# On Orientations and Shortest Paths

Refael Hassin
*Department of Statistics*
*Tel Aviv University*
*Tel Aviv, Israel*

and

Nimrod Megiddo
*IBM Research*
*Almaden Research Center*
*San Jose, California 95120-6099*
*and*
*School of Mathematical Sciences*
*Tel Aviv University*
*Tel Aviv, Israel*

Dedicated to Alan J. Hoffman on the occasion of his 65th birthday.

Submitted by Uriel G. Rothblum

ABSTRACT

An orientation of an undirected graph is a choice of direction for each of its edges. An orientation is called ideal with respect to a given set of pairs of vertices if it does not increase the shortest-path distances between the members of any of the pairs. A polynomial-time algorithm is given for constructing an ideal orientation with respect to two given pairs and any positive edge-lengths, or else recognizing that no such orientation exists. Moreover, we show that this problem is in the class NC. For a general number of pairs the problem is proven NP-complete even with unit edge-lengths.

## 1. INTRODUCTION

This note is concerned with orientations of undirected graphs:

Definition 1.1. An *orientation* of an undirected graph $G$ is a directed graph $G'$ obtained from $G$ by replacing each edge of $G$ with a single directed edge.

Robbins [16] suggested an application in the context of traffic control, where orientation means restriction of all roads to be one-way. More results on orientation of graphs can be found in [1], [2], [5], [6], [7], [8], [10], [11], [12], [13], [14], [15], and [19]. We confine attention to orientations that maintain reachability:

Definition 1.2. Given an undirected graph $G = (V, E)$ and $m$ pairs of vertices $(s_i, t_i)$ $(i = 1, \ldots, m)$, an orientation $G'$ of $G$ is called *feasible* if for every $i$ $(i = 1, \ldots, m)$ there exists in $G'$ a path from $s_i$ to $t_i$. The set of pairs $(s_i, t_i)$ is also said to be feasible in this case.

One may view the orientation problem as some kind of a multicommodity flow problem. However, there are substantial differences between this and the usual multicommodity flow problem. The main difference is that there are no capacity constraints in the context of orientation, while multicommodity flow becomes trivial in the absence of such constraints. Furthermore, in multicommodity flow in an undirected network the direction of flow in an edge plays no role (it is only the total flow in both directions that matters). In orientation all the flows in an edge must have the same direction.

## 2. FEASIBILITY OF ORIENTATIONS

In this section we discuss the problem of recognizing feasibility of a set of pairs of vertices:

Problem 2.1. Given is an undirected graph $G = (V, E)$ together with a set S consisting of $m$ pairs of vertices $(s_i, t_i)$ $(i = 1, \ldots, m)$. Find a *feasible* orientation, i.e., an orientation $G'$ of $G$ in which there is a path from $s_i$ to $t_i$ $(i = 1, \ldots, m)$, or conclude that no such orientation exists.

We prove below that feasibility of any set $S$ of pairs of vertices can be recognized in polynomial time. We first discuss the case where S consists of two pairs.

Proposition 2.2. *Given a connected undirected graph $G = (V, E)$ and two pairs of vertices $(i, i')$, $(j, j')$, a feasible orientation $G'$ of $G$ exists if and only if the following condition holds: for every partition of the vertex set $V$ into two disjoint sets $V_1$, $V_2$, if $i, j' \in V_1$ and $i', j \in V_2$, then there must exist at least two edges of $G$ connecting vertices of $V_1$ with ones in $V_2$.*

*Proof.* The condition is obviously necessary. To prove sufficiency, suppose there exists no orientation of $G$ with paths as required. Consider any path $L$ connecting $i$ with $i'$. Let $G_L$ denote a mixed graph (i.e., a graph with both directed and undirected edges) obtained from $G$ by orienting all the edges along $L$ in the direction from $i$ to $i'$. By assumption, there is no path in $G_L$ from $j$ to $j'$. Thus, there is a partition of $V$ into two disjoint sets $V_1$, $V_2$ such that $j' \in V_1$ and $j \in V_2$, and there are no edges of $G_L$ leading from a vertex in $V_2$ to a vertex in $V_1$. However, since there is in $G$ some path connecting $j$ with $j'$, there is at least one edge leading from a vertex in $V_1$ to a vertex in $V_2$. It follows that there exists precisely one such edge and, moreover, $i \in V_1$ and $i' \in V_2$. ∎

It follows from the proof of Proposition 2.2 that the following is true:

