Bridging a Small Gap in the Gap Amplification of Assignment Testers

Oded Goldreich Or Meir

August 17, 2019

Abstract

Irit Dinur's proof of the PCP theorem via gap amplification (*J. ACM*, Vol. 54 (3) and *ECCC* TR05-046) has an important extension to Assignment Testers (a.k.a PCPPs). This extension is based on a corresponding extension of the gap amplification theorem from PCPs to Assignment Testers (a.k.a PCPPs). Specifically, the latter extension states that the rejection probability of an Assignment Tester can be amplified by a constant factor, at the expense of increasing the output size of the Assignment Tester by a constant factor (while retaining the alphabet). We point out a gap in the proof of this extension, and show that this gap can be bridged. We stress that the gap refers to the amplification of Assignment Testers, and the underlying issue does not arise in the case of standard PCPs. Furthermore, it seems that the issue also does not arise with respect to the applications in Dinur's paper, but it may arise in other applications.

This note appeared as Comment Nr 3 on TR05-046 of *ECCC*. (The comment was posted in Oct. 2007.) The current revision is intentionally minimal, but does include some minor corrections and clarifications of the original note as well as some stylistic improvements. Section 4 is a digest that was added in the current revision.

1 Background

(We make references to specific items as numbered both in the journal version of Dinur's work [D07] and in the version posted on ECCC [D05], since both are cited in the literature.)

1.1 Assignment Testers and Gap Amplification

We begin by recalling the definition of Assignment Testers, as stated in [D05, D07] (following [DR06]), while commentging that this notion is closely related to the notion of PCPs of Proximity (as defined in [BGHSV04]). The current formulation of Assignment Testers refers to the notion of a constraint graph, which is a graph $G = (V, E)$ augmented with binary constraints that are associated with its edges. These constraints refer to an assignment α of values in Σ_0 to the vertices (i.e., $\alpha : V \to \Sigma_0$), and $\mathrm{UNSAT}_\alpha(G)$ denotes the fraction of edge-contraints that are violated by this assignment.

Definition 1 (Assignment Testers, See [D05, Definition 3.1] and [D07, Definition 2.8]): *An assignment-tester with alphabet Σ_0 and rejection probability $\varepsilon > 0$ is a polynomial-time transformation $\mathcal{P}$ that, on input a circuit Φ over Boolean variables X, outputs a constraint graph $G = \langle (V, E), \Sigma_0, \mathcal{C} \rangle$ such that $X \subseteq V$ and the following hold with respect to any assignment $a : X \to \{0,1\}$.*

- (Completeness): *If $a \in SAT(\Phi)$, then there exists $b : (V \backslash X) \to \Sigma_0$ such that $UNSAT_{a \cup b}(G) = 0$.*
- (Soundness): *If $a \notin SAT(\Phi)$, then for all $b : (V \backslash X) \to \Sigma_0$, $UNSAT_{a \cup b}(G) \geq \varepsilon \cdot \mathtt{dist}(a, SAT(\Phi))$.*[1]

The main technical result of [D05, D07] is a gap amplification theorem for PCPs. The following important extension of this theorem to Assignment Testers is also provided in [D05, D07]:

Theorem 2 (Gap Amplification for Assignment Testers, See [D05, Theorem 8.1] and [
There exists $t \in \mathbb{N}$ such that given an assignment-tester with constant-size alphabet Σ and rejection probability ε, one can construct an assignment-tester with the same alphabet such that

- *the rejection probability of the new assignment-tester is at least $\min(2\varepsilon, 1/t)$; and*
- *the output size of the new assignment-tester is at most a constant factor larger than the output size of the given assignment-tester.*

The size of a graph (e.g., an output of an assignment-tester) *is defined as the number of edges in it.*

[1] Indeed, $\mathrm{SAT}(\Phi)$ denotes the set of assignments that satisfy Φ, and $\mathtt{dist}(a, S)$ denotes the relative Hamming distance of the assignment a from the set S.

