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 \( \frac{1}{2} \). 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, \( \frac{1}{2} \)). 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 \( L \)-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.
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 \( \mathbb{F}_2 \) 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 \( O(n^c) \) but cannot be testable in query complexity \( o(n^c) \) for any constant \( c > 0 \) [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 \( \Omega(n^{\frac{1}{2}}) \) queries even if the approximation ratio is \( 2 - \frac{\gamma}{2} \) 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 \( X \) 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 \( X \in X \) is called \( \epsilon \)-far from (satisfying) a property \( P \) if an \( \epsilon \)-fraction of \( f \) should be changed to make \( X \) having the property \( P \). The concrete definition of \( \epsilon \)-farness depends on the concerned combinatorial objects.

Let \( A \) be a decision problem that decides whether an object \( X \in X \) satisfies a property \( P \) or not. A testing algorithm with an error parameter \( \epsilon > 0 \) (or \( \epsilon \)-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 \( \epsilon \)-far from \( P \), rejects \( X \) with probability at least \( \frac{2}{3} \).

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

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 \( f_U : U \times [d] \rightarrow C \cup \{\emptyset\} \) and \( f_C : C \times [3] \rightarrow U \) where \([n] = \{1, 2, \ldots, n\} \). \( f_U(u,i) \) represents the \( i \)th clause at which \( u \) occurs. If no such clause exists, the value of it is \( \emptyset \). \( f_C(c,i) \) represents the \( i \)th 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 \( \epsilon \)-far from being satisfiable if at least \( \frac{dn}{2} \) 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 \( f : V \times [d] \rightarrow V \cup \{\emptyset\} \). \( f(v,i) \) represents the \( i \)th vertex incident to \( v \). If no such vertex exists, the value of it is \( \emptyset \). \( G \) is called \( \epsilon \)-far from a property \( P \) if at least \( \frac{dn}{2} \) 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 \( \epsilon \)-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 in-degrees and out-degrees are bounded by \( d \)) and \( n \) be the number of vertices in \( G \). \( G \) is represented by a function \( f_{in} : V \times [d] \rightarrow V \cup \{\emptyset\} \) and \( f_{out} : V \times [d] \rightarrow V \cup \{\emptyset\} \).
defines a set of instances that satisfy \( \varphi \) with \( |\varphi(X)| \leq c_4|X| \) 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 \( \varphi \) from \( A \) to \( B \) with constants \( c_1, c_2, c_3 > 0 \). If there exist constants \( \epsilon \) and \( \delta \) such that every \( \epsilon \)-approximator for \( A \) must have approximation complexity at least \( \epsilon n \) where \( n \) is the size of an instance of \( A \), then every \( \frac{1}{c_1} \)-approximator for \( B \) must have approximation complexity at least \( \frac{\alpha}{c_2} \), 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 \( P_e \) by deciding whether \( \varphi(X) \) is in \( Q_0 \) or \( Q_{c_1} \). Suppose that there is an \( \frac{1}{c_1} \)-approximator for \( B \) with approximation 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 \( P_e \) 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 \( f(n') \geq \frac{\alpha}{c_2} \geq \frac{\alpha c_3}{c_4} \).

**Lemma 2.4.** Let \( A \) and \( B \) be optimization problems. Suppose that there is a strong \( L \)-reduction \( \varphi \) from \( A \) to \( B \) with constants \( c_1, c_2, c_3 > 0 \). If there exist \( \epsilon \) and \( \delta \) such that every \( \epsilon \)-approximator for \( A \) must have approximation complexity at least \( \delta n \) where \( n \) is the size of an instance of \( A \), then every \( \frac{1}{c_1} \)-approximator for \( B \) must have approximation complexity at least \( \frac{\alpha}{c_2} \)

**Proof.** Suppose that there is an \( \frac{1}{c_1} \)-approximator for \( B \) with approximation 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 \( |OPT_B(\varphi(X)) - g_B(x_B)| \leq c_2 |OPT_B(\varphi(X)) - g_B(x_B)| \leq \frac{1}{c_1} OPT_B(\varphi(X)) \leq OPT_B(X) \). By considering that one query to \( \varphi(X) \) is simulated by \( c_2 \) queries to \( X \), it follows that we can approximate \( OPT_B(X) \) within \( \epsilon \) error with at most \( c_2 f(n') \) queries. From the assumption of the hardness of approximating \( A \), \( c_2 f(n') \geq \delta n \) must hold. From \( n' \leq c_3 n \), we obtain that \( f(n') \geq \frac{\delta}{c_2} \geq \frac{\alpha c_3}{c_4} \).