COROLLARY 2.3. *Under the conditions of Proposition 2.2, if there exists a feasible orientation, then for every path L connecting i with i′ there exists a feasible orientation G′ which is consistent with the directions on L (from i to i′).*

COROLLARY 2.4. *The existence of a feasible orientation with respect to two given pairs of vertices (including the construction in case one exists) can be decided within two runs of any path-finding algorithm.*

REMARK 2.5. A simple consequence of Corollary 2.3 is as follows. Suppose the edges of $G$ have lengths associated with them. It follows that if there is a feasible orientation, then there is a feasible one where the path from $i$ to $i'$ is shortest with respect to the original graph $G$. By symmetry, there is also one where the path from $j$ to $j'$ is shortest in $G$. This raises the question under what conditions there exists an orientation that allows *both* pairs to be connected with shortest paths. We call such an orientation *ideal*. Trivially, if $G$ has only two vertices $i = j'$ and $j = i'$ which are connected with two edges of distinct lengths, then there exists a feasible orientation but not an ideal one. We discuss below the problem of recognizing whether a graph has an ideal orientation.

In the rest of the present section we discuss the general feasibility problem with respect to any set S of pairs of vertices. We first give some standard definitions and cite a basic theorem of Robbins [16]. A graph has *edge connectivity* of at least 2 if for every two vertices $i$, $j$, there exist at least two edge-disjoint paths between $i$ and $j$. It is well known that a graph has edge connectivity of at least 2 if and only if it remains connected after the removal of any edge. (This characterization is used in Robbins's paper). A *bridge* is an edge whose removal makes the graph disconnected. Tarjan [21] (extending [20]) gives a linear-time algorithm for finding all the bridges of a graph. He calls a connected graph *bridge-connected* if it has no bridges or, equivalently, if it has edge-connectivity of at least 2. A directed graph is called *strongly connected* if for every two vertices $i$, $j$, there exist paths from $i$ to $j$ and from $j$ to $i$. An undirected graph is called *orientable* if it has a strongly connected orientation.

THEOREM 2.6 (Robbins). *A graph is orientable if and only if it is bridge-connected.*

*Proof.* The proof we give here is different from Robbins's and is more constructive. The interesting part of Theorem 2.6 is the sufficiency. We describe a linear-time orientation algorithm for a bridge-connected graph $G$. (A different proof, which also implies that a bridge-connected graph can be oriented in linear time, can be found in [15].) Starting from any vertex, conduct a depth-first search until some cycle is generated. Now, orient the edges of the cycle and contract the latter into a single vertex. Continue the search on the resulting graph, recursively. The algorithm terminates when the remaining graph consists of a single vertex. ■

A *bridge-connected component* of a graph $G$ is a maximal (with respect to inclusion) bridge-connected subgraph of $G$. Given the bridges of a graph, the bridge-connected components can be found in linear time. The bridge-connected components of a graph induce a partition on the set of vertices. The bridges of the graph do not belong to any of the bridge-connected components, and every nonbridge is contained in some bridge-connected component.

Consider a pair $(i, i')$ of vertices which belong to distinct bridge-connected components of a connected graph $G'$. Consider any path $L$ from $i$ to $i'$. Clearly, every bridge $e$ on $L$ must also be on every other path between $i$ and $i'$. Thus, if $G'$ is an orientation of $G$ in which $i'$ is reachable from $i$, then necessarily all these bridges must be oriented in the direction from $i$ to $i'$. On the other hand, by Theorem 2.6, every bridge-connected component has a strongly connected orientation. This suggests the following algorithm for Problem 2.1:

ALGORITHM 2.7.

*Step 1.* Find all the bridges of the graph.

*Step 2.* For every pair $(s_i, t_i)$, find a path from $s_i$ to $t_i$ and orient the bridges on the path accordingly; if a bridge is already oriented in the opposite direction, STOP (there is no feasible orientation).

*Step* 3. Orient each bridge-connected component in a strongly connected way.

The validity of the algorithm follows from the preceding discussion. A straightforward implementation of step 2 takes $O(m|E|)$ time, while steps 1 and 3 take only $O(|E|)$ time. We will prove a better bound later, but at least for small values of $m$ it is useful to have the following theorem:

THEOREM 2.8. *The feasible-orientation problem on a connected graph $G = (V, E)$ with respect to a set of $m$ pairs can be solved in $O(m|E|)$ time.*

REMARK 2.9. In a more efficient implementation of step 2 of Algorithm 2.7, one would first construct a tree whose vertices represent the bridge-connected components, and whose edges represent the bridges. Thus, step 2 amounts to finding a feasible orientation of a tree with respect to a given set of pairs of vertices. An obvious algorithm for the latter runs in $O(mn)$ time, which is already an improvement over the obvious algorithm that does not take advantage of the tree structure, running in $O(m|E|)$ time.

