# Systematized Approaches to the Complexity of Subgraph Problems

Miyano, Satoru
Research Institute of Fundamental Information Science Kyushu University



出版情報：RIFIS Technical Report. 25, 1990-04-09. Research Institute of Fundamental Information Science, Kyushu University

RIFIS-TR-CS-25

# RIFIS Technical Report

## Systematized Approaches to the Complexity of Subgraph Problems

Satoru Miyano

April 9, 1990

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan
E-mail: miyano@rifis.sci.kyushu-u.ac.jp Phone: 092(641)1101 Ext.4471

# Systematized Approaches to the Complexity of Subgraph Problems

Satoru Miyano
Research Institute of Fundamental Information Science
Kyushu University 33, Fukuoka 812, Japan

April 9, 1990

**Abstract**

This is a survey on complexity issues of subgraph problems proved in a systematic way. We deal with vertex-deletion and edge-deletion problems which can be viewed as subgraph problems. General NP-completeness theorems are presented for these problems. We also present a systematized result which shows polynomial time algorithms for these problems restricted to series-parallel graphs. Another problem we consider in this paper is the lexicographically first maximal subgraph problems which appear in connection with parallel complexity theory.

## 1 Introduction

A number of NP-complete problems have been shown in the literatures [8]. Most of these NP-complete problems are proved by giving reductions problem by problem. On the other hand, there are approaches to systemitize their reductions so that one need not prove the completeness individually.

Such systematized approaches have been very successful for subgraph problems which are to find a specified subgraph from a given graph. Formally, for a property $\pi$ on graphs (digraphs), the subgraph problem for $\pi$ is described as follows:

**Instance:** A graph (digraph) $G = (V, E)$.

**Problem:** Find a subgraph of $G$ satisfying the property $\pi$ if there is any such subgraph.

Problems of this kind involves many important problems in combinatorial graph algorithms. For example, the problem of finding a maximum independent set (or clique) in a graph is a subgraph problem for $\pi$="no two vertices are adjacent" (or $\pi$="any two vertices are adjacent") [4], [12]. The problem of computing a depth-first search tree is also of this kind [24].

This paper surveys general results on the complexity of subgraph problems with which we can determine their complexity simply by examining given properties.

This paper is organized as follows: Section 2 is for necessary definitions and terminologies. In Section 3 we deal with vertex deletion and edge deletion problems which can be formulated as maximum subgraph problems. We present important results by [2], [1], [15], [28], [29], [27] which systematize NP-completeness proofs for these problems. We also present an interesting result on linear time algorithms for series-parallel graphs by [23]. Section 4 is concerned with the lexicographically first maximal subgraph problems which can be solved by greedy algorithms. Based on [19], we first present general P-completeness results analogous to the results on NP-completeness. We also deal with a $\Delta_2^p$-completeness theorem which yields a new series of $\Delta_2^p$-complete problems [20].

## 2 Preliminaries

This section introduces some terminologies in graph theory ([9], [10]). Throughout this paper, a graph and digraph mean an undirected graph and a directed graph, respectively. Unless stated, all graphs and digraphs are simple except series-parallel graphs, i.e., no parallel edges are allowed.

Let $G = (V, E)$ be a graph (digraph). For a subset $U$ of vertices, the *induced subgraph of* $U$, denoted $G(U)$, is the graph defined by $G(U) = (U, E(U))$, where $E(U)$ consists of edges whose endpoints are both in $U$. For a subset $F$ of edges, the *edge-induced subgraph of* $F$, denoted $G[F]$, is the graph defined by $G[F] = (V(F), F)$, where $V(F)$ is the set of vertices appearing as endpoints of edges in $F$.

An edge $e$ is said to be *contracted* in $G$ if $e$ and all of its parallel edges, if they appear, are deleted and its endpoints are identified. A graph $H$ is a *contraction* of $G$ if $H$ can be obtained from $G$ by a sequence of edge contractions. A graph $H$ is called a *subcontraction* of $G$ if $H$ is isomorphic to a contraction of some subgraph of $G$.

Let $G = (V, E)$ be a connected graph. An *articulation point* is a vertex $v$ of $G$ whose deletion disconnects $G$. A connected graph $G$ is called *biconnected* if $G$ has no articulation point. The *biconnected components* of $G$ are the maximal biconnected subgraphs of $G$.

A pair $\{u, v\}$ of vertices is called a *separation pair* if there exist subgraphs $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ satisfying the following conditions:

(a) $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \{u, v\}$.

