数理解析研究所講究録
943 巻 1996 年 73-78

A supergraph technique for search problems

Koichi Yamazaki (山崎 浩一)
University of Electro-Communications
1-5-1 Chofugaoka, Chofu
Tokyo 182, Japan

Abstract

An algorithm D is a decision algorithm if D decide whether a problem has an explanation or not (D outputs "yes" or "no"). An algorithm S is a search algorithm if S find an explanation for a problem (when the problem has an explanation). An explanation is called a natural evidence.

We consider the following three graph problems : k-COLORABILITY, k-BANDWIDTH, and k-TREE. In this paper, we discuss the efficiency of finding a natural evidence for the graph problems when its search algorithm can employ its decision algorithm as oracle. We will show that for each the three problem if the decision problem is solvable in time $D(n)$ then the corresponding search problem is solvable in $O(n^2 \times D(n))$ time.

1 Introduction

In this paper, we discuss the efficiency of finding a natural evidence for the graph problems when its search algorithm can employ its decision algorithm as oracle. Such subject is investigated as gap between decision and search [1, 2, 3, 4, 5, 6]. Problems in NP have a natural evidence. For example, for k-COLORABILITY its natural evidence is a function $f : V(X) \rightarrow \{1, \cdots, k\}$ such that $f(u) \neq f(v)$ whenever $\{u, v\} \in E$. For k-COLORABILITY its decision problem is deciding whether input graph G satisfies k-COLORABILITY or not. In the decision problem, it is only shown to exist a its natural evidence. Its search problem is finding its a natural evidence. We will call an algorithm which solves a decision problem by *decision algorithm*. Similarly, the term *search algorithm* can be defined. The gap is considered as efficiency of a search algorithm with help of its decision algorithm as oracle. In other words if there exists a search algorithm which employs a corresponding decision algorithm as an oracle, and which computes a natural evidence in $O(f(n))$ time, then we say that the gap is at most $O(f(n))$.

In this paper, we will show that for some graph problems, critical graphs are useful to obtain a natural evidence. Critical graphs for some properties have information about a natural evidence. Furthermore those natural evidences are easy to get from the information of critical graphs. Hence if we obtain a critical graph for a property Π then we can efficiently find a natural evidence concerned with Π using the information. We will demonstrate only two problems k-COLORABILITY and k-BANDWIDTH, but the technique discussed in this paper is effective for k-TREE. For search problem of k-TREE, it is known there exists a linear time algorithm when k is fixed, hence for k-TREE there is no gap between decision and search.

2 Preliminaries

We consider *finite undirected and connected graphs* without *loops* and without *multiple* edges. Let $G = (V, E)$ and $G' = (V, E')$ be graphs. G' is a *supergraph of* G if $E \subseteq E'$.

Definition 2.1 Let $G = (V, E)$ be a graph.

(1) G satisfies k-COLORABILITY if there is a function $f : V \to \{1, \cdots, k\}$ such that $f(u) \neq f(v)$ whenever $\{u, v\} \in E$,

(2) G satisfies k-BANDWIDTH if there is a bijection $f : V \to \{1, \cdots, |V|\}$ such that for all $\{u, v\} \in E$, $|f(u) - f(v)| \leq k$, and

(3) G satisfies k-TREE if there is a pair $D = (S, T)$ with $S = \{X_i | i \in I\}$ a collection of subsets of V and $T = (I, F)$ a tree, with one node for each subset of S, such that the following three conditions are satisfied:

 (1) $\bigcup_{i \in I} X_i = V$, for each $1 \leq i \leq |I|$ $|X_i| - 1 \leq k$,

 (2) for all edges $\{u, w\} \in E$, there is a subset $X_i \in S$ such that both v and w are contained in X_i,

 (3) for each vertex x the set of nodes $\{i | x \in X_i\}$ forms a subtree of T.