PROPOSITION 2.10. *The feasible-orientation problem on a tree with respect to $m$ pairs of vertices can be solved in $O(m \log n + n \min(m, \log n))$ time.*

*Proof.* If $m \leqslant \log n$, the claim follows from a previous discussion. Suppose $m > \log n$. Let $c$ denote a centroid of the tree $T$, i.e., $c$ is a vertex with the property that each of the subtrees $T_1, \ldots, T_p$ rooted at $c$ has no more than $\lceil n/2 \rceil$ vertices. For each input pair $(s_i, t_i)$, if $s_i \in T_j$, $t_i \in T_k$, and $j \neq k$, then replace $(s_i, t_i)$ by the two pairs $(s_i, c)$ and $(c, t_i)$. This step takes $O(m)$ time. It separates the problem into $p$ independent problems on the subtrees $T_j$. However, the total number of pairs may be doubled. Thus, instead of continuing this procedure recursively, we do the following. Consider any subtree $T_j$ and the pairs of the forms $(u, c)$ and $(c, v)$ associated with it. If there exist at least one pair of the form $(u, c)$ and at least one pair of the form $(c, v)$ (where both $u$ and $v$ are vertices of $T_j$), then the problem is infeasible and we stop. If there are no such pairs at all, we do not do anything with $T_j$ at this stage. So, suppose there are only pairs of the form $(u, c)$. We first direct the tree $T_j$ towards $c$. This is a preprocessing step rather than the actual orientation, and it takes linear time in the size of $T_j$. The algorithm then labels a vertex $u$ as "done" if and only if the path from $u$ to $c$ has already been oriented. (Thus, all the vertices on the path from $u$ to $c$ are also labeled as "done.") A typical step goes as follows. The algorithm picks any pair $(u, c)$ such that $u$ is not done yet. It follows the path from $u$ to $c$, orienting every edge accordingly and labeling every vertex as done, until it reaches an already done vertex. It then labels $u$ as done. Clearly, this step of processing all the subtrees $T_j$ takes $O(n)$ time. We now have to solve the problem independently over the $p$ partially oriented subtrees and the total number of pairs to consider is not greater than $m$. We continue recursively. It is easy to see that this process works in $O(\log n)$ phases, each of which takes $O(m + n)$ time, not counting the operation of replacing a pair by two pairs involving a centroid. It is easy to see that the total effort of these replacements is $O(m)$. It follows that in this case the running time is $O((m + n)\log n)$. ■

Corollary 2.11. *The feasibility of $m$ pairs can be decided in $O(|E| + m\log n + n\min(m, \log n))$ time.*

As a consequence of the validity of Algorithm 2.7 we have:

Corollary 2.12. *Given a connected undirected graph $G = (V, E)$ and a set $S$ of $m$ pairs of vertices $(s_i, t_i)$ $(i = 1, \ldots, m)$, the set $S$ is feasible (see Definition 1.2) if and only if every set of two pairs from $S$ is feasible.*

*Proof.* The condition is obviously necessary. Obviously, if every set of two pairs is feasible, then Algorithm 2.7 succeeds, since all the orientations of bridges succeed. ■

Finally, the following "greedy" algorithm also solves Problem 2.1:

Algorithm 2.13.

*Step 0.* Initialize $\Gamma = G$; $i = 1$.

*Step 2.* If there is in $\Gamma$ a path from $s_i$ to $t_i$ then (i) orient all the undirected edges of $\Gamma$ on this path in the direction from $s_i$ to $t_i$, (ii) for each portion of this path which lies within a bridge-connected component, say a path $L$ from $u$ to $v$, pick another path $L'$ from $u$ to $v$ within that component so that $L$ and $L'$ are edge-disjoint, and orient its undirected edges from $v$ to $u$, and (iii) name the resulting mixed graph $\Gamma$; else STOP (there is no feasible orientation).

*Step 3.* If $i = m$, orient all the undirected edges of $\Gamma$ arbitrarily and terminate; else set $i = i + 1$ and go to step 1.

The validity of Algorithm 2.13 is based on the fact that by orienting along an arbitrary path between two members of any of the given pairs, the algorithm cannot lose feasibility. Certainly, no mistake can be made while orienting bridges this way. While orienting along a path within a bridge-connected component, again, there can be made no mistakes, since every oriented portion is immediately complemented into an oriented cycle.

