

of Z it suffices to show that $(e_{i_{j-2}}, e_{i_{j-1}}) \in K$. Choose path q such that $p = qe_{i_{j-1}}$ is a simple s-path (cf. Fig. 5).

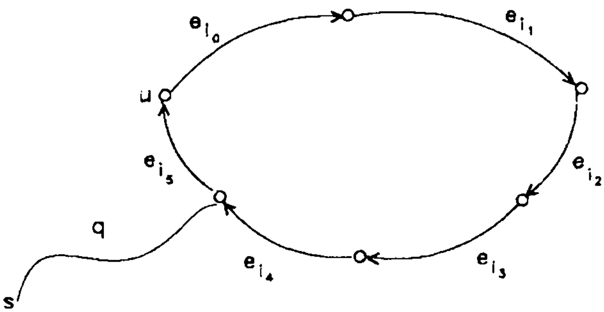

Fig. 5. The cycle C in the case $j = 6$.

The path q must enter cycle C in a node different from u (simple p is simple). Since u is the first entry point, q must enter C in node $v = sink(e_{i_0})$, say $q = q_1 q_2$ with q_1 ending in v. But then $q_1 e_{i_1} \cdots e_{i_{j-2}} e_{i_{j-1}}$ is a simple s-path and hence $(e_{i_{j-2}}, e_{i_{j-1}}) \in K$. This proves part (a) and also the claim that C has two entry points.

(b) We have to show that the only vertex in $V(C) \cap B(Z)$ is u. In order to simplify notations we may assume w.l.o.g. that $i_l = l$ for $0 \leq l \leq j-1$, i.e., $C = e_0, e_1, \ldots, e_{j-1}$.

Let $e_i = (v_i, v_{i+1})$ with $v_j = v_0 = u$. Assume that $Z(C) \cap B(Z)$ contains a vertex different from u. Choose $z \in V(C) \cap B(Z)$, $z \neq u$, $z = v_{r+1}$ with minimal r, i.e., z is a double-point different from u but as close to u as possible. Then there are edges e_r, e_{r+1} of C with $sink(e_r) = z = source(e_{r+1})$ and there are edges e_t, e_{t+1} of $p(Z) - C$ with $sink(e_t) = z = source(e_{t+1})$; (cf. Fig. 6.)

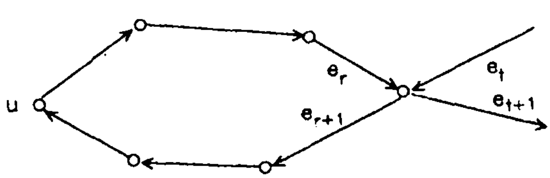

Fig. 6. A double point.

We observe first that $(e_r, e_{t+1}) \notin K$ and $(e_t, e_{r+1}) \notin K$. Assume otherwise. If $(e_r, e_{t+1}) \in K$, then $e_0, \ldots, e_r, e_{t+1}, \ldots, e_{k-1}$ is a simple cycle in G' and if $(e_t, e_{r+1}) \in K$, then $e_{r+1}, \ldots, e_t$ is a simple cycle in G'. In both cases we have a contradiction to the minimality of Z.

Let $qe_t e_{t+1}$ be a simple s-path. Then $qe_t e_{r+1}$ is not simple since $(e_t, e_{r+1}) \notin K$. We can factor q as $q = q_1 q_2$ with q_1 minimal ending in an entry pont of C, i.e., q_1 ends in v_h for $h \in \{0, 1\}$. Consider the path $p = q_1 e_h e_{h+1} \cdots e_r e_{t+1}$.

This path cannot be simple since $(e_r, e_{t+1}) \notin K$. By the choice of q_1 we conclude that $sink(e_{t+1})$ lies on $q_1 e_h \cdots e_r$. Since $q_1 q_2 e_t e_{t+1}$ is simple the vertex $sink(e_{t+1})$ must be an endpoint of an edge in $\{e_{h+1}, \dots, e_{r-1}\}$, i.e., $sink(e_{t+1})$ must belong to vertex set $\{v_{h+1}, \dots, v_r\} \subseteq \{v_1, \dots, v_r\}$. Thus vertex $sink(e_{t+1})$ belongs to $V(C) \cap B(C)$. This is a contradiction to the choice of z, because $sink(e_{t+1}) \neq u$, $sink(e_{t+1}) \neq z$ and $sink(e_{t+1})$ is closer to u than z.

