A generic greedy algorithm, partially-ordered graphs and NP-completeness*

Antonio Puricella and Iain A. Stewart,
Department of Mathematics and Computer Science,
University of Leicester, Leicester LE1 7RH, U.K.

Abstract

Let π be any fixed polynomial-time testable, non-trivial, hereditary property of graphs. Suppose that the vertices of a graph G are not necessarily linearly ordered but partially ordered, where we think of this partial order as a collection of (possibly exponentially many) linear orders in the natural way. We prove that the problem of deciding whether a lexicographically first maximal subgraph of G satisfying π, with respect to one of these linear orders, contains a specified vertex is **NP**-complete.

1 Introduction

Miyano [6] proved that the problem of computing the lexicographically first maximal subgraph of a given graph, where this subgraph should satisfy some fixed polynomial-time testable, non-trivial, hereditary property π, is **P**-hard (even when the given graph is restricted to be either bipartite or planar and π is non-trivial on the class of bipartite or planar graphs, respectively). Because of the stipulations on π, the lexicographically first maximal subgraph satisfying the property π can be computed by a generic greedy algorithm. Note that Miyano's result is widely applicable; to *any* polynomial-time testable, non-trivial, hereditary property π, such as whether a graph is planar, bipartite, acyclic, of bounded degree, an interval graph, chordal, and so on. Miyano states that his work was inspired by that of Asano and Hirata [1], Lewis and Yannakakis [5], Watanabe, Ae and Nakamura [7] and Yannakakis [8] on node- and edge-deletion problems in **NP**. Typical is this work is the result of Lewis and Yannakakis [5] that the problem of finding the minimum number of nodes needing to be deleted from a graph so that the graph satisifes a fixed polynomial-time testable, non-trivial, hereditary property π is **NP**-hard.

*An extended abstract of this paper is to appear in *Proceedings of 27th International Workshop on Graph-Theoretic Concepts in Computer Science* (WG'01), Lecture Notes in Computer Science.

Of course, a tacit assumption in Miyano's work is that the vertices of any graph are linearly ordered. In this paper, inspired by Miyano's results, we return to the setting of **NP** in that we consider computing lexicographically first maximal subgraphs of given graphs, where these subgraphs should satisfy some given polynomial-time testable, non-trivial, hereditary property π, except that now the graphs come equipped with not just one linear ordering of their vertices but several. Hence, for a given graph we will be involved with a collection of lexicographically first maximal subgraphs and not just one. Note that if we gave our linear orderings explicitly then a graph on n vertices could only come with a polynomial (in n) number of such linear orderings (as otherwise it would be unreasonable to define that the whole input has size n) and we would still be working within **P**. In order to work with an exponential number of linear orderings, we present our collection of linear orderings in the form of a partial order, *i.e.*, an acyclic digraph, with a source vertex providing the (common) least element of any of the linear orderings. Similarly to as in Miyano's deterministic scenario, a non-deterministic polynomial-time greedy algorithm computes all lexicographically first maximal subgraphs.

Our main result is that the problem of deciding whether a lexicographically first maximal subgraph of a given partially ordered graph, where this subgraph should satisfy some fixed polynomial-time testable, non-trivial, hereditary property π, contains some specified vertex is **NP**-complete (even when the given graph is restricted to be planar bipartite and π is non-trivial on this class of graphs). We use similar techniques to Miyano although our proofs are comparatively simpler and the combinatorics is very different.

It is not at all obvious as to how we might use Miyano's result for **P**-completeness directly to prove an analogous result for **NP**-completeness but with partial orderings replacing linear orderings (indeed, we have failed with this approach and have had to revert back to 'first principles'). Furthermore, the **NP**-completeness results from [1, 5, 7, 8] can not be formulated in our framework. For example, the result of Lewis and Yannakakis [5], mentioned above, is concerned with sizes of maximal subgraphs satisfying a specific property and as such is unrelated to our problems.

2 Basic definitions

For standard graph-theoretic definitions, the reader is referred to [2]. A property π on graphs is *hereditary* if whenever we have a graph with the property π, the deletion of any vertex and its incident edges does not produce a graph violating π, *i.e.*, π is preserved by induced subgraphs. A property π is called *non-trivial* on a class of graphs if there are infinitely many graphs from this class satisfying π but π is not satisfied by all graphs of the class.