## 3. THE COMPLEXITY OF IDEAL ORIENTATIONS

The general problem of ideal orientation (see Remark 2.5) is as follows.

PROBLEM 3.1. Given are an undirected graph $G=(V,E)$ with edge-lengths (and assume there are no negative cycles), and with $m$ pairs of vertices $(s_i, t_i)$ $(i=1,\dots,m)$. Let $l_i(G)$ denote the length of the shortest path in $G$ between $s_i$ and $t_i$ $(i=1,\dots,m)$. Similarly, for any orientation $G'$ of $G$, let $l_i(G')$ denote the length of the shortest path in $G'$ between $s_i$ and $t_i$ $(i=1,\dots,m)$. Find an *ideal* orientation, i.e., an orientation $G'$ of $G$ such that $l_i(G')=l_i(G)$ $(i=1,\dots,m)$, or conclude that no such orientation exists.

REMARK 3.2. In Problem 3.1, if the first member is the same in all the pairs, the problem is easy. If the common vertex is $s$, then an ideal orientation is obtained from any spanning tree which consists of shortest paths from $s$ to any vertex of the graph. A similar observation applies to the case where the second member is the same in all the pairs.

PROPOSITION 3.3. *The problem of recognizing whether a graph with unit edge lengths has an ideal orientation with respect to a given set of pairs of vertices is* NP-*complete.*

*Proof.* We describe a reduction from 3-SAT (see [9]). Consider a formula $\phi = C_1 \wedge \cdots \wedge C_m$, where $C_j = x_j \vee y_j \vee z_j$ $(j=1,\dots,m)$ are the given clauses and $\{x_j, y_j, z_j\} \subset \{v_1, \bar{v}_1, \dots, v_n, \bar{v}_n\}$. We construct a graph $G=(V,E)$ as follows. To each clause $C_j$ assign a vertex also denoted by $C_j$. Similarly, to each member of the set $\{v_1, \bar{v}_1, \dots, v_n, \bar{v}_n\}$ assign a vertex denoted by the same literal, and add $2n+1$ more vertices $R$, $U_1,\dots,U_n$, $V_1,\dots,V_n$. Edges of the graph are as follows. All the $V_i$'s are connected to $R$ by strings of $2n$ edges in series. For each $i$ $(i=1,\dots,n)$ both $v_i$ and $\bar{v}_i$ are connected to both $U_i$ and $V_i$. Also, $v_i$ is connected to $\bar{v}_{i+1}$ $(i=1,\dots,n-1)$ by two parallel edges. For each $j$ $(j=1,\dots,m)$ the vertex $C_j$ is connected to the three vertices corresponding to the literals $x_j, y_j, z_j$ by strings of $2n$ edges in series. Now, consider $m+2$ pairs of vertices $(s_j, t_j)$ as follows. First, let $s_j = C_j$ and $t_j = R$ $(j=1,\dots,m)$. Also, let $s_{m+1} = t_{m+2} = \bar{v}_1$ and $s_{m+2} = t_{m+1} = v_n$. It is easy to verify that in $G$ there are paths of length $4n+1$

between each $C_j$ and $R$. Also, there are paths of length $3n - 1$ between $\bar{v}_1$ and $v_n$. It is also easy to verify that $\phi$ is satisfiable if and only if there exists an orientation $G'$ of $G$ where all these pairs are connected with paths which are shortest in $G$. ■

COROLLARY 3.4. *The problem of finding an orientation which minimizes the sum of shortest-path distances between members of given pairs of vertices is* NP-*hard even when edges have unit length.*

In view of Proposition 3.3, it is interesting to consider the problem of ideal orientations with respect to a fixed number of pairs of vertices. At the present we have a positive answer for the case of two pairs. It is interesting to note that an analog of Corollary 2.12 does not exist for ideal orientations; namely, we have the following:

PROPOSITION 3.5. *There exists an undirected graph $G$ and three pairs of vertices for which no ideal orientation of $G$ exists, such that for every two of the three pairs there exists an ideal orientation of $G$.*

