A Completeness Theorem for Polynomial-Time Local Search Problems

Shimozono, Shininchi
Department of Information Systems, Kyushu University

出版情報：RIFIS Technical Report. 33, 1990-12-28. Research Institute of Fundamental Information Science, Kyushu University

RIFIS-TR-CS-33

RIFIS Technical Report

A Completeness Theorem for Polynomial-Time Local Search Problems

Shininchi Shimozono

December 28, 1990

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

E-mail: sin@rifis.sci.kyushu-u.ac.jp Phone: 092 (641)1101 Ex. 2329

A Completeness Theorem for Polynomial-Time Local Search Problems

Shinichi Shimozono

Department of Information Systems,
Kyushu University 39, Kasuga 816, Japan†

December 27, 1990

Abstract

We give a series of combinatorial optimization problems defined by graph properties on vertex weighted graphs and allowing the local search methods. We show that the weighted vertex-induced subgraph problem for any nontrivial hereditary property is complete for the class PLS of polynomial-time local search problems, which are defined to formalize the local search algorithms and their complexity of finding locally optimal solutions. Our result yields, without any specific discussions, the PLS-completeness of weighted vertex-induced subgraph problems for many well-known properties.

1 Introduction

In the last twenty years a lot of heuristic approaches have been developed for NP-hard combinatorial optimization problems. The local search method, well known as the Lin-Kernighan algorithm for "Travelling Salesperson Problem" [11], is one of the efficient approximation approaches for optimization problems. Basically the method is based on iterations of deterministic improving process which searches better combinations in polynomial time. Although the algorithm may be trapped in a locally optimal solution far from the optimum, there are a lot of extended researches which try to escape from local optima and seek near optimum solutions by nondeterministic improving procedures [1], [8].

On the other hand, from the computational complexity point of view, Johnson et al. [5] defined the class PLS of polynomial-time local search problems to formalize complexity of finding locally optimal solutions by the local search methods.

†Mailing address: Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka 812, Japan (e-mail: sin@rifis.sci.kyushu-u.ac.jp).

As a remarkable result, they have shown that the Lin-Kernighan algorithm with a P-complete local search procedure is PLS-complete.

In this paper, we prove the PLS-completeness of the weighted vertex-induced subgraph problem for any nontrivial hereditary property. The techniques we employ for the proof are already used for proving NP-completeness or P-completeness of generalized subgraph problems [10], [13], [14], [15]. Even though it is natural to guess a similar completeness result for PLS from the former results, the proof of our result is rather complicated. Our completeness result covers many new PLS-complete problems since a lot of properties such as planar, acyclic, complete, bipartite and chordal [4] are all hereditary and nontrivial.

2 Preliminaries on PLS

First we review some definitions for the class PLS and the first PLS-complete problem FLIP [5].

Definition 1. Let Σ be a finite alphabet. A *polynomial-time local search problem* L is either a maximization or minimization problem specified as follows:

(a) D^L: A subset of Σ^* whose elements are called *instances.*

(b) For each instance $\Pi \in D^L$, we associate it with the following:

  (i) S^L_Π: This is a finite subset of Σ^* called the *solution space.* An element s in S^L_Π is called a *solution* of Π. We assume that $|s|$ is polynomially bounded with respect to $|\Pi|$.

  (ii) $N^L_\Pi(s)$: This is a subset of S^L_Π called the *neighborhood* of s, where s is a solution in S^L_Π. We call a solution in $N^L_\Pi(s)$ a *neighborhood solution* of s.

  (iii) $F^L_\Pi : S^L_\Pi \to \mathrm{N}$: This function is called the *cost function* for S^L_Π, where N is the set of nonnegative integers. The value $F^L_\Pi(s)$ is called the *cost* of s.

We require that D^L, S^L_Π, N^L_Π and F^L_Π are polynomial-time computable with respect to $|\Pi|$. A solution s in S^L_Π is called *locally optimal* if s has no better neighborhood solution, i.e., $F^L_\Pi(s') \leq F^L_\Pi(s)$ (resp., $F^L_\Pi(s') \geq F^L_\Pi(s)$) for all s' in $N^L_\Pi(s)$ when L is a maximization (resp., minimization) problem. We denote by PLS the class of polynomial-time local search problems.

From now on, we consider only maximization problems without loss of generality.

