PAPER *Special Section on Foundations of Computer Science*

Query-Number Preserving Reductions and Linear Lower Bounds for Testing

Yuichi YOSHIDA[†a)], ***Nonmember and*** **Hiro ITO**[†b)], ***Member***

SUMMARY In this paper, we study lower bounds on the query complexity of testing algorithms for various problems. Given an oracle that returns information of an input object, a testing algorithm distinguishes the case that the object has a given property P from the case that it has a large distance to having P with probability at least $\frac{2}{3}$. The query complexity of an algorithm is measured by the number of accesses to the oracle. We introduce two reductions that preserve the query complexity. One is derived from the gap-preserving local reduction and the other is from the L-reduction. By using the former reduction, we show linear lower bounds on the query complexity for testing basic NP-complete properties, i.e., 3-edge-colorability, directed Hamiltonian path/cycle, undirected Hamiltonian path/cycle, 3-dimensional matching and NP-complete generalized satisfiability problems. Also, using the second reduction, we show a linear lower bound on the query complexity of approximating the size of the maximum 3-dimensional matching.
key words: *property testing, lower bounds*

1. Introduction

To decide whether a huge object has some predetermined property, a concept called *property testing* was proposed [3], [11]. In the setting of property testing, a testing algorithm is supposed to distinguish that an object has a predetermined property P and that it has a large distance to having P with high probability (say, $\frac{2}{3}$). The definition of farness varies depending on problems. In order to construct algorithms that run in constant time, i.e., independent of the size of the object, we do not want to even read the whole object. Thus, we assume the existence of an oracle that represents the object and a testing algorithm obtains information of the object by accessing it. The efficiency of the testing algorithm is measured by the *query complexity*, i.e., the number of accesses to the oracle.

In the last decade, a lot of constant-time testing algorithms were developed. However, lower bounds on the query complexity were less investigated. In this paper, we present a general technique using a reduction to prove (linear) lower bounds on the query complexity.

We introduce two reductions that preserve the query complexity by adding new constraints to known reductions. One of our reductions is *strong gap-preserving local reduction*, which maps a decision problem to another decision problem. This is derived from the gap-preserving local reduction introduced by [1]. The other one is *strong L-reduction*, which maps an optimization problem to another optimization problem. The L-reduction [9] is originally designed to show the nonexistence of PTAS (polynomial time approximation scheme).

Using these reductions, we show that various problems have linear lower bounds on the query complexity for testing. That is, all problems listed below have linear lower bounds. We call a generalized satisfiability problem *Schaefer* if it is NP-complete [12] (the concrete definition is deferred to Sect. 6). In particular, the Schaefer generalized satisfiability problem includes one-in-three 3SAT and not-all-equal 3SAT as special cases.

3EC-d (*Bounded* 3-*Edge-Colorability*)
Instance: An undirected graph with a degree upper bound d.
Question: Can the edges be colorable by 3 colors such that no two edges with the same color share a common vertex?

DHP-d and DHC-d (*Bounded Directed Hamiltonian Path/Cycle*)
Instance: A directed graph with a degree upper bound d.
Question: Does the graph contain a Hamiltonian path/cycle, i.e., a path/cycle that visits each vertex exactly once?

UHP-d and UHC-d (*Bounded Undirected Hamiltonian Path/Cycle*)
Instance: An undirected graph with a degree upper bound d.
Question: Does the graph contain a Hamiltonian path/cycle, i.e., a path/cycle that visits each vertex exactly once?

3DM-d (*Bounded* 3-*Dimensional Matching*)
Instance: Set $E \subseteq U \times V \times W$ where U, V and W are disjoint sets. The number of occurrences in E of an element of U, V and W is bounded from above by a constant d.
Question: Does E contain a matching, i.e., $E' \subseteq E$ such that $|E'| = \min(|U|, |V|, |W|)$ and no two elements of E' agree in any coordinate?

Schaefer-3SAT-d (*Bounded Generalized Satisfiability Problem*)
Instance: An instance of Schaefer generalized satisfiability problem such that each clause has exactly 3 variables and each variable occurs at most d times.

Manuscript received March 23, 2009.
Manuscript revised June 3, 2009.
[†]The authors are with School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
a) E-mail: yyoshida@lab2.kuis.kyoto-u.ac.jp
b) E-mail: itohiro@kuis.kyoto-u.ac.jp
DOI: 10.1587/transinf.E93.D.233

Question: Is there a truth assignment to variables that satisfies all clauses?

Max-3DM-d (*Bounded Max-3-Dimensional Matching*) is an optimization problem, for which given an instance of 3DM-d $E \subseteq U \times V \times W$, we are to find the maximum number of subsets of E that have no element in common. We show that approximating Max-3DM-d (Bounded Max-3-Dimensional Matching) has a linear lower bound on the query complexity.

(1) Related Work:

In [5], it is shown that any testing algorithm for bipartiteness and being expander of a graph with a degree bound requires $\Omega(\sqrt{n})$ queries where n is the number of vertices in a graph. In [1], linear lower bounds on the query complexity are shown for testing 3SAT-d (solvability of a SAT such that each clause has exactly 3 variables and each variable occurs at most d times) and E3LIN-2 (solvability of linear equations over $\{0, 1\}$ with 3 variables in each equation) such that each variable occurs in at most d equations. A linear lower bound on the query complexity of testing 3-colorability for a graph with a degree bound is also shown in the same paper. There are other results for one-sided testers, i.e., testing algorithms that always accept graphs satisfying the concerned property. It is known that any one-sided tester requires $\Omega(n)$ queries to test cycle-freeness [5] and the property of having a perfect matching [14]. Also, there is a property of graphs with/without a degree bound that is testable in query complexity $O(n^\alpha)$ but cannot be testable in query complexity $o(n^\alpha)$ for any constant $\alpha > 0$ [4].