Let $\Pi_c(k)$ be the property k-COLORABILITY, $\Pi_b(k)$ be k-BANDWIDTH, and $\Pi_t(k)$ be k-TREE. Each critical graph for $\Pi_c(k)$ is a complete k-partite graph. The critical graph with n nodes for $\Pi_b(k)$ is the kth power of a path on n nodes, denoted by P_n^k (see [8, 7]). In the lemma 2.1, it is shown that every critical graph with n nodes for $\Pi_b(k)$ is isomorphic to P_n^k, that is, the critical graph is uniquely decidable if its isomorphisms are identified. For any positive integer k and n, P_n^k is a diameter critical graph and a interval graph. Each critical graph for $\Pi_t(k)$ is a k-tree.

Lemma 2.1 *Let $G = (V, E)$ be a critical graph for $\Pi_b(k)$ and $|V| = n$. Then G is isomorphic to P_n^k.*

Proof. Since G satisfies $\Pi_b(k)$, there is a bijection $f : V \to \{1, 2, \cdots, n\}$ such that for all $\{u, v\}$, $|f(u) - f(v)| \leq k$. For convenience, v_i expresses the node $u \in V$ such that $f(u) = i$. Since G is a critical graph for $\Pi_b(k)$, $\{v_i, v_j\}$ is in E for all i and j such that $|i - j| \leq k$. This means that G is isomorphic to P_n^k. □

Definition 2.2 Let Π be a graph property. A graph $X = (V, E)$ is a *critical graph for* Π if

(1) X satisfies Π, and

(2) X' dose not satisfies Π for any supergraph X' of X.

Definition 2.3 A graph property Π is *hereditary* if it cannot be destroyed by removing edges from the graph; i.e., whenever a graph $G = (V, E)$ satisfies the property Π then also a graph $G' = (V, E')$ satisfies Π, where $E' \subseteq E$.

3 Results

Our search algorithm has two phases:

(1) Constructing a critical graph from the input graph,

(2) Obtaining a natural evidence from the constructed critical graph.

In oder to construct a critical graph, search algorithm adds edges according to answer of decision algorithm. The order of adding edges does not dominat whether the output is a critical graph or not if its property is hereditary.

Let $\overline{E}$ be the set of pairs $e = \{u, v\}$ such that $e \notin E$. For phase (1), the pairs in $\overline{E}$ are numbered in any order. Let $(e_1, e_2, \cdots, e_m)$ be the set $\overline{E}$ which is numbered by an arbitrary order. Then, for phase (1), our search algorithm is as follows:

Procedure PHASE1

input a graph $G = (V, E)$
output a critical graph $G' = (V, E')$ for a property Π

$E' := E$;
for $i := 1$ **to** m **do**
(* Check whether G' satisfies Π or not by the decision algorithm *)
if $(V, E' \cup \{e_i\})$ satisfies the property **then** $E := E' \cup \{e_i\}$;

The following lemma guarantees correctness of **PHASE1**.

Lemma 3.1 *Let $G = (V, E)$ be a input graph such that $|V| = n$. If a graph property Π is hereditary, then* **PHASE1** *outputs a critical graph for* Π.

Proof. Let E' be the set of pairs $e = \{u, v\}$ such that $e \notin E$, and $(e_1, e_2, \cdots, e_m)$ be the set $\overline{E}$ which is numbered by an arbitrary order. Suppose, to the contrary, that the output graph $G' = (V, E \cup Add)$ is not a critical graph for Π, where $Add = \{e_{i_1}, e_{i_2}, \cdots, e_{i_p}\}$ is the set of the added edges in the prosess of **PHASE1** and $i_s < i_t$ iff $s < t$. Then, there is a nonempty subset of $E' - Add$ $Fill = \{e_{i_{j_1}}, e_{i_{j_2}}, \cdots, e_{i_{j_q}}\}$ such that $(V, E \cup Add \cup Fill)$ is a critical graph for Π. Note that $(Add \cap Fill) = \emptyset$ and $(Add \cup Fill) \subseteq \overline{E}$. Let us pay attention to the i_{j_1}-th loop of the **for** loop in **PHASE1**. In the just before the i_{j_1}-th loop, G represents the graph $(V, E \cup \{e_1, e_2, \cdots, e_{i_{j_1}-1}\})$. In the i_{j_1}-th loop, it is checked that whether $(V, E \cup \{e_1, e_2, \cdots, e_{i_{j_1}-1}\} \cup \{e_{i_{j_1}}\})$ satisfies Π or not. Since the answer of the check is "no", $(V, E \cup \{e_1, e_2, \cdots, e_{i_{j_1}-1}\} \cup \{e_{i_{j_1}}\})$ dose not satisfies Π. On the other hand, $(V, E \cup \{e_1, e_2, \cdots, e_{i_{j_1}-1}\} \cup \{e_{i_{j_1}}\})$ is a subgraph of the critical graph $(V, E \cup Add \cup Fill)$ such that $(E \cup \{e_1, e_2, \cdots, e_{i_{j_1}-1}\}) \subseteq (E \cup Add \cup Fill)$. This contradicts that Π is hereditary. □