Let π be some property of graphs. Let G be a graph, let H be a partial order of the vertices of G, and let s and t be vertices of G. We assume that the partial order H is given in the form of an acyclic digraph detailing the immediate predecessors, *i.e.*, the *parents*, and the immediate successors, *i.e.*, the *children*, of each vertex. We think of a partial order H as encoding a collection of linear orders of the form $s = s_0, s_1, s_2, \ldots, s_k$, where s_{j+1} is a child of s_j and s_k has no children. Note that a partial order can encode an exponential number of linear orders.

The algorithm GREEDY(π) is as follows:

```
input(G,H,s)
  S := ∅
  current-vertex := s
  if π(S∪{current-vertex},G) then                    (*)
    S := S∪{current-vertex}
    fi
  while current-vertex has at least one child in H do
    current-vertex := a child of current-vertex in H
    if π(S∪{current-vertex},G) then                  (**)
      S := S∪{current-vertex}
      fi
    od
output(S)
```

where $\pi(S \cup \{\textit{current-vertex}\}, G)$ is a predicate evaluating to 'true' if, and only if, the subgraph of G induced by the vertices of $S \cup \{\textit{current-vertex}\}$ satisfies π. We say that a vertex v is the *current-vertex* if we have 'frozen' an execution of the algorithm GREEDY(π) immediately prior to executing either line $(*)$ or line $(**)$ and the value of the variable *current-vertex* at this point is v. Note that in general the algorithm GREEDY(π) is non-deterministic and produces a collection of sets of vertices as outputs. If the property π is hereditary and can be checked in polynomial-time then the algorithm GREEDY(π) non-deterministically computes, in polynomial-time, the lexicographically first maximal subgraphs of the graph G satisfying π with respect to the linear orders encoded within the partial order H.

Let $\mathcal{C}$ be a class of graphs and let π be some property of graphs. The problem GREEDY(partial order, $\mathcal{C}$, π) has: as its instances tuples (G, H, s, t), where G is a graph from $\mathcal{C}$, H is a partial order of the vertices of G and s and t are vertices of G; and as its yes-instances those instances for which there exists an execution of the algorithm GREEDY(π) on input (G, H, s) resulting in the vertex t being output. The problem GREEDY(linear order, $\mathcal{C}$, π) is defined similarly. Miyano's result from [6] can be stated as follows.

Theorem 1 *Let π be a polynomial-time testable, non-trivial, hereditary property on the class of graphs $\mathcal{C}$, where $\mathcal{C}$ is the class of all graphs, the class of planar graphs or the class of bipartite graphs. Then the problem GREEDY(linear order, $\mathcal{C}$, π) is* **P***-complete.* □

3 Our results

In order to prove our main result, we need to first establish a completeness result for the specific problem GREEDY(partial order, planar bipartite, independent set) (we only sketch the proof due to space limitations).

Theorem 2 *The problem GREEDY(partial order, planar bipartite, independent set) is* **NP***-complete.*

Proof We reduce from the known **NP**-complete problem Directed Hamiltonian Path (DHP): whose instances are triples (G, s, t), where G is a digraph and s and t are vertices of G; and whose yes-instances are instances for which there is a Hamiltonian path in G from s to t (see [3]).

Let $(G = (V, E), s, t)$ be an instance of DHP of size n. W.l.o.g. we assume that $|V| > 2$, that the vertex set of G is $\{1, 2, \ldots, n\}$ and that $s = 1$ and $t = n$. Corresponding to this instance, we build an instance (G', H', s', t') of GREEDY(partial order, planar bipartite, independent set). The vertex set V' of G' and H' is

$$\{u_{i,j}, v_{i,j}, w_{i,j}, z_j : i, j = 2, 3, \ldots, n-1\} \cup \{x, s', t'\}.$$

