SOME NP-COMPLETE PROBLEMS ON GRAPHS

Fánicz Gavril
Département d'Informatique, Université de Montréal

Abstract
In this paper we add a number of new graph problems to the long list of the known NP-complete problems. Among them are the Optimal linear cut arrangement, Optimal directed tree arrangement and arrangements on a grid.

1. Introduction.

As proved by Cook [3] and Karp [8], there exists a family of problems called NP-complete no member of which is known to have a polynomial time algorithm, but if any of them does have one, then they all have. It seems unlikely that the NP-complete problems have polynomial time algorithms, and therefore knowing that a problem is NP-complete may spare some work to researchers.

Any problems were proved NP-complete in [4] and [8]. The purpose of this paper is to prove that some previously unknown problems are NP-complete.

Let us see a few definitions. We consider only finite graphs with no parallel edges and no self-loops. In a graph \( G(V,E) \) \( V \) denotes the set of vertices and \( E \) the set of edges. By \( (u,v) \) we denote the undirected edge between the vertices \( u \) and \( v \) or the directed edge from \( u \) to \( v \) when \( G \) is directed. In a digraph (directed graph) we allow both edges \( (u,v) \) and \( (v,u) \). A given graph is considered undirected, if not otherwise stated.

A set of vertices in a graph \( G(V,E) \) is called a clique if every two of its elements are adjacent.

A set of vertices is called independent if no two of its elements are adjacent. A coloring is a partition of the set of vertices of a graph into disjoint independent sets. The chromatic number of a graph is the minimum number of independent sets necessary for its coloring. A cut is a partition of \( V \) into two disjoint subsets \( S,V-S \); we denote by \( g(S,V-S) \) the number of edges having one endpoint in \( S \) and the other in \( V-S \). Consider an arrangement \( v_1, v_2, \ldots, v_n \) of the vertices of \( G \) on a line. For \( 1 \leq i \leq n \) we denote by \( c(v_i, v_{i+1}) \) the number of edges of \( G \) crossing over the open segment (of the line) between \( v_i \) and \( v_{i+1} \).

A graph \( G(V,E) \) is said to be the intersection graph of a family of sets \( S \) if and only if there is a 1-1 correspondence between the vertices of \( G \) and the elements of \( S \) such that two vertices of \( G \) are adjacent exactly when the corresponding sets in \( S \) have a non-empty intersection. The intersection graph of a family of intervals on a line is called an interval graph. The reference [9] is a survey on intersection graphs.

A digraph \( H \) is called a line digraph if and only if there is a digraph \( G(V,E) \) and a 1-1 correspondence between the edges of \( G \) and the vertices of \( H \) such that two vertices \( u,v \) of \( H \) are connected by an edge \((u,v)\) exactly when the corresponding edges in \( G \) are in the position \( e \). The line digraphs were discussed in [7]. It is easy to see that a clique of a line digraph has at most three vertices.

On a tree \( T \) the distance \( d_T(u,v) \) between two vertices \( u,v \) of \( T \) is the number of edges in the unique simple path between \( u \) and \( v \).

As known [1,3,4,8] for proving that a problem \( P \) is NP-complete it is enough to prove that \( P \in \text{NP} \) and to find a known NP-complete problem \( P_2 \) such that \( P_2 \) is reducible to \( P \) in polynomial time (by a deterministic Turing machine). We denote the fact that \( P_2 \) is reducible to \( P \) in polynomial time by \( P \leq P_2 \).

All the problems presented in this paper are clearly in \( \text{NP} \). The known NP-complete problems that will reduce to our problems are the following:

Pl.1. Optimal linear arrangement [4].
Input: A graph \( G(V,E) \) and a positive integer k.
Property: There exists a linear arrangement \( v_1, v_2, \ldots, v_n \) of the vertices of \( G \) such that
\[
\sum_{1 \leq i < j \leq n} E(v_i, v_j) \leq k.
\]

Pl.2. Simple max cut [4].
Input: A graph \( G(V,E) \) and a positive integer k.
Property: There exists a set \( S \subseteq V \) such that \( g(S,V-S) \geq k \).