$\Pi_c(k)$, $\Pi_b(k)$ and $\Pi_u(k)$ are hereditary. Next we will show that natural evidences of G for $\Pi_c(k)$, $\Pi_b(k)$ and $\Pi_u(k)$ are computable from each critical graphs in $O(n^2)$ time in lemma 3.2, 3.3, and 3.4 respectively.

Lemma 3.2 *Let $G = (V, E)$ be a input graph such that $|V| = n$, and $G' = (V, E')$ be a critical graph for Π_c such that $E \subseteq E'$. A natural evidence of G for Π_c is computable from the critical graphs G' in $O(n^2)$ time.*

Proof. To show this lemma, it is sufficient to construct a function $f : V \rightarrow \{1, \cdots, k\}$ such that $f(u) \neq f(v)$ whenever $\{u, v\} \in E$ in $O(n^2)$ time. Let M' be the adjacent matrix of G'. Since every critical graph for $\Pi_c(k)$ is a complete k-partite graph, the complement graph $\overline{G'}$ of G' has k components. And each component in $\overline{G'}$ is a complete graph. In the procedure COLORING, coloring function f is constructed as follows : nodes u and v are in C iff $f(u) = f(v)$, where C is a component of $\overline{G'}$, u and v are nodes in V. It is clear that the function f created by COLORING is a natural evidence for $\Pi_c(k)$ with input G, and the running time of COLORING is $O(n^2)$ time.

Procedure COLORING

input M'
output a natural evidence f

$R := \{1, 2, \cdots, n\}$;
$i := 1$;
$j := 1$;

```
for i := 1 to n do
  if i ∈ R then
    begin
      Define f(h) = j for all h such that M'(i,h) = 0;
      (* Note that M'(i,i) = 0 for 1 ≤ i ≤ n *)
      Remove h from R for all h such that M'(i,h) = 0;
      j := j + 1;
    end.
```

□

Lemma 3.3 *Let $G = (V, E)$ be a input graph such that $|V| = n$, and $G' = (V, E')$ be a critical graph for Π_b such that $E \subseteq E'$. A natural evidence of G for Π_b is computable from the critical graphs G' in $O(n^2)$ time.*

Proof. To show this lemma, it is sufficient to construct a bijection $f : V \rightarrow \{1, \cdots, n\}$ such that for all $\{u, v\} \in E$, $|f(u) - f(v)| \leq k$. Let M' be the adjacent matrix of G'. We will show that f can be obtained from M' in $O(n^2)$ time.

From lemma 2.1, G' is isomorphic to P_n^k. The correctness of procedure BIJECTION is based on the following claims.

claim 1 Let n be a integer such that $k \leq n$. Then P_n^k has exactly two nodes with k degree.

claim 2 Let n be a integer such that $k \leq n-1$, and H be a graph such that H can be cotained by deletion of a node with k degree (and its incident edges) from P_n^k. Then H is isomorphic to P_{n-1}^k.