The edges of G' are

$$\begin{aligned}&\{(u_{i,j}, v_{i,j}), (u_{i,j}, w_{i,j}) : i, j = 2, 3, \ldots, n-1\}\\ &\quad \cup\{(w_{i,j}, u_{i+1,j}) : i = 2, 3, \ldots, n-2; j = 2, 3, \ldots, n-1\}\\ &\quad \cup\{(w_{n-1,j}, z_j) : j = 2, 3, \ldots, n-1\}\\ &\quad \cup\{(z_j, w_{n-1,j+1}) : j = 2, 3, \ldots, n-2\}\\ &\quad \cup\{(z_{n-1}, t')\}\end{aligned}$$

and the edges of H' are

$$\begin{aligned}&\{(v_{i,j}, v_{i+1,j'}) : i = 2, 3, \ldots, n-2; j, j' = 2, 3, \ldots, n-1; (j, j') \in E\}\\ &\quad \cup\{(s', v_{2,j}) : j = 2, 3, \ldots, n-1; (1, j) \in E\}\\ &\quad \cup\{(v_{n-1,j}, x) : j = 2, 3, \ldots, n-1; (j, n) \in E\}\\ &\quad \cup\{(x, u_{2,2})\}\\ &\quad \cup\{(u_{i,j}, w_{i,j}) : i = 2, 3, \ldots, n-2; j = 2, 3, \ldots, n-1\}\\ &\quad \cup\{(w_{i,j}, u_{i+1,j}) : i = 2, 3, \ldots, n-2; j = 2, 3, \ldots, n-1\}\\ &\quad \cup\{(u_{n-1,j}, u_{2,j+1} : j = 2, 3, \ldots, n-2\}\\ &\quad \cup\{(u_{n-1,n-1}, w_{n-1,2})\}\\ &\quad \cup\{(w_{n-1,j}, z_j) : j = 2, 3, \ldots, n-1\}\\ &\quad \cup\{(z_j, w_{n-1,j+1}) : j = 2, 3, \ldots, n-2\}\\ &\quad \cup\{(z_{n-1}, t')\}\end{aligned}$$

The construction of the instance (G', H', s', t') is illustrated in Figs. 1, 2 and 3 which depict: a digraph G; the resulting graph G'; and the resulting partial order H', respectively.

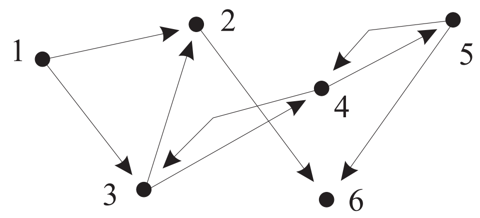

Figure 1. A digraph G.

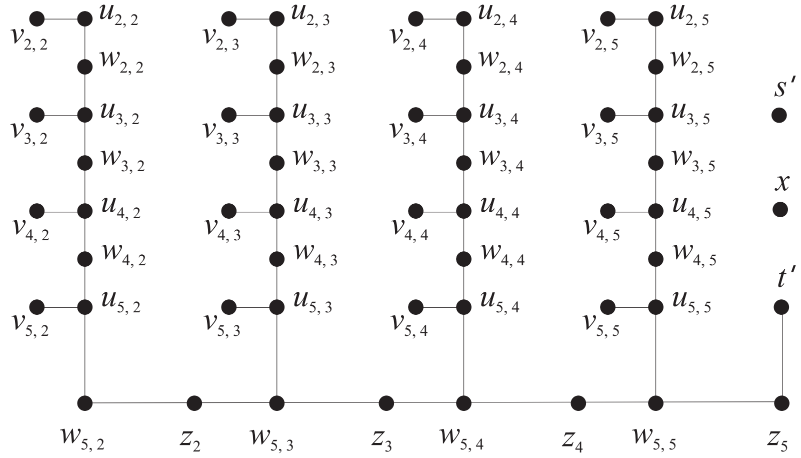

Figure 2. The graph G' corresponding to G.