Let us look at the problems whose inputs I contain only positive integers. Denote by \( \max I \) the magnitude of the largest number appearing in the input \( I \). As defined in [6], an NP-complete problem is called strongly NP-complete if there exists a polynomial \( p(x) \) such that if we restrict the problem only to inputs I having \( \max I \leq p(\text{length}(I)) \), then the restricted problem is also NP-complete. For example, as proved in [6], the Bin packing problem is strongly NP-complete. Therefore there exists a polynomial \( p(x) \) such that the following problem is NP-complete.

Pl.3. p-Bin packing.
Input: A set of positive integers \( I=x_1, x_2, \ldots, x_n \), \( n, k, b \) such that \( \max I \leq p(\text{length}(I)) \).
Property: The set \( x_1, x_2, \ldots, x_n \) can be partitioned into \( k \) subsets (called bins) such that the sum of the numbers in every subset is at most \( b \).


Alia, Frosini and Maestrini [2] considered the following problem in circuit design:
P.2.1. Linear arrangement of modules.
Input: A set of modules (vertices) \( V \), a set of pairs of modules \( E \) representing the wire connections, a positive integer \( k \).
Property: There exists a linear arrangement of the modules such that the wire connections can be placed on \( k \) layers, no two wires intersecting except at the endpoints.

For example, when \( V = \{v_1, v_2, v_3, v_4, v_5\} \), \( E = \{(v_1, v_2), (v_3, v_5)\} \), and \( k = 3 \) the linear arrangement of the modules and the three layers of the wire connections is given in Figure 1a.

![Figure 1a: Linear Arrangement of Modules](image)

Figure 1.

In the above problem we can consider \( V \) as the set of vertices and \( E \) as the set of edges of a graph \( G(V, E) \).

Lemma 2.1. Given \( G(V, E) \) and \( k \), the set \( V = \{v_1, \ldots, v_n\} \) has a linear arrangement as in P.2.1. (see Figure 1a) if and only if \( V \) has a linear arrangement \( v_{i1}, v_{i2}, \ldots, v_{in} \) such that for every \( 1 \leq j < n \)

\[ c(v_{ij}, v_{ij+1}) < k \]  

(Figure 1b).

Proof: Let us assume that \( G \) has a linear arrangement as in P.2.1. Since all the wire connections, i.e., the edges of \( G \), are on at most \( k \) layers, it follows that for every \( 1 \leq j < n \)

\[ c(v_{ij}, v_{ij+1}) < k \]  

(Figure 1b). Every edge \( (v_{ij}, v_{ij+1}) \in E \) corresponds to the open interval between \( v_{ij} \) and \( v_{ij+1} \) on \( L \). Construct the interval graph \( G_1 \) of the set of open intervals corresponding to the edges of \( G \). In this interval graph, the maximum clique, i.e., the maximum number of intersecting intervals, is at most \( k \). As known \([9]\) an interval graph has the size of the maximum clique equal to the chromatic number. Hence the chromatic number of \( G_1 \) is at most \( k \). In a minimum coloring of \( G_1 \), every color is an independent set of vertices and no two of the corresponding intervals intersect, hence they can form a layer. Therefore the edges of \( G \) can be arranged on at most \( k \) layers, every layer corresponding to a color in the minimum coloring of \( G_1 \), and this is a solution to P.2.1. \( \square \)

Therefore P.2.1. is equivalent to the following problem:

P.2.2. Optimal linear cut arrangement.
Input: A graph \( G(V, E) \), \( V = \{v_1, \ldots, v_n\} \), a positive integer \( k \).
Property: There is a linear arrangement \( v_{i1}, \ldots, v_{in} \) of the vertices of \( G \) such that for every \( 1 \leq j < n \)

\[ c(v_{ij}, v_{ij+1}) < k. \]

We will prove that P.2.2. is NP-complete by reducing to it the Simple max cut problem. Independence from the present paper, the NP-completeness of P.2.2. was proved by L. Stockmeyer in a letter to M.R. Garey and D.S. Johnson \([5]\), but the result was never published.