Definition 2. Let L and K be problems in PLS. We say that L is PLS-*reducible* to K if there are polynomial-time computable functions f and g such that (a), (b) and (c) hold for each instance Π of L :

(a) $f(\Pi)$ is an instance of K.

(b) Let s be a solution of $f(\Pi)$. Then $g(f(\Pi), s)$ is a solution of Π.

(c) If s is a locally optimal solution in $S^K_{f(\Pi)}$, then $g(f(\Pi), s)$ is also a locally optimal solution in S^L_{Π}.

Let $C = (x_1, \ldots, x_n, g_1, \ldots, g_N, y_1, \ldots, y_m)$ be an acyclic boolean circuit with n inputs and m outputs. The gates $x_1, \ldots, x_n$ are the *input gates* and $y_1, \ldots, y_m$ are the *output gates.* The indegree of an input gate is 0. Each g_i is either an AND-gate, an OR-gate, or a NOT-gate. The inputs to g_i come from x_j $(1 \leq j \leq n)$ and g_k $(1 \leq k < i)$, The indegree of an output gate y_i is 1 and the value for y_i comes from either x_j $(1 \leq j \leq n)$ or g_k $(1 \leq k \leq N)$. The following problem is known as the first standard PLS-complete problem.

Definition 3. An instance of FLIP is a boolean circuit $C = (x_1, \ldots, x_n, g_1, \ldots, g_M, y_1, \ldots, y_m)$ with n inputs and m outputs. The solution space S_C is the set of boolean assignments to the input gates $x_1, \ldots, x_n$, and the cost function F_C is given by

$$F_C(s) = \sum_{j=1}^{m} y_j \cdot 2^j,$$

where y_j is the j-th output of the circuit with an input $s = (s_1, \ldots, s_n)$. The neighborhood $N_C(s)$ of s is all the assignments obtained by flipping a single bit of the current input, i.e., $N_C(s) = \{(s_1, \ldots, \bar{s}_k, \ldots, s_n) | 1 \leq k \leq n\}$.

Lemma 1. (Johnson, Papadimitriou and Yannakakis [5]) *FLIP is* PLS*-complete.*

3 Main Result

We say that a property π is *hereditary* on induced subgraphs if a graph G satisfies π then all vertex-induced subgraphs of G satisfy π. We say that a property π is *nontrivial* on a family Γ of graphs if infinitely many graphs in Γ satisfy π and some in Γ violates π.

Definition 4. Let π be a hereditary property on graphs. The *weighted greedy maximal π problem* (WGM-π) is defined as follows. An instance is a vertex-weighted graph $G = (V, E, W)$, where W is a function $W : V \to \mathrm{N}$ of weights on vertices. We assume a linear order on vertices V. A solution V^* is a subset of vertices inducing a subgraph satisfying π, and the cost $F_G(V^*)$ of V^* is defined by

$$F_G(V^*) = \sum_{v \in V^*} W(v).$$

A neighborhood solution $U(u, V^*)$ of V^* shall be generated for each $u \in V - V^*$ as follows:

$U(u, V^*) \leftarrow \{u\} \cup (V^* - \{v | v \text{ is adjacent to } u\})$.
$T \leftarrow V - V^*$.
while $T \neq \emptyset$
  Choose the first $t \in T$ of the largest weight.
  If the subgraph induced by $U(u, V^*) \cup \{t\}$ does not violate π
    then $U(u, V^*) \leftarrow U(u, V^*) \cup \{t\}$.
  $T \leftarrow T - \{t\}$.
end of while.

Our main result is the following theorem.

Theorem 1. *If a property π is hereditary, nontrivial and polynomial-time testable, then the WGM-π is* PLS*-complete.*

For the proof of Theorem 1, we first show the PLS-completeness of the *weighted greedy maximal independent set problem* (WGMIS) that is the WGM-π problem defined by setting $\pi =$ "independent set", where an independent set of a graph is a set of vertices such that no two vertices are adjacent.

Without formal discussion, Johnson et al. [5] have already mentioned the PLS-completeness of the weighted independent set problem with a "Kernighan-Lin-like" local search algorithm that is defined by slightly modifying the original Kernighan-Lin algorithm [7]. However, since our neighborhood of WGM-π is different from their neighborhood, we need to prove the PLS-completeness of our WGMIS.