In the procedure BIJECTION, HEAD and TAIL represent the two nodes with k-degree. In BIJECTION, P_n^k is reduced in P_{n-1}^k and P_{n-2}^k is reduced in P_{n-3}^k and so on. Once TAIL is fixed, TAIL is never changed in BIJECTION. On the other hand HEAD is changed whenever critical graph is reduced. We can get f by finding HEAD in the reduced graph for each level. Therefor we can get a natural evidence f with the following procedure BIJECTION in $O(n^2)$ time.

```
Procedure BIJECTION

  Procedure MAKE-DEGREE-TABLE

    output table of degrees

    DEG := 0;
    for i := 1 to n do
      DEG := 0;
      for j := 1 to n do
        if M'(i,j) = 1 then DEG := DEG + 1;
      DEGREE-TABLE(i) := DEG;
    end;
  end.

  Procedure DELETE-NODE(i)

    input i : node
    function Modification of M' and DEGREE-TABLE

    for j := 1 to n do
```

```
        begin
          if M'(i,j) = 1 then
            begin
                Rewrite M'(i,j) = 1 to M'(i,j) = 0;
                Rewrite M'(j,i) = 1 to M'(j,i) = 0;
                DEGREE-TABLE(j) := DEGREE-TABLE(i)-1;
            end
        end;
end.

main
  HEAD := undef;
  TAIL := undef;
  MAKE-DEGREE-TABLE;
  for i := 1 to n do
    if DEGREE-TABLE(i) = k then
      if HEAD = undef then HEAD := i else TAIL := i;
  Define f(HEAD) = 1 and f(TAIL) = n;
  DELETE-NODE(HEAD);

  for h := 2 to n - k do
    begin
      i := 1;
      while (DEGREE-TABLE(i) ≠ k) or (i = TAIL) do i := i + 1;
      HEAD := i;
      Define f(HEAD) = h;
      DELETE-NODE(HEAD);
    end;

  for h := k - 1 downto 1 do
    begin
      i := 1;
      while (DEGREE-TABLE(i) ≠ h) or (i = TAIL) do i := i + 1;
      HEAD := i;
      Define f(HEAD) = n - h;
      DELETE-NODE(HEAD);
    end
end.
```

□

Lemma 3.4 *Let $G = (V, E)$ be a input graph such that $|V| = n$, and $G' = (V, E')$ be a critical graph for Π_u such that $E \subseteq E'$. A natural evidence of G for Π_u is computable from the critical graphs G' in $O(n^2)$ time.*

□

References

[1] K. Abrahamson, M.R. Fellows, M.A. Langston, and B.M.E. Moret, Constructive complexity, *Discrete Applied Math.*, 34 (1991), pp.3-16.

[2] D.J. Brown, M.R. Fellows and M.A. Langston, POLYNOMIAL-TIME SELF-REDUCIBILITY: THEORETICAL MOTIVATIONS AND PRACTICAL RESULTS, *Intern. J. Computer Math.*, 31 (1989), pp.1-9.

[3] M.R. Fellows and M.A. Langston, Fast self-reduction algorithms for combinatorial problems of VLSI design, *Proceedings 3rd Aegean Workshop on Computing* (1988), pp.278-287.

[4] M.R. Fellows and M.A. Langston, On search, decision, and the efficiency of polynomial-time algorithms, *Proceedings 21th ACM STOC* (1989), pp.501-512.

[5] R. Impagliazzo and G. Tardos, Decision Versus Search Problems in Super-Polynomial Time, *Proceedings 30th IEEE FOCS* (1989), pp.222-227.

[6] R.M. Karp, E. Upfal, and A. Wigderson, Are Search and Decision Problem Computationally Equivalent?, *Proceedings 17th ACM STOC* (1985), pp.464-475.

[7] F.R.K. Chung, Labelings of graphs, in *Selected topics in graph theory 3* (ed. L.W. Beineke and R.J. Wilson), (1988), pp.151-168.

[8] P.Z. Chinn, J. Chvátalová, A.K. Dewdney, and N.E. Gibbs, The Bandwidth problem for Graphs and Matrices - A Survey, *J. Graph Theory*, 6 (1982), pp.223-254.