We have so far shown that $V(C) \cap B(Z) \subseteq \{u\}$. Since $p(Z)$ is not simple we have $V(C) \cap B(Z) \neq \emptyset$ and hence $V(C) \cap B(Z) = \{u\}$. $\square$

Lemma 4 gives us detailed information about the structure of $p(Z)$. Only consecutive edges of $p(Z)$ can form a simple cycle; every such cycle has two entry points and exactly the first entry point is a double point of $p(Z)$. In particular every edge e of $p(Z)$ is part of at most one such cycle and two such cycles are either vertex-disjoint or agree in exactly their first entry point. We call an edge e of $p(Z)$ a cycle edge if it is a part of a simple cycle formed from edges of $p(Z)$ and a base edge otherwise. Consider a base edge $e = (v, w)$. Since $p(Z)$ is a cycle there must be a simple subpath of $p(Z)$ which starts in w and ends in v and since e is a base edge this subpath can consist of exactly one edge. Thus, if e is a base edge, then e^{-1} is also an edge of $p(Z)$. We claim that e^{-1} is also a base edge. Assume otherwise, i.e., there is a simple cycle C formed from edges of $p(Z)$ and containing e^{-1}. Then $\{source(e), sink(e)\} \subseteq V(C) \cap B(Z)$, a contradiction to Lemma 4. This proves that e^{-1} is also a base edge.

Let B be the set of base edges and let $V(B)$ be the set of endpoints of base edges. If we view a pair $\{e, e^{-1}\}$ of base edges as an undirected edge connecting $sink(e)$ and $source(e)$, then $(V(B), B)$ is a forest. Since every cycle (formed from edges of $p(Z)$) has exactly one double point it is even a tree. We call $(V(B), B)$ the base tree, (cf. Fig. 7).

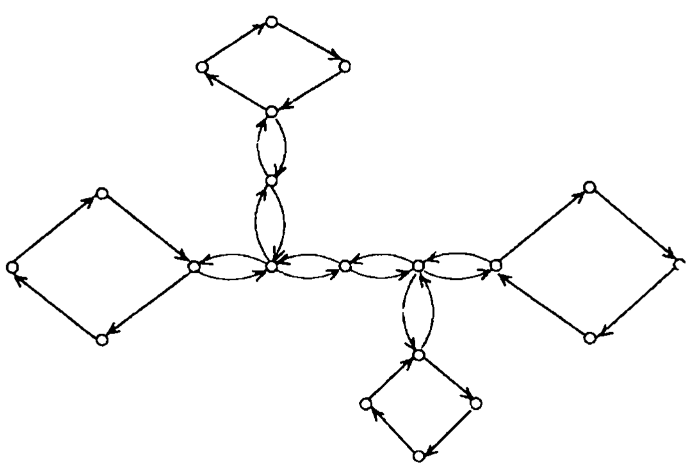

Fig. 7. The base tree and the attached cycles.

Let us summarize what we achieved at this point. Cycle $p(Z)$ consists of a base tree $(V(B), B)$ and simple cycles attached to some of the vertices of the base tree. Clearly, there must be a simple cycle attached to every leaf of the base tree because $(e, e^{-1}) \in K$ for some base edge of e otherwise. But $(e, e^{-1}) \in K$ is impossible. Thus there are at least two cycles attached to the base tree.

Assume next that there are two simple cycles C and C' with first entry points u and u' and second entry points v and v' such that v and v' are entry points of $V(p(Z))$. Let p be the simple path of base edges from u to u'. Then p^{-1} is a path of base edges from u' to u and hence C, C' together with path p form a contradiction to property BF3. In view of this argument the proof of Theorem 2 is now completed by

Lemma 5. *Let C with first entry point u and second entry point v be one of the simple cycles formed by the edges of $p(Z)$. Then v is an entry point of $V(p(Z))$.*

Proof. We assume otherwise and derive a contradiction. Let $Q = \{q; q$ is a simple s-path ending in v and not going through $u\}$. Then $Q \neq \emptyset$ since v is an entry point of C. Let $q \in Q$ be arbitrary. Since v is not an entry point of $V(p(Z))$ we can write $q = q_1 q_2$ where $source(q_2) \in V(p(Z))$, $q_2 \neq \varepsilon$, and no intermediate vertex of q_2 is in $V(p(Z))$. Define vertex $a(q)$ as $source(q_2)$. Then $a(q) \neq v$ and $a(q) \neq u$.

We will next select a particular $p \in Q$ as follows. If there is a $q \in Q$ with $a(q) \notin B(Z)$, then let p be such a path. If $a(q) \in B(Z)$ for all $q \in Q$, then let $p \in Q$ be such that there is no $q' \in Q$ such that the simple base path from u to $a(q')$ goes through $a(q)$. In other words, if we view the base tree as a rooted tree with root u, then the subtree with root $a(p)$ does not contain $a(q')$ for any $q' \in Q$.