Lemma 2. *WGMIS is* PLS*-complete.*
Proof. We PLS-reduce FLIP to WGMIS. Let $C = (x_1, \ldots, x_n, g_1, \ldots, g_N, y_1, \ldots, y_m)$ be a boolean circuit with n inputs and m outputs as an instance of FLIP. We construct a weighted graph $G' = (V', E', W')$ that simulates the computation of C for the current input and its neighborhood solutions. From now on, without loss of generality, we may assume that C contains only NAND-gates.

At first, we construct the subgraphs G'^k for $0 \leq k \leq n$. For an input $s = (s_1, \ldots, s_n)$ of C, G'^k $(1 \leq k \leq n)$ (resp., G'^0) simulates the computation of C with the neighborhood solution $(s_1, \ldots, \bar{s}_k, \ldots, s_n)$ (resp., $(s_1, \ldots, s_n)$) as an input. $G'^k = (V'^k, E'^k, W'^k)$ is given as follows:

> For each input gate x_i $(1 \leq i \leq n)$, G'^k has an edge $\{x_i^k, \bar{x}_i^k\}$. For each gate g_i $(1 \leq j \leq N)$, G'^k has an edge $\{g_i^k, \bar{g}_i^k\}$ (Fig. 2 (a)). We call these edges the *gate value pairs*. For each NAND-gate $g_j \leftarrow v \wedge w$, G'^k contains a triangle $\{\alpha_j^k, \beta_j^k\}, \{\beta_j^k, \gamma_j^k\}, \{\gamma_j^k, \alpha_j^k\}$ (Fig. 2 (b)) called the *gate triangle* of g_j, where v and w are in $\{x_1, \ldots, x_n, g_1, \ldots, g_{j-1}\}$. In addition to this triangle, G'^k has edges $\{v^k, \alpha_j^k\}$, $\{\bar{v}^k, \gamma_j^k\}$, $\{w^k, \beta_j^k\}$, $\{\bar{w}^k, \gamma_j^k\}$, $\{\gamma_j^k, g_j^k\}$, $\{\bar{v}^k, \bar{g}_j^k\}$ and $\{\bar{w}^k, \bar{g}_j^k\}$ as shown in Fig. 2 (c). We call the pairs $\{v^k, \bar{v}^k\}$ and $\{w^k, \bar{w}^k\}$ (resp., $\{g_j^k, \bar{g}_j^k\}$ the inputs (resp., output) of the gate triangle. For each output gate y_l $(1 \leq l \leq m)$, G'^k contains an edge $\{y_l^k, \bar{v}_l^k\}$, where v_l is the gate directly connected to y_l.

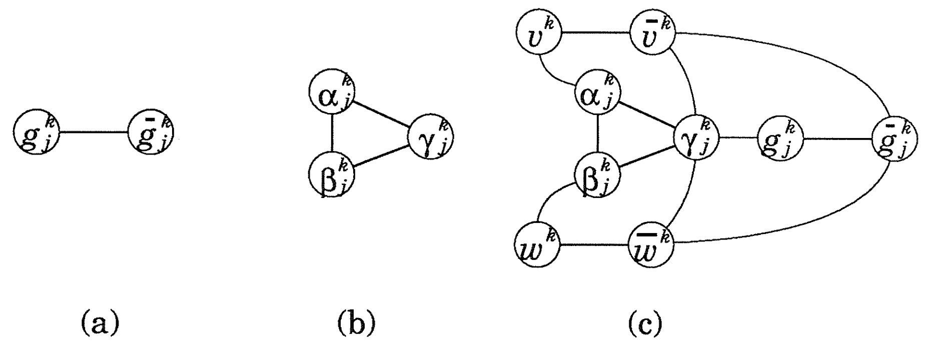

Figure 1: The graph representation of (a) a gate value pair, (b) a gate triangle and (c) a NAND-gate calculating $g_j \longleftarrow \overline{v \wedge w}$.

The weights of the newly added vertices are given as follows: (i) $\sigma^0 \rightarrow 2^{8N} + 2^{2N}$, (ii) $\sigma^1, \ldots, \sigma^n \rightarrow 2^{8N}$, (iii) $\mu^j, \bar{\mu}^j \rightarrow 2^{2N}$, (iv) $x_i, \bar{x}_i \rightarrow 2^N$, (v) $\nu_i^k, \bar{\nu}_i^k \rightarrow 1$.