For inapproximability results, linear lower bounds on the query complexity of approximating the size of Max-2SAT, Max-3SAT, Vertex Cover and Max-Cut with some constant approximation ratios are known [1]. In particular, approximating the size of Vertex Cover requires $\Omega(\sqrt{n})$ queries even if the approximation ratio is $2 - \gamma$ for any $\gamma > 0$ [10].

(2) Organization:

This paper is organized as follows. In Sect. 2, we formally give the definition of the testing and the models for each problem. Also, we state the new reductions and basic properties of the reductions. A linear lower bound on the query complexity for testing 3EC-d is shown in Sect. 3. Linear lower bounds on the query complexity for testing DHP-d, DHC-d, UHP-d and UHC-d are shown in Sect. 4. Furthermore, linear lower bounds on the query complexity for testing 3DM-d and Schaefer-3SAT-d are shown in Sects. 5 and 6, respectively. Section 7 is devoted to describe the direction of future research.

2. Definitions and Preliminaries

2.1 Testing and Problems

First, we define 3SAT-d, which will be used as a source problem of our reductions.

3SAT-d (*Bounded 3SAT*)

Instance: A CNF such that each clause has exactly 3 variables and each variable occurs at most d times.

Question: Is there an assignment to variables such that every clause is satisfied?

Max-3SAT-d (*Bounded Max-3SAT*) is an optimization problem, for which given a instance of 3SAT-d we are to maximize the number of satisfied clauses.

Let $\mathcal{X}$ be combinatorial objects with a functional representation f. The *query complexity* of an algorithm is measured by the number of queries to f made by the algorithm. An instance $X \in \mathcal{X}$ is called *ϵ-far* from (satisfying) a property P if an ϵ-fraction of f should be changed to make X having the property P. The concrete definition of ϵ-farness depends on the concerned combinatorial objects.

Let A be a decision problem that decides whether an object $X \in \mathcal{X}$ satisfies a property P or not. A *testing algorithm* with an error parameter $\epsilon > 0$ (or *ϵ-tester*) for A is a randomized algorithm that, given an oracle f that represents X:

- if X satisfies P, accepts X with probability at least $\frac{2}{3}$.
- if X is ϵ-far from P, rejects X with probability at least $\frac{2}{3}$.

Let A be an optimization problem on $\mathcal{X}$. For $X \in \mathcal{X}$, let $OPT_A(X)$ denote the optimal value of X for A. An approximation algorithm with an error parameter ϵ (or *ϵ-approximator*) for A, given an oracle f that represents X, returns $r(X)$ such that $\frac{OPT_A(X)}{1+\epsilon} \leq r(X) \leq (1 + \epsilon)OPT_A(X)$.

We describe the model used for 3SAT-d and Schaefer-3SAT-d. Let $X = (U, C)$ be an instance of 3SAT-d or Schaefer-3SAT-d where U is a set of variables and C is a set of clauses. We assume that X is represented by functions $f_U : U \times [d] \to C \cup \{\emptyset\}$ and $f_C : C \times [3] \to U$ where $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$. $f_U(u, i)$ represents the ith clause at which u occurs. If no such clause exists, the value of it is $\emptyset$. $f_C(c, i)$ represents the ith literal in c. A testing algorithm can obtain information of X by making a query to f_U and f_C. For simplicity, we assume that we know the relations of clauses as a prior knowledge. X is called ϵ-far from being satisfiable if at least $\frac{\epsilon dn}{3}$ clauses must be removed to make it satisfiable where n is the number of variables in X.

Next, we describe the model for 3EC-d and DHP-d. Let $G = (V, E)$ be an undirected graph with a degree bound d and n be the number of vertices in G. G is represented by a function $f : V \times [d] \to V \cup \{\emptyset\}$. $f(v, i)$ represents the ith vertex incident to v. If no such vertex exists, the value of it is $\emptyset$. G is called ϵ-far from a property P if at least $\frac{\epsilon dn}{2}$ edges must be added or removed to make the graph satisfy the property P preserving the degree bound d.

The representation and the definition of ϵ-farness for directed graphs are similar to those for undirected graphs. Let $G = (V, E)$ be a directed graph with a degree bound d (i.e, both of in-degrees and out-degrees are bounded by d) and n be the number of vertices in G. G is represented by a function $f_{in} : V \times [d] \to V \cup \{\emptyset\}$ and $f_{out} : V \times [d] \to V \cup \{\emptyset\}$.

$f_{in}(v, i)$ represents the end-vertex of the ith in-coming edge of v. $f_{out}(v, i)$ represents the end-vertex of the ith out-going edge of v. If there is no such vertex, the value of it is $\emptyset$. G is called ϵ-far from a property P if at least ϵdn edges must be added or removed to make the graph satisfy the property P preserving the degree bound d.

Finally, we describe the model for 3DM-d. Let $G = (U, V, W, E), E \subseteq W \times X \times Y$ be an instance of 3DM-d and $n = |U| = |V| = |W|$. G is represented by functions $f_U : U \times [d] \to E \cup \{\emptyset\}, f_V : V \times [d] \to E \cup \{\emptyset\}, f_W : W \times [d] \to E \cup \{\emptyset\}$ and $f_E : E \times [3] \to U \cup V \cup W$. f_U, f_V and f_W returns the ith element of E that contains an element of U, V and W, respectively. f_E returns an element of an edge. G is called ϵ-far from a property P if at least ϵdn edges must be added or removed to make G satisfy the property P preserving the property that each element of U, V and W occurs at most d edges.

