ORIENTING GRAPHS TO OPTIMIZE REACHABILITY∗

S. Louis Hakimi,† Edward F. Schmeichel,‡ Neal E. Young§

Abstract

It is well known that every 2-edge-connected graph can be oriented so that the resulting
digraph is strongly connected. Here we study the problem of orienting a connected graph
with cut edges in order to maximize the number of ordered vertex pairs (x,y) such that
there is a directed path from x to y. After transforming this problem, we prove a key
theorem about the transformed problem that allows us to obtain a quadratic algorithm
for the original orientation problem. We also consider how to orient graphs to minimize
the number of ordered vertex pairs joined by a directed path. After showing this problem
is equivalent to the comparability graph completion problem, we show both problems are
NP-hard, and even NP-hard to approximate to within a factor of 1 + ǫ, for some ǫ > 0.

1 Introduction

Our terminology and notation is standard except as indicated. We mention only that if X is
a subset of the vertices in a graph, we use ⟨X⟩ to denote the subgraph induced by X. Good
references for any other undefined terms are [1,2].

Let G be any connected graph. Given an orientation ⃗G of G, we will use R(⃗G) to denote
the number of ordered vertex pairs (x,y) such that there is a directed path from x to y in ⃗G.
We call R(⃗G) the reachability of ⃗G.

Robbins [12] showed that G can be oriented so that ⃗G is strongly connected (i.e., R(⃗G) = |V|(|V| − 1)) if and only if G is 2-edge-connected. In the following section (Section 2), we
study the problem of how to orient an arbitrary graph G to obtain a digraph ⃗G with R(⃗G) as
large as possible. We first transform the problem into an equivalent orientation problem on
vertex-weighted trees. Although this transformed version is NP-hard, we prove a key theorem
(Theorem 2.2) which allows us to get a quadratic algorithm for the original problem, as well
as a fully polynomial approximation scheme for the transformed problem.

In Section 3, we consider the analogous problem of how to orient G so as to minimize
R(⃗G) for the resulting digraph ⃗G. We show this problem is equivalent to comparability graph
completion (adding the fewest edges so the resulting graph can be transitively oriented), and
then show that both problems are NP-hard, and even NP-hard to approximate within a factor
of 1 + ǫ, for some ǫ > 0. Related hardness results appear in [6,10].

∗This work was supported by the National Science Foundation under Grants NCR-95-05551 and DMS-92-
06991
†Department of Electrical and Computer Engineering, University of California, Davis, CA 95616
‡Department of Mathematics and Computer Science, San Jose State University, San Jose, CA 95192
§Department of Computer Science, Dartmouth College, Hanover, NH 03755
2 Orienting Graphs to Maximize Reachability

Suppose we are given a connected graph $G$ on $n$ vertices with cut edges. Our goal is to orient $G$ to obtain a digraph $\vec{G}$ with $R(\vec{G})$ as large as possible.

It will be useful to first transform this basic problem. Let $C_1, C_2, \ldots, C_b$ denote the components left when the cut edges of $G$ are removed from $G$. By Robbins Theorem [12] each $C_i$ can be oriented into a strongly connected digraph. Let us contract each non-trivial $C_i$ into a single vertex $x_i$, giving $x_i$ weight $wt(x_i) = |V(C_i)|$. The resulting contracted graph is, of course, a $b$-vertex tree $T = T(G)$ with integer weights on the vertices. Our original problem is now easily seen to be equivalent to the following problem: How should we orient $T$ to maximize $\sum wt(x_i) \cdot wt(x_j)$, the sum being taken over all vertex pairs $(x_i, x_j)$ in $T$ such that there is a directed path from $x_i$ to $x_j$ in $\vec{T}$? Note that the input size of the transformed problem may be exponentially smaller than the input size of the original problem (roughly, $b(1 + \log n)$ versus $n$).