1.2 Overview of the proof of Theorem 2

The assignment tester of Theorem 2 is constructed in two steps: First, for a fixed constant $d \in \mathbb{N}$ and an arbitrary constant $t \in \mathbb{N}$ (to be determined later), an intermediate assignment tester with alphabet $\Sigma^{d^{t/2}}$ and rejection probability $p = \Omega(\min(\sqrt{t}\cdot\varepsilon, 1/t))$ is constructed. Then, a composition theorem of Dinur and Reingold [DR06] is applied to the intermediate assignment tester in order to reduce its alphabet's size, resulting in an assignment tester with alphabet Σ and rejection probability $\Omega(p) = \Omega(\min(\sqrt{t}\cdot\varepsilon, 1/t))$. The number t is then fixed to some sufficiently large natural number that yields the desired rejection probability.

The subject of this note is a gap in the first step of the foregoing construction; namely, the construction of the intermediate assignment tester. Specifically, we show that under certain circumstences, the intermediate assignment tester has output size that is quadratic in the output size of the input assignment tester, failing to establish Theorem 2. Such an increase in the output size can not be afforded by the applications of Theorem 2 presented in [D05] and [D07]. We comment that those circumstences do not seem to occur in the applications of Theorem 2 presented in of [D05]. In this note we show that the proof of Theorem 2 can be corrected so the theorem holds *under any circumstences*.

Outline of the construction of the intermediate assignment tester. Let Φ be a circuit over Boolean variables X.

1. First, the intermediate assignment tester runs the given assignment tester on input Φ, yielding a constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$.

   For any vertex $v \in V$, let $\deg_G(v)$ denote the degree of v in G.

2. Next, the intermediate assignment tester constructs the constraint graph $H = (\mathrm{prep}(G))^t$, where $\mathrm{prep}(G)$ is the graph in which every vertex v of G is replaced by a $\deg_G(v)$-vertex expander graph, denoted $[v]$, whose vertices represent "copies" of v and whose edges correspond to equality constraints. We denote the set of vertices of H by V_H. (In addition, a larger expander, with trivial constraints, is superimposed on V_H; but we ignore it here.)[2]

   We stress that the $X \not\subseteq V_H$, since each $x \in X \subseteq V$ was replaced by $[x]$.

[2] We also ignored the contraints placed on the edges of H, which are a key issue in [D05, D07]. We do so since our focus is on the added constraints of Step 3.

3. Finally, the intermediate assignment tester constructs and outputs a constraint graph H', whose set of vertices is $V_H \cup X$ and whose edges consist of the edges of H and of "consistency edges" that check consistency between X and V_H (i.e., between each $x \in X$ and each vertex in $[x]$). The edges are reweighted such that the latter consistency edges form half of the edges of H'.

   For every $v \in V_H \cup X$, let $\deg_{H'}(v)$ denote the degree of v in H'.

(We stress that the definition of the latter consistency edges, which are added in the last step, was not fully specified above. The question of how to actually define these edges is the issue that we address in this note.)

2 The gap

The gap in the proof is in the way the consistency edges between X and V_H are defined. Specifically, we show that if the graph G is highly non-regular, then the construction of H' may contain too many consistency edges. For simplicity, let us assume that $t = 1$, but note that the argument holds for any value of t. For $t = 1$, it holds that $H = \mathrm{prep}(G)$ and $V_H = \bigcup_{v \in V} [v]$, where $[v]$ is the set of vertices that represent "copies" of the vertex v of G.

The consistency edges as defined in [D05, D07]. The natural way to define the consistency edges, which is the way taken in [D05, D07], is based on the natural randomized testing procedure (to be described next). This procedure is given oracle access to an assignment $A : V_H \cup X \to \Sigma$ to H', and is allowed to make two queries to A, which it selects uniformly at random (see below). The procedure then decides whether to accept or reject A. Assuming that $t = 1$, the aforementioned procedure is as follows:

1. Select $x \in X$ uniformly at random.
2. Select $z \in [x]$ uniformly at random (recall that $[x]$ is the set of vertices in H that are copies of x).
3. Accept if and only if $A(x) = A(z)$.