Theorem 2.2. Simple max cut is \textit{optimal linear cut arrangement}, hence the \textit{optimal linear cut arrangement problem} is NP-complete.

Proof. Consider a graph \( G(V, E) \), \( |V| = m \), and a positive integer \( w \) as input for the Simple max cut problem. Let \( r = m^5 \) and let \( U = \{u_1, \ldots, u_r\} \) be a set of new vertices. We construct a new graph \( G(V, E) \) by taking \( V = V \cup U \) and \( E = \{(u, v) | u, v \in V, (u, v) \not\in E \} \).

For a completely connected graph with \( x \) vertices \( Y_1, \ldots, Y_x \) all the linear arrangements are equivalent and for any arrangement \( Y_{i1}, \ldots, Y_{in} \), we can calculate in polynomial time the value \( f(x) = \max_{1 \leq j < i} c(Y_{ij}, Y_{ij+1}) \). Let \( k = f(r^m) \) and consider the graph \( G(V, E) \), \( k \) as input for the \textit{optimal linear cut arrangement}.

Assume that \( G(V, E) \) has a cut \( V_1 = \{a_1, \ldots, a_t\}, V_2 = \{b_1, \ldots, b_{n-t}\} \) such that \( g(V_1, V_2) \geq w \). Consider the following linear arrangement of the vertices of \( V \): \( a_1, \ldots, a_t, b_1, \ldots, b_{n-t} \). It is easy to see that the biggest number of edges crossing an open interval between two consecutive vertices appears among the vertices \( u_1, \ldots, u_r \) and this number is less or equal to \( k \). Hence we have a solution of \( G(V, E) \), \( k \), for the \textit{optimal linear cut arrangement}.

Conversely, consider a linear arrangement \( Y_{i1}, Y_{i2}, \ldots, Y_{in} \) of the vertices of \( G \) giving a solution to P.2.2. Consider the complete graph \( \overline{G}_1 \) on the vertices \( Y_1, \ldots, Y_{n+r} \). In \( \overline{G}_1 \) let \( Y_{i1}, Y_{i1+1} \) be the place in which \( f(n+r) = \max_{1 \leq j < i} c(Y_{ij}, Y_{ij+1}) \). The edges of \( \overline{G}_1 \) crossing over the open segment \( Y_{i1}, Y_{i1+1} \) are those of \( E \), their number being at
most k (by the linear arrangement of $\overline{E}$), and those of $E$. Let $E_1$ be the subset of edges of $E$ crossing in $\overline{G}$ over the open segment $Y_{i_1}Y_{i_1+1}$. Thus $f(n+r)<k+|E_1|$ and $|E_1|>f(n+r)-k=f(n+r)-f(n+r)+w=w$. Therefore the open segment $Y_{i_1}Y_{i_1+1}$ defines on $G$ a cut $S,V-S$ such that $g(S,V-S)=w$, and this is a solution of the Simple max cut. □

Arrangements on a grid.

In this Section we consider grids formed in the plane by two sets of lines, one parallel to the vertical axis and the other to the horizontal axis, such that the distance between every two consecutive lines is the same. A grid of size $a \times b$ has exactly a horizontal and $b$ vertical lines. A place in the grid in which two lines intersect is called a node.

We will prove that the following problems are NP-complete.

P3.1. Intersection graph of segments on a grid.

Input: A graph $G(V,E)$ and two positive integers $a,b$.

Property: In the grid of size $a \times b$ there exists a set of segments $S$, every segment being on a line of the grid such that $G$ is the intersection graph of $S$.

P3.2. Intersection graph of parallelograms on a grid.

Input: A graph $G(V,E)$ and two positive integers $a,b$.

Property: In the grid of size $a \times b$ there exists a set of parallelograms $S$ whose edges are on the lines of the grid such that $G$ is the intersection graph of $S$.

The following problem was proved NP-complete in collaboration with A. Pnueli.

P3.3. Edge embedding on a grid.

Input: A graph $G(V,E)$ and two positive integers $a,b$.

Property: The vertices of $G$ can be arranged in the nodes of a grid $a \times b$ such that for every edge $(u,v)\in E$, the two vertices $u$ and $v$ are on the same line of the grid.