We begin by showing that this transformed problem is NP-hard. In particular, consider the following decision problem:

**WEIGHTED TREE ORIENTATION (WTO)**

**Instance:** Tree $T$, weight function $wt: V(T) \to \mathbb{Z}^+$, integer $B > 0$.

**Question:** Is there an orientation $\vec{T}$ such that $\mu(\vec{T}) \geq B$, the sum taken over all pairs $(v, w)$ with a directed path from $v$ to $w$ in $\vec{T}$?

**Theorem 2.1.** WTO is NP-complete.

**Proof.** WTO is obviously in NP, and so it suffices to show it is NP-hard. For this we reduce PARTITION [4,p.233] to WTO. Given positive integers $a_1, a_2, \ldots, a_m$ with even sum $S$, consider the weighted tree $T$ in Fig. 1 and set $B = 5(S/2)^2$. Given any $I \subseteq \{1, 2, \ldots, m\}$, consider the orientation $\vec{T}$ obtained by orienting toward (resp, away from) the vertex with weight $S$ each edge whose other end vertex has weight $a_i$ for $i \in I$ (resp, $i \in \{1, 2, \ldots, m\} - I = \bar{I}$). We then find

$$\mu(\vec{T}) = S \left( \sum_{i \in I} a_i + \sum_{i \in \bar{I}} a_i \right) + \sum_{i \in I} a_i \cdot \sum_{i \in \bar{I}} a_i \leq S^2 + \left( \frac{S}{2} \right)^2 = 5 \left( \frac{S}{2} \right)^2$$

with equality precisely if $\sum_{i \in I} a_i = \sum_{i \in \bar{I}} a_i = \frac{S}{2}$. $\blacksquare$

We now wish to develop a key result which will allow us to obtain a quadratic algorithm for the original orientation problem, as well as a fully polynomial approximation scheme for WTO. First, however, we need some terminology and notation. Let $T$ be any tree with positive integer vertex weights. An optimal orientation $\vec{T}$ is one which maximizes $\mu(\vec{T})$. Given an orientation $\vec{T}$ and a vertex $w \in V(T)$, define

$$\text{In}_{\vec{T}}(w) \doteq \{ x \in V(T) \mid \text{there exists a directed path from } x \text{ to } w \text{ in } \vec{T} \}$$
$$\text{Out}_{\vec{T}}(w) \doteq \{ x \in V(T) \mid \text{there exists a directed path from } w \text{ to } x \text{ in } \vec{T} \}$$
In particular, \( w \in \text{In}_{\vec{T}}(w) \cap \text{Out}_{\vec{T}}(w) \). Define
\[
\text{In}_{\vec{T}}'(w) = \text{In}_{\vec{T}}(w) - \{w\} \quad \text{and} \quad \text{Out}_{\vec{T}}'(w) = \text{Out}_{\vec{T}}(w) - \{w\}.
\]
We will usually drop the subscript \( \vec{T} \) if it is clear from the context. Finally, given any subgraph with vertex set \( X \subseteq V(T) \), let \( ||X|| = \sum_{x \in X} wt(x) \) be called the total weight of that subgraph. A centroid in a vertex weighted tree is any vertex \( c \) whose removal minimizes the maximum total weight of any component in \( T - c \).

Our goal now is to prove the following.

**Theorem 2.2.** Let \( c \) be a centroid of \( T \). In every optimal orientation \( \vec{T} \), we have \( \text{In}_{\vec{T}}(c) \cup \text{Out}_{\vec{T}}(c) = V(T) \).

Before proving this, we require the following result.

**Lemma.** Let \( \vec{T} \) be any optimal orientation of \( T \), and let \( P \) be any undirected path in \( T \). Then there are no two vertices \( x, y \) on \( P \) such that the edges of \( P \) incident to \( x \) (resp, \( y \)) are both directed toward \( x \) (resp, away from \( y \)) in \( \vec{T} \).

Briefly, there are no "double reversals" on any path in \( \vec{T} \) (see Fig. 2).

