Δ_2^p-Complete Lexicographically First Maximal Subgraph Problems

Satoru Miyano

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

Abstract

The lexicographically first maximal (lfm) induced path problem is shown Δ_2^p-complete. The lfm rooted tree problem is also Δ_2^p-complete. But when restricted to topologically sorted directed acyclic graphs (dags), the lfm rooted tree problem allows a polynomial time algorithm. Moreover, the problem restricted dags with degree 3 is shown in NC^2 while the problem for degree 4 is P-complete.

1. Introduction

Papadimitriou [Pa] is the first who gave a natural problem complete for Δ_2^p, which is the class of problems solvable in polynomial time using oracles in NP. He proved that the uniquely optimum traveling salesman problem is Δ_2^p-complete. Afterwards, Wagner [Wa] has found some Δ_2^p-complete problems related to optimization problems. This paper gives Δ_2^p-complete problems of a new kind. The importance of the Δ_2^p-completeness is not only due to its high complexity but also due to the observation that any Δ_2^p-complete problem is hard to efficiently parallelize even if NP-oracles are available.

For a given hereditary property π on graphs, we consider the problem of finding the lexicographically first maximal (abbreviated to lfm) subset U of vertices of a graph $G = (V, E)$ such that U induces a *connected* subgraph satisfying π, where we assume that V is linearly ordered as $V = \{1, ..., n\}$. Problems of this kind have been extensively studied in [AM], [Ma], [M1], [M2]. In particular, without the connectedness restriction, the P-completeness of the lfm subgraph problem for *any* nontrivial polynomial time testable hereditary property is proved in [M1] as an analogue of the results in [LY], [Y2]. However, since the connectedness is not necessarily inherited by subgraphs, a new analysis is required.

In this paper we are involved in the complexity analysis of problems of this kind. Our main result is that the lfm induced path problem is Δ_2^p-complete. We should here note that this problem is different from the lfm maximal path problem discussed in [AM] which was shown P-complete.

Some of the lfm connected subgraph problems for hereditary properties are polynomial time solvable. For example, the lfm clique problem is obviously in P. In Section 3, we prove a general theorem asserting that the lfm connected subgraph problem for a hereditary property is NP-hard if the property is satisfied by graphs with arbitrarily large diameters and is determined by blocks. Hence the connectedness makes the problem harder.

In Section 4, we concentrate on a special problem, the lfm rooted tree problem. It is also possible to prove that this problem is Δ_2^p-complete even if the instances are directed acyclic graphs (abbreviated to dags). But if vertices of a dag are topologically sorted, it allows a polynomial time algorithm. Moreover, our analysis shows that the problem restricted to topologically sorted dags with degree 4 is P-complete and the degree bound 4 is proved to be optimal in the sense that the problem for degree 3 is, interestingly, solvable in NC^2. Finally, the complexity analysis of the lfm forest problem is given in comparison with the rooted tree problem.

2. The Lexicographically First Maximal Induced Path Problem is Δ_2^p-Complete

The *lfm induced path problem* is to find the lfm subset of vertices that induces a path, a connected graph of degree at most 2 with no cycle. We prove that this is Δ_2^p-complete.

For any graph property π, the lexicographically first maximal subgraph satisfying π is computed by the following greedy algorithm:

```
begin
    U ← ∅;
    for j = 1 to n do
      if U ∪ {j} can be extended to a subgraph of G satisfying π then U ← U ∪ {j}
end
```

From the algorithm it is clear that the lfm subgraph problem for π is in Δ_2^p if π is polynomial time testable.

Following [Pa], we give the definition of the *deterministic satisfiability problem* as follows:

Let $x_1, ..., x_{k-1}$ be $k-1$ variables and $Y_1, ..., Y_k$ be k sets of variables. A formula $F(x_1, ..., x_{k-1}, Y_1, ..., Y_k)$ in conjunctive normal form is said to be *deterministic* if F consists of the following clauses:

(1) Either (y) or $(\bar{y})$ is a clause of F for each y in $Y_1 \cup Y_k$.