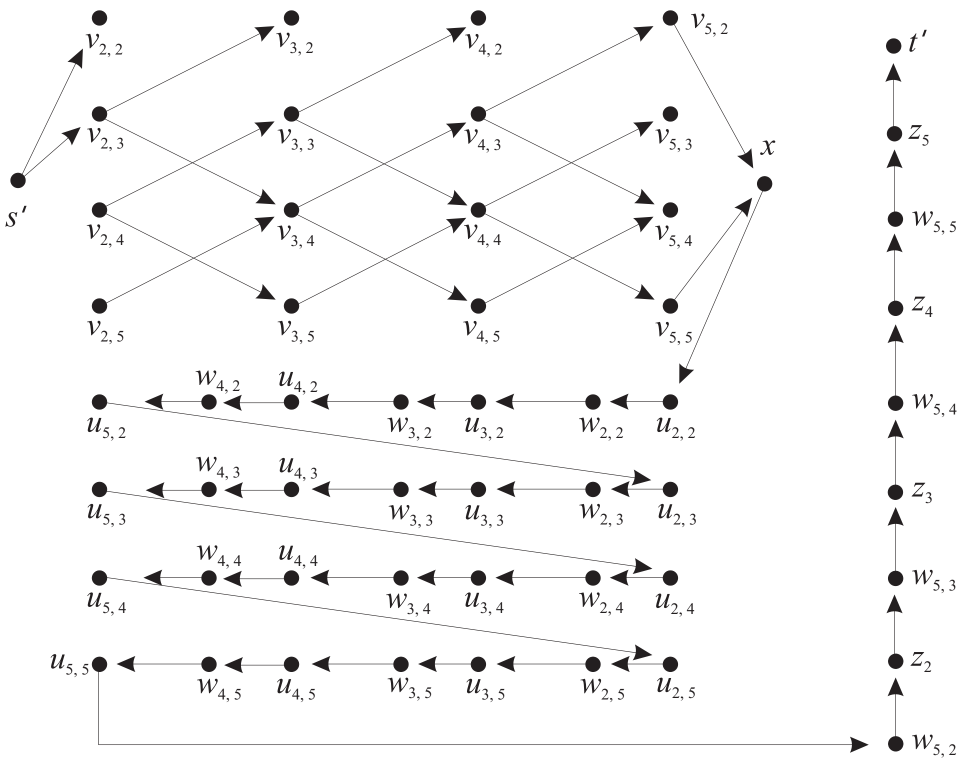

Figure 3. The partial order H' corresponding to G.

Suppose that (G, s, t) is a yes-instance of DHP. Then there is a Hamiltonian path $s = s_1, s_2, s_3, \ldots, s_{n-1}, s_n = t$ in G. Consider the following path in H':

$$s', v_{2,s_2}, v_{3,s_3}, \ldots, v_{n-1,s_{n-1}}, x$$

(note that this is indeed a path in H'). In the execution of the algorithm GREEDY (independent set) on (G', H', s', t'), following this path in H' clearly results in the vertices of $\{s', v_{2,s_2}, v_{3,s_3}, \ldots, v_{n-1,s_{n-1}}, x\}$ all being output.

Henceforth, the path chosen in H' is fixed. With reference to Fig. 2, following this path we work down the first column of u- and w-vertices of G' (that is, the column with index 2, *i.e.*, involving vertices of the form $u_{-,2}$ and $w_{-,2}$), then the second column (the column with index 3), until having worked down the last column (the column with index $n-1$), we work along the bottom row of w- and z-vertices. For every $j = 2, 3, \ldots, n-1$, a vertex $v_{i,j}$, for some i, has been output by the algorithm GREEDY(independent set); that is, there is exactly one v-vertex output from every column. Hence, as we work down the columns of u- and w-vertices, the vertex $w_{n-2,j}$ is output by the algorithm GREEDY(independent set) but the vertex $u_{n-1,j}$ is not, for all $j = 2, 3, \ldots, n-1$. Consequently, when we work along the bottom row of w- and z-vertices of G', the vertex $w_{n-1,j}$ is output but the vertex z_j is not, for all $j = 2, 3, \ldots, n-1$. Finally, the vertex t' is output. Hence, (G', H', s', t') is a yes-instance of $\mathcal{H}$.

Conversely, suppose that (G', H', s', t') is a yes-instance of $\mathcal{H}$ and consider an execution of the algorithm GREEDY(independent set) witnessing this fact. The path chosen in H' from s' to x yields a path of length $n-1$ in G from 1 to n. Suppose that this path in G is such that a vertex j appears on it more than once. This means that vertices $v_{i,j}$ and $v_{i',j}$ appear on the path in H' from s' to x, where $i \neq i'$. Hence, with reference to Fig. 2, there must be some column in G' for which a v-vertex has not been output by the algorithm GREEDY(independent set). Let the largest index of any such column be k. When we work down the u- and w-vertices of column k in G' in our execution of the algorithm GREEDY(independent set), the result is that all of the u-vertices are output and none of the w-vertices are. When we work down the u- and w-vertices of column m, for any $m > k$, in our execution of the algorithm GREEDY(independent set), the result is that the vertex $u_{n-1,m}$ is not output. Hence, when we work along the bottom row of w- and z-vertices in our execution of the algorithm GREEDY(independent set), the vertices $z_k, z_{k+1}, \ldots, z_{n-1}$ are all output but not the vertex t'. This yields a contradiction; and so we have a Hamiltonian path in G from 1 to n. Hence, (G, s, t) is a yes-instance of DHP.