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 \( (\frac{1}{3} - \alpha) \)-approximator for 3SAT-\( d \) must have linear query complexity.

**Theorem 2.6.** [1] For every real number \( \alpha \), there is a constant \( d \) such that every \( (\frac{1}{3} - \alpha) \)-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 $\varphi_{3EC}$. 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 $\varphi_{EC}$ creates a graph $G = \varphi_{3EC}(X)$ 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 2$d$ inverters and $d$ outputs. We can verify that all the values (true or false) of outputs of 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 $ui$, we create a variable-setter $Ui$ with 2$d$ inverters and $d$ outputs, and for each clause $cj$ we create a satisfaction tester $Cj$. If the $k$th clause including $ui$ is $cj$, then we connect the $k$th output of $Ui$ to the input of $Cj$ if $ui$ appears as a positive literal in $cj$ and insert an inverter between the $k$th output of $Ui$ and the input of $Cj$ if $ui$ appears as a negative literal in $cj$. 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 $G = \varphi_{3EC}(X)$. Then $n' < \frac{9d n}{3}$.

**Proof.** For each variable in $X$, there is a variable setter with 2$d$ 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 $(7 \cdot 2d)n + (7 \cdot 3 + 7)d n' = \frac{9d n}{3}$ vertices in total.

**Lemma 3.2.** The reduction $\varphi_{3EC}$ 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 $\epsilon$-far instance of 3SAT-$d$ and $G = \varphi_{3EC}(X)$. Let $\frac{3e'n'}{2}$ 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 $cj$ in $X$ as variable-setters that represent variables occurring in $cj$, a satisfaction-tester that represents $cj$, and inverters inserted between those variable-setters and the satisfaction-tester. We call a clause $cj$ alive if no edge deletion occurred at the territory of $cj$. 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 $\frac{3en'}{2}$ clauses are turned to be dead in total by removing $\frac{3e'n'}{2}$ edges.

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

![Fig. 1 An inverter and its symbolic representation.](image1)

![Fig. 2 Left: A variable setter with 8 inverters and 4 outputs. Right: A satisfaction tester.](image2)
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 $u_i$ 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 $U_2$. 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, $n' < \left(\frac{10d+5m}{3}\right)$ and $d' \leq d$.

**Proof.** There are $3d + 5$ vertices for each variable setter. Since there are at most $\frac{d}{3}$ clauses in $X$, there exist at most $\frac{dn}{3}$ vertices representing clauses in $G$. Thus, $n' \leq n \cdot (3d + 5) + \frac{d}{3} = \left(\frac{10d+5m}{3}\right)$ 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 $\varphi_{DHP}$ 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
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 $e'd'n'$ edges, at most $2e'd'd'n'$ 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 $\varphi_{\text{DHP}}$ from a 3SAT-$d$ instance created by collecting living clauses.

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

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 $\epsilon$-tester for DHP-d must have linear query complexity.

We can make a reduction $\varphi_{\text{DHC}}$ for DHC-d from $\varphi_{\text{DHP}}$ by connecting exit vertex of $U_a$ 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 $\epsilon$-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 $\varphi_{\text{DHP}}(X)$ and $\varphi_{\text{DHC}}(X)$ 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 $\epsilon$-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 $\varphi_{\text{DM}}$, maps an instance of Max-3SAT-d to an instance of Max-3DM-3. Let $X$ be an instance of Max-3SAT-d and $Y = (U, V, W, E) = \varphi_{\text{DM}}(X)$. 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

\[
OPT(Y) = 6Km - 3m + OPT(X),
\]

\[
n' = 6Km - 2m,
\]

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

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

**Lemma 5.1.** $\varphi_{\text{DM}}$ is a strong L-reduction from Max-3SAT-d to Max-3DM-3.

From Lemmas 2.4 and 5.1, and Theorem 2.5, we get the following theorem.