Finally, the linear order on vertices V can be given appropriately by the names and the indices of vertices. We omit the details. It can be defined, for example, $\sigma^1 < \ldots < \sigma^n$, $\alpha_i^k < \beta_i^k < \gamma_i^k$, $v_i^k < \bar{v}_i^k$, and so forth.

The graph G' given above simulates the neighborhood-searching steps of FLIP by simulating G'^0 and G'^k $(1 \leq k \leq n)$ alternatingly. The connections between $x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n$ and $\nu_1^k, \bar{\nu}_1^k, \ldots, \nu_n^k, \bar{\nu}_n^k$ imply that the gate value pair $\{\nu_i^k, \bar{\nu}_i^k\}$ represents the i-th value of $s_1, \ldots, \bar{s}_k, \ldots, s_n$ for $1 \leq i \leq n$. Next we look into the following three claims about locally optimal solutions.

Claim 1. If an independent set $V'^* \subseteq V'$ is a locally optimal solution of G', it contains exactly one switch vertex σ^k and the vertices of the subgraph G'^k simulates the computation of the circuit C for the input represented by the gate value pairs $\{\nu_1^k, \bar{\nu}_1^k\}, \ldots, \{\nu_n^k, \bar{\nu}_n^k\}$.

Proof. Observe the following facts:

(1) One of σ^k $(0 \leq k \leq n)$ must be chosen in V'^*. If not, we have a neighborhood solution by adding any σ^k to V'^* and removing all vertices in $\cup_{j \neq k} V'^j$ from V'^*. This neighborhood solution results in positive gain more than $2^{8N} - n \cdot n \cdot 2 \cdot 2^{7N}$, and since $\sigma^0, \ldots, \sigma^n$ form the complete graph K_{n+1}, exactly one switch vertex is in V'^*.

(2) The gate value pair $\{x_i^k, \bar{x}_i^k\}$ represents either 1 or 0, i.e., exactly one of $x_i^k, \bar{x}_i^k$ is in V'^*. If none of $x_i^k, \bar{x}_i^k$ is in V'^*, we have a neighborhood solution by adding one of $x_i^k, \bar{x}_i^k$ to V'^* and removing all adjacent vertices. This results in positive gain more than $2^{7N} - 2^{6N}$.

(3) All gate value pairs and gate triangles of G'^k represent the computations of NAND-gates. Let $\{v^k, \bar{v}^k\}$ and $\{w^k, \bar{w}^k\}$ be the inputs and let $\{g_j^k, \bar{g}_j^k\}$ be the output of a NAND-gate triangle $\{\alpha_j^k, \beta_j^k, \gamma_j^k\}$ as shown in Fig. 2 (c). Then exactly one of $\alpha_j^k, \beta_j^k, \gamma_j^k$ (resp., $g_j^k, \bar{g}_j^k$)is in V'^*. Consider the following two cases:

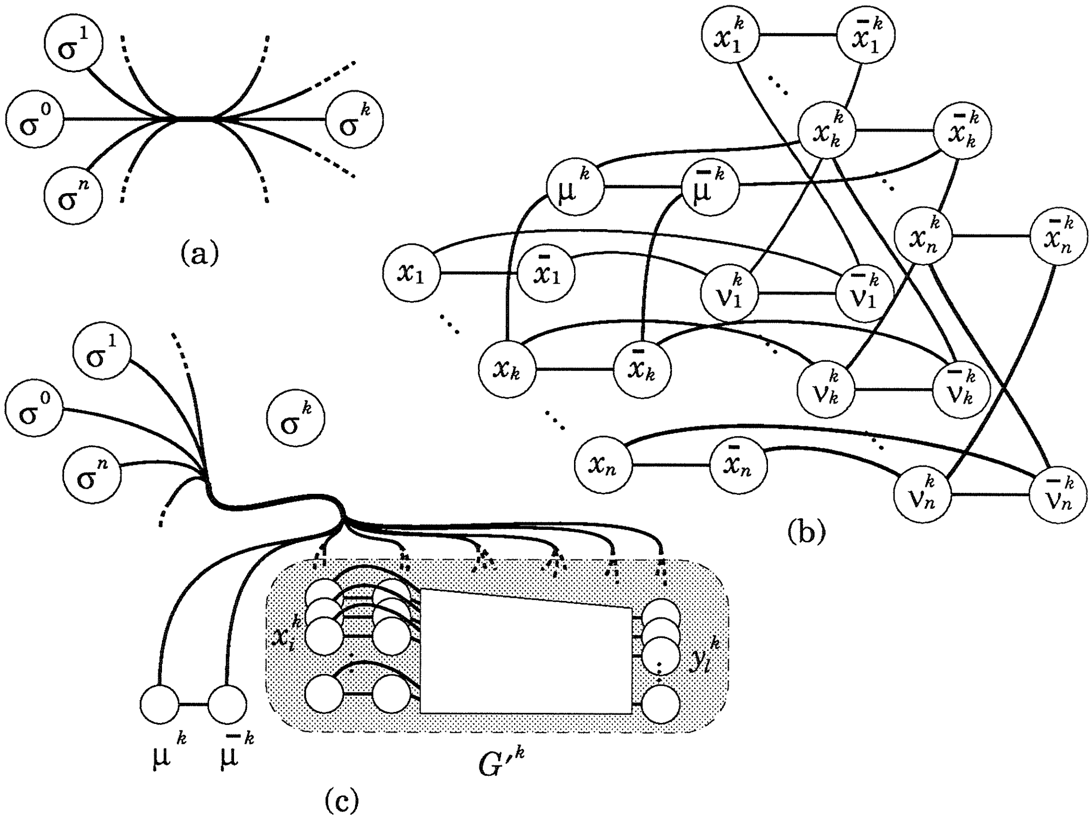

Figure 2: The constructions of the graph G' simulating the FLIP.

Case 1. None of $\alpha_j^k, \beta_j^k, \gamma_j^k$ is in V'^*:
If v^k or w^k is not in V'^*, we can add α_i^k or β_i^k to V'^* and get weight $2^{3(N-j)+3N+1}$. Otherwise, v^k and w^k are both in V'^*. Then we add γ_i^k to V'^* and get weight $2^{3(N-j)+3N+1}$ with loss at most $2^{3(N-j)+3N}$.

Case 2. None of $g_j^k, \bar{g}_j^k$ is in V'^*:
If v^k and w^k are both in V'^*, we add $\bar{g}_j^k$ to V'^* and get weight $2^{3(N-j)+3N}$ with loss less than $2 \cdot 2^{3(N-j-1)+3N+1}$. Otherwise, we add g_j^k to V'^* and also get positive gain.

Claim 2. If a solution V'^* is locally optimal and the selection switch σ^k is chosen in V'^*, the value represented by $\{x_i, \bar{x}_i\}$ is the same as that of $\{x_i^k, \bar{x}_i^k\}$ for $1 \le i \le n$.

Proof. As we mentioned above, the truth values obtained by the gate value pairs in V'^k represents the computation of C. Then we can show that the value represented by $\{x_i, \bar{x}_i\}$ is the same as that of $\{x_i^k, \bar{x}_i^k\}$. If not, we can reach a new neighborhood solution with a better cost as follows:

Case 1. Suppose that the value represented by the gate value pair $\{x_k, \bar{x}_k\}$ differs from that of $\{x_k^k, \bar{x}_k^k\}$. For producing the better neighborhood solution, choose μ^k or $\bar{\mu}^k$ that is not adjacent to the chosen vertex of the pair $\{x_k^k, \bar{x}_k^k\}$ and reject both x_k and $\bar{x}_k$ with gain at least $2^{2N} - 2^N$. Then for the next better neighborhood solution, choose x_k or $\bar{x}_k$ in the same way (and this implies that both ν_i^k and $\bar{\nu}_i^k$ are rejected) with gain $2^N - 1$.

Case 2. Otherwise, neither ν_i^k nor $\bar{\nu}_i^k$ is not in V'^*. For producing the better neighborhood solution, choose one of $x_i, \bar{x}_i$ in the same way as the choice of $\{x_i^k, \bar{x}_i^k\}$ and reject $\nu_i^j, \bar{\nu}_i^j$ for all $0 \le j \le n$, then add ν_i^j or $\bar{\nu}_i^j$ for all $0 \le j \le n$ to V'^* in the linear order. We get positive gain 1.