P3.4. Clique embedding on a grid.

Input: A graph $G(V,E)$ and two positive integers $a,b$.

Property: The vertices of $G$ can be arranged in the nodes of a grid $a \times b$ such that for every maximal clique $A$ of $G$, the vertices of $A$ are in one line and for two maximal cliques $A,B$ whose vertices appear on the same line, the two segments defined by $A$ and $B$ are disjoint.

Lemma 3.1. (a) p-Bin packing a Intersection graph of segments on a grid; (b) Intersection graph of segments on a grid a Intersection graph of parallelograms on a grid.

Proof. (a) Let $I=\{x_1,\ldots,x_n, k, b\}$ be an input for the p-Bin packing. Without loss of generality we can assume that $x_i>b$ for every $1<i<n$ (by the appropriate multiplication). Construct a graph $G(V,E)$ as follows:

$v=\{u_1,\ldots,u_n\} \cup \{\bigcup_{i=1}^{n} v_i^1, v_i^2, \ldots, v_i^k\}

E=\{(u_i,v_i^j) | 1<i<n, 1<j<k\}$. Let $G(V,E)$, $k$, $b-1$ be the input for P3.1.

Consider a solution of the p-Bin packing for I. We correspond to every $u_i$ a segment $u_i$ of length $x_i-1$, and to every $v_i^j$ a point $v_i^j$. Now we arrange the segments $u_i$ in the horizontal lines of the grid of size $k \times (b-1)$ according to the repetition of the $x_i$'s in the bins and along every $v_i$ we put the points $\{v_i^1, \ldots, v_i^k\}$ on the nodes of the grid. Clearly this is a solution for P3.1.

Conversely, it is easy to see that a solution of P3.1 with $G(V,E)$, $k$, $b-1$, gives a solution for the p-Bin packing. (b) Trivial. □

Lemma 3.2. (a) p-Bin packing a edge embedding on a grid; (b) p-Bin packing a Clique embedding on a grid.

Proof. Let $I=(x_1,\ldots,x_n, k, b)$ be an input for the p-Bin packing. We assume that $x_i>b$ for every $1<i<n$. For every $x_i$ let $\overline{x_i}=\{v_i^1,\ldots,v_i^k\}$.

Construct a graph $G(V,E)$ where $v=\bigcup_{i=1}^{n} \overline{x_i}$ and $E=\{(v_i^j,v_i^j) | 1<i<n, 1<j,k\}$. In fact for every $1<i<n$ the set $\overline{x_i}$ is a clique and $G$ is a collection of cliques. Consider $G(V,E)$, $k$, $b-1$ as an input for P3.3. or P3.4.

Consider a solution of the p-Bin packing for I. There is a correspondence between the $k$ bins and the $k$ horizontal lines of the $k \times (b-1)$ grid. For every bin, and every $x_i$ in the bin we put the clique $\overline{x_i}$ in the horizontal line of the grid corresponding to the bin. We put the vertices of $\overline{x_i}$ consecutively in the nodes of the grid. In this way we obtain a solution for P3.3. and P3.4.

Similarly, a solution of P3.3 or P3.4 with $G(V,E)$, $k$, $b-1$ gives a solution for the p-Bin packing. □

Theorem 3.3. The problems P3.1 - P3.4 are NP-complete.

4. NP-complete problems for line digraphs.

As proved in [8] (see also [1]), the following graph problems are NP-complete: independent set, node cover, feedback edge set, feedback node set. We will prove that these problems are NP-complete even when restricted to line-digraphs.

The following reduction was communicated to me by D. Knuth [10].

Lemma 4.1. Simple Max Cut a Independent set for line digraphs.

Proof. Let be given $G(V,E)$ and $w$ as input for the Simple Max Cut. Construct a digraph $G(V,E)$ by replacing each edge $(u,v)$ of $G$ with the two directed edges $(u,v)$ and $(v,u)$. Consider the line digraph $H$ of $G$ and $w$ as input for the Independent set for line digraphs.