**Proof of the Lemma.** Otherwise, let \( x, y \) be a closest such pair on \( P \), so the edges of \( P \) between \( x \) and \( y \) form a directed path from \( y \) to \( x \). Let \( e_x \) denote the edge at \( x \) on the path from \( y \) to \( x \). Let \( X = \{ w \in V(T) \mid \text{there is an undirected path in } T \text{ between } w \text{ and } x \text{ which does not contain } e_x \} \) and let \( X' = X \cap \text{In}(x) \).

Let \( z \) denote the neighbor of \( y \) on \( P \) which does not occur on the directed path from \( y \) to \( x \) (see Fig. 2). Thinking of \( T \) as rooted at \( z \), consider the subtree \( T_y \) rooted at \( y \). Note that \( ||\text{In}'(y)|| \geq ||\text{Out}'(y) \cap T_y|| \) (else we could reverse the orientations of all edges in \( T_y \) to obtain an orientation better than \( \vec{T} \)). Reverse all edges in \( \langle X' \rangle \), and consider the gain and loss in \( \mu(\vec{T}) \) in doing so, where the loss in \( \mu(\vec{T}) \) is the sum of the terms in \( \mu(\vec{T}) \) which no longer exist under the new orientation, and the gain in \( \mu(\vec{T}) \) is defined analogously. Since \( \text{Out}'(x) \subseteq \text{Out}'(y) \cap T_y \), we find
\[
\text{Loss in } \mu(\vec{T}) \leq ||X'|| \cdot ||\text{Out}'(x)|| < ||X'|| \cdot ||\text{Out}'(y) \cap T_y|| \\
\leq ||X'|| \cdot ||\text{In}'(y)|| \leq \text{Gain in } \mu(\vec{T})
\]
and thus we would have a better than optimal orientation. This proves the Lemma.

**Proof of Theorem 2.2.** Throughout, think of $T$ as rooted at centroid $c$. If the theorem fails for some optimal orientation $\vec{T}$, there must be a vertex $x \notin \text{In}(c) \cup \text{Out}(c)$ adjacent to a vertex $v \in \text{In}'(c) \cup \text{Out}'(c)$. (We will call $(v, x)$ a dangling edge at $v$.) We now consider two cases, assuming $||\text{Out}(c)|| \geq ||\text{In}(c)||$ (else reverse the orientation of all edges of $\vec{T}$).

**Case 1.** There is a dangling edge $(v, x)$ at $v \in \text{In}'(c)$.

Note that since $c \in \text{In}(c) - (T_v \cap \text{In}(c))$, we have

$$||\text{Out}(c)|| \geq ||\text{In}(c)|| > ||T_v \cap \text{In}(c)||.$$

Reverse $(v, x)$ and the edges in $T_x$. We find

$$\text{Gain in } \mu(\vec{T}) \geq ||\text{Out}(c)|| \cdot ||\text{Out}(x)|| > ||T_v \cap \text{In}(c)|| \cdot ||\text{Out}(x)|| \geq \text{Loss in } \mu(\vec{T})$$

contradicting the optimality of $\vec{T}$.

**Case 2.** There are no dangling edges at any vertex of $\text{In}'(c)$.

Let $(x, v)$ be a dangling edge at $v \in \text{Out}'(c)$. If $\text{outdeg}(c) \geq 2$, then $\vec{T}$ contains a path which violates the Lemma (with $v, c$ playing the roles of $x, y$ in the Lemma). Hence we may assume $\text{outdeg}(c) = 1$. But then, since $c$ is a centroid and there are no dangling edges off vertices in $\text{In}'(c)$, and since $\{x\} \cup (T_v \cap \text{Out}(c)) \subset V(T) - \text{In}(c)$, we find

$$||\text{In}(c)|| \geq ||V(T) - \text{In}(c)|| > ||T_v \cap \text{Out}(c)||.$$

Reverse $(x, v)$ and the edges in $T_x$. We find

$$\text{Gain in } \mu(\vec{T}) \geq ||\text{In}(c)|| \cdot ||\text{In}(x)|| > ||T_v \cap \text{Out}(c)|| \cdot ||\text{In}(x)|| \geq \text{Loss in } \mu(\vec{T})$$

contradicting the optimality of $\vec{T}$. ■

It is well known that the centroid of a vertex-weighted tree can be found in linear time, and that the centroid consists of either a single vertex or two adjacent vertices [9]. In the latter case, Theorem 2.2 implies that the optimal orientation may be represented schematically as in Figure 3, where $c_1$ and $c_2$ denote the adjacent centroids.