As the construction (G', H', s', t') from (G, s, t) can clearly be completed using log-space, the result follows. $\square$

Now we consider the problem GREEDY(partial order, planar bipartite, π) where π is *any* polynomial-time testable, non-trivial, hereditary property. We begin with some specific graph-theoretic definitions.

A *cut-point* of a connected graph G is a vertex c such that its removal (along with its incident edges) from G results in a graph with at least 2 connected components. A *component relative to a cut-point* c is a subgraph consisting of c, one of the derived connected components and all those edges of G joining c and a vertex of the component. If a connected graph does not have any cut-points then it is *biconnected*.

Let $\mathbf{a} = (a_1, a_2, \ldots, a_s)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_t)$ be two tuples of positive integers. We order these tuples lexicographically as follows. We say that $\mathbf{a} >_L \mathbf{b}$ if either:

- there exists some $i \in \{1, 2, \ldots, \min\{s, t\}\}$ such that $a_j = b_j$, for all $j \in \{1, 2, \ldots, i-1\}$, and $a_i > b_i$; or
- $s > t$ and $a_j = b_j$, for all $j \in \{1, 2, \ldots, t\}$.

The *α-sequence* α_G of a connected graph G is defined as follows. Suppose that G is not biconnected. If c is a cut-point of G whose removal results in a graph with k connected components then define $\alpha_{c,G} = (n_1, n_2, \ldots, n_k)$, where $n_1 \geq n_2 \geq \ldots \geq n_k$ are the numbers of vertices in the components relative to c. We define α_G to be the lexicographically-minimal tuple of the (non-empty) set $\{\alpha_{c,G} : c \text{ is a cut-point of } G\}$, and we define c_G to be any cut-point for which $\alpha_G = \alpha_{c_G,G}$. If G is biconnected then we define $\alpha_G = (|G|)$ and c_G as any vertex.

Given a graph G with connected components $G_1, G_2, \ldots, G_k$, the *β-sequence* β_G of G is defined as $(\alpha_{G_1}, \alpha_{G_2}, \ldots, \alpha_{G_k})$, where $\alpha_{G_1} \geq_L \alpha_{G_2} \geq_L \ldots \geq_L \alpha_{G_t}$. A β-sequence is therefore a tuple of tuples of integers.

Theorem 3 *Let π be a property satisfying the following conditions:*

(i) π is non-trivial on planar bipartite graphs;

(ii) π is hereditary on induced subgraphs;

(iii) π is satisfied by all sets of independent edges; and

(iv) π is polynomial-time testable.

The problem GREEDY(partial order, planar bipartite, π) is **NP***-complete.*

Proof For brevity, we refer to the problem GREEDY(partial order, planar bipartite, π) as $\mathcal{G}$. The property π is, by assumption, non-trivial on planar bipartite graphs. It follows that amongst all planar bipartite graphs violating π, there must be (at least) one with smallest β-sequence, where β-sequences are ordered lexicographically but where the comparison of components, i.e., α-sequences, is according to $\geq_L$. Let us call such a graph J; that is,

$$\beta_J = \min\{\beta_G : G \text{ is a planar bipartite graph violating } \pi\}.$$

Let $J_1, J_2, \ldots, J_k$ be the connected components of J ordered according to $\alpha_{J_1} \geq_L \alpha_{J_2} \geq_L \ldots \geq_L \alpha_{J_k}$. It follows that J has β-sequence $\beta_J = (\alpha_{J_1}, \alpha_{J_2}, \ldots, \alpha_{J_k})$.

Let $c = c_{J_1}$ and let the connected components of J_1 relative to c be $I_0 \cup \{c\}, I_1 \cup \{c\}, \ldots, I_m \cup \{c\}$, where $|I_0| \geq |I_1| \geq \ldots \geq |I_m|$. Denote by I_* the subgraph of J_1 induced by the vertices of $\{c\} \cup I_1 \cup \ldots \cup I_m$. By (ii) and (iii) it follows that π is satisfied by any independent set of vertices, and so I_0 must contain at least one edge (otherwise J would be a set of independent vertices).