For a cut $S,V-S$ of $G$ such that $g(S,V-S)=w$,
the set of edges \( \{(u,v) \mid u \in S, v \in V-S\} \) of \( G \) gives an independent set of at least \( w \) vertices in \( H \). Conversely, consider an independent set of vertices \( F_1 \) of \( H \) such that \( |F_1| > w \). Let

\[
S = \{u \in V, (u,v) \in F_1\}, \quad S_2 = \{v \notin V, (u,v) \in F_1\}.
\]

Since \( F_1 \) is an independent set of \( H \) it follows that \( S \neq S_2 \). Therefore \( g(S_1,V-S_1) > w \) in \( G \). \( \square \)

For every graph \( G(V,E) \) a set \( ACV \) is independent if and only if \( V-A \) is a node cover of \( G \).

Therefore:

**Lemma 4.2.** Independent set for line digraphs a Node cover for line digraphs.

**Lemma 4.3.** Feedback node set (f.n.s.) for line digraphs a Feedback edge set (f.e.s.) for line digraphs.

**Proof.** Let \( H(U,F) \) be a line digraph of a digraph \( G \) and let \( H(U,F) \) be a digraph defined as follows:

\[
U = \{x, y, z\}, \quad F = \{(x, y), (y, z), (z, x)\} \mid e \in E, F = \{(e, 0), (e, 1)\} \mid e \in E
\]

Hence, every vertex in \( H \) is replaced by two vertices \( e, 0 \) and \( e, 1 \) such that the incoming edges \( e, 0 \) correspond to the outgoing edges \( e, 1 \) and \( e \) is an edge from \( e, 0 \) to \( e, 1 \). Every vertex \( v \) corresponds to an edge \( (u,v) \) of \( G \). Let us construct a digraph \( G \) by inserting the middle of every edge \( e \) of \( G \) and \( e \) is a new vertex and let \( v \) denote the new edges by \( (e, 0, e, 1) \). It is easy to see that \( H \) is the line digraph of \( G \).

Consider a digraph \( G(V,E) \) and its line digraph \( H \). It is easy to see that a subset of edges \( E_1 \subseteq E \) is a feedback edge set of \( G \) if and only if the set of vertices corresponding to \( E_1 \) in \( H \) is a feedback node set of \( H \).

**Theorem 4.5.** The following problems for line digraphs are NP-complete: independent set, node cover, feedback node set, feedback edge set.

**Theorem 4.6.** The following problems are NP-complete even for graphs having every clique of size at most 3: independent set, node cover, feedback node set, feedback edge set.

5. Optimal directed tree arrangement.

Similarly to the Optimal Linear arrangement we can define the following problems:

**P5.1.** Optimal directed tree arrangement.

**Input:** A graph \( G(V,E) \) and a positive integer \( k \).

**Property:** The vertices of \( G \) can be arranged as a directed tree \( T \) such that the edges of \( G \) are along the directed paths of \( T \) and \( \sum_{(u,v)} d_T(u,v) \leq k \).

Given a set \( X = \{u_1, u_2, \ldots, u_n\} \) and a family \( Y = \{S_j\} \) of subsets of \( X \) we encounter sometimes the problem of arranging the elements of \( X \), in a specific pattern such that for every \( S_j \in Y \) the elements of \( S_j \) appear consecutively on the pattern.

For example this problem appears in information retrieval when \( X \) is a set of records and \( Y \) a set of queries. With \( X \) and \( Y \) we can construct the bipartite graph \( B(X,Y) \) whose set of vertices is \( X \cup Y \), two vertices \( u \in X \), \( v \in Y \) being adjacent if and only if \( u \sim v \). Since in most occurrences of \( X \) and \( Y \) the above arrangement is in pattern is impossible it is of interest to know the minimum number of "dummy" edges (between \( X \) and \( Y \) to be added to \( B \) such that the above arrangement is possible.

Relative to this we have the following problem:

**P5.2.** Augmented directed tree arrangement.

**Input:** A bipartite graph \( B(X,Y) \) and a positive integer \( k \).

**Property:** There is a bipartite graph \( B(X,Y) \) obtained by adding at most \( k \) edges to \( B(X,Y) \) fulfilling: the elements of \( X \) can be arranged as a direct tree \( T \) such that for every \( S_j \) the set of vertices adjacent to \( S \) in \( B \) form a directed path in \( T \).