(b) $E = E_1 \cup E_2$, $E_1 \cap E_2 = \emptyset$, $|E_1| \geq 2$, $|E_2| \geq 2$.

(c) There are edges $e_1 \in E_1$ and $e_2 \in E_2$ such that there is a cycle in $G$ containing both $e_1$ and $e_2$.

A biconnected graph $G$ is called *3-connected* if it contains no separation pair. The *3-connected components* of $G$ are the maximal 3-connected subcontractions of $G$.

## 3 Vertex-Deletion and Edge-Deletion Problems

Many combinatorial graph problems can be formulated as vertex-deletion and edge-deletion problems. This section surveys some very general NP-completeness theorems on these problems. The importance of the results lie on the point that they systematized NP-completeness proofs and released us from attacking each problem individually.

### 3.1 Problems

Let $\pi$ be a property on graphs (digraphs) such as "planar". The *vertex deletion problem for* $\pi$ is the problem of finding a set of vertices of minimum size such that deletion of these vertices together with the edges adjacent to them results in a subgraph satisfying $\pi$. Equivalently, the vertex deletion problem is to find a set $U$ of vertices of *maximum size* such that the induced subgraph of $U$ satisfies $\pi$. By this correspondence, the vertex deletion problem for $\pi$ is also called the *maximum induced subgraph problem for* $\pi$.

Examples of graph (digraph) properties are listed below:

(1) Independent set (or null): No two vertices are adjacent.

(2) Clique (or complete): Every vertex is adjacent to all other vertices.

(3) Planar: A *planar* graph is a graph which has a layout on the plane in which no edges cross.

(4) Outerplanar: An *outerplanar* graph is a planar graph with a planar layout such that all vertices lie on the same face.

(5) Bipartite: A *bipartite graph* is a graph $G = (V, E)$ such that the vertex set $V$ is partitioned as $V = N \cup M$ and every edge in $E$ has one endpoint in $N$ and the other in $M$.

(6) Acyclic: Without any cycles. An acyclic graph is also called a *forest.* A set of vertices whose deletion results in an acyclic graph (digraph) is called a *feedback vertex set.*

(7) Maximum degree $k$: Every vertex is adjacent to at most $k$ vertices.

(8) Chordal: A graph $G$ is *chordal* if for every circuit of length greater than 3 there is an edge joining two nonconsecutive vertices of the circuit. A chordal graph is also called a *triangulated graph.*

(9) Line-invertible (or edge graph): A graph $G$ is *line-invertible* if there is a graph $H = (V, E)$ such that $G$ is isomorphic to the graph having $E$ as a vertex set and an edge set consisting of $\{e, e'\}$ such that $e$ and $e'$ share a common endpoint in $H$.

(10) Without cycles of length $l$: This property is for both graphs and digraphs.

(11) Without cycles of length $\leq l$: This property is for both graphs and digraphs.

(12) Transitive: A digraph $G = (V, E)$ is *transitive* if $(u, v) \in E$ and $(v, w) \in E$ implies $(u, w) \in E$.

(13) Symmetric: A digraph $G = (V, E)$ is *symmetric* if $(v, u) \in E \Leftrightarrow (u, v) \in E$.

(14) Antisymmetric: A digraph $G = (V, E)$ is *antisymmetric* if $(u, v) \in E \Rightarrow (v, u) \notin E$.

(15) Transitively orientable: A graph is *transitively orientable* if there is an assignment of directions to the edges such that the resulting digraph is transitive. A transitively orientable graph is also called a *comparability graph.*

(16) Interval graph: A graph is an *interval graph* if there is a one-to-one correspondence between the vertex set and a set of intervals such that

two vertices are adjacent if and only if their corresponding intervals have nonempty intersection.

(17) Nonseparable: A graph $G$ is *nonseparable* if it is connected, has more than one vertex and has no articulation points.

(18) With a singleton $k$-basis: We say that a graph $G$ has a *singleton $k$-basis* if each connected component of $G$ contains a vertex $v$ such that every vertex in the connectd component is of distance at most $k$ from $v$.

(19) Eulerian: A graph is called *Eulerian* if there is a path which passes through all edges exactly once.

The *edge deletion problem for $\pi$* is to find a set of edges of minimum size whose deletion results in a subgraph satisfying $\pi$. As in the case of the vertex deletion problem, the edge deletion problem can be regarded as the problem of finding a set $F$ of edges of mimimum size such that the edge-induced subgraph of $F$ satisfies $\pi$. We also call this problem the *maximum edge-induced subgraph problem for $\pi$*. However, there is a slight difference between the edge deletion problem and the maximum edge-induced subgraph problem. Edge deletions may produce vertices of degree 0 but every vertex of an edge-induced subgraph is of degree at least one. But we confuse these problems since only differeces are on vertices of degree 0.