Let p be as defined above. Let $a = a(p)$, write $p = p_1(a, x)p_2$ and let (b, a) be the last edge on the base path from u to a (if $a \in B(Z)$) or the unique edge in $p(Z)$ ending in a (if $a \notin B(Z)$), cf. Fig. 8.

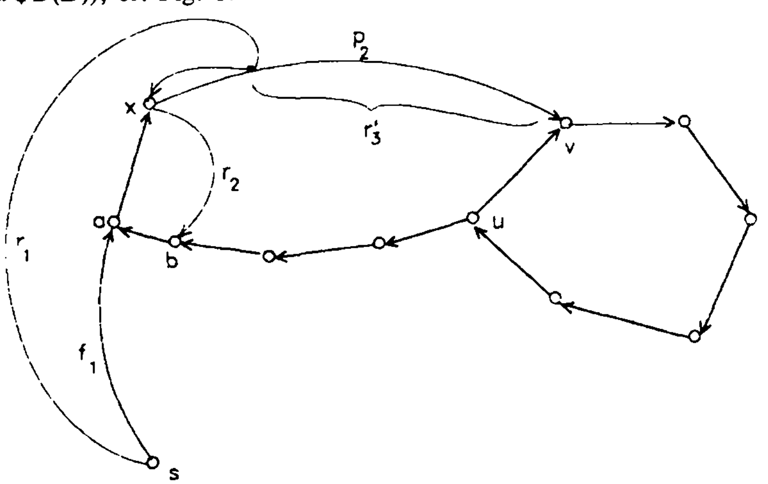

Fig. 8. The notation in lemma 5.

Claim. $((b,a),(a,x)) \in K$.

Proof. If there is a simple s-path having (b,a) as its last edge and not going through x, then $((b,a),(a,x)) \in K$. So let us assume that every simple s-path having (b,a) as its last edges goes through x. Since $((b,a),(a,c)) \in K$ where (a,c) is the edge following (b,a) on $p(Z)$ there is at least one such path. Let r be any such path. Write $r = r_1 r_2$ with $x = source(r_2)$.

Consider s-path $r_1 p_2$. This path is not necessarily simple. We can write $r_1 p_2 = r_1' r_2' r_3'$ where $r_1' r_3'$ is a simple s-path to v and hence $q := r_1' r_3' \in Q$. Also r_1' is a prefix of r_1 and r_3' is a suffix of p_2.

Let $d = a(q)$ and write $q = q_1 q_2$ with $source(q_2) = d$. We observe first that q_1 is a proper prefix of r_1 (if $p_2 = \varepsilon$ and hence $x = v$ this follows from the fact that $q_2 \neq \varepsilon$ by definition of $a(q)$. If $p_2 \neq \varepsilon$ and hence $x \notin V(p(Z))$ this follows from the fact that only the last point of p_2 and hence r_3' is in $V(p(Z))$.). We observe next that q_1 does not go through either x, a or b. (It does not go through x since $x = sink(r_1)$ and q is a proper prefix of r_1 and it does not go through a and b because (b,a) was the last edge of the simple s-path $r = r_1 r_2$.)

Let q_3 be the unique simple path from d to a formed from edges of $p(Z)$. Then (b,a) is the last edge of q_3. This can be seen as follows. If $a \notin B(Z)$, then (b,a) is clearly the last edge of q_3 since (b,a) is the only edge of $p(Z)$ ending in a. If $a \in B(Z)$, then $d \in B(Z)$ by our choice of a and q_3 in a path of base edges. Also, if (a,b) were not the last edges of q_3, then d lies in the subtree of the base tree rooted at a, contradicting the choice of a. Thus (b,a) is the last edge of q_3.

Finally, consider path $q_1 q_3$. It does not go through x and has (b,a) as its last edge. Thus $((b,a),(a,x)) \in K$ and the claim is proved. □

The proof of Lemma 5 is now readily completed. Let t be the subpath of $p(Z)$ starting in v and having (b,a) as its last edge. Consider cycle $t(a,x)p_2$. Then $(e,e') \in K$ for consecutive edges of this cycle except maybe for the last edge of p_2, say e, and the first edge of t, say e'. But pe' is a simple path and hence $(e,e') \in K$ since p is a simple path ending in v, not going through u and since u and v are the only entry points of C. Thus $t(a,x)p_2 = p(Z')$ where Z' is a simple cycle in G'. Also $b(Z') < b(Z)$ since no vertex of p_2 is a double point of Z' and since the multiplicity of u as a double point is one less in Z' then in Z. This contradicts the choice of Z and hence completes the proof of Lemma 5 and Theorem 2. □□

4. An alternative characterization

In this section we derive an alternative characterization.