To prove the **NP**-completeness of the problem $\mathcal{G}$, we reduce from the problem GREEDY(partial order, planar bipartite, independent set), which, for brevity, we denote by $\mathcal{H}$, and which was proven to be **NP**-complete in Theorem 2. That is, from an instance (G, H, s, t) of $\mathcal{H}$, we create an instance (G', H', s', t') of $\mathcal{G}$ (with the appropriate properties).

We will divide the construction of G' from G into three phases. For any subset of vertices U of J, we denote by $\langle U \rangle$ the subgraph of J induced by the vertices of U. Note that as $\langle I_0 \cup \{c\} \rangle$ contains at least one edge and is connected, there exists a vertex d of $I_0 \cup \{c\}$ such that (c, d) is an edge of $\langle I_0 \cup \{c\} \rangle$.

<u>Phase 1</u> For each vertex u of G, we attach a copy of $\langle I_* \cup \{c\} \rangle$ by identifying u with c (all such copies are disjoint). Call the resulting graph $\tilde{G}$. Note that the vertex set of $\tilde{G}$ consists of the vertices of G, which we call the *G-vertices*, together with disjoint copies of the vertices of I_*. As both $\langle I_* \rangle$ and G are planar and bipartite, $\tilde{G}$ maintains these properties.

<u>Phase 2</u> We replace each edge (u, v) of $\tilde{G}$, where u and v are G-vertices, by a copy of $\langle I_0 \cup \{c\} \rangle$ by identifying u with c and v with d (all such copies are disjoint). Note that our choice of d results in the graph so formed being planar and bipartite.

<u>Phase 3</u> We add disjoint copies of $J_2, J_3, \ldots, J_k$ to obtain G', which is clearly planar and bipartite.

The partial order H' consists of a linear order onto which is concatenated the partial order H (of the G-vertices). The linear order consists of: all vertices of G' that are vertices of some copy of $\langle I_0 \rangle$; followed by all vertices in the copies of $\langle I_* \rangle$; followed by all vertices of $J_2, J_3, \ldots, J_k$. It does not matter how we order the vertices of some copy of $\langle I_0 \rangle$, for example, in the linear order. We concatenate this linear order prior to H by including an edge from the last vertex of the linear order to the vertex s of H. Denote the vertex s' to be the first vertex of the above linear order, and denote the vertex t' to be the G-vertex of G' formerly known as t. Our construction can be visualised in Fig. 4.

We will now state three lemmas to be used in the remainder of the proof (the proofs are straightforward and are omitted due to space limitations).

Lemma 4 *Any graph K consisting of any number of disjoint copies of $\langle I_0 \setminus \{d\} \rangle$ plus any number of disjoint copies of $\langle I_* \rangle$ plus a disjoint copy of each of $J_2, J_3, \ldots, J_k$ satisfies π.*

Proof The connected components of K consist of $J_2, J_3, \ldots, J_t$ together with the connected components of the copies of $\langle I_0 \setminus \{d\} \rangle$ and $\langle I_* \rangle$. Consider the α-sequence α of a connected component of either $\langle I_0 \setminus \{d\} \rangle$ or $\langle I_* \rangle$. All components of α are strictly less than $|I_0|+1$; and so α is strictly less than α_{J_1}. Hence, β_K has one less component equal to α_{J_1} than β_J, with all other components strictly less than α_{J_1}; and so K satisfies π by minimality of β_J. $\square$