*Proof.* Consider an undirected graph on eight vertices $1,\dots,8$ with the edges $(i, i+1)$ $(i = 1,\dots,7)$, $(1,8)$, $(2,6)$, and $(4,8)$. Consider the following three ordered pairs of vertices: $(1,5)$, $(3,7)$, and $(6,2)$. The shortest-path distances are 3 between 1 and 5, 3 between 3 and 7, and 1 between 6 and 2. If the edge $(2,6)$ is oriented from 6 to 2 (which is necessary for preserving the distance of 1 from 6 to 2), then the pair $(1,5)$ needs the edge $(4,8)$ to be oriented from 8 to 4, while the pair $(3,7)$ requires the opposite orientation. Thus, there is no orientation under which all the three pairs are content. It is easy to check that for any two of the pairs an ideal orientation exists. ■

## 4. IDEAL ORIENTATIONS FOR TWO PAIRS OF VERTICES

In this section we consider the problem of finding an ideal orientation of a graph $G = (V, E)$ with positive edge lengths with respect to two given pairs of vertices $(i, i')$, $(j, j')$, or conclude that no such orientation exists. If an ideal orientation exists, we call the quadruple $(i, i'; j, j')$ *content*. We prove that this problem can be solved in polynomial time.

For any two vertices $u$, $v \in V$, denote by $L(u, v)$ the subgraph of $G$ consisting of all the vertices and directed edges of $G$ that lie on any shortest path connecting $u$ and $v$. The following facts are obvious:

FACT 4.1. *If $(i, i'; j, j')$ is content and $i \neq i'$, then there exists an edge $(x, i')$ in $L(i, i')$ such that $(i, x; j, j')$ is content.*

FACT 4.2. *Suppose $(x, i')$ is an edge in $L(i, i')$ such that $(i, x; j, j')$ is content. Under these conditions, if the edge $(x, i')$ is not in $L(j, j')$, then $(i, i'; j, j')$ is content.*

Notice that Facts 4.1 and 4.2 do not suffice for designing a good algorithm, since they do not provide a good recursive characterization of "content."

FACT 4.3. *A quadruple $(i, i'; i', i)$ is content if and only if there exist two edge-disjoint shortest paths between $i$ and $i'$.*

The following lemma gives rise to a characterization of content:

LEMMA 4.4. *Suppose $(i, i'; j, j')$ is a quadruple such that*

(i) *both $i$ and $i'$ are vertices of $L(j, j')$,*
(ii) *both $j$ and $j'$ are vertices of $L(i, i')$, and*
(iii) $(i, i') \neq (j', j)$.

*Under these conditions $(i, i'; j, j')$ is content.*

*Proof.* For any two vertices $s$, $t$, denote by $d(s, t)$ the shortest-path distance between $s$ and $t$, and denote by $\pi(s, t)$ a shortest path from $s$ to $t$. Suppose $(i, i'; j, j')$ satisfies conditions (i)–(iii). Thus, there exists a shortest path $\pi(i, i')$ which visits $j$. Also, there exists a shortest path $\pi(j, j')$ which visits $i'$. Without loss of generality, assume these two paths contain the same shortest path $\pi(j, i')$. By the same argument it follows that there exist shortest paths $\pi(i, j)$, $\pi(j, i')$, $\pi(i', j')$, and $\pi(j', i)$ such that the union of any two of these paths with a common endpoint is a shortest path [for example, the union of $\pi(j, i')$ and $\pi(i', j')$ is a shortest path between $j$ and $j'$]. It follows that the union of any two of the above paths which have a common endpoint is simple (that is, contains no cycles). The concatenation of these four paths is of course a cycle. We claim that it is a simple cycle. In view of the symmetry, it suffices to prove that $\pi(i, j)$ and $\pi(j', i')$ are vertex-disjoint. Suppose, to the contrary, that these two paths have a common vertex $k$. Since $k$ lies on a shortest path from $i$ to $j$ and $j$ lies on a shortest path from $i$ to $i'$, it follows that $j$ lies on a shortest path from $k$ to $i'$, so

$$d(k, j) + d(j, i') = d(k, i').$$

Analogously,

$$d(j,i')+d(i',k)=d(j,k).$$

It follows that $d(j,i')=0$, so $i'=j$. By symmetry, also $j'=i$. Thus, $(i,i')=(j',j)$, contradicting assumption (iii). We have established the existence of a simple cycle on which the vertices $i$, $j$, $i'$, $j'$ occur in this order. Now, direct all the edges along the cycle according to their directions in the above four paths. Clearly, the distances from $i$ to $i'$, and from $j$ to $j'$, subject to this orientation, equal the shortest-path distances. This partial orientation can obviously be extended into an ideal orientation. ■

Note that if $(i,i')=(j',j)$ the problem is easy and one has just to check that the graph has two $i-i'$ edge-disjoint paths.