In the former case, we have essentially an instance of PARTITION. Let $T$ be rooted at $c$, and let $v_1, \ldots, v_k$ denote the neighbors of $c$. Consider an optimal orientation of $\vec{T}$ of $T$. By Theorem 2.2, the subtree $T_{v_i}$ must be oriented entirely toward (respectively, away from) $c$ if and only if the edge between $v_i$ and $c$ is oriented toward (respectively, away from) $c$. Thus,

![Figure 2. Double Reversal on P](image-url)
\[ \mu(\vec{T}) = ||c|| \cdot ||T - c|| + \left( \sum_{i \in I} ||T_{v_i}|| \right) \left( \sum_{i \in \bar{I}} ||T_{v_i}|| \right) + \sum_{i=1}^{k} \mu(T_{v_i}^*), \]

where \( I = \{ i \mid T_{v_i} \text{ is oriented toward } c \text{ in } \vec{T} \} \) and \( T_{v_i}^* \) denotes \( T_{v_i} \) oriented entirely towards (respectively, away from) \( c \) if \( i \in I \) (respectively, \( i \notin I \)). To maximize \( \mu(\vec{T}) \), we need to find a partition \( I \cup \bar{I} \) of \( \{1, 2, \ldots, k\} \) such that \( \sum_{i \in I} ||T_{v_i}|| \) and \( \sum_{i \in \bar{I}} ||T_{v_i}|| \) are as equal as possible.

There is, of course, a well-known dynamic programming algorithm to find such a partition [4, Section 4.2]. Since \( \sum_{i=1}^{k} ||T_{v_i}|| \leq n \), the running time of this dynamic programming algorithm is \( O(kn) = O(n^2) \). Since all other tasks in the original orientation algorithm (e.g., finding the 2-edge-connected components in \( G \), giving these components strongly-connected orientations, etc.) can easily be done in \( O(|E|) \) time using standard depth-first search techniques, we see that the original orientation problem can be solved in \( O(n^2) \) time; i.e., quadratic in the original input size \( n \). On the other hand, there is also a well-known fully polynomial approximation scheme for the above partition problem [7], which in turn provides a fully polynomial approximation scheme for the NP-complete problem WTO.

3 Orienting Graphs to Minimize Reachability

Let \( G \) be a connected graph, and let \( r(G) = \min R(\vec{G}) \), the minimum being taken over all orientations of \( G \). We call an orientation \( \vec{G} \) a minimal orientation if \( R(\vec{G}) = r(G) \). We begin with the following result.

**Theorem 3.1.** Every minimal orientation of \( G \) is acyclic.

**Proof.** Suppose to the contrary there exists a minimal orientation \( \vec{G} \) which is not acyclic. Since \( \vec{G} \) is not acyclic, at least one of the strongly-connected components of \( \vec{G} \), say \( C \), is not a single vertex. Let \( E' \) denote a set of edges in \( C \) whose reversal renders \( \langle V(C) \rangle \) not strongly connected, and let \( \vec{G}' \) denote \( \vec{G} \) with the edges in \( E' \) reversed. It is easy to see every ordered pair of vertices joined by a directed path in \( \vec{G}' \) is joined by a directed path in \( \vec{G} \), while clearly there is a pair of vertices in \( V(C) \) which are joined by a directed path in \( \vec{G} \) but not in \( \vec{G}' \). Thus, \( R(\vec{G}') < R(\vec{G}) \), which violates the minimality of \( \vec{G} \). \( \blacksquare \)