The consistency edges are defined using the procedure as follows: For every possible outcome of the coin tosses ω, let v_1^ω and v_2^ω denote the vertices that the procedure queries on coin tosses ω. Then, a consistency edge is placed between v_1^ω and v_2^ω, and this edge accepts an assignment $A : V_H \cup X \to \Sigma$ if and only if the procedure accepts on coin tosses ω when given oracle

access to A (i.e., $A(v_1^\omega) = A(v_2^\omega)$). Note that for every $x \in X$, it holds that $\deg_{H'}(x)$ equals to the number of consistency edges connected to x using the foregoing procedure. The problem is now as follows:

- Since the procedure chooses $x \in X$ uniformly at random (at Step 1), every variable $x \in X$ must have the same degree in H'. That is, for every two variables $x, y \in X$, it holds that $\deg_{H'}(x) = \deg_{H'}(y)$.
- Since the procedure chooses $z \in [x]$ uniformly at random (at Step 2), every variable $x \in X$ must satisfy $\deg_{H'}(x) \geq |[x]| = \deg_G(x)$.
- Combining the previous two items, it follows that the degree of every variable $x \in X$ is at least $\max_{x \in X} \{\deg_G(x)\}$, and therefore the number of consistency edges added by the foregoing procedure is at least $|X| \cdot \max_{x \in X} \{\deg_G(x)\}$.

Now, suppose that $|X| = \Omega(\text{size}(G))$ and that there exists $x_0 \in X$ for which $\deg_G(x_0) = \Omega(\text{size}(G))$; this can be the case if G is highly non-regular. In such a case, the number of consistency edges that will be added in the construction of H' will be at least $|X| \cdot \deg_G(x_0) = \Omega(\text{size}(G)^2)$, and therefore we will have $\text{size}(H') = \Omega(\text{size}(G)^2)$, contradicting the claim of Theorem 2. Note that this problem does not occur if G is a regular graph (i.e., $\deg_G(x) = \deg_G(y)$ for every $x, y \in X$), since in such case we have that

$$N \stackrel{\text{def}}{=} |X| \cdot \max_{x \in X} \{\deg_G(x)\} = \sum_{x \in X} \deg_G(x) \leq \text{size}\,(G)$$

and therefore we will have $\text{size}(H') = 2 \cdot \max(\text{size}(H), N) = O(\text{size}(G))$, as required. Ditto if G is almost regular (i.e., $\deg_G(x) = \Theta(\deg_G(y))$ for every $x, y \in X$).

It seems that the assignment testers to which this construction is applied in [D05, D07] are regular, and in such a case the gap we discuss does not occur. However, envisioning possible applications in which the regularity condition does not hold or is hard to verify, we wish to establish the result also for such (general) cases.

3 Bridging the gap

We turn to describe how the gap can be bridged. In order to bridge the gap, we modify the foregoing randomized procedure as follows. For every $x \in X$, let $[x]'$ to be an arbitrary subset of $[x]$ that has size $\min(|[x]|, \text{size}(H)/|X|)$.

The modified procedure is the same as the original procedure, except for that in Step 2, it chooses z uniformly at random from *the set* $[x]'$ *instead of in* $[x]$. Observe that this modification indeed solves the problem, since now the degree of every variable $x \in X$ in H' is at most $\text{size}(H)/|X|$, and therefore the total number of consistency edges is at most $\text{size}(H) = O(\text{size}(G))$.