We now have

THEOREM 4.5. *If $(i,i';j,j')$ is a quadruple such that $(i,i')\neq(j',j)$, then it is content if and only if at least one of the following conditions holds*:

(i) *$(i,i')$ is an edge in $L(i,i')$, and $(j,j')$ is an edge of $L(j,j')$.*

(ii) *There is an edge $(x,i')\in L(i,i')\backslash L(j,j')$ such that $(i,x;j,j')$ is content.*

(iii) *There is an edge $(i,x)\in L(i,i')\backslash L(j,j')$ such that $(x,i';j,j')$ is content.*

(iv) *There is an edge $(x,j')\in L(j,j')\backslash L(i,i')$ such that $(i,i';j,x)$ is content.*

(v) *There is an edge $(j,x)\in L(j,j')\backslash L(i,i')$ such that $(i,i';x,j')$ is content.*

(vi) *Both $i$ and $i'$ are vertices of $L(j,j')$, and both $j$ and $j'$ are vertices of $L(i,i')$.*

*Proof.* Sufficiency of (i) is obvious. Sufficiency of each of (ii)–(v) was already stated in Fact 4.2. Sufficiency of (vi) was proven in Lemma 4.4. Now, suppose $(i,i';j,j')$ is content, $(i,i')\neq(j',j)$, and condition (i) does not hold. Without loss of generality assume $(i,i')$ is not an edge of $L(i,i')$. By Fact 4.1, there exists an edge $(x,i')$ in $L(i,i')$ such that $(i,x;j,j')$ is content. If $(x,i')$ is not an edge of $L(j,j')$, then condition (ii) holds. Suppose this is not the case so, in particular, $i'$ is in $L(j,j')$. Notice the symmetry of conditions (ii)–(v). The same argument applies to all of them. If we do not succeed in showing that one of them holds, then we finally get that condition (vi) holds. ■

Theorem 4.5 suggests a recursive procedure for deciding whether a given quadruple $(i,i';j,j')$ is content. To actually implement the algorithm, one would rather use a "bottom-up" approach. Thus, given a quadruple $(i, i'; j, j')$, one would consider not only this quadruple, but also every quadruple $(u, u'; v, v')$ such that there exists a shortest path from $i$ to $i'$ which first visits $u$ and then $u'$, and similarly there exists a shortest path from $j$ to $j'$ which first visits $v$ and then $v'$. In other words, $L(u, u')$ is a subgraph of $L(i, i')$, and $L(v, v')$ is a subgraph of $L(j, j')$. We call such quadruples *legal*. Obviously, the set of legal quadruples is a cartesian product of sets of pairs of vertices. Recall that a quadruple $(u, u'; v, v')$ is called content if there is an orientation in which $(u, u')$ and $(v, v')$ are connected by shortest paths.

The first step of the algorithm is to mark as "content" all the quadruples $(u, u'; v, v')$ where $(u, u')$ is an edge of $L(i, i')$, $(v, v')$ is an edge of $L(j, j')$, and $(u, u') \neq (v', v)$ The algorithm then attempts to extend the set of quadruples known to be content as follows. Consider any legal quadruple $(u, u'; v, v')$ known to be content such that $u' \neq i'$. Consider any edge $(u', x)$ of $L(i, i')$. Clearly, the quadruple $(u, x; v, v')$ is legal. Moreover, if $(x, u') \notin L(v, v')$ then $(u, x; v, v')$ is also content. The extension of a feasible quadruple is also attempted in the other three directions, that is, the algorithm considers quadruples of the form $(x, u'; v, v')$, $(u, u'; x, v')$ and $(u, u'; v, x)$ where in each case $x$ is chosen as a neighbor of the vertex replaced by it. Similar extensions are performed based on Lemma 4.4 and Fact 4.3.

It is obvious that the algorithm is polynomial, but we have not attempted to obtain the best time bound for it. We find it more interesting to note the following:

THEOREM 4.6. *The problem of recognizing whether a given quadruple is content is in the class* NC, *i.e., can be solved in polylogarithmic time on a polynomial number of processors.*