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

Next, we show that $\varphi_{\text{DM}}$ is also a strong gap-preserving local reduction.

**Theorem 5.3.** The reduction $\varphi_{\text{DM}}$ 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 $Y = \varphi_{\text{DM}}(X)$. Suppose that $X$ is satisfiable, i.e., $OPT(X) = m$. 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 $\epsilon$-far, i.e., $OPT(X) \leq (1 - \epsilon)m$. 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 $|M| \leq n' - em$ and (2), we must modify at least $n' - (n' - em) = em = \frac{\epsilon}{18K - d}$ edges to transform $Y$ into $Y'$. Hence $Y'$ is an $\frac{\epsilon}{18K - d}$-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 $\epsilon$-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 \(\text{Sat}(S)\), is the problem of deciding whether a given conjunction of \(S\)-clauses is satisfiable. Let \(A\) be a formula. \(\text{Var}(A)\) denotes the set of variables occurring in \(A\). Denote by \(\overline{A}(A)\) the set of all assignments \(s: \text{Var}(A) \rightarrow [0, 1]\) that satisfy \(A\). Two formulas \(A\) and \(B\) are logically equivalent if \(\text{Var}(A) = \text{Var}(B)\) and \(\overline{A}(A) = \overline{A}(B)\).

**Schaefer’s dichotomy theorem**[12] states a necessary and sufficient condition for a set of relations \(S\) under which \(\text{Sat}(S)\) is polynomially-solvable assuming \(P \neq NP\).

**Theorem 6.1.** (Schaefer’s dichotomy theorem) Let \(S\) be a set of relations. If \(S\) satisfies one of the conditions (a)-(f), then \(\text{Sat}(S)\) is in \(P\). Otherwise, \(\text{Sat}(S)\) 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 (biffunctive).
- 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 \(\text{Sat}(S)\) 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 \(\varphi_{\text{sat}}\). Due to the space limit, we cannot describe the whole process of \(\varphi_{\text{sat}}\). The most important part of the reduction is that it has “locality.” Let \(X\) be an instance of 3SAT and \(Y = \varphi_{\text{sat}}(X)\). 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 \(\varphi_{\text{sat}}\) is a strong gap-preserving local reduction from 3SAT-\(d\) to Schaefer-3SAT-\(d'\).

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

Let \(X\) be an \(\epsilon\)-far instance of 3SAT-\(d\) and \(Y = \varphi_{\text{sat}}(X)\). Let \(\epsilon_{d'} = \frac{1}{3}\) 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 \(\frac{\epsilon_{d'}}{3}\) clauses of \(Y\). If \(\epsilon_{d'} < \frac{c}{3n}\) holds, \(Y'\) contains clauses that are reduced from an instance created by removing at most \(\frac{c}{3n}\) clauses from \(X\). This contradicts the satisfiability of \(Y'\). Thus, \(\epsilon_{d'} \geq \frac{c}{3n}\) must hold.

Since there exist some constants \(c\) and \(c_4\) such that \(d' \leq cd\) and \(n' \leq c_4n\), there exists some constant \(c_2\) such that \(\epsilon' \geq \frac{c_2}{c_4}\) holds. \(\Box\)

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 \(\epsilon\)-tester for Schaefer-3SAT-\(d\) must have linear query complexity. \(\Box\)

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 \(\epsilon\)-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 \(\Omega(n^{1/2})\) queries [5]. Since it can be reduced to SAT(S) such that S is bijunctive, testing such SAT(S) requires \(\Omega(n^{1/2})\) 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 MaxSAT(S) is polynomially-solvable assuming \(P \neq NP\). It is possible to show linear lower bounds of approximations for such problems using strong L-reductions.
References


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.

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.