(2) For each $i = 1, ..., k-1$ and each y in Y_{i+1}, there are sets C_y^i and D_y^i of conjunctions of literals from $Y_i \cup \{x_i\}$ with the following properties:

(i) Exactly one of the conjunctions in $C_y^i \cup D_y^i$ is true for any truth assignment.

(ii) F contains clauses $(\alpha \rightarrow y)$ and $(\beta \rightarrow \bar{y})$ for each $\alpha \in C_y^i$ and each $\beta \in D_y^i$.

DSAT(Deterministic Satisfiability)

Instance: A deterministic formula $F_0(x_1, ..., x_{k-1}, Y_1, ..., Y_k)$ and formulas $F_1(Y_1, Z_1)$, ... , $F_{k-1}(Y_{k-1}, Z_{k-1})$ in 3-conjunctive normal form, where $\{x_1, ..., x_{k-1}\}$, $Y_1, ..., Y_k$, $Z_1, ..., Z_{k-1}$ are mutually disjoint sets of variables.

Question: Is there a truth assignment $\hat{x}_1, ..., \hat{x}_{k-1}, \hat{Y}_1, ..., \hat{Y}_k$ satisfying (i) and (ii).

(i) $F_0(\hat{x}_1, ..., \hat{x}_{k-1}, \hat{Y}_1, ..., \hat{Y}_k) = t$.

(ii) $F_i(\hat{Y}_i, Z_i)$ is satisfiable $\iff \hat{x}_i = t$ for $i = 1, ..., k-1$.

Lemma 1 [Pa]. *DSAT is Δ_2^p-complete.*

Lemma 2. *For a formula F in conjunctive normal form, we can construct a graph G_F with specified vertices a, w_0, w_1 of degree 1 such that F is satisfieable (resp. not satisfiable) if and only if the lfm induced path of G_F is a path from a to w_1 (resp. w_0).*

Proof. For simplicity we assume that F is in 3-conjunctive normal form. Let $c_1, ..., c_m$ be the clauses of F and let $x_1, ..., x_n$ be the variables occurring in $c_1, ..., c_m$. For each variable x_i, we use the graph in Fig. 1(a) called the *variable graph*, where $k_i = \max\{|\{c_j : c_j \text{ contains } x_i\}|, |\{c_j : c_j \text{ contains } \bar{x}_i\}|\}$. We call the subgraph induced by $d_i, x_i, \bar{x}_i$ the *value assignment part* for x_i. For each clause $c_j = \alpha_j + \beta_j + \gamma_j$, we use the graph in Fig. 1(b) called the *clause graph* and c_j a *clause vertex*. We call vertices $\alpha_j[c_j]_p$, $\beta_j[c_j]_p$, $\gamma_j[c_j]_p$ $(p = 0, 1)$ *literal vertices*. An example of construction for a formula $F = (x_1 + x_2)(x_1 + \bar{x}_2)(\bar{x}_1 + x_2)(\bar{x}_1 + \bar{x}_2)$ in 2-conjunctive normal form is given in Fig. 2 together with the numbering of vertices, where some edges are not drawn since they make the figure ugly. As shown in Fig. 2, the variable graphs are concatenated in the order of $x_1, ..., x_n$ and the clause graphs are connected using square vertices $z_1, ..., z_m$. It also has special vertices a, b, h_0, h_1, h_2, w_0 and w_1 which are wired as shown. We put edges $\{h_1, z_j\}$ for $j = 1, ..., m$. Due to these edges, $z_1, ..., z_m$ are forbidden to be chosen when h_1 has been chosen before. We also add edges $\{h_0, u\}$ for all literal vertices u. These edges separate the clause vertices from the variable graphs when h_0 is chosen. We connect the vertex x_i (resp. $\bar{x}_i$) to vertices $\bar{x}_i[c_j]_0$, $\bar{x}_i[c_j]_1$ (resp. $x_i[c_j]_0$, $x_i[c_j]_1$) if clause c_j contains the literal $\bar{x}_i$ (resp. $\bar{x}_i$). The vertex x_i (resp. $\bar{x}_i$) is also connected to