A number of graph problems can also be viewed as maximum edge-induced subgraph problems. The followings are some examples: The maximum matching problem is the case for $\pi$="degree$\leq$ 1". The maximum cut problem is defined by setting $\pi$="bipartite". The Chinese postman problem is for $\pi$="Eulerian".

The *edge contraction problem for $\pi$* is to find a set of edges of minimum size whose contraction produces a subgraph satisfying $\pi$. This problem is not exactly a subgraph problem. But we deal with edge contraction problems since they are deeply related to edge deletion problems.

To discuss the complexity issues on these problems, we consider the following decision problems.

1. MAXIMUM INDUCED SUBGRAPH PROBLEM FOR $\pi$

**Instance:** A graph (digraph) $G = (V, E)$ and an integer $K \leq |V|$.

**Problem:** Decide whether there is a set $U$ of vertices with $|U| \geq K$ whose induced subgraph satisfies $\pi$.

2. MAXIMUM EDGE-INDUCED SUBGRAPH PROBLEM FOR $\pi$

   **Instance:** A graph (digraph) $G = (V, E)$ and an integer $K \leq |V|$.

   **Problem:** Decide whether there is a set $F$ of edges with $|F| \geq K$ such that the edge-induced subgraph of $F$ satisfies $\pi$.

3. EDGE-CONTRACTION PROBLEM FOR $\pi$

   **Instance:** A graph (digraph) $G = (V, E)$ and an integer $K \leq |V|$.

   **Problem:** Decide whether there is a set $F$ of edges with $|F| \leq K$ whose contracion results in a subgraph satisfiying $\pi$.

## 3.2 General NP-Completeness Results

### 3.2.1 Vertex-Deletion Problems

The vertex cover problem, which is known to be NP-complete [12], is regarded as the vertex deletion problem for $\pi =$"independent set".

Krishnanmoorthy and Deo [13] showed that the maximum induced subgraph problems are NP-complete for 17 explicit properties. For the NP-completeness proofs, they developed a rather unified approach for reductions from the vertex cover problem using forbidden subgraphs.

Then a more general NP-completeness theorem was obtained by Lewis and Yannakakis [15]. We need some definitions for stating their result.

Let $D$ be a class of graphs (digraphs). We say that a property $\pi$ is *nontrivial* on $D$ if infinitely many graphs (digraphs) in $D$ satisfy $\pi$ and infinitely many graphs (digraphs) in $D$ violates $\pi$. A property $\pi$ is said to be *hereditary* (resp., *hereditary on induced subgraphs*, *hereditary on contractions*) if, whenever a graph $G$ satisfies $\pi$, all subgraphs of $G$ (resp., induced subgraphs of $G$, contractions of $G$) satisfy $\pi$. Obviously, if a property is hereditary, then it is hereditary on induced subgraphs. A property $\pi$ is called *polynomial time testable* if there is a polynomial time algorithm deciding whether a graph (digraph ) $G$ satisfies $\pi$ or not.

The graph (digraph) properties of (1)-(16) in the above list are nontrivial, herediatry on induced subgraphs and polynomial time testable. But the property $\pi$="transitively orientable" is hereditary on induced subgraphs but not hereditary.

**Theorem 1** (Lewis and Yannakakis [15]) *Let $\pi$ be a property on graphs (digraphs). If $\pi$ is*

1. *nontrivial,*
2. *hereditary on induced subgraphs, and*
3. *polynomial time testable,*

*then MAXIMUM INDUCED SUBGRAPH PROBLEM FOR* $\pi$ *is NP-complete.*

If $\pi$ satisfies the conditions of Theorem 1 for planar graphs, then the problem whose instances are restricted to planar graphs is also NP-complete. Moreover, for digraphs, the problem restricted to acyclic digraphs is NP-complete under the same conditions on $\pi$ for acyclic digraphs [15].

Theorem 1 covers a large number of, in fact infinitely many, NP-complete maximum induced subgraph problems. As we have seen that the properties (1)-(16) of the list satisfy the conditions of Theorem 1, the corresponding maximum induced subgraph problems are all NP-complete. Theorem 1 was proved by reducing the vertex cover problem but requires different reductions according to graphs or digraphs.

For properties which are not hereditary, the maximum induced subgraph problems need not be NP-complete. For example, the maximum induced subgraph problem for $\pi =$"biconnected" is solvable in linear time [24].