Obviously \( r(G) \geq |E(G)| \), and it is known that \( r(G) = |E(G)| \) if and only if \( G \) is a comparability graph [1,5], or, equivalently, if \( G \) is transitively orientable [3]. It is known that comparability graphs can be recognized in polynomial time [3].
Given a connected graph \( G \), let \( \chi(G) \) (respectively, \( \tau(G) \)) denote the minimum number of edges which must be deleted from (respectively, added to) \( G \) to obtain a comparability graph. We now apply Theorem 3.1 to establish a connection between \( r(G) \) and \( \tau(G) \).

**Theorem 3.2.** \( r(G) = |E(G)| + \tau(G) \).

**Proof.** Suppose we add \( \tau(G) \) edges to \( G \) to obtain a comparability graph \( G' \), and then orient \( G' \) so that \( R(\vec{G'}) = |E(G')| = |E(G)| + \tau(G) \). But, of course \( r(G) \leq R(\vec{G'}) \), and so \( r(G) \leq |E(G)| + \tau(G) \).

On the other hand, consider a minimal orientation \( \vec{G} \) of \( G \), so that \( r(G) = R(\vec{G}) \). By Theorem 3.1, \( \vec{G} \) is acyclic. Consider the transitive closure \( \text{cl}(\vec{G}) \) of \( \vec{G} \). We then find

\[
 r(G) = R(\vec{G}) = R(\text{cl}(\vec{G})) \\
 = |E(G)| + (\text{No. of Edges Added to } \vec{G} \text{ to obtain } \text{cl}(\vec{G})) \\
 \geq |E(G)| + (\text{Min. No. of Edges which need to be added to } G \text{ to obtain a transitively orientable graph}) \\
 = |E(G)| + \tau(G).
\]

Thus, \( r(G) = |E(G)| + \tau(G) \), as asserted.

We now turn to the complexity of computing \( r(G) \). Consider the following decision problem.

**MINIMUM REACHABILITY ORIENTATION (MRO)**

**Instance:** Graph \( G \), integer \( k \geq |E(G)| \)

**Question:** Is \( r(G) \leq k \)?

In a moment we will prove

**Theorem 3.3.** \( \text{MRO is NP-complete} \).

It follows immediately from Theorems 3.2 and 3.3 that the following problem is also NP-complete.

**COMPARABILITY GRAPH COMPLETION (CGC)**

**Instance:** Graph \( G \), integer \( k \geq 0 \)

**Question:** Is there a superset \( E' \) of \( E \) such that \( |E' - E| \leq k \) and \( G = (V, E') \) is a comparability graph (i.e., \( \tau(G) \leq k \))? (Previously, it was known only that \( \text{COMPARABILITY SUBGRAPH} \) (i.e., deciding if \( \chi(G) \leq k \) is NP-complete [4,p.197]. For the optimization versions of MRO and CGC, the goal is to compute (or approximate) the minimum \( k \) such that \( \langle G, k \rangle \) is a positive instance of the decision problem.)

**Proof of Theorem 3.3.** Clearly \( \text{MRO} \in \text{NP} \). To show \( \text{MRO} \) is NP-hard, we will reduce \( \text{NOT-ALL-EQUAL 3SAT} \) [4,p.259].
Let \( I \) be an instance of NAE3SAT with \( m \) clauses. Construct a graph \( G_I \) as follows. Each variable \( x \) will be represented by an edge \((x_T, x_F)\) in \( G_I \), and orienting this edge towards \( x_T \) (respectively, \( x_F \)) will correspond to setting \( x \) to \( T \) (respectively, \( F \)). Each clause \( C \) will be represented by a 9-cycle in \( G_I \). The three literals in \( C \) will be assigned to three equally spaced edges \((1T, 1F), (2T, 2F), (3T, 3F)\) on \( C \)'s 9-cycle, as shown in Fig. 3. If the literal assigned to the edge \((1T, 1F)\) is \( X \) (respectively, \( \overline{X} \)), add two 2-paths joining \( 1T \) to \( xF \) and \( 1F \) to \( xT \) (respectively, joining \( 1T \) to \( xT \) and \( 1F \) to \( xF \)) to \( G_I \). Make analogous connections for the other two literals of \( C \), as well as for the remaining clauses of \( I \), to complete \( G_I \).