2.2 New Reductions and Key Lemmas

We introduce two reductions by adding some restrictions to known reductions, i.e., a gap-preserving local reduction [1], that reduces a decision problem and an L-reduction [9], that reduces an optimization problem. These reductions describe the relation between the query complexity of a problem and the query complexity of the reduced problem. For a property P, let P_0 denote a set of instances that satisfy P, and let P_ϵ denote a set of instances that are ϵ-far from having P for $\epsilon > 0$.

Definition 2.1 (Strong gap-preserving local reduction)**.** *Let A and B be decision problems for properties P and Q, respectively. We say that a mapping φ is a* strong gap-preserving local reduction *from A to B if there exist universal constants $c_1, c_2, c_3 > 0$ such that the following properties hold:*

1. *If $X \in P_0$, then $\varphi(X) \in Q_0$.*
2. *If $X \in P_\epsilon$, then $\varphi(X) \in Q_{\epsilon/c_1}$.*
3. *The answer to an oracle query to $\varphi(X)$ can be computed by making at most c_2 oracle queries to X.*
4. *$|\varphi(X)| \leq c_3|X|$ for any X.*

The condition (4) is added to the previous definition of a gap-preserving local reduction.

Definition 2.2 (Strong L-reduction)**.** *Let A and B be two optimization problems with objective functions g_A and g_B, respectively. We say that a mapping φ is a* strong L-reduction *from A to B if there exist universal constants $c_1, c_2, c_3, c_4 > 0$ such that the following properties hold:*

1. *For any X of an instance of A, $OPT_B(\varphi(X)) \leq c_1 OPT_A(X)$.*
2. *For every solution x_B of $\varphi(X)$ we can find a solution x_A of X such that $|OPT_A(X) - g_A(x_A)| \leq c_2|OPT_B(\varphi(X)) - g_B(x_B)|$ without any query.*
3. *The answer to an oracle query to $\varphi(X)$ can be computed by making c_2 oracle queries to X.*
4. *$|\varphi(X)| \leq c_4|X|$ for any X.*

The conditions (3) and (4) are added to the previous definition of L-reduction. Also, the condition (2) is different from that of L-reduction. The solution of X must be calculated from the solution of $\varphi(X)$ without any query. L-reductions permit polynomial time calculation for this.

A strong gap-preserving local reduction and a strong L-reduction preserve a linear lower bound on the query complexity of testing and approximating as shown in the following lemmas, which are key lemmas of this paper.

Lemma 2.3. *Let A and B be decision problems for properties P and Q, respectively. Suppose that there is a strong gap-preserving local reduction φ from A to B with constants $c_1, c_2, c_3 > 0$. If there exist constants ϵ and δ such that every ϵ-tester for A must have query complexity at least δn where n is the size of an instance of A, then every $\frac{\epsilon}{c_1}$-tester for B must have query complexity at least $\frac{\delta n'}{c_2 c_3}$, where n' is the size of an instance of B.*

Proof. From (1) and (2), we can decide whether X is in P_0 or in P_ϵ by deciding whether $\varphi(X)$ is in Q_0 or Q_{ϵ/c_1}. Suppose that there is an $\frac{\epsilon}{c_1}$-tester for B with query complexity at most $f(n')$. Since one query to $\varphi(X)$ is simulated by c_2 queries to X, we can decide whether X is in P_0 or in P_ϵ by at most $c_2 f(n')$ queries. From the assumption of the hardness of testing A, $c_2 f(n') \geq \delta n$ must hold. From $n' \leq c_3 n$, we obtain that $f(n') \geq \frac{\delta n}{c_2} \geq \frac{\delta n'}{c_2 c_3}$. □

Lemma 2.4. *Let A and B be optimization problems. Suppose that there is a strong L-reduction φ from A to B with constants $c_1, c_2, c_3 > 0$. If there exist ϵ and δ such that every ϵ-approximator for A must have query complexity at least δn where n is the size of an instance of A, then every $\frac{\epsilon}{c_1 c_2}$-approximator for B must have query complexity at least $\frac{\delta n'}{c_3 c_4}$ where n' is the size of an instance of B.*

Proof. Suppose that there is an $\frac{\epsilon}{c_1 c_2}$-approximator for B with query complexity $f(n')$. We convert the solution x_B of $\varphi(X)$ obtained by this algorithm to the solution x_A of X so that $|OPT_A(X) - g_A(x_A)| \leq c_2|OPT_B(\varphi(X)) - g_B(x_B)| \leq \frac{\epsilon}{c_1} OPT_B(\varphi(X)) \leq \epsilon OPT_A(X)$. By considering that one query to $\varphi(X)$ is simulated by c_3 queries to X, it follows that we can approximate $OPT_A(X)$ within ϵ error with at most $c_3 f(n')$ queries. From the assumption of the hardness of approximating A, $c_3 f(n') \geq \delta n$ must hold. From $n' \leq c_4 n$, we obtain that $f(n') \geq \frac{\delta n}{c_3} \geq \frac{\delta n'}{c_3 c_4}$. □

A lower bound of the query complexity of testing 3SAT-d and approximating Max-3SAT-d is already known.

Theorem 2.5. *[1] For every real number $\alpha > 0$ there is a constant d such that every $(\frac{1}{8} - \alpha)$-tester for 3SAT-d must have linear query complexity.* □

Theorem 2.6. *[1] For every real number α, there is a constant d such that every $(\frac{1}{7} - \alpha)$-approximator for Max-3SAT-d must have linear query complexity.* □

3. A Linear Lower Bound of Testing 3-Edge-Colorability

In this section, we show a linear lower bound of testing 3-edge-colorability.