Claim 3. If V'^* is locally optimal, it must include the selection switch σ^0, and the sequence of the values represented by $\{\nu_1^j, \bar{\nu}_1^j\}, \ldots, \{\nu_n^j, \bar{\nu}_n^j\}$ is the neighborhood solution of FLIP given by flipping the j-th bit for $0 \le j \le n$ (if $j = 0$, it means that no flip occurs).

Proof. If σ^k with $k \neq 0$ is chosen in V'^*, we can reach a better neighborhood solution $U(\sigma^0, V'^*)$. We simply choose σ^0 and reject all in $\{\sigma^k, \mu^k, \bar{\mu}^k\} \cup V'^k$, then choose vertices in V'^0 by weight descending order. Since the value represented by $\{x_i, \bar{x}_i\}$ is the same as the previous input $\{x_i^k, \bar{x}_i^k\}$ of G'^k from Claim 2, the new solution $U(\sigma^0, V'^*)$ simulates the computation for the same input by V'^0. Moreover, we obtain positive gain 1 by adding one of $\nu_k^k, \bar{\nu}_k^k$ that was rejected by $\{x_k, \bar{x}_k\}$ and $\{x_k^k, \bar{x}_k^k\}$. It means the pairs $\{\nu_i^j, \bar{\nu}_i^j\}$ for $0 \le j \le n$ represent the neighborhood solution.

Now we are ready to show that an independent set V'^* has no improved neighborhood solution only if the corresponding input of C has no better solutions, i.e., the solution of FLIP determined by V'^* is a locally optimal solution. For V'^*, we define a sequence of bits $s_1, \ldots, s_n$ as follows: If the pair $\{x_i, \bar{x}_i\}$ represents 1, i.e., the vertex x_i is in V'^*, then the i-th input s_i is 1; otherwise, s_i is 0. Suppose that V'^* is a locally optimal solution for G' and the solution $(s_1, \ldots, s_n)$ of FLIP determined by the values of $\{x_1, \bar{x}_1\}, \ldots, \{x_n, \bar{x}_n\}$ is not. Then we have at least one better neighborhood solution for C which can be obtained by flipping a single bit. Let $(s_1, \ldots, \bar{s}_k, \ldots, s_n)$ be one of those improved solutions. Since we

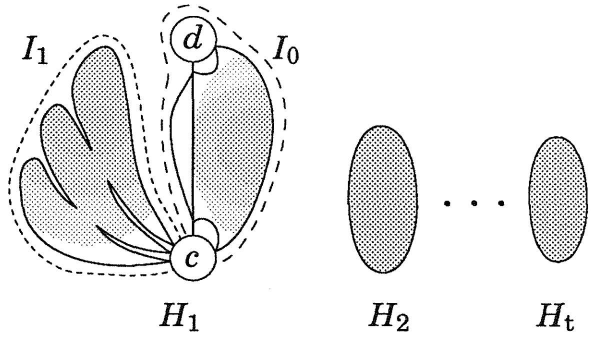

Figure 3: The general form of a critical graph as connected components.

have σ^0 in V'^* from Claim 3 and the value of $\{x_i, \bar{x}_i\}$ is the same as $\{x_i^0, \bar{x}_i^0\}$ for all $1 \leq i \leq n$, we can get a new better neighborhood solution $U(\sigma^k, V'^*)$. We add σ^k to V'^* and reject all of V'^0, then choose vertices in V'^k by the weight descending order. At first for all $1 \leq i \leq n$ we must chose vertices of $\{x_i^k, \bar{x}_i^k\}$ that are not adjacent to already chosen $\nu_i^k, \bar{\nu}_i^k$, so the new solution V'^* simulates the computation of the circuit with inputs $(s_1, \ldots, \bar{s}_k, \ldots, s_n)$. The vertices $y_1^k, \ldots, y_m^k$ simulates the cost of the solution $(s_1, \ldots, \bar{s}_k, \ldots, s_n)$ of C, so the solution $U(\sigma^k, V'^*)$ contributes more weights than V'^*. This contradicts the local optimality of the solution V'^*. □

Before we look into our main result, we review some notions from [10] and [13]. A vertex c is called a cutpoint of graph if deletion of c separates the graph into at least two connected components. A subgraph consisting of a resulting connected component together with c and the edges between c and the component is called a component relative to c.