The vertex cover problem allows a polynomial time algorithm by matching technique if instances are restricted to bipartite graphs [14]. Hence the restriction to bipartite graphs may make a problem easier. Yannakakis [27] analyzed the complexity of maximum induced subgraph problems restricted to bipartite graphs. He proved a very beautiful classification theorem exploiting complicated arguments.

For a graph $G = (V, E)$ and a vertex $u$, the neighborhood $N(u)$ of $u$ is defined by $N(u) = \{v \mid \{u, v\} \in E\}$. Then let $\nu(G)$ be the number of different neighborhoods of its nodes, i.e., $\nu(G) = |\{N(u) \mid u \in V\}|$. Then for a property $\pi$ on graphs we define $\nu(\pi) = \sup\{\nu(G) \mid G \text{ is a graph satisfying } \pi\}$.

**Theorem 2** (Yannakakis [27]) *Let* $\pi$ *be a nontrivial property on bipartite graphs which is hereditary on induced subgraphs and polynomial time testable. Then MAXIMUM INDUCED SUBGRAPH PROBLEM FOR* $\pi$ *restricted to bipartite graphs is,*

(1) *if* $\nu(\pi) = \infty$, *then NP-complete,*

(2) *if* $\nu(\pi) < \infty$, *then polynomial time computable.*

Yannakakis [25] considered how the connectedness condition affects the complexity of maximum induced subgraph problems.

MAXIMUM CONNECTED SUBGRAPH PROBLEM FOR $\pi$ is, given a graph (digraph) $G$ and an integer $K$, to decide whether there is a subset $U$ of vertices with $|U| \geq K$ whose induced subgraph is *connected* and satisfies $\pi$.

A property $\pi$ is *interesting on connected graphs* if there are arbitrarily large connected graphs satisfying $\pi$.

The following result asserts that the connectedness does not affect the complexity.

**Theorem 3** (Yannakakis [25]) *Let $\pi$ be a property on graphs. If $\pi$ is*

1. *hereditary on induced subgraphs,*
2. *nontrivial and interesting on connected graphs, and*
3. *polynomial time testable,*

*then MAXIMUM CONNECTED SUBGRAPH PROBLEM FOR $\pi$ is NP-complete.*

The same result is also shown for digraphs but we require the following additional condition [25]: There is a polynomial time algorithm which finds a digraph of $n$ vertices satisfying $\pi$ for every $n$.

The property $\pi$="maximum degree 2 and acyclic" satisfies the conditions of Theorem 3 and the connected graphs satisfying $\pi$ are paths. Therefore, the problem of finding a maximum induced path is NP-complete.

### 3.2.2 Edge-Deletion Problems

Yannakakis [27] showed that maximum edge-induced subgraph problems for some properties on graphs and digraphs are NP-complete. He proved the NP-completeness of the maximum edge-induced subgraph problems for the following properties by giving reductions individually: (a) without cycles of specified length $l$, or of any length $\leq l$, (b) connected and maximum degree $k$ ($k \geq 2$), (c) outerplanar, (d) transitive, (e) line-invertible, (f) bipartite, (g) transitively orientable.

It is natural to ask whether a result similar to Theorem 1 holds for maximum edge-induced subgraph problems. It is well-known that the maximum matching problem [14] and the Chinese postman problem [6] are solvable in polynomial time but the maximum cut problem is NP-complete [12]. Hence the situation is rather different from the vertex deletion problems. However, Watanabe, Ae and Nakamura [28], [29] have successed in proving a result analogous to Theorem 1.

Let $S$ be a set of graphs. We say that a property $\pi$ is *characterizable by forbidden subgraphs* (resp., *forbidden subcontractions, forbidden homeomorphic subgraphs, forbidden induced subgraphs*) in $S$ if a graph $G$ satisfies $\pi$ if and only if $G$ has no subgraph isomorphic to (resp., no subgraph homeomorphic to, no subcontraction isomorphic to, no induced subgraph isomorphic to) any graph in $S$. A graph property is said to be *finitely characterizable by 3-connected forbidden subcontractions* if there exists a finite nonempty set $S$ of 3-connected graphs such that $\pi$ is characterizable by forbidden subcontractions in $S$.

For example, the property $\pi$="planar" is characterizable by forbidden homeomorphic subgraphs in $\{K_{3,3}, K_5\}$.