We use a slight modification of the reduction introduced by [6] from 3SAT to 3-edge-colorability, and call this reduction φ_{EC}. The original reduction aims at creating a 3-regular graph. Since we do not need to create such a graph, the reduction described here is simpler. The reduction φ_{EC} creates a graph $G = \varphi_{EC}(X)$ with maximum degree at most 3 (i.e., an instance of 3EC-3) from an instance X of 3SAT-d. In the graph G, values (true or false) are conveyed by pairs of edges. In a 3-edge-coloring of G, such a pair of edges represents true (resp., false) if the edges have the same color (resp., different colors),

A gadget shown in Fig. 1 is an inverter. If an inverter is 3-edge-colored, it is easily checked that the one of pairs of edges a, b or c, d must have the same color and remaining 3 edges c, d, e or a, b, e must have different colors. Thus, it can be regarded that an inverter takes a value as an input from a pair of edges a, b, negates it and emits it as an output to a pair of edges c, d.

The value of a variable in X is represented by a variable setter. An example is depicted in the left side of Fig. 2, which is constructed for the case $d = 4$. In general, we construct a variable setter by cyclically placing d pairs of inverters. Thus, a variable setter has $2d$ inverters and d outputs. We can verify that all the values (true or false) of outputs of

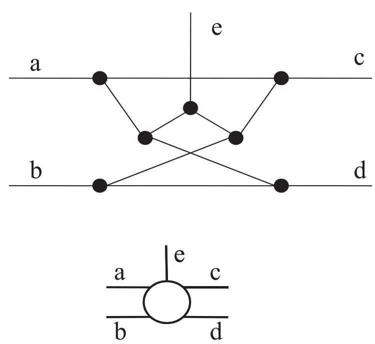

Fig. 1 An inverter and its symbolic representation.

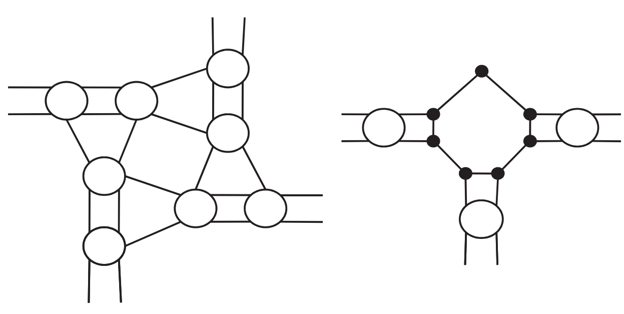

Fig. 2 Left: A variable setter with 8 inverters and 4 outputs. Right: A satisfaction tester.

a variable setter are the same if this component is 3-edge-colored. Each output will be connected to a component that represents a clause where the variable occurs.

The satisfiability of each clause in X is tested by a satisfaction tester depicted in the right side of Fig. 2. This component can be 3-edge-colored if and only if at least one of the values of the inputs is true.

G is constructed from X as follows. For each variable u_i we create a variable setter U_i with $2d$ inverters and d outputs, and for each clause c_j we create a satisfaction tester C_j. If the kth clause including u_i is c_j, then we connect the kth output of U_i to the input of C_j if u_i appears as a positive literal in c_j and insert an inverter between the kth output of U_i and the input of C_j if u_i appears as a negative literal in c_j. When there are edges unaccounted for, we just remove them.

Lemma 3.1. *Let X be an instance of* 3SAT-*d with n variables and n' be the number of vertices in $G = \varphi_{EC}(X)$. Then* $n' < \frac{91dn}{3}$.

Proof. For each variable in X, there is a variable setter with $2d$ inverters. For each clause in X, there is a satisfaction-tester with 3 inverters and 7 vertices. For each occurrence of a negative literal, there is one more inverter. Since one inverter have seven vertices, there are at most $(7 \cdot 2d)n + (7 \cdot 3 + 7)\frac{dn}{3} + 7 \cdot dn = \frac{91dn}{3}$ vertices in total. □

Lemma 3.2. *The reduction φ_{EC} is a strong gap-preserving local reduction from* 3SAT-*d to* 3EC-3.

Proof. The conditions (1) and (3) of a strong gap-preserving local reduction obviously hold. The condition (4) holds from Lemma 3.1. We show that the condition (2) holds in the following.

Let X be an ϵ-far instance of 3SAT-d and $G = \varphi_{EC}(X)$. Let $\frac{3\epsilon' n'}{2}$ be the minimum number of edges to be removed in order to make G 3-edge-colorable where n' is the number of vertices in G. Let G' be the resulting graph obtained by removing such edges. Note that adding edges is meaningless when considering the minimum number of edge modifications for 3-edge-colorability.

We define the territory of a clause c_j in X as variable-setters that represent variables occurring in c_j, a satisfaction-tester that represents c_j, and inverters inserted between those variable-setters and the satisfaction-tester. We call a clause c_j alive if no edge deletion occurred at the territory of c_j. Otherwise, we call the clause dead. Removing an edge of a variable-setter makes at most d clauses dead. Removing an edge of a satisfaction-tester makes at most one clause dead. Removing an edge of an inverter makes at most one clause dead. Thus, at most $\frac{3\epsilon' dn'}{2}$ clauses are turned to be dead in total by removing $\frac{3\epsilon' n'}{2}$ edges.