The reason that the modified procedure works is roughly as follows: Consider some assignment to $X \cup V_H$. Ideally, we would like that if a variable $x \in X$ is assigned a value that is inconsistent with most of $[x]$, then this variable violates $\Omega(1/|X|)$-fraction of the edges of H'. Suppose now that some variable $x \in X$ is assigned a value that is inconsistent with most of the vertices in $[x]$. Then, either that x is inconsistent with most of the set $[x]'$, or most of the set $[x]'$ is inconsistent with most of the set $[x]$. In the first case, at least $\Omega(1/|X|)$-fraction of the edges are violated, since the modified procedure chooses x with probability $1/|X|$ and then chooses with probability at least $1/2$ a vertex $z \in [x]'$ that is inconsistent with x. In the second case, the inconsistency of the two majorities yields sufficiently many violated edges by virture of the mixing property of the expander $[x]$. Details follow.

Indeed, the case where x is consistent with most of $[x]'$ is more problematic, since the procedure is likely to choose $z \in [x]'$ that is consistent with x, whereas this value is inconsistent with the majority in $[x]$. Indeed, this is possible only when $[x]' \neq [x]$, which in particular implies that $|[x]| > s \stackrel{\text{def}}{=} \text{size}(H)/|X| = |[x]'|$. Hence, there is an $(s/2)$-subset of $[x]$ (i.e., the majority in $[x]'$) that is inconsistent with most of $[x]$, and therefore by the mixing property of the expander $[x]$ at least $\Omega(s/2)$ inner edges of $[x]$ are violated. It follows that the fraction of violated edges that are incident at x is at least

$$\frac{\Omega(s)}{\text{size}(H')} = \frac{\Omega(s)}{O(\text{size}(H))} = \Omega(1/|X|)$$

as required. Below we give a rigorous proof of this argument.

The modified procedure for arbitrary t. We first describe the modified procedure for an arbitrary value of t (rather than just $t = 1$):

1. Select $x \in X$ uniformly at random.
2. Select $z \in [x]'$ uniformly at random (recall that $[x]'$ is an arbitrary s-subset of $[x]$, where $s = \min(|[x]|, \text{size}(H)/|X|)$).
3. Take a $\lfloor t/2 \rfloor$-step random walk in $\text{prep}(G)$ starting from z, and let w be the endpoint of the walk. Accept if and only if $A(w)_z = A(x)$.

Recall that $A : V_H \cup X \to \Sigma^{d^{t/2}} \cup \{0,1\}$ assigns each $v \in V_H$ a sequence of values, one per each vertex u at distance at most $t/2$ from v. Hence, $A(v)_u$ denotes the value that v attributes to u.

We now use the procedure to define the consistency edges as before, and then reweight the edges of H' such that the consistency edges form half of the edges of H'. Observe that this modification *solves the problem*: Indeed, this construction requires placing at most $\text{size}(H)/|X|$ consistency edges on H' for each variable in X, but this sums-up to only $O(\text{size}(H)) = O(\text{size}(G))$ consistency edges.

It remains to show that the intermediate assignment tester that uses the modified randomized procedure has rejection probability $\Omega(\min(\sqrt{t} \cdot \varepsilon, 1/t)$. In order to do it, we prove a result analogous to [D05, Lemma 8.2] and [D07, Lemma 9.2]. The reason that we prove again such a result is that Dinur [D05, D07] proves the result for her construction of H', while we prove it for the modified version of this construction. The following lemma also differs from [D05, Lemma 8.2] and [D07, Lemma 9.2] in some (hidden) constant factors.

Lemma 3 *Assume that $\varepsilon < 1/t$ and fix an assignment $a : X \to \{0,1\}$. Then*

- *If $a \in SAT(\Phi)$, then there exists $b : V_H \to \Sigma^{d^{t/2}}$ such that $UNSAT_{a \cup b}(H') = 0$.*
- *If $\delta = \texttt{dist}(a, SAT(\Phi)) > 0$, then for every $b : V_H \to \Sigma^{d^{t/2}}$ it holds that $UNSAT_{a \cup b}(H') = \Omega(\sqrt{t} \cdot \varepsilon) \cdot \delta$.*