![Figure 4. The Graph \( G_I \)](image)

Note that however we orient \( G_I \), each of the \( m \) 9-cycles will contain a directed 2-path, and thus \( r(G_I) \geq |E(G_I)| + m \). In fact, \( r(G_I) = |E(G_I)| + m \) if and only if \( I \) has a satisfying truth assignment in the not-all-equal sense. Indeed, given a satisfying truth assignment for \( I \), we can obtain such an orientation for \( G_I \) as follows: Orient the edges \((x_T, x_F)\) to correspond to the truth assignment, and orient all the 6-cycles in \( G_I \) containing these edges so that none contains a directed 2-path. Note that the edges in each 9-cycle to which a literal was assigned are now oriented to reflect the truth value of that literal under the truth assignment. Since each clause contains both a true and a false literal, it is trivial to complete the orientation of the 9-cycles so each contains exactly one directed 2-path passing through a darkened vertex on the 9-cycle. Conversely, any orientation of \( G_I \) with only \( |E(G_I)| + m \) reachable pairs must have exactly one directed 2-path per 9-cycle, and so corresponds to a satisfying truth assignment.

We also observe that it remains NP-hard even to approximate \( r(G) \) to within a factor of \( 1 + \epsilon \), for some \( \epsilon > 0 \). Consider the following optimization problem.

**MAX NOT-ALL-EQUAL 3SAT**

**Instance:** Boolean formula in 3CNF

**Question:** What is the maximum number of clauses that can be satisfied (in a not-all-equal sense) by a truth assignment?
It was established recently [8,11] that it is NP-hard to approximate MAX NAE3SAT to within a factor of 1.013 (that is, finding an assignment that satisfies, in the not-all-equal sense, 1/1.013 of the maximum possible number of clauses is NP-hard). Using this, we can strengthen Theorem 3.3 as follows:

**Theorem 3.4.** There exists a constant \( \epsilon > 0 \) such that approximating the optimization versions of MRO or CGC to within a factor of \( 1 + \epsilon \) is NP-hard.

**Proof Sketch.** Let \( I \) be an instance of NAE3SAT with \( m \) clauses. In the following, “satisfying” a clause of \( I \) refers to making at least one of its literals true and at least one false. Recall that \( R(G_I) \) denotes the number of ordered pairs of vertices \((x, y)\) with a directed path from \( x \) to \( y \) in an orientation \( \vec{G}_I \) of \( G_I \).

**Claim 1:** Given any assignment satisfying \( x \) of the clauses of \( I \), there is an orientation \( \vec{G}_I \) of \( G_I \) such that
\[
R(\vec{G}_I) = |E(G_I)| + 3m - 2x.
\]

Claim 1 follows from the proof of Theorem 3.3.

**Claim 2:** Given any orientation \( \vec{G}_I \), there is an assignment satisfying at least \( \frac{1}{2}(|E(G_I)| + 3m - R(\vec{G}_I)) \) clauses of \( I \). Claim 2 holds because if any clause-subgraph (see Fig. 4) in \( G_I \) has more than one directed 2-path, then it has at least three. Thus, any orientation can be converted into an equally good orientation corresponding to a truth assignment (where the only directed 2-paths are 2-paths through the darkened vertices on the 9-cycles). By simple algebra, the claimed bound holds for this assignment.