Let H' be a subgraph of G' induced by the territories of living clauses. If $\frac{3\epsilon' dn'}{2} < \frac{\epsilon dn}{3}$, H' is not 3-edge-colorable, since H' equals $\varphi_{EC}(X')$ where X' is a CNF such that less than $\frac{\epsilon dn}{3}$ clauses are removed from X. Since a graph is not 3-edge-colorable when a subgraph is not 3-edge-colorable, G'

is not 3-edge-colorable. It contradicts the assumption. Thus, $\frac{3\epsilon' dn'}{2} \geq \frac{\epsilon dn}{3}$ holds. From Lemma 3.1, $\epsilon' \geq \frac{2\epsilon dn}{9dn'} \geq \frac{2\epsilon}{273d}$. □

From Lemmas 2.3 and 3.2, and Theorem 2.5, we obtain the next.

Theorem 3.3. *There exist constants $\epsilon < 1$ and $d \geq 3$ such that every ϵ-tester for 3EC-d must have liner query complexity.* □

4. A Linear Lower Bound of Testing Hamiltonian Path/Cycle

In this section, we show a linear lower bound of the query complexity of testing that a directed or undirected graph has a Hamiltonian path/cycle.

First, we consider DHP. We use the reduction from 3SAT to DHP given by [13], and call the reduction φ_{DHP}. Let X be an instance of 3SAT-d and $G = \varphi_{DHP}(X)$ be the graph created by this reduction. In G, a variable u_i in X is represented by a rhombus-shaped component U_i depicted in the left of Fig. 3. We call this component a *variable setter*. A variable setter has an entrance vertex in the upper side and an exit vertex in the bottom side and $3d+3$ auxiliary vertices in the middle. A proper Hamiltonian path is supposed to enter from the entrance vertex and pass through the middle vertices and exit from the exit vertex. If the Hamiltonian path passes the middle vertices from left to right (resp., right to left), then it is regarded as representing true (resp., false). Each clause that includes u_i has two consecutive vertices within the middle vertices as its territory. For each $i (1 \leq i \leq n-1)$, the exit vertex of U_i and the entrance vertex of U_{i+1} is connected by an edge.

Each clause c_j has a corresponding *clause vertex* v_j. Assume that c_j is the kth clause including u_i. If it is a positive literal, then we connect an edge from the $(3k)$th vertex of the middle vertices of U_i to v_j and connect an edge from v_j to the $(3k+1)$th vertex of them. Otherwise, we connect an edge from the $(3k+1)$th vertex of the middle vertices of U_i to v_j and connect an edge from v_j to the $(3k)$th vertex of them. The right of Fig. 3 shows a part of a graph reduced from $(u_1 \vee u_3 \vee u_4) \wedge (\bar{u_1} \vee \bar{u_2} \vee u_3) \wedge (u_2 \vee u_4 \vee u_5)$.

Suppose that there is a variable assignment that satisfies X. Then, we can make a Hamiltonian path of G as follows. The path starts at the entrance vertex of U_1. If u_1 is true on the assignment, it passes through the middle vertices from left to right. If u_i is false, it passes through the middle vertices from right to left. After that, it goes to the exit vertex of U_1 and enter the entrance vertex of U_2. Repeat this process until it reaches the exit vertex of U_n. Every v_j can be passed while the path goes through the middle vertices of some U_i such that u_i appears in c_j.

We can see that if there is no assignment that satisfies X, then no Hamiltonian path exists in G. See [13] for detailed discussion.

Lemma 4.1. *Let X be an instance of 3SAT-d with n variables and n' and d' be the number of vertices and the maximum degree of $G = \varphi_{DHP}(X)$, respectively. Then, $n' < \frac{(10d+5)n}{3}$ and $d' \leq d$.*

Proof. There are $3d+5$ vertices for each variable setter. Since there are at most $\frac{dn}{3}$ clauses in X, there exist at most $\frac{dn}{3}$ vertices representing clauses in G. Thus, $n' \leq n \cdot (3d+5) + \frac{dn}{3} = \frac{(10d+15)n}{3}$ holds. The maximum degree of a vertex in a variable setter is at most 3. The maximum degree of a vertex representing a clause is at most d. Thus, $d' \leq d$ holds. □

Lemma 4.2. *The reduction φ_{DHP} is a strong gap-preserving local reduction from 3SAT-d to DHP-d.*

Proof. The conditions (1) and (4) of a strong gap-preserving local reduction are verified by the above discussion and Lemma 4.1. Also, the condition (3) is easily verified. We show that the condition (2) holds in the following.

Let X be an instance of 3SAT-d and $G = \varphi_{DHP}(X)$. Let n' and d' denote the number of vertices of G and the maximum degree of G, respectively. Let $\epsilon' d' n'$ be the minimum number of edges to be added or removed in order to make G having a Hamiltonian path and let the resulting graph be G'.

We define a territory of a clause c_j in X as v_j and variable-setters for variables occurring in c_j. We call a clause c_j alive if no edge modification occurred at the territory of c_j. Otherwise, we call the clause dead. Modifying