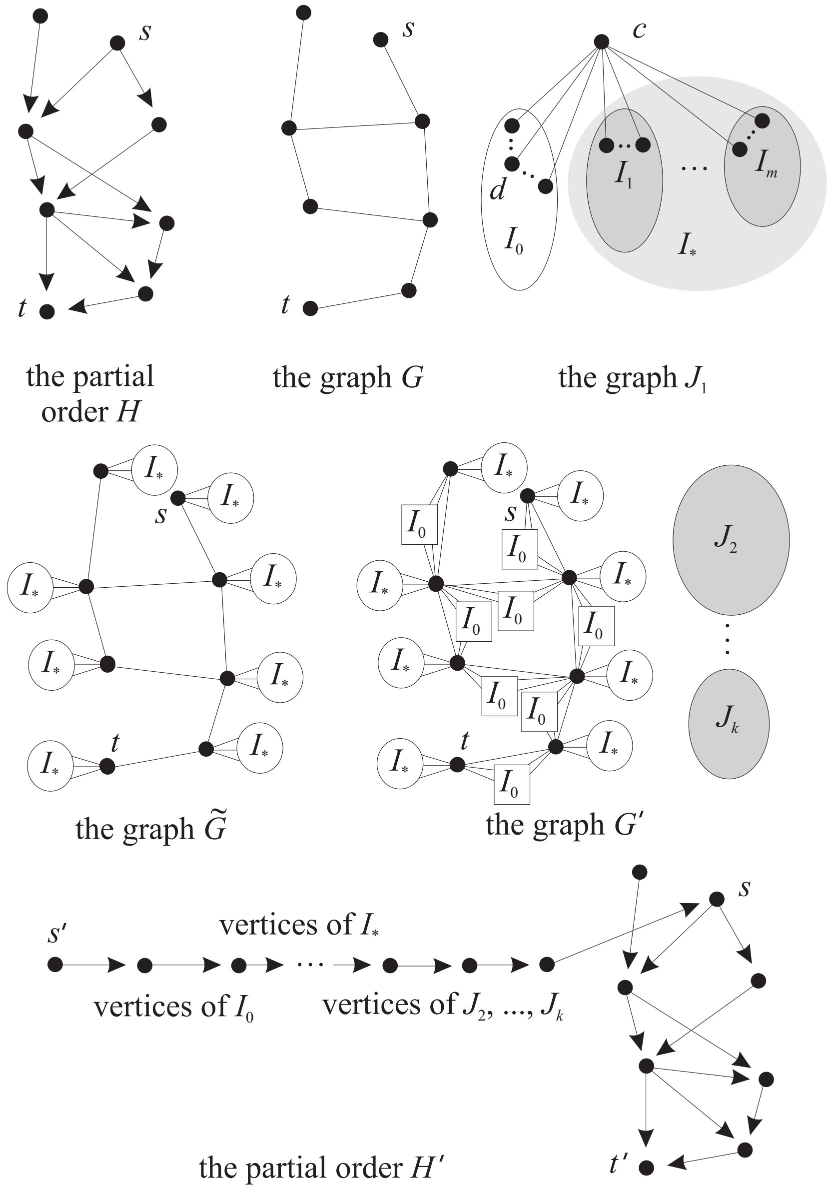

Figure 4. Our basic construction.

Lemma 5 *Take a single copy of $\langle I_* \cup \{c\}\rangle$ and any number of disjoint copies of $\langle (I_0 \setminus \{d\}) \cup \{c\}\rangle$, and identify the vertices named c in all of these graphs. Then the resulting graph M satisfies π.*

Proof Let M' be the connected component of M containing c. We begin by remarking that any other connected component of M has an α-sequence strictly less than the α-sequence $(|I_0|+1)$; and so strictly less than α_{J_1}.

Suppose that c is a cut-point of M'. Then $\alpha_{c,M'}$ has components $|I_1|+1, |I_2|+1, \ldots, |I_m|+1$ as well as possibly some other components which are all strictly less than $|I_0|+1$. Hence, by arguing similarly to as in the proof of Lemma 4, $\alpha_{M'}$ is strictly less than α_{J_1}. By the remark above, β_M is strictly less than β_J and so M satisfies π by the minimality of β_J.

Suppose that c is not a cut-point of M'. Then $I_* = I_1$, *i.e.*, $m=1$, and $M' = \langle I_* \rangle$; hence, $\alpha_{M'}$ is at most $(|I_1|+1)$. Any connected component of M different from M' has size at most $|I_0|-2$, and so $\alpha_{J_1} = (|I_0|+1, |I_1|+1)$ is strictly greater than the α-sequence of any connected component of M. Consequently, β_M is strictly less than β_J; and M satisfies π by the minimality of β_J. □

Lemma 6 *Any graph N consisting of disjoint copies of $J_2, J_3, \ldots, J_k$ plus any number of disjoint copies of the graph M from Lemma* 5 *satisfies π.*

Proof By the proof of Lemma 5, the graph M is such that the maximal component of β_M is strictly less than α_{J_1}. By reasoning as we did in the proof of Lemma 4, it follows that β_N is strictly less than β_J and so N satisfies π by the minimality of β_J. □

Throughout, we refer to a G-vertex in G' and the corresponding vertex in G by the same name (and also to a vertex of H and the corresponding vertex in the portion of the partial order H' corresponding to H by the same name).