Proof. The first item of the lemma can be proved using the same proof as in [D05, D07]. Turning to the second item, assume that $\delta = \texttt{dist}(a, \text{SAT}(\Phi)) > 0$ and fix an assignment $b : V_H \to \Sigma^{d^{t/2}}$ to H. We shall prove that $\text{UNSAT}_{a \cup b}(H') = \Omega(\sqrt{t} \cdot \varepsilon) \cdot \delta$. As in [D05, D07], let b_1 be the assignment to $\text{prep}(G)$ decoded from b using a plurality vote, and let b_0 the assignment to G decoded from b_1 using plurality vote. The case where $\texttt{dist}(b_0|_X, a) \leq \delta/2$ can be proved using the same proof as in [D05, D07], since in this case $\texttt{dist}(b_0|_X, \text{SAT}(\Phi)) \geq \delta/2$ (by the triangle inequality), and the argument focuses on this fact only (while ignoring a). therefore by the definition of G it holds that $\text{UNSAT}_{b_0}(G) \geq \varepsilon \cdot \delta/2$. Specifically, in this case, by the definition of G, it holds that $\text{UNSAT}_{b_0}(G) \geq \varepsilon \cdot \delta/2$, and applying the reasoning of [D05, D07] which rely on the properties of pre-processing and graph powering), it holds that $\text{UNSAT}_b(H) = \Omega(\sqrt{t} \cdot \varepsilon) \cdot \delta$.

Finally, since the edges of H form half of the edges of H', it follows that $\mathrm{UNSAT}_{a\cup b}(H') = \Omega(\sqrt{t}\cdot\varepsilon)\cdot\delta$, as required.

We turn to the case where $\mathtt{dist}(b_0|_X, a) > \delta/2$, where we must rely on a and refer to the current modified construction. We shall prove that $\mathrm{UNSAT}_{a\cup b}(H') = \Omega(\delta)$, which implies the required result. Recall that $b_0(v)$ is defined by plurality vote of b_1 in $[v]$. In contrast, we define b'_0 to be an assignment for G such that for every $v \in V$ the value $b'_0(v)$ is the plurality vote among the values assigned by b_1 to the vertices in $[v]'$ (i.e., $b'_0(v)$ is the value that maximizes $\Pr_{u\in[v]'}[b_1(u) = b'_0(v)]$). Indeed, the key question is whether these two assignments are close or not, and we consider the two possible cases: (1) $\mathtt{dist}(b_0|_X, b'_0|_X) \leq \delta/4$ and (2) $\mathtt{dist}(b_0|_X, b'_0|_X) > \delta/4$.

1. Suppose that $\mathtt{dist}(b_0|_X, b'_0|_X) \leq \delta/4$. We show that in such case $a\cup b$ violates at least $\delta/16$ of the consistency edges of H', by considering the action of the modified randomized procedure defined above. Using the triangle inequality, it holds that $\mathtt{dist}(b'_0|_X, a) > \delta/4$; hence, in this case, with probability at least $\delta/4$, the procedure chooses in Step 1 a vertex $x \in X$ such that $b'_0(x) \neq a(x)$. Recall that the value $b'_0(x)$ is defined to be the most popular value assigned by b_1 to the vertices of $[x]'$, and therefore with probability at least $\frac{1}{2}$ the procedure chooses in Step 2 a vertex $z \in [x]'$ such that $b_1(z) \neq a(x)$. Similiarly, conditioned on $b_1(z) \neq a(x)$, with probability at least $\frac{1}{2}$, the procedure chooses in Step 3 a vertex w such that $b(w)_z \neq a(x)$. Putting all of these together, it follows that in this case the randomized procedure rejects $a\cup b$ with probability at least

$$\frac{\delta}{4}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{\delta}{16}$$

   and therefore $\mathrm{UNSAT}_{a\cup b}(H') = \Omega(\delta)$, as required.