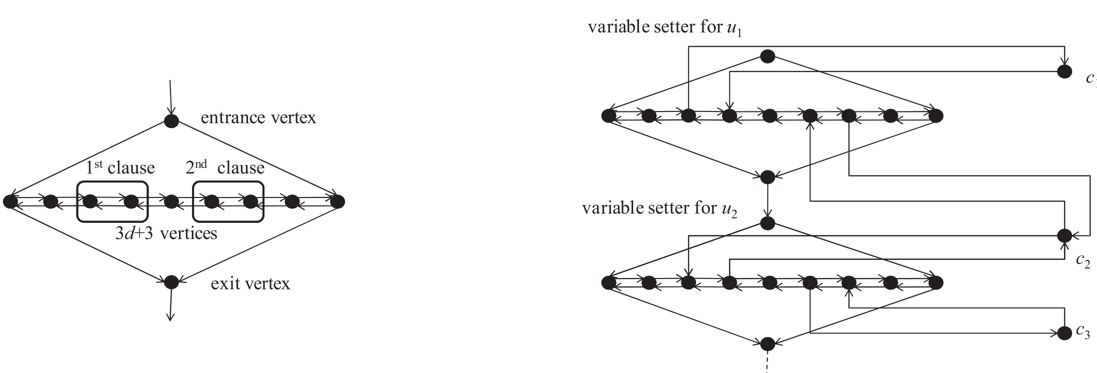

Fig. 3 Left: A variable setter with $d = 2$. Right: A part of a graph reduced from $(u_1 \vee u_3 \vee u_4) \wedge (\bar{u_1} \vee \bar{u_2} \vee u_3) \wedge (u_2 \vee u_4 \vee u_5)$.

an edge of a variable-setter makes at most d clauses dead. Modifying an edge of a clause vertex makes at most one clause dead. Since one edge modification causes at most two components to be changed, by modifying $\epsilon' d' n'$ edges, at most $2\epsilon' dd' n'$ clauses are turned to be dead.

We show that a Hamiltonian path in G' must enter a living variable setter from its entrance vertex and exit from its exit vertex, i.e., cannot go to or come from other variable setters via clause vertices. Suppose there exists a Hamiltonian path P such that P goes to a clause vertex via a vertex v in the middle of a variable setter U and after that P goes to another variable setter U' without returning to U. Then, one of adjacent vertices of v in the middle of U cannot be passed by P anymore since there is just one vertex adjacent to v which is not passed by P. It contradicts that P is a Hamiltonian path. Thus, if we ignore the route while passing the territory of dead clauses, the Hamiltonian path should form a Hamiltonian path of a graph reduced by φ_{DHP} from a 3SAT-d instance created by collecting living clauses.

If $2\epsilon' dd' n' < \frac{\epsilon dn}{3}$ holds, there is no such Hamiltonian path in the reduced graph and contradicts that G' has a Hamiltonian path. Thus, $2\epsilon' dd' n' \geq \frac{\epsilon dn}{3}$. From Lemma 4.1, it follows that $\epsilon' \geq \frac{\epsilon n}{6d'n'} \geq \frac{\epsilon}{20d+30}$. □

From Lemmas 2.3 and 4.2, and Theorem 2.5, we obtain the next theorem.

Theorem 4.3. *There exist constants $\epsilon < 1$ and d such that every ϵ-tester for DHP-d must have linear query complexity.* □

We can make a reduction φ_{DHC} for DHC-d from φ_{DHP} by connecting exit vertex of U_n and entrance vertex of U_1. The rest of the proof is the same, and we obtain the following result for DHC-d.

Theorem 4.4. *There exist constants $\epsilon < 1$ and d such that every ϵ-tester for DHC-d must have linear query complexity.* □

To get an instance of UHP-d and UHC-d from 3SAT-d, we replace each edge of $\varphi_{DHP}(X)$ and $\varphi_{DHC}(X)$ by two undirected edges and one vertex that connects them. As long as we consider Hamiltonian paths and Hamiltonian cycles, this replacement plays the same role. Thus, this new reductions from 3SAT-d to UHP-d and UHC-d are also strong gap-preserving local reductions.

Theorem 4.5. *There exist constants $\epsilon < 1$ and d such that every ϵ-tester for UHP-d and UHC-d must have linear query complexity.* □

5. A Linear Lower Bound of Testing and Approximating 3-Dimensional Matching

We argue about a linear lower bound of 3-dimensional matching problem.

Unlike the previous problems, we cannot use a standard polynomial reduction such as [8]. To prove that a reduction satisfies the condition (2) of a strong gap-preserving local reduction, we used the monotonicity of a property P, i.e., "if a subinstance of X does not satisfy P, then the whole X does not satisfy P either." 3DM-d, however, does not have monotonicity. Thus, we must employ another method. Fortunately, the L-reduction given by [7] used for showing inapproximability of maximum 3-dimensional matching is fit for our purpose. The reduction, called φ_{DM}, maps an instance of Max-3SAT-d to an instance of Max-3DM-3. Let X be an instance of Max-3SAT-d and $Y = (U, V, W, E) = \varphi_{DM}(X)$. Let $OPT(X), OPT(Y)$ denote the optimal value of X for Max-3SAT-d and the optimal value of Y for Max-3SAT-d, respectively. It is shown in [7] that

$$\mathrm{OPT}(Y) = 6Km - 3m + \mathrm{OPT}(X), \tag{1}$$

$$n' = 6Km - 2m, \tag{2}$$

where $K = 2^{\lfloor \log_2(\frac{3}{2}d+1) \rfloor}$, m be the number of clauses of X and $n' = \min(|U|, |V|, |W|)$.

First, we show that φ_{DM} is a strong L-reduction. The condition (1) holds since φ_{DM} is an L-reduction. Using (1), the condition (2) is also achieved. We can easily verify that φ_{DM} satisfies the conditions (3) and (4).

Lemma 5.1. *φ_{DM} is a strong L-reduction from Max-3SAT-d to Max-3DM-3.* □

From Lemmas 2.4 and 5.1, and Theorem 2.6, we get the following theorem.

Theorem 5.2. *There exist some constant $\epsilon < 1$ and d such that every ϵ-approximator for Max-3DM-3 must have linear query complexity.* □

Next, we show that φ_{DM} is also a strong gap-preserving local reduction.