Consider the algorithm GREEDY(π) on input (G', H', s', t'). The partial order H' consists of a linear order, whose vertices are S_0, say, concatenated with the partial order H. The subgraph of G' induced by the vertices of S_0 is as is the graph K of Lemma 4 and consequently every vertex of S_0 is always placed in every output from GREEDY(π). Note that the algorithm GREEDY(π) on input (G', H', s', t') with current-vertex s is working with exactly the same partial order, namely H, as is the algorithm GREEDY(independent set) on input (G, H, s, t) with current-vertex s.

Suppose, as our induction hypothesis, that:

- the algorithm GREEDY(independent set) on input (G, H, s, t) has current-vertex u, for some ancestor u of s in H, and has so far output the set of vertices S;
- the algorithm GREEDY(π) on input (G', H', s', t') has current vertex u in H' and has so far output the set of vertices $S_0 \cup S$; and
- the subgraph of G' induced by the vertices of $S_0 \cup S'$ is in the form of a subgraph of the graph N in Lemma 6.

Note that the induction hypothesis clearly holds, in the base case, when the vertex u is actually s.

Suppose that the algorithm GREEDY(π) outputs the vertex u. If u is such that adding u to $S_0 \cup S'$ completes a copy of I_0 then we would have a copy of J within the subgraph of G' induced by the vertices of $S_0 \cup S \cup \{u\}$. This would yield a contradiction because this subgraph satisfies π (by definition), π is hereditary on induced subgraphs, and J would then have to satisfy π. Hence, the vertex u is not joined to any vertex of S in G and so u is output by the algorithm GREEDY(independent set).

Conversely, if the algorithm GREEDY(independent set) outputs u then this is because $S \cup \{u\}$ is an independent set in G; and consequently $S_0 \cup S \cup \{u\}$ induces in G' a subgraph of the form of a subgraph of the graph N in Lemma 6. Hence, by Lemma 6, u is output by the algorithm GREEDY(π).

By induction, we obtain that if S is a set of vertices output by the algorithm GREEDY(independent set) on input (G, H, s, t) then $S_0 \cup S$ is output by the algorithm GREEDY(π) on input (G', H', s', t'), and conversely. Hence, we have a log-space reduction from $\mathcal{H}$ to $\mathcal{G}$. □

Just as Miyano did in [6], we can actually remove the necessity in Theorem 3 for π to be satisfied by all sets of independent edges. Ramsey theory can be applied to show that any graph property that is non-trivial and hereditary on a class of graphs is either satisfied by all independent sets or by all cliques. We can then use this fact to eliminate the need for π to be satisfied by all independent edges (we omit the details here but point out that we proceed exactly as Miyano did). Hence, we obtain the following.

Corollary 7 *Let π be a polynomial-time testable, hereditary graph property that is non-trivial on planar bipartite graphs. The problem GREEDY(partial order, planar bipartite, π) is complete for* **NP**. □

We conclude by remarking that in future we will extend Corollary 7 so that it applies to directed graphs and we will examine degree bounds on graphs so as to delineate when a problem GREEDY(partial order, graphs, π) becomes solvable in polynomial-time (for specific properties π).

References

[1] T. Asano and T. Hirata, Edge-deletion and edge-contraction problems, *Proceedings of 14th Annual ACM Symposium on the Theory of Computing* (1982) 245–254.

[2] C. Berge, *Graphs and Hypergraphs*, North-Holland (1973).

[3] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman (1979).

[4] A.M. Gibbons and W. Rytter, *Efficient Parallel Algorithms*, Cambridge University Press (1988).

[5] J.M. Lewis and M. Yannakakis, The node deletion problem for hereditary properties is NP-complete, *Journal of Computer and System Sciences* **20** (1980) 219–230.

[6] S. Miyano, The lexicographically first maximal subgraph problems: P-completeness and NC algorithms, *Mathematical Systems Theory* **22** (1989) 47–73.

[7] T. Watanabe, T. Ae and A. Nakamura, On the removal of forbidden graphs by edge-deletion or by edge-contraction, *Discrete Applied Mathematics* **3** (1981) 151–153.

[8] M. Yannakakis, Node-deletion problems on bipartite graphs, *SIAM Journal of Computing* **10** (1981) 310–327.