2. Suppose that $\mathtt{dist}(b_0|_X, b'_0|_X) > \delta/4$. We show that in such case $\mathrm{UNSAT}_b(H) = \Omega(\delta)$, due to the violation of the equality constraints of $\mathrm{prep}(G)$. Recall that $\mathrm{prep}(G)$ is constructed by replacing every vertex v of G with a set of copies $[v]$ of size $\deg_G(v)$, placing the edges of an expander on $[v]$ and associating those edges with equality constraints. Observe that the inequality $b_0(x) \neq b'_0(x)$ can only hold for variables $x \in X$ for which $[x]' \neq [x]$, and in this case it follows that $|[x]'| = \mathrm{size}(H)/|X|$, since $|[x]'| = \min(|[x]|, \mathrm{size}(H)/|X|)$ and $[x]' \subseteq [x]$. Now, observe for every $x \in X$ that satisfies $b_0(x) \neq b'_0(x)$, it holds that $\Omega(|[x]'|)$ equality edges of $[x]$ (i.e., edges between the majority vertices of $[x]'$ and the majority vertices of $[x]$) are violated

by b_1, due to the mixing property of the expander that was used for the construction of prep(G). It follows that in this case the number of edges of prep(G) that are violated by b_1 is at least

$$\begin{aligned}\left|\left\{x \in X : b_0(x) \neq b'_0(x)\right\}\right| \cdot \Omega\left(\frac{\text{size}(H)}{|X|}\right) &= \texttt{dist}(b_0|_X, b'_0|_X) \cdot \Omega(\text{size}\,(H)) \\ &= \Omega(\delta \cdot \text{size}\,(H)).\end{aligned}$$

The latter equality implies that $\text{UNSAT}_b\,(H) = \Omega(\delta)$, and therefore $\text{UNSAT}_{a\cup b}\,(H') = \Omega\,(\delta)$, as required.

This completes the proof of the lemma. ■

4 Digest

The issue at hand is augmenting the construction of the intermediate constraint graph, which is the pivot of the gap amplification, used in order to adapt it from the PCP setting to the setting of assignment-testers (resp., PCPPs). The natural augmentation calls for connecting each input variable (i.e., a variable representing a bit in the input assignment (resp., a location in the input oracle)) to all auxiliary copies of this variable (i.e., auxiliart variables (resp., locations in the proof oracle)) that were produced in the preprocessing step.

The problem is that this natural augmentation may yield too large of an overhead in the case that the original constraint graph is not (almost) regular. The solution is to connect each input variable only to a number of copies that does not exceed the average degree of input variables in the original graph. The reason that this works is that expanders were placed (in the preprocessing step) among the copies of each variable, and so a mismatch between the majority of the "connected copies" and the majority of all copies will violate a large number of edges (i.e., a number that is linearly related to the number of connected copies). Hence, the number of violated edges is linearly related to the minimum between the degree of the input variable in the original constraint graph and the average degree of all input variables in that graph. The latter term suffices since the distance between assignments treats all input variables equally.

The last assertion also explians why it is unlikely that the given assignment tester will contain input variables of significantly different degrees. Hence, we expect that in most applications, all input variables will have (almost) the same degree, and in that case the original analysis of [D05, D07]

suffices. Still, it feels better not to augment the definition of assignment testers to to require this.

References

[BGHSV04] E. Ben-Sasson, O. Goldreich, P. Harsham, M. Sudan and S. Vadhan, Robust PCPs of Proximity, Shorter PCPs and Applications to Coding, *SIAM Journal of Computing*, Vol. 36 (4), pages 889–974, 2006. Preliminary version in STOC 2004, pages 120-134.

[D05] I. Dinur. The PCP theorem by gap amplification, ECCC TR05-046.

[D07] I. Dinur. The PCP theorem by gap amplification, *Journal of ACM*, Vol. 54 (3), 2007. Preliminary version in STOC 2006, pages 241–250.

[DR06] I. Dinur and O. Reingold. Assignment testers: Towards combinatorial proofs of the PCP theorem. *SIAM Journal of Computing*, Vol. 36 (4), pages 975–1024, 2006. Preliminary version in FOCS 2004, pages 155–164.