Theorem 5.3. *The reduction φ_{DM} is a strong gap-preserving local reduction from 3SAT-d to 3DM-3.*

Proof. The conditions (3) and (4) are the same as those of a strong L-reduction.

Let X be an instance of 3SAT-d and $Y = \varphi_{DM}(X)$. Suppose that X is satisfiable, i.e., $\mathrm{OPT}(X) = m$. By (1) and (2), $OPT(Y) = 6Km - 3m + m = n'$ holds. Thus, the condition (1) of the gap-preserving local reduction follows.

Suppose that X is ϵ-far, i.e., $\mathrm{OPT}(X) \leq (1-\epsilon)m$. By (1), $OPT(Y) \leq 6Km - 3m + (1-\epsilon)m = n' - \epsilon m$ holds. We modify the minimum number of edges of Y so that the resulting instance Y' has a 3-dimensional matching M' with size n'. Let M be edges of Y contained in M'. Since $|M| \leq n' - \epsilon m$ and (2), we must modify at least $n' - (n' - \epsilon m) = \epsilon m = \frac{\epsilon}{18K-6} 3n'$ edges to transform Y into Y'. Hence Y is an $\frac{\epsilon}{18K-6}$-far instance of 3DM-d. Thus, the condition by (2) of the gap-preserving local reduction follows. □

From Lemmas 2.3 and 5.3, and Theorem 2.5, we get the following theorem.

Theorem 5.4. *There exist some constant $\epsilon < 1$ and d such that every ϵ-tester for 3DM-3 must have linear query complexity.* □

6. A Linear Lower Bound of Testing NP-Complete Generalized Satisfiability Problem

A logical relation is defined as a non-empty subset of $\{0,1\}^d$ for some $d \geq 1$. Let $S = \{R_1, R_2, \ldots, R_m\}$ be a finite set of logical relations. An S-clause is a clause of the form $r_i(v_1, v_2, \ldots, v_d)$ where v_i is a variable and r_i is a relation symbol representing R_i. S-satisfiability, denoted by Sat(S), is the problem of deciding whether a given conjunction of S-clauses is satisfiable. Let A be a formula. Var(A) denotes the set of variables occurring in A. Denote by $\mathfrak{S}(A)$ the set of all assignments $s : \mathrm{Var}(A) \to \{0,1\}$ that satisfy A. Two formulas A and B are logically equivalent if $\mathrm{Var}(A) = \mathrm{Var}(B)$ and $\mathfrak{S}(A) = \mathfrak{S}(B)$

Schaefer's dichotomy theorem [12] states a necessary and sufficient condition for a set of relations S under which Sat(S) is polynomially-solvable assuming P ≠ NP.

Theorem 6.1. *(Schaefer's dichotomy theorem) Let S be a set of relations. If S satisfies one of the conditions (a)-(f), then Sat(S) is in P. Otherwise, Sat(S) is NP-complete.*

- *every relation in S is satisfied if all variables are assigned* 0 *(*0*-valid).*
- *every relation in S is satisfied if all variables are assigned* 1 *(*1*-valid).*
- *every relation in S is logically equivalent to a CNF with at most two literals (bijunctive).*
- *every relation in S is logically equivalent to a CNF with at most one positive literal (weakly negative).*
- *every relation in S is logically equivalent to a CNF with at most one negative literal (weakly positive).*
- *every relation in S is logically equivalent to a system of linear equations over the two-element field* $\{0,1\}$ *(affine).* □

Schaefer-Sat denotes Sat(S) such that S satisfies *none* of the conditions of Theorem 6.1 or in other words is NP-complete. Schaefer-3SAT is a Schaefer-Sat in which each clause contains exactly three variables.

To show a lower bound on Schaefer-3SAT, We use a reduction from 3SAT to Schaefer-3SAT given by [12]. We call the reduction φ_{ssat}. Due to the space limit, we cannot describe the whole process of φ_{ssat}. The most important part of the reduction is that it has "locality." Let X be an instance of 3SAT and $Y = \varphi_{ssat}(X)$. A clause of X is converted to a clause of Y without any information of other clauses of X. Thus, if the number of occurrences of a variable in X is at most d, then the number of occurrences of a variable in Y is also bounded by some constant d'. Let n and n' be the number of variables in X and Y, respectively. Then, by the same arguments, there is some constant c such that n' is bounded by cn.

Lemma 6.2. *For any d there exists some constant d' such that φ_{ssat} is a strong gap-preserving local reduction from* 3*SAT-d to Schaefer-*3*SAT-d'*

Proof. The conditions (1), (3) and (4) are verified by the locality of φ_{ssat}. We show that the condition (2) actually holds as follows.

Let X be an ϵ-far instance of 3SAT-d and $Y = \varphi_{ssat}(X)$. Let $\frac{\epsilon' d' n'}{3}$ be the minimum number of clauses to be removed in order to make Y satisfiable where n' and d' is the number of variables and the upper bound of the occurrences of a variable of Y. Let Y' be the resulting instance obtained by removing such clauses.

We define the territory of a clause c_j in X as clauses in Y that is reduced from c_j. We call a clause c_j is alive if none of clauses of the territory of c_j is removed. Otherwise, we call the clause dead.

Since removing a clause of Y makes at most 1 clause of X dead, at most $\frac{\epsilon' d' n'}{3}$ clauses in X are turned to be dead in total by removing $\frac{\epsilon' d' n'}{3}$ clauses of Y. If $\frac{\epsilon' d' n'}{3} < \frac{\epsilon d n}{3}$ holds, Y' contains clauses that are reduced from an instance created by removing at most $\frac{\epsilon d n}{3}$ clauses from X. It contradicts the satisfiability of Y'. Thus, $\frac{\epsilon' d' n'}{3} \geq \frac{\epsilon d n}{3}$ must hold.