**Theorem 4** (Watanabe, Ae and Nakamura [28], [29]) *If $\pi$ be a nontrivial property on graphs which is finitely characterizable by 3-connected forbidden subcontractions, then the following problems are NP-complete.*

(1) *MAXIMUM EDGE-INDUCED SUBGRAPH PROBLEM FOR $\pi$.*

(2) *EDGE CONTRACTION PROBLEM FOR $\pi$.*

Asano and Hirata [2] improved Theorem 4 as follows: A property $\pi$ on graphs is *determined by the 3-connected components* if a graph $G$ satisfies $\pi$ if and only if every 3-connected component of $G$ satisfies $\pi$.

It can be seen that if $\pi$ is characterizable by 3-connected forbidden subcontractions then it is hereditary on subgraphs and determined by the 3-connected components but the converse is not true.

Examples of properties $\pi$ which are hereditary on subgraphs and determined by the 3-connected components are $\pi$ ="planar" and $\pi$ ="series-parallel".

**Theorem 5** (Asano and Hirata [2]) *Let $\pi$ be a nontrivial property on graphs which is hereditary, determined by the 3-connected components and polynomial time testable. Then the following problems are NP-complete.*

(1) *MAXIMUM EDGE-INDUCED SUBGRAPH PROBLEM FOR $\pi$.*

(2) *EDGE CONTRACTION PROBLEM FOR $\pi$.*

Furthermore, Asano [1] extended the arguments in [28], [29] and showed that the problem remains NP-complete even if instances are restricted to planar graphs.

As to edge contraction problems, Asano [3] also showed that if a property $\pi$ is nontrivial on connected graphs, hereditary on contractions, determined by the biconnected components and polynomial time testable then EDGE-CONTRACTION PROBLEM FOR $\pi$ is NP-complete.

The following results due to El-Mallah and Colbourn [7] also covers quite large NP-hard families.

**Theorem 6** (El-Mallah and Colbourn [7]) *Let $S$ be a set of biconnected graphs with minimum degree at least 3. If a property $\pi$ is characterizable by forbidden homeomorphic subgraphs (resp., forbidden subcontractions) in $S$, then MAXIMUM EDGE-INDUCED SUBGRAPH PROBLEM FOR $\pi$ is NP-hard.*

### 3.3 Restriction to Series-Parallel Graphs

Most of NP-complete graph problems fall in P when instances are appropriately restricted. For example, the restriction to bipartite graphs allows the vertex cover problem a polynomial time algorithm. Such restrictions that make NP-complete problems solvable in polynomial time are found for each problem in the appendix of [8]. It is not our purpose of this paper to enumerate these restrictions but to show a class of graphs for which subgraph problems can be solved in polynomial time.

Takamizawa, Nishizeki and Saito [23] showed in a unified way that maximum induced subgraph and maximum edge-induced subgraph problems are linear time computable for series-parallel graphs.

In this section we deals with graphs (digraphs) with multiple edges since we consider series-parallel graphs. We say that two edges are *series* (resp., *parallel*) if they are incident to a vertex of degree 2 (resp., if they join the same pair of distinct vertices). A *series-parallel graph* is defined recursively as follows:

(a) A graph consisting of two vertices joined by two parallel edges is a series-parallel graph.

(b) If $G$ is a series-parallel graph, then a graph obtained by replacing any edge of $G$ by series or parallel edges is a series-parallel graph.

**Theorem 7** (Takamizawa, Nishizeki and Saito (1982)) *Let $\pi$ be a property on graphs characterizable by a finite number of forbidden subgraphs (resp., forbidden induced subgraphs). Then the maximum edge-induced subgraph problem (resp., maximum induced subgraph problem) for $\pi$ is linear time computable for series-parallel graphs.*

# 4 Problems Solvable by Greedy Algorithms

Instead of finding a maximum size subgraphs, there is a way of finding a *maximal* subgraph satisfying a given property. One of the simplest ways of finding a maximal subgraph is to employ greedy methods. This section considers algorithms which finds the *lexicographically first maximal* subgraphs.

Let $G = (V, E)$ be a graph (digraphs) with $V = \{1, ..., n\}$. The vertices in $V$ are linearly ordered as $1 < \cdots < n$. For a hereditary property $\pi$, consider the following greedy algorithm:

**begin** /* $G = (V, E)$ is given, where $V = \{1, ..., n\}$ */
  $U \leftarrow \emptyset$;
  **for** $i \leftarrow 1$ **to** $n$ **do**
    **if** the subgraph induced by $U \cup \{i\}$ satisfies $\pi$
      **then** $U \leftarrow U \cup \{i\}$
**end**

Algorithm 1: Greedy algorithm for maximal subgraphs

The set $U$ of vertices computed by the above algorithm is the lexicographically first maximal set of vertices whose induced subgraph of $U$ satisfies $\pi$. Formally, the lexicographic order on the set $2^V$ of all subsets of $V$ is defined as follows, where $V = \{1, ..., n\}$:

$\emptyset < \{1\} < \{1,2\} < \cdots < \{1,2,...,n\} < \{2\} < \{2,3\} < \cdots < \{2,3,...,n\}$
$< \{3\} < \cdots < \{3,...,n\} < \cdots < \{n-1,n\} < \{n\}$

For example, the lexicographically first maximal independent set is shown in Figure 1 as the gray vertices.

The problem we consider is the following decision problem, where LF is an abbreviation for "lexicogaphically first":

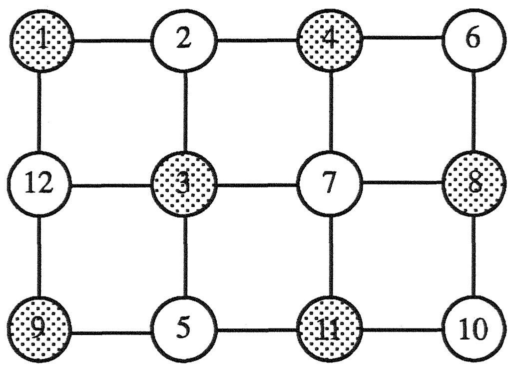


図 1: The lexicographically first maximal independent set

4. LF MAXIMAL SUBGRAPH PROBLEM FOR $\pi$ (LFMSP($\pi$))

   **Instance:** A graph (digraph) $G = (V, E)$ and a vertex $v$, where $V = \{1, ..., n\}$.

   **Problem:** Decide whether the vertex $v$ is in the lexicographically first maximal subset $U$ of vertices whose induced subgraph satisfies $\pi$.

### 4.1 General P-Completeness Theorems for LFMSP($\pi$)

P denotes the class of sets accepted by polynomial time deterministic Turing machines. A set $S$ is called *P-complete* [11] if (1) $S$ is in P and (2) every problem in P is log-space reducible to $S$. Another definition of P-completeness is given by NC-reducibility by [5] but the difference is not important in this paper. Recently, P-complete problems have received considerable attensions since any P-complete problem does not seem to allow efficient parallel algorithms [5], [18]. Some amount of P-complete problems are also reported [21].

**Theorem 8** (Miyano [19]) *Let $\pi$ be a nontrivial property on graphs (digraphs) which is hereditary on induced sugraphs and polynomial time testable. Then LFMSP($\pi$) is P-complete.*

The above theorem also holds when the instances are restricted to planar (resp., bipartite) graphs and $\pi$ satisfies the conditions of Theorem 8 for planar (resp., bipartite) graphs. These results are proved by reducing the lexicographicaly first maximal independent set problem restricted to planar (resp., bipartite) graphs which is P-complete [19].

Unfortunately, the lexicographically first maximal independent set problem restricted to planar bipartite graphs is not known to be P-complete. By this reason, we need a new analysis for simultaneously planar and bipartite graphs. We call a collection of disjoint edges *independent edges.* With an additional condition for independent edges, LFMSP($\pi$) restricted to planar bipartite graphs becomes P-complete.

**Theorem 9** (Miyano [19]) *Let $\pi$ be a nontrivial property on planar bipartite graphs. If $\pi$ is satisfied by all independent edges, hereditary on induced subgraphs and polynomial time testable, then LFMSP($\pi$) is P-complete.*

By Theorems 8, 9, the problem of finding the maximal induced subgraph by Algirithm 1 for many hereditary properties is seen to be P-complete, hence, hardly efficiently parallelizable.

When a linear order is given on the edge set as $E = \{e_1 < e_2 < \cdots < e_m\}$, we can also consider the lexicographically first maximal edge-induced subgraph satisfying a given property $\pi$. As we have seen in Secton 3, general NP-completeness results are known for the maximum edge-induced subgraph problems. But we do not know such general P-completeness results for the lexicographically first maximal edge-induced subgraph problems. The situation is rather different from that of induced subgraphs. We have the following observations:

(1) For the properties $\pi$="acyclic" and $\pi$="bipartite", the lexicographically first maximal edge-induced subgraph problems have efficient parallel algorithms. Hence they do not seem to be P-complete [19].

(2) For the property $\pi$="without cycles of length $k$" ($k \geq 3$), the lexicographically first maximal edge-induced subgraph problem is P-complete [19].

(3) For the property $\pi$="maximum degree 1", the problem is the lexicographically first maximal matching problem. This problem is shown CC-complete [16]. This fact implies that this problem may be neither P-complete and nor efficiently parallelizable.

## 4.2 General $\Delta_2^p$-Completenes Theorem

A typical nonhereditary graph property is "connected". Theorem 3 shows that the connectedness neither increases nor decreases the complexity of many maximum subgraph problems. However, the complexity of LFMSP($\pi$) changes drastically when the connectedness is added to the property. The class we consider here is $\Delta_2^p$ (also denoted $\mathrm{P}^{\mathrm{NP}}$), which is the class of sets accepted by deterministic polynomial time oracle Turing machines using oracles in NP (see [8]). This class obviously contains NP and co-NP.

Algorithm 1 computes the lexicographically first maximal set when the property $\pi$ is hereditary on induced subgraphs. In general, for any property $\pi$ (not necessarily hereditary), the lexicographically first maximal set of vertices which induces a subgraph satisfying $\pi$ is computed by the following algorithm (Algorithm 2):

**begin** /* $G = (V, E)$ is given, where $V = \{1, ..., n\}$ */
    $U \leftarrow \emptyset$;
    **for** $i \leftarrow 1$ **to** $n$
        **if** there exists a set $W$ satisfying
                1. $W \supseteq U \cup \{i\}$
                2. the induced subgraph of $W$ satisfies $\pi$
            **then** $U \leftarrow U \cup \{i\}$
**end**

Algorithm 2: General LFMSP($\pi$) algorithm

Again we consider the following decison problem:

5. LF MAXIMAL CONNECTED SUBGRAPH PROBLEM FOR $\pi$ (LFMCSP($\pi$))

   **Instance:** A graph (digraph) $G = (V, E)$ and a vertex $v$, where $V = \{1, ..., n\}$.

   **Problem:** Decide whether the vertex $v$ is in the lexicographically first maximal subset $U$ whose induced subgraph is *connected* and satisfies $\pi$.

If $\pi$ is polynomial time testable, it is easy to see that LFMCSP($\pi$) is in solvable by a deterministic polynomial time oracle Turing machine using the NP-oracle which decide the **if**-condition of Algorithm 2. Hence it is in $\Delta_2^p$. For this problem we also have a general completeness theorem.

We say that a graph property $\pi$ is *determined by the blocks* if for any graphs $G_1$ and $G_2$ satisfying $\pi$ the graph formed by identifying any vertex of $G_1$ and any vertex of $G_2$ also satisfies $\pi$.

**Theorem 10** (Miyano [20]) *Let $\pi$ be a nontrivial property on graphs which is hereditary on induced sugraphs, determined by the blocks and polynomial-time testable. Then LFMCSP($\pi$) is $\Delta_2^p$-complete.*

Theorem 10 is proved by reducing the deterministic satisfiability problem [22] that was shown $\Delta_2^p$-complete.

One of the interesting properties not covered by Theorem 10 is $\pi_0$="maximum degree 2 and acyclic" for which the connected induced subgraphs are paths. For the property $\pi_0$, LFMCSP($\pi_0$) is also shown $\Delta_2^p$-complete by giving an individual reduction [18]. Hence a more general result seems to hold.

For a property $\pi$, we define the *diameter* $\delta(\pi)$ by $\sup\{\delta(G) \mid G$ is a connected graph satisfying $\pi\}$, where $\delta(G)$ is the diameter of $G$. For example, $\delta$("planar")$=\infty$, $\delta(\pi_0) = \infty$ and $\delta$("clique")=1. By Theorem 8 LFMCSP("clique") is P-complete. On the other hand, LFMCSP($\pi_0$) and LFMCSP("planar") (by Theorem 10) are $\Delta_2^p$-complete. By these observations, we conjecture the following:

**Conjecture 1** If a property $\pi$ is nontrivial on connected graphs and satisfies $\delta(\pi) = \infty$, then LFMCSP($\pi$) is $\Delta_2^p$-hard.

## 5 Conclusions

We surveyed some general theorems for showing completeness for NP, P and $\Delta_2^p$. Since thousands of natural NP-complete problems have been reported, a single NP-complete problem may not be very attractive. However, the approaches presented in this paper cover a large class of problems in a systematic way. Hence the systematic approaches will increase importance in the analysis of complexity.

## 6 Acknowledgements

The author would like to thank Toshimasa Watanabe for valuable informations.

## References


[1] ASANO, T., An application of duality of edge-deletion problems, *SIAM J. Comput.*, **16**, 2 (1987), 312-331.

[2] ASANO, T. and HIRATA, T., Edge-deletion and edge-contraction problems, *Proc. 14th ACM Symposium on Theory of Computing* (1982), 245-254.

[3] ASANO, T. and HIRATA, T., Edge-contraction problems, *J. Comput. System Sci.*, **26**, 2 (1983), 197-208.

[4] BALAS, E. and CHANG, S.-Y., Finding a maximun clique in an arbitrary graph, *SIAM J. Comput.*, **15**, 4 (1986), 1054-1068.

[5] COOK, S.A., A taxonomy of problems with fast parallel algoritms, *Inf. Comput.*, **64**, 1 (1985), 2-22.

[6] EDMONDS, J. and JOHNSON, E.L., Matching, Euler tours, and the Chinese postman, *Math. Program.*, **5**, (1973), 88-124.

[7] EL-MALLAH, E.S. and COLBOURN, C.J., The complexity of some edge deletion problems, *IEEE Trans. Circuits and Systems*, **35**, 3 (1988), 354-362.

[8] GAREY, M.R. and JOHNSON, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completenes*, W.H. Freeman and Company, San Francisco, 1979.

[9] GOLUMBIC, M.C., *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[10] HARARY, F., *Graph Theory*, Addison-Wesley, Reading, Mass., 1969.

[11] JONES, N.D. and LAASER, W.T., Complete problems for deterministic polynomial time, *Theor. Comput. Sci.*, **3**, 1 (1977), 105-117.

[12] KARP, R.M., Reducibility among combinatorial problems, in *Complexity of Computer Computation* (R.E. Miller and J.W. Thatcher, eds.), Plenum Press, New York, 1972, 85-103.

[13] KRISHNAMOORTHY, M.S. and DEO, N., Node-deletion NP-complete problems, *SIAM J. Comput.*, **8**, 4 (1979), 619-625.

[14] LAWLER, E.L., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart & Winston, New York, 1976.

[15] LEWIS, J.M. and YANNAKAKIS, M., The node-deletion problem for hereditary properties is NP-complete, *J. Comput. Syst. Sci.*, **20**, 2 (1980), 219-230.

[16] MAYR, E.W. and SUBRAMANIAN, A., The complexity of circuit value and network stability, *Proc. 4th Annual Conference on Structure in Complexity Theory*, (1989), 114-123.

[17] MIYANO, S., $\Delta_2^p$-complete lexicographically first maximal subgraph problems, *Proc. 13th Mathematical Foundation of Computer Science* (Lecture Notes in Computer Science **324**), (1988), 454-462.

[18] MIYANO, S., Parallel complexity and P-complete problems, *Proc. Int. Conf. Fifth Generation Computer Systems 1988*, (1988), 532-541.

[19] MIYANO, S., The lexicographically first maximal subgraph problems: P-completeness and NC algorithms, *Math. Systems Theory*, **22**, 1 (1989), 47-73.

[20] MIYANO, S., A new series of $\Delta_2^p$-complete problems, *Proc. Int. Workshop on Discrete Algorithms and Complexity*, (1989), 187-194.

[21] MIYANO, S., SHIRAISHI, S. and SHOUDAI, T., A list of P-complete problems, RIFIS-TR-CS-17, Research Institute of Fundamental Information Science, Kyushu University, 1989.

[22] PAPADIMITRIOU, C.H., On the complexity of unique solution, *J. ACM*, **31**, 2 (1984), 392-400.

[23] TAKAMIZAWA, K., NISHIZEKI, T. and SAITO, N., A linear-time computability of combinatorial problems on series-parallel graphs, *J. ACM*, **29**, 3 (1982), 623-641.

[24] TARJAN, R.E., Depth-first-search and linear graph algorithms, *SIAM J. Comput.*, **1**, 2 (1972), 146-160.

[25] YANNAKAKIS, M., The effect of a connectivity requirement on the complexity of maximum subgraph problems, *J. ACM*, **26**, 4 (1979), 618-630.

[26] YANNAKAKIS, M., Edge-deletion problems, *SIAM J. Comput.*, **10**, 2 (1981), 297-309.

[27] YANNAKAKIS, M., Node-deletion problems on bipartite graphs, *SIAM J. Comput.*, **10**, 2 (1981), 310-327.

[28] WATANABE, T, AE, T. and NAKAMURA, A., On the removal of forbidden graphs by edge-deletion or by edge-contraction, *Discrete Appl. Math.*, **3**, 2 (1981), 151-153.

[29] WATANABE, T, AE, T. and NAKAMURA, A., On the NP-hardness of edge-deletion and -contraction problems, *Discrete Appl. Math.*, **6**, 1 (1983), 63-78.