For a connected graph H, we define the *α-sequence* α_H of H as follows. If H is not biconnected, let c be any cutpoint of H and let $H_1, \ldots, H_{j(c)}$ be connected components relative to c. Then $\alpha_{c,H} = \langle |H_1|, \ldots, |H_{j(c)}| \rangle$, where $|H_i|$ represents the number of vertices in H_i, and we assume $|H_1| \geq \cdots \geq |H_{j(c)}|$. Then α_H is defined by $\alpha_H = \min\{\alpha_{c,H} | c \text{ is a cutpoint of } H\}$, where min is the minimum with respect to the lexicographic order on sorted lists of positive integers. Let c_H be any cut point with $\alpha_H = \alpha_{c_H,H}$. If H is biconnected, we define $\alpha_H = \langle |H| \rangle$ and let c_H be any vertex. For a graph G with connected components $G_1, \ldots, G_t$, the *β-sequence* β_G of G is $\langle \alpha_{G_1}, \ldots, \alpha_{G_t} \rangle$, where $\alpha_{G_1} \geq \cdots \geq \alpha_{G_t}$.

Proof of Theorem 1. We PLS-reduce WGMIS G' to WGM-π problem G. There is a *critical graph* H such that $\beta_H = \min\{\beta_{H'} | H' \text{ is a graph violating } \pi\}$. β_H, the β-sequence of H, can be expressed as the sequence of connected components $\langle \alpha_{H_1}, \alpha_{H_2}, \ldots, \alpha_{H_t} \rangle$. Let c be a cutpoint of H_1. Since the deletion of c produces at least two connected components if H_1 is not biconnected, let I_0 be the largest connected component of H_1 relative to c and let I_1 be the graph obtained by removing I_0 except c. I_0 has a vertex d ($\neq c$) such that the distance between c and d is one.

At first, we define the indices of vertices for graph I_0 and I_1 by assigning unique integers, for calculating the weight W. Let $\delta(u,v)$ be the distance between vertices u and v on the graph. For each graph I_0 and I_1, the depth of the vertex v in I_0 is defined as

$$\delta'(v) = \min\{\delta(v,c), \delta(v,d)\},$$

and the depth of v in I_1 is

$$\delta'(v) = \delta(v,c).$$

We define the index $\tau(v)$ of vertices based on the depth of v for both graphs I_0 and I_1 as follows. No two vertices in I_0 (resp., I_1) have the same index, and if two vertices v and u satisfy relation $\delta'(v) > \delta'(u)$ then these vertices must satisfy relation $\tau(v) > \tau(u)$.

Now we construct a graph $G = (V, E, W)$ for a given graph $G' = (V', E', W')$ as follows. A copy of I_1 is attached to each vertex u of G' by identifying u with c. Then each edge $\{u, v\}$ in E' is replaced by a copy of I_0 by identifying u with c, and v with d. Finally, independent graphs $H_2, \ldots, H_t$ are added.

Moreover, the weight function W is defined as follows.

(1) If vertex $v \in V$ corresponds to one of vertices of V', the weight $W(v)$ is the same as $W'(v)$.

(2) If vertex v is one of vertices of $H_2, \ldots, H_t$, the weight $W(v)$ is some constant, for example, 1.

(3) Otherwise, vertex v corresponds to one of I_0 or I_1. In this case its weight is $2^{\tau(v)} \cdot ||V'||$, where $\tau(v)$ is the index of v and $||V'||$ is the total weight of the vertices in V'.

The linear order on V is given as follows: If $v \in V$ corresponds to one of V', then the order of v is determined by the order on V'; Otherwise, the order is determined arbitrary.

Finally, the mapping g of solutions calculates V'^* of G' as follows: if vertex $v \in V \cap V'$ of G is in V^*, then the corresponding vertex v' of G' is in V'^*; otherwise vertex v' is not in V'^*.

For the proof of the theorem, we have to consider two cases.

(i) The critical graph H consists of only one connected component.

(ii) The critical graph H consists of at least two connected components.

Case (i). First, we show the following claim.

Claim. The solution V'^* of G' induced by V^* is locally optimal if V^* is an optimal solution of G.

We show that all vertices in $V - V'$ must be chosen in V^* if it is an optimal solution. Suppose that some $u \in V - V'$ is not in V^*. Since there is at least one vertex which is adjacent to u and weight at most $W(u)/2$, we have an improved neighborhood solution $U(u, V^*)$, that adds u to V^* and rejects vertices of smaller weights than u.