Since there exist some constants c and c_4 such that $d' \leq cd$ and $n' \leq c_4 n$, there exists some constant c_2 such that $\epsilon' \geq \frac{\epsilon}{c_2}$ holds. □

From Lemmas 2.3 and 6.2, and Theorem 2.5, we obtain the following theorem.

Theorem 6.3. *There exist some constants $\epsilon < 1$ and d such that every ϵ-tester for Schaefer-*3*SAT-d must have linear query complexity.* □

7. Concluding Remarks

We introduce two reductions to show linear lower bounds on the query complexity of testing algorithms for various NP-complete problems. One is a strong gap-preserving local reduction used with the monotonicity of the problem itself (3EC-d, DHP-d and Schaefer-3SAT-d). The other is a strong L-reduction (3DM-d). It might be interesting to consider that these techniques can be applied to a wider class of problems, e.g., NP-hard problems with monotonicity or MAX SNP-hard problems.

Another problem we want to mention is how large the query complexity of testing non-Schaefer Sat(S) is. If all relations in S are 0-valid or 1-valid, it is trivial since there is no ϵ-far instance. Also, we can show that there is a linear lower bound when S is affine since it is equivalent to E3LIN-2, which requires linear query [1]. Testing bipartiteness of a graph with a bounded degree takes at least $\Omega(\sqrt{n})$ queries [5]. Since it can be reduced to SAT(S) such that S is bijunctive, testing such SAT(S) requires $\Omega(\sqrt{n})$ queries.

The dichotomy theorem for maximum generalized satisfiability problems (Max-SAT(S)) is already known [2]. It gives a necessary and sufficient condition under which Max-SAT(S) is polynomially-solvable assuming P ≠ NP. It is possible to show linear lower bounds of approximations for such problems using strong L-reductions.

References

[1] A. Bogdanov, K. Obata, and L. Trevisan, "A lower bound for testing 3-colorability in bounded-degree graphs," FOCS '02: Proc. 43rd Symposium on Foundations of Computer Science, pp.93–102, IEEE Computer Society, 2002.
[2] N. Creignou, "A dichotomy theorem for maximum generalized satisfiability problems," J. Comput. Syst. Sci., vol.51, no.3, pp.511–522, 1995.
[3] O. Goldreich, S. Goldwasser, and D. Ron, "Property testing and its connection to learning and approximation," J. ACM, vol.45, no.4, pp.653–750, 1998.
[4] O. Goldreich, M. Krivelevich, I. Newman, and E. Rozenberg, "Hierarchy theorems for property testing," Electronic Colloquium on Computational Complexity (ECCC), vol.15, no.097, 2008.
[5] O. Goldreich and D. Ron, "Property testing in bounded degree graphs," STOC '97: Proc. 29th Annual ACM Symposium on Theory of Computing, pp.406–415, ACM, 1997.
[6] I. Holyer, "The np-completeness of edge-coloring," SIAM J. Comput., vol.10, no.4, pp.718–720, 1981.
[7] V. Kann, "Maximum bounded 3-dimensional matching is max snp-complete," Inf. Process. Lett., vol.37, no.1, pp.27–35, 1991.
[8] R.M. Karp, "Reducibility among combinatorial problems," in Complexity of Computer Computations, ed. R.E. Miller and J.W. Thatcher, pp.85–103, Plenum Press, 1972.
[9] C. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes," STOC '88: Proc. 20th annual ACM symposium on Theory of computing, pp.229–234, ACM, 1988.
[10] M. Parnas and D. Ron, "Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms," Theor. Comput. Sci., vol.381, no.1-3, pp.183–196, 2007.
[11] R. Rubinfeld and M. Sudan, "Robust characterizations of polynomials with applications to program testing," SIAM J. Comput., vol.25, no.2, pp.252–271, 1996.
[12] T.J. Schaefer, "The complexity of satisfiability problems," STOC '78: Proc. 10th annual ACM symposium on Theory of computing, pp.216–226, ACM, 1978.
[13] M. Sipser, Introduction to the Theory of Computation, International Thomson Publishing, 1996.
[14] Y. Yoshida, M. Yamamoto, and H. Ito, "An improved constant-time approximation algorithm for maximum matchings," STOC '09: Proc. 41st ACM Symposium on Theory of Computing, pp.225–234, ACM, 2009.

Hiro Ito received the B.E., M.E., and Dr. of Engineering degrees in the Department of Applied Mathematics and Physics from the Faculty of Engineering, Kyoto University in 1985, 1987, and 1995, respectively. From 1987 to 1996 and from 1996 to 2001 he was a member of NTT Laboratories and Toyohashi University of Technology, respectively. Since 2001, he has been an associate professor in the Department of Communications and Computer Engineering, Graduate School of Informatics at Kyoto University. He has been engaged in research on discrete algorithms mainly on graphs and networks, discrete mathematics, and recreational mathematics. Dr. Ito is a member of the Operations Research Society of Japan, the Information Processing Society of Japan, and the European Association for Theoretical Computer Science.

Yuichi Yoshida is a student of Graduate School of Informatics, Kyoto University. He cofounded Preferred Infrastructure Inc. in 2006. He received B.Eng. and M.Info. degrees from Kyoto University in 2007 and 2009, respectively. His main research interests include approximation algorithms, randomized algorithms and property testing.