*Proof.* To describe the algorithm, we introduce an auxiliary graph $G^*$ whose vertices correspond to legal quadruples of vertices in the original graph $G$. Consider any quadruple $q = (u, u'; v, v')$. We construct directed edges from $q$ into all the quadruple of the form $(u, x; v, v')$ where $(u', x) \in L(i, i')$ and $(x, u') \notin L(v, v')$. Analogously, we construct edges from $q$ into all the quadruples of the form $(x, u'; v, v')$ where $(x, u) \in L(i, i')$ and $(u, x) \notin L(v, v')$, all the quadruples of the form $(u, u'; v, x)$ where $(v', x) \in L(j, j')$ and $(x, v') \notin L(u, u')$, and all the quadruples of the form $(u, u'; x, v')$ where $(x, v) \in L(j, j')$ and $(v, x) \notin L(u, u')$. In other words, an edge $(q_1, q_2)$ in $G^*$ represents the case that if $q_1$ is content, then so is $q_2$ by a simple extension. Now, it takes polylog time on a polynomial number of processors to identify all the pairs $(u, u')$ where there are two edge-disjoint shortest paths between $u$ and $u'$. Thus, all the content quadruples of the form $(u, u'; u', u)$ can be identified in advance. Similarly, it takes polylog time to identify all the content quadruples $(u, u'; v, v')$ where both $u$ and $u'$ are vertices of $L(v, v')$ and both $v$ and $v'$ are vertices of $L(u, u')$, and obviously all those where $(u, u')$ is an edge of $L(u, u')$ and $(v, v')$ is an edge of $L(v, v')$. Now the problem is reduced to recognizing whether there exists a path in $G^*$ from one of the quadruples which were identified in advance as content, into the given quadruple $(i, i'; j, j')$. Finding such a path is obviously in NC. ■

## 5. ON THE COMPLEXITY OF SOME EXTENSIONS

In this section we prove NP-completeness of some related problems. First, it is interesting to note that an extension of the two-pair ideal-orientation problem in the following direction gives an NP-complete problem:

PROBLEM 5.1. Given a graph $G$ with integral edge lengths, two pairs of vertices $(s_i, t_i)$ $(i = 1,2)$, and two integers $k_1, k_2$, recognize whether there exists an orientation $G'$ such that the shortest-path distance from $s_i$ to $t_i$ in $G'$ is not greater than $k_i$ $(i = 1,2)$.

PROPOSITION 5.2. *Problem* 5.1 *in* NP-*complete.*

*Proof.* Membership in NP is obvious. The rest of the proof is by reduction from PARTITION (see [9]). In an instance of the latter one is given a set of positive integers $\{w_1,\dots, w_n\}$, and one has to recognize whether there exists a subset $S \subset \{1,\dots, n\}$ such that $\sum_{i \in S} w_i = \frac{1}{2}W$, where $W = \sum_{i=1}^{n} w_i$. Now, construct a graph $G$ with $n + 1$ vertices $0, 1,\dots, n$. Join vertices $i - 1$ and $i$ with two edges, one of length 1 and the other of length $w_i + 1$ $(i = 1,\dots, n)$. Let $s_1 = t_2 = 0$ and $s_2 = t_1 = n$. Let $k_1 = k_2 = \lfloor n + W/2 \rfloor$. It is easy to verify that the set $\{1,\dots, n\}$ can be partitioned into two sets of equal sums of weights if and only if there exists an orientation of $G$ as required in Problem 5.1. ■

Another direction of possible extension is given in the following problem:

PROBLEM 5.3. The set $E$ of edges of a graph $G$ is given as the union $E = E_1 \cup E_2$ of two directed acyclic graphs corresponding to two pairs of vertices, $(s_i, t_i)$ $(i = 1,2)$. Recognize whether there exists an orientation $G'$ such that for $i = 1,2$ there exists a path in $G'$ which uses only edges of $E_i$.