Now suppose that V^* is an optimal solution and the solution V'^* induced by V^* is not. Then we have at least one better solution of G' rather than V'^*, so let $U'(u', V'^*)$ be such solution. We choose the vertex u in V corresponding to u' for producing the new neighborhood solution, so all the vertices adjacent to u are rejected. Therefore, we can add $u \in V$ to V^*, and all other vertices corresponding to the vertices added to $U'(u', V'^*)$ will be added in the same manner. This introduces a new improved solution of G, and contradicts the assumption. Thus the claim holds.

Case (ii). If we have more than two connected components in H, we need a special restriction on the initial solution: the connected components $H_2, \ldots, H_t$ and subgraphs corresponding to H_1 are separated from each other, so we can not select all vertices from $H_2, \ldots, H_t$ if vertices of one of the copies of $I_0 \cup I_1$ have completely included in V^*. Therefore the initial solution which will be produced by the algorithm must have all vertices of $H_2, \ldots, H_t$ in V^*. This condition is enough for this case. In the neighborhood structure of *WGM-π* we can reject only adjacent vertices, so the vertices of $H_2, \ldots, H_t$ will not be rejected. The mapping of solution g is calculated by the vertices corresponding to the vertices of H_1, so the restriction makes no affects to PLS-reduction. The same proof as Case (i) will be applied, and the mapping of solutions g is also the same as Case (i). □

4 Concluding Remarks

Johnson et al. [5] conjectured that every PLS-complete problem requires a P-complete local search procedure. Krentel [9] has disproved this conjecture by showing PLS-complete problems with LOGSPACE local search procedures. Although the verification of local optimality of WGMIS is P-complete, we conjecture that similar problems which use NC algorithms for the maximal independent set problem such as [3], [12], [6] instead of the lexicographically first maximal independent set are also PLS-complete. The techniques for the reduction given here do not work directly with these NC local search procedures. We leave the completeness of the problem with NC local search as an open problem.

References

[1] Aarts, E. H. L. and Korst, J. H. M., Boltzmann machines for traveling salesman problems, *European J. Operational Research* **39**, 79–95, 1989.

[2] Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and company, 1978.

[3] Goldberg, M. and Spencer, T., A new parallel algorithm for the maximal independent set problem, *SIAM J. Comput.* **18** (2), 419–427, 1989.

[4] Golumbic, M. C., *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.

[5] Johnson, D. S., Papadimitriou, C. H. and Yannakakis, M., How easy is local search?, *J. Comput. System Sci.* **37**, 79–100, 1988.

[6] Karp, R. M. and Wigderson, A., A fast parallel algorithm for the maximal independent set problem, *J. Assoc. Comput. Mach.* **32** (4), 762–73, 1985.

[7] Kernighan, B. W. and Lin, S., An efficient heuristic procedure for partitioning graphs, *Bell Systems Technical Journal* **49**, 291–307, 1972.

[8] Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., Optimization by simulated annealing, *Science* **13** (230), 671–680, 1983.

[9] Krentel, M. W., On finding locally optimal solutions (extended abstract), In *Structure in Complexity Theory Fourth Annual Conference*, pages 132–137, 1989.

[10] Lewis, J. M. and Yannakakis, M., The node-deletion problem for hereditary properties is NP-complete, *J. Comput. System Sci.* **20**, 219–230, 1980.

[11] Lin, S. and Kernighan, B. W., An effective heuristic algorithm for the traveling salesman problem, *Operations Research* **21**, 1973.

[12] Luby, M., A simple parallel algorithm for the maximal independent set problem, *SIAM J. Comput.* **15** (4), 1036–53, 1986.

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

[14] Miyano, S., A new series of Δ_2^p-complete problems, In *Proc. Int. Workshop on Discrete Algorithms and Complexity*, pages 187–194, 1989.

[15] Yannakakis, M., Node-deletion problems on bipartite graphs, *SIAM J. Comput.* **10**, 310–327, 1981.

About the Author

Shinichi Shimozono (下薗 真一) was born on April 7, 1966 and graduated from Department of Physics, Kyushu University in 1989. He is now a graduate student at Department of Information Systems, Kyushu University. He is studying the computational complexity theory.

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