Suppose one could find an orientation \( \vec{G}_I \) of \( G_I \) with \( R(G_I) \leq (1 + \epsilon)r(G_I) \). By Claim 2, this would yield an assignment satisfying at least
\[
\frac{1}{2}(|E(G_I)| + 3m - (1 + \epsilon)r(G_I))
\]

clauses of \( I \). Let \( \max(I) \) denote the maximum number of clauses in \( I \) which can be simultaneously satisfied in a not-all-equal sense. By Claim 1, \( r(G_I) \leq |E(G_I)| + 3m - 2\max(I) \). Thus expression (1) is at least \( \frac{1}{2}(|E(G_I)| + 3m - (1 + \epsilon)(|E(G_I)| + 3m - 2\max(I))) \), which equals
\[
\max(I) - \frac{\epsilon}{2}[|E(G_I)| + 3m - 2\max(I)].
\]

Using \(|E(G_I)| \leq 24m\) and \( m \leq 2\max(I) \) (assuming without loss of generality that every clause has at least two literals, a random assignment satisfies at least half the clauses on average), expression (2) is at least \( \max(I) - \frac{\epsilon}{2}[48\max(I) + 6\max(I) - 2\max(I)] = (1 - 26\epsilon)\max(I) \). From this the claimed hardness of approximating MRO follows, with \( \epsilon > 0.00049 \).

A similar argument, using Theorem 3.2 and omitting the \(|E(G_I)|\) term in expression (2) (and subsequent expressions), establishes the claimed hardness of approximating CGC, with \( \epsilon > 0.0064 \). □
4 Concluding Remarks

It would be interesting to determine the algorithmic complexity of completing the orientation of a partially oriented graph to maximize reachability. Thus far we have made little progress on this problem. The analogous completion problem to minimize reachability is, of course, NP-hard.

On the other hand, it is very easy to characterize partial orientations which can be completed into strong orientations. An edge-cut \((X, \overline{X})\) in a partially oriented graph is called one-way if all the edges in the cut are already oriented, and all are oriented from \(X\) to \(\overline{X}\) or all from \(\overline{X}\) to \(X\). We have

**Theorem 4.1.** A partial orientation of a graph \(G\) can be completed into a strong orientation if and only if \(G\) is 2-edge-connected and there are no one-way edge cuts in the partial orientation.

5 References

[1] C. Berge, *Graphs and Hypergraphs*, North-Holland, Amsterdam, 1973.
[2] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, North-Holland, Amsterdam, 1976.
[3] S. Even, A. Pnueli, and A. Lempel, "Permutation graphs and transitive graphs," *J. Assoc. Comp. Mach.* **19** (1972), 400-410.
[4] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to NP-Completeness*, W.H. Freeman, San Francisco, 1979.
[5] P.C. Gilmore and A.J. Hoffman, "A characterization of comparability graphs and of interval graphs," *Can. J. Math.* **16** (1964), 539-548.
[6] M.C. Golumbic, H. Kaplan and R. Shamir, "Graph sandwich problems," *J. Algorithms* **19** (1995), 449-473.
[7] O.H. Ibarra and C.E. Kim, "Fast approximation algorithms for the knapsack and sum of subset problems," *J. Assoc. Comp. Mach.* **22** (1975), 463-468.
[8] V. Kann, J. Lagergren, and A. Panconesi, "Approximability of maximum splitting of k-sets and some other APX-complete problems," *Tech. Report TRITA-NA-P9509*, Dept. of Numer. Analysis and Comp. Sci., Royal Inst. of Technology, Stockholm.
[9] O. Kariv and S.L. Hakimi, "An algorithmic approach to network location problems, II: the p-medians," *SIAM J. Appl. Math.* **37**(1979), 539-560.
[10] P. Dell’Olmo, M. Grazia Speranza and Z. Tuza, "Comparability graph augmentation for some multiprocessor scheduling problems," *Discrete Appl. Math.* **72** (1997), 71-84.
[11] C.H. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes," *J. Comp. Systems Sci.*, **43** (1991), 425-440.
[12] H.E. Robbins, "A theorem on graphs with an application to a problem of traffic control," *Amer. Math. Monthly* **46** (1939), 281-283.