PROPOSITION 5.4. *Problem* 5.3 *in* NP-*complete.*

*Proof.* The proof goes by reduction from 3-SAT, and we use the notation established in the proof of Proposition 3.3. We construct two directed acyclic graphs as follows. The first graph corresponds to the boolean variables. For each variable $v_i$ we construct two vertex-disjoint paths of length $2m$ between the same endpoints: $(c_{i-1} = a_0^i, a_1^i, \dots, a_{2m}^i = c_i)$ and $(c_{i-1} = b_0^i, b_1^i, \dots, b_{2m}^i = c_i)$. The pair $(s_1, t_1)$ is the same as $(c_0, c_n)$. The set $E_1$ is the set of edges $(a_{j-1}^i, a_j^i)$ and $(b_{j-1}^i, b_j^i)$ $(i = 1, \dots, n,\ j = 1, \dots, 2m)$. A path from $c_0$ to $c_n$ which uses only edges in $E_1$ must select for every $i$ either the path of $a_j^i$'s (which corresponds to $v_i$ being true) or the path of $b_j^i$'s ($v_i$ false). We now turn to the construction of the second acyclic set of directed edges, representing the clauses. Clause $C_j = x_j \vee y_j \vee z_j$ is represented by three internal vertex-disjoint paths of length 5: $(d_{j-1} = x_0^j,\ x_1^j, \dots, x_5^j = d_j)$, $(d_{j-1} = y_0^j,\ y_1^j, \dots, y_5^j = d_j)$, and $(d_{j-1} = z_0^j,\ z_1^j, \dots, z_5^j = d_j)$. If $x_j$ is $v_i$, then we identify $x_2^j = a_{2j}^i$ and $x_3^j = a_{2j-1}^i$. If $x_j$ is $\bar{v}_i$, the identification is $x_2^j = b_{2j}^i$ and $x_3^j = b_{2j-1}^i$. For $y_j$ and $z_j$ the identification is analogous. The edges in $E_2$ are of the form $(x_{i-1}^j, x_i^j)$, $(y_{i-1}^j, y_i^j)$, and $(z_{i-1}^j, z_i^j)$ $(j = 1, \dots, m,\ i = 1, \dots, 5)$, and the pair $(s_2, t_2)$ is the same as $(d_0, d_m)$. It is easy to check that an orientation as required exists if and only if the formula is satisfiable. ■

*The authors are grateful to Tali Eilam for some good comments on an earlier version of the paper. Also, conversations with Dorit Hochbaum are gratefully acknowledged.*

## REFERENCES

1 F. Boesch and R. Tindell, Robbins's theorem for mixed graphs, *Amer. Math. Monthly* 86:716–719 (1980).

2 V. Chvátal and C. Thomassen, Distances in orientations of graphs, *J. Combin. Theory Ser. B* 24:61–65 (1978).

3 S. Even, A. Itai, and A. Shamir, Timetable and multicommodity flow problems, *SIAM J. Comput.* 5:691–703 (1976).

4 S. Fortune, J. E. Hopcroft, and J. Wiley, The directed subgraph homeomorphism problem, *Theoret. Comput. Sci.* 10:111–121 (1980).

5 A. Frank, On the orientation of graphs, *J. Combin. Theory Ser. B* 28:251–261 (1980).

6 A. Frank, An algorithm for submodular functions on graphs, *Ann. Discrete Math.* 16:97–120 (1982).

7 A. Frank and A. Gyarfas, How to orient the edges of a graph? in *Combinatorics* Colloq. Math. Soc. János Bolyai 18, 1976, pp. 352–364.

8 A. Frank, E. Tardos, and A. Sebö, Covering, directed and odd cuts, *Math. Programming Stud.* 22:99–112 (1984).
9 M. R. Garey and D. S. Johnson, *Computers and Intractability*, Freeman, San Francisco, 1979.
10 P. C. Gilmore and A. J. Hoffman, A characterization of comparability graphs and of interval graphs, *Canad. J. Math.* 16:539–548 (1964).
11 S. L. Hakimi, On the degrees of the vertices of a directed graph, *J. Franklin Inst.* 279:290–308 (1965).
12 E. L. Johnson and P. Pierroni, A linear programming approach to the optimum network orientation problem, *Math. Programming Stud.* 26:215–217 (1986).
13 C. St. J. A. Nash-Williams, Well-balanced orientations of finite graphs and unobtrusive odd-vertex-pairing, in *Recent Progress in Combinatorics* (W. T. Tutte, Ed.), Academic, New York, 1969.
14 A. Pnueli, A. Lempel, and S. Even, Transitive orientation of graphs and identification of permutation graphs, *Canad. J. Math.* 23:160–175 (1971).
15 F. S. Roberts, *Applied Combinatorics*, Prentice-Hall, 1984.
16 H. E. Robbins, A theorem on graphs with an application to a problem of traffic control, *Amer. Math. Monthly* 46:281–283 (1939).
17 P. D. Seymour, Disjoint paths in graphs, *Discrete Math.* 29:293–309 (1980).
18 Y. Shiloach, A polynomial solution to the undirected two paths problem, *J. Assoc. Comput. Mach.* 27:445–456 (1980).
19 D. Soroker, Fast parallel strong orientation of mixed graphs and related augmentation problems, *J. Algorithms*, to appear.
20 R. E. Tarjan, Depth-first search and linear graph algorithms, *SIAM J. Comput.* 1:146–160 (1972).
21 R. E. Tarjan, A note on finding the bridges of a graph, *Inform. Process. Lett.* 2:160–161 (1974).

*Received 30 November 1987; final manuscript accepted 19 September 1987*