Definition 5. A source graph G has property BF4 if there is no pair e, e' of edges such that there are simple s-paths $p = p_1 e p_2 e'$ and $p' = p_1' e' p_2' e$. □

Theorem 3. *A graph is BF-orderable iff it satisfies conditions BF1 and BF4.*

Proof. If a graph does not satisfy BF4, then it is clearly not BF-orderable. So let us assume that source graph $G=(V,E,s)$ satisfies BF1 and BF4. We will show that G satisfies BF2 and BF3 and hence is BF-orderable.

Assume first that it does not satisfy BF2. Then there is a cycle C having entry points x_1, x_2 which are not neighbors on C. Let e and e' be the edges of C starting in x_1 and x_2 respectively and let p_1 and p_1' be paths which verify that x_1 and x_2 are entry points of C. Let $C=ep_2e'p_2'$. Then the paths p_1ep_2e' and $p_1'e'p_2'$ are simple and hence BF4 is violated.

Assume next that G does not satisfy BF3. Let $C_1, C_2, u_1, u_2, v_1, v_2$, and p be defined as in the definition of property BF3. Let $p_1(p_2)$ be an s-path which verifies that $v_1(v_2)$ is an entry point of set $V(C_1) \cup V(C_2) \cup V(p)$ and let $e(e')$ be the edge of $C_1(C_2)$ starting in $v_1(v_2)$. Then the paths p_1e (part of C_1 from v_1 to u_1) $p(u_2, v_2)e'$ and p_2e' (part of C_2 from v_2 to u_2) $p\ ^{1}(u_1, v_1)e$ are simple and hence BF4 is violated. □

Theorem 3 can also be formulated as follows. Consider the following auxiliary graph. $G_* = (E, E_*)$ where $(e, e') \in E_*$ if there is a simple s-path p_1ep_2e'. It is clear that G is BF-orderable iff G_* is acyclic. Theorem 3 states: if G satisfies BF1, then G_* is acyclic if it contains no cycle of length two.

5. Complexity

We gave two characterizations of BF-orderable graphs by forbidden subgraphs. The proof of Theorem 2 also yields an $O(|E|^2)$ algorithm for deciding whether a graph is BF-orderable and computing a BF-order if it is. This can be seen as follows. In the proof of Theorem 2 we considered the auxiliary graph $G' = (E, K)$ where $(e_1, e_2) \in K$ iff there is a simple s-path pe_1e_2. The graph G' has size $O(|E|^2)$ and G is BF-orderable iff G' is acyclic. Also a topological sort of G' yields a BF-order of G. It is therefore sufficient to show that G' can be computed in time $O(|E|^2)$. For a fixed edge $e_2 \in E$ let $Pred(e_2) = \{e_1 \in E; (e_1, e_2) \in K\}$. We show how to compute $Pred(e_2)$ by depth-first-search (cf. [4, Section IV.5]) on a graph $\tilde{G}$ in time $O(|E|)$. The graph $\tilde{G}$ is obtained from G by deleting all edges leaving $source(e_2)$ and by deleting vertex $sink(e_2)$ with all its incident edges. Next perform a depth-first-search with start vertex s on $\tilde{G}$. Then $source(e_2)$ is a leaf of the DFS-tree and hence $Pred(e_2)$ is exactly the set of tree, forward and cross edges ending in $source(e_2)$. This shows that $Pred(e_2)$ can be computed in time $O(|E|)$ and hence G' can be computed in time $O(|E|^2)$. Finally, a topological sort of G' can be computed in time $O(|E| + |K|) = O(|E|^2)$; cf. [4, Section IV.2]. Thus we can decide in time $O(|E|^2)$ whether a graph $G=(V,E,s)$ is BF-orderable. We believe that this time bound can be improved considerably by checking directly whether properties BF1, BF2, and

BF3 are satisfied. However, we have not been able to do so yet and leave it as a challenge to the reader.

References

[1] R.E. Bellmann, On a routing problem, Quart. Appl. Math. 16 (1958) 87–90.
[2] C.E. Leiserson and R.Y. Pinter, Optimal placement for river routing, SIAM J. Comput. 12 (3) (1983) 447–462.
[3] Th. Lengauer and K. Mehlhorn, VLSI complexity theory, efficient VLSI algorithms and the HILL design system, to appear in: J. Trullemans, ed., Algorithmics in VLSI, Proc. of the International Professorship in Computer Science (Academic Press, New York).
[4] K. Mehlhorn, Data Structures and Efficient Algorithms, Vol 2: Graph Algorithms and NP-completeness, EATCS Monographs in Theoretical Computer Science (Springer, Berlin, 1984).