**Theorem 5.1.** Optimal linear arrangement a Optimal directed tree arrangement, hence P5.1 is NP-complete.

**Proof.** Let \( G(V,E) \) and \( k \) be an input for the Optimal linear arrangement problem. Let \( r = |V| \) and let \( U = \{u_1, \ldots, u_r\} \) be a set of new vertices. Construct a graph \( G(V,E) \) having \( V = V \cup U \) and \( E = E \cup \{(u,v) \mid u \in U, v \in V\} \). Let \( k = r + \binom{|V|}{2} \). Let \( G(V,E) \) and \( k \) be an input for P5.1.

Consider a linear arrangement \( v_1, \ldots, v_n \) of the vertices of \( G \) as required by the Optimal linear arrangement. Then, the directed tree of Figure 2 is a solution for the Optimal directed tree arrangement with \( G(V,E), k \).

Conversely, let \( T \) be a solution of P5.1 with \( G(V,E) \) and \( k \). Let \( U \) be a vertex of \( G \) terminal (a sink) of \( T \). Then, all the vertices of \( V \) must be on the path from the root of \( T \) to \( v \) since all the edges of \( G \) must be along the directed paths of \( T \). Hence

\[
\sum_{(u,v)} d_T(u,v) = 1 + 2 + \ldots + r + \binom{|V|}{2} > k, \tag{1}
\]

contradicting the fact that \( T \) is a solution of P5.1 with \( G(V,E) \) and \( k \). Therefore in \( T \) no vertex of \( V \) is terminal. Let \( v \in V \) be a terminal vertex of \( T \). All the vertices of \( V \) must be on the directed path \( p \) from the root of \( T \) to \( u \) since the edges of \( G \) must be along the directed paths of \( T \). Let \( v \) be the element of \( V \) farthest from the root on
p. Assume that the vertices of \( V \) are not consecutive, and let \( u \in U \) be a vertex appearing on \( p \) between two vertices \( v_1, v_2 \) of \( V \) (\( v_1, v_2 \) need not necessarily be neighbours of \( u \) on \( p \)). Since every other vertex of \( U \) is connected to both \( v_1, v_2 \) and at least one of these edges passes over \( u \), it follows that the appearance of \( u \) between \( v_1, v_2 \) adds to the solution of P5.1. at least \( n - 1 \). The edges of \( u \) contribute to the solution by at most \( n^2 \). Hence, by deleting \( u \) from \( p \) and adding it to \( V \) by an edge \((V, u)\) as a terminal vertex of \( T \) we obtain a better solution. Doing the same with every vertex of \( U \) appearing on \( p \) between two vertices of \( V \) we obtain a better solution in which all the vertices of \( V \) appear consecutively on a path. Now it is easy to see that this path is a solution of the optimal linear arrangement with \( G(V, E) \) and \( k \). \( \square \)

**Theorem 5.2.** Optimal directed tree arrangement

An augmented directed tree arrangement, hence P5.2 is NP-complete.

**Proof.** Let \( G(V, E) \), \( k \) be an input for P5.1. Construct the bipartite graph \( B(V, E) \) whose set of vertices is \( V \cup E \), two vertices being adjacent if and only if they correspond to a vertex and an incident edge in \( G \). Let \( B(V, E) \) and \( k = k - |E| \) be an input for P5.2.

Consider a solution \( T \) of \( G(V, E) \), \( k \) for P5.1. In \( T \), if an edge \( e \) passes over a vertex \( v \), we add to \( B(V, E) \) the edge \((v, e)\). In this way, by adding to \( B(V, E) \) at most \( k - |E| \) edges we obtain a bipartite graph \( B(V, E) \) having \( T \) as solution for P5.2.

Conversely, a solution of P5.2. with \( B(V, E) \) and \( k \) is clearly a solution of P5.1. with \( G(V, E) \) and \( k \). \( \square \)

**References**


