Locally Testable Codes and Expanders

Irit Dinur* Tali Kaufman†

April 19, 2012

Abstract

A locally testable code is a code defined by a robust set of local constraints. Namely, the distance of a vector from the code is well approximated by the fraction of local constraints that it violates. A constraint graph of an LTC is a graph whose vertices are labeled by the coordinates of the code, and vertex i is adjacent to j whenever they occur together in a constraint. We study the relation between the topology of this graph and the structure of the code.

We show that every constraint graph of an LTC must be a *small set expander*, in which all sets up to some linear size expand. Moreover, a constraint graph of an LTC can be decomposed into *constantly* many *expander* graphs on which the induced codes are approximately LTCs.

Our work suggests a subtle relation between LTCs and expanders. It is known [BSHR05] that codes defined by strongly expanding (e.g. random) sets of constraints are not locally testable. In contrast, we show that every constraint graph of an LTC must be weakly expanding (i.e., small set expanders). Our result provides a necessary condition for LTCs that can be applied toward proving that certain codes are *not* LTCs.

On the way to our result we prove that every small-set expander (i.e., a graph where small sets up to some linear size are guaranteed to expand) can be decomposed into a constant number of "standard" expanders.

1 Introduction

A constraint satisfaction problem is given by a set of variables, and a set of local constraints on them. It is conveniently viewed as a hypergraph G whose vertices are the variables, and where for each constraint we place a hyperedge labeled by the constraