Invited Survey Paper

Systematized Approaches to the Complexity of Subgraph Problems

SATORU MIYANO*

This is a survey of issues related to the complexity of subgraph problems proved in a systematic way. It deals with vertex deletion and edge deletion problems that can be viewed as subgraph problems. General NP-completeness theorems for these problems are presented, as well as a systematized result that shows polynomial time algorithms for these problems restricted to series-parallel graphs. Another problem considered in this paper is the lexicographically first maximal subgraph problems that appear in connection with parallel complexity theory.

1. Introduction

A number of NP-complete problems have been shown in the literature [8]. Most of them are proved by giving reductions problem by problem. On the other hand, there are approaches for systematizing their reductions so that their completeness need not be proved individually.

Such systematized approaches have been very successful for subgraph problems involving finding a specified subgraph from a given graph. Formally, for a property π on graphs (digraphs), the subgraph problem for π is described as follows:

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

Problem: Find a subgraph of G satisfying the property π 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 that allow the complexity of subgraph problems to be determined simply by examining given properties.

This paper is organized as follows: Section 2 gives the necessary definitions and terminology. In Section 3, we deal with vertex deletion and edge deletion problems that can be formulated as maximum subgraph problems. We present various important results [1, 2, 15, 27, 28, 29] that systematize NP-completeness proofs for these problems. We also present an interesting result on linear time algorithms for series-parallel graphs [23]. Section 4 is concerned with lexicographically first maximal subgraph problems that can be solved by greedy algorithms. Following the approach taken in a previous paper [19], we present general P-completeness results analogous to the results on NP-completeness. We also deal with a Δ_2^p-completeness theorem that yields a new series of Δ_2^p-complete problems [20].

*Research Institute of Fundamental information Science, Faculty of Science, Kyushu University, 33, Fukuoka 812, Japan.

2. Preliminaries

This section introduces some terminology related to 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, that is, 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 by $G(U)$, 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 by $G[F]$, 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 the endpoints of e 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 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) $E=E_1 \cup E_2$, $E_1 \cap E_2=0$, $|E_1| \geq 2$, $|E_2| \geq 2$.

(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 is that they systematize NP-completeness proofs and remove the need to solve each problem individually.

3.1 Problems

Let π be a property on graphs (digraphs) such as "planar." The *vertex deletion problem for* π 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 π. Equivalently, the vertex deletion problem is to find a set U of vertices of *maximum size* such that the induced subgraph of U satisfies π. By this correspondence, the vertex deletion problem for π is also called the *maximum induced subgraph problem for* π.

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 that has a layout on a 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 $(v, u) \in E \Longleftrightarrow (u, v) \in E$.

(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 connected component is at a distance of at most k from v.

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

The *edge deletion problem for* π is to find a set of edges of minimum size whose deletion results in a subgraph satisfying π. As in the case of the vertex deletion problem, the edge deletion problem can be regarded as a problem of finding a set F of edges of maximum size such that the edge-induced subgraph of F satisfies π. We also call this problem the *maximum edge-induced subgraph problem for* π. 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. These problems are often confused, since the only differences are in vertices of degree 0.

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

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

In discussing the issues related to the complexity of these problems, we consider the following decision problems:

1. MAXIMUM INDUCED SUBGRAPH PROBLEMS FOR π

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 π.

2. MAXIMUM EDGE-INDUCED SUBGRAPH PROBLEM FOR π

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 π.

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 contraction results in a subgraph satisfiying π.

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 π="independent set".

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

A more general NP-completeness theorem was obtained by Lewis and yannakakis [15]. We need some definitions before stating their result.

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

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

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

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

then the MAXIMUM INDUCED SUBGRAPH PROBLEM FOR π is NP-complete.

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

Theorem 1 covers infinitely many NP-complete maximum induced subgraph problems. As we have seen, properties (1)-(16) of the list satisfy the conditions of Theorem 1, and thus 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 whether it is used for graphs or digraphs.

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

The vertex cover problem allows a polynomial time algorithm by the matching technique if instances are restricted to bipartite graphs [14]. This restriction 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 by complicated arguments.

For a graph $G=(V, E)$ and a vertex u, the neighborhood $N(u)$ of u is defined by $N(u)=\{v \mid \{u, v\} \in E\}$. 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 π on graphs we define $\nu(\pi)=\sup\{\nu(G) \mid G$ is a graph satisfying $\pi\}$.

Theorem 2 (Yannakakis [27]) *Let π be a nontrivial property on bipartite graphs that is hereditary on induced subgraphs and polynomial time testable. The the MAXIMUM INDUCED SUBGRAPH PROBLEM FOR π restricted to bipartite graphs is*

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

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

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

The MAXIMUM CONNECTED SUBGRAPH PROBLEMS FOR π 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 π.

A property π is *interesting on connected graphs* if there are arbitrarily large connected graphs satisfying π.

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

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

1. *hereditary on induced subgraphs,*
2. *nontrivial and interesting on connected graphs, and*

3. polynomial time testable,
then the MAXIMUM CONNECTED SUBGRAPH PROBLEM FOR π is NP-complete.

The same result is also shown for digraphs, but requires the following additional condition [25]: There is a polynomial time algoprithm that finds a digraph on n vertices satisfying π for every n.

The property π= "maximum degree 2 and acyclic" satisfies the conditions of Theorem 3 and the connected graphs satisfying π 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 in vertex deletion problems. However, Watanabe, Ae and nakamura [28, 29] have succeeded in proving a result analogous to Theorem 1.

Let S be a set of graphs. We say that a property π is *characterizable by forbidden subgraphs* (resp., *forbidden subcontractions, forbidden homemorphic subgraphs, forbidden induced subgraphs*) in S if a graph G satisfies π 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 π is characterizable by forbidden subcontractions in S.

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

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

(1) *MAXIMUM EDGE-INDUCED SUBGRAPH PROBLEM FOR π*

(2) *EDGE CONTRACTION PROBLEM FOR π.*

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

It can be seen that if π 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 π that are hereditary on subgraphs and determined by the 3-connected components are π= "planar" and π= "series-parallel".

Theorem 5 (Asano and Hirata [2]) *Let π be a nontrivial property on graphs that 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 π*

(2) *EDGE CONTRACTION PROBLEM FOR π.*

Furthermore, Asano [1] extended the arguments in Watanabe *et al.* [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 π is nontrivial on connected graphs, hereditary on contractions, determined by the biconnected components, and polynomial time testable, then the EDGE CONTRACTION PROBLEM FOR π is NP-complete.

The following results obtained by El-Mallah and Colbourn [7] also cover 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 π is characterizable by forbidden homeomorphic subgraphs (resp., forbidden subcontractions) in S, then the MAXIMUM EDGE-INDUCED SUBGRAPH PROBLEM FOR π is NP-hard.*

3.3 Restriction to Series-Parallel Graphs

Most NP-complete graph problems fall in P when instances are appropriately restricted. For example, the restriction to bipartite graphs allows a polynomial time algorithm to be used for the vertex cover problem. Such restrictions that make NP-complete problems solvable in polynomial time are found for each problem in the appendix of Garey and Johnson [8]. It is not the 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

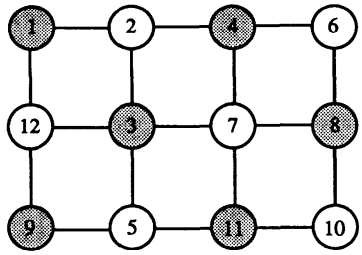

Fig. 1 The lexicographically first maximal independent set.

edges is a series-parallel graph.

Theorem 7 (Takamizawa, Nishizeki, and Saito [23]) *Let π 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 π is linear time computable for series-parallel graphs.*

4. Problems Solvable by Greedy Algorithms

Instead of finding a maximum size subgraph, 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 that find the *lexicographically first maximal* subgraphs.

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

begin/*$G=(V, E)$ is given, where $V=\{1, \ldots, n\}$*/
  $U \leftarrow \emptyset$;
  for $i \leftarrow 1$ **to** n **do**
    if the subgraph induced by $U \cup \{i\}$ satisfies π
      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 π. Formally, the lexicographic order on the set 2^V of all subsets of V is defined as follows, where $V=\{1, \ldots, n\}$:

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

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

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

4. LF MAXIMAL SUBGRAPH PROBLEM FOR π (LFMSP(π))

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

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

4.1 General P-Completeness Theorems for LFMSP (π)

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 [5] but the difference is not important in this paper. Recently, P-complete problems have received considerable attention, since any P-complete problem does not seem to allow efficient parallel algorithms [5, 18]. Some P-complete problems have also been reported [21].

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

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

Unfortunately, the lexicographically first maximal independent set problem restricted to planar bipartite graphs is not known to be P-complete. For 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(π) restricted to planar bipartite graphs becomes P-complete.

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

From Theorems 8 and 9, the problem of finding the maximal induced subgraph by Algorithm 1 for many hereditary properties is seen to be P-complete, and thus difficult to parallelize efficiently.

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 π. As we saw in Section 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 may make the following observations:

(1) For the properties π=''acyclic'' and π=''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 π=''without cycles of length k'' ($k \geq 3$), the lexicographically first maximal edge-induced subgraph problem is P-complete [19].

(3) For the property π=''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 nor efficiently parallelizable.

4.2 General Δ_2^p-Completeness 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(π) changes drastically when the connectedness is added to the property. The class we consider here is Δ_2^p (also denoted by P^{NP}), which is the class of sets accepted by deterministic polynomial time oracle Turing machines using oracles in NP [8]. This class obviously contains NP and co-NP.

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

begin/*$G=(V, E)$ is given, where $V=\{1, \ldots, n\}$*/
  $U \leftarrow 0$;
  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 π
    then $U \leftarrow U \cup \{i\}$
end

Algorithm 2: General LFMSP(π) algorithm

Again we consider the following decision problem:

5. LF MAXIMAL CONNECTED SUBGRAPH PROBLEM FOR π (LFMCSP(π))

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

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

If π is polynomial time testable, it is easy to see that LFMCSP(π) is solvable by a deterministic polynomial time oracle Turing machine using the NP-oracle that decides the **if**-condition of Algorithm 2. Hence it is in Δ_2^p. For this problem we also have a general completeness theorem.

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

Theorem 10 (Miyano [20]) *Let π be a nontrivial property on graphs that is hereditary on induced sugraphs, determined by the blocks and polynomial time testable. Then LFMCSP(π) is Δ_2^p-complete.*

Theorem 10 is proved by reducing the deterministic satisfiability problem [22], which was shown to be Δ_2^p-complete.

One of the interesting properties not covered by Theorem 10 is π_0=''maximum degree 2 and acyclic'', for which the connected induced subgraphs are paths. For the property π_0, LFMCSP(π_0) is also shown Δ_2^p-complete by giving an individual reduction [18]. Hence a more general result seems to hold.

For a property π, 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, δ(''planar'') $=\infty$, $\delta(\pi_0)=\infty$ and δ(''clique'')$=1$. By Theorem 8 LFMCSP (''clique'') is P-complete. On the other hand, LFMCSP(π_0) and LFMCSP(''planar'') (by Theorem 10) are Δ_2^p-complete. From these observations, we may make the following conjecture:

Conjecture 1 If a property π is nontrivial on connected graphs and satisfies $\delta(\pi)=\infty$, then LFMCSP(π) is Δ_2^p-hard.

5. Conclusions

We surveyed some general theorems for showing completeness for NP, P, and Δ_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. Such systematic approaches will have increasing importance in the analysis of complexity.

Acknowledgements

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

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. Syst. Sci.*, **26**, 2 (1983), 197-208.
4. Balas, E. and Chang, S.-Y. Finding a maximum clique in an arbitrary graph. *SIAM J. Comput.*, **15**, 4 (1986), 1054-1068.
5. Cook, S. A. A taxonomy of problems with fast parallel algorithms, *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-Completeness*, 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. KRISHNANMOORTHY, 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. Δ_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. on 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 Δ_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. 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.

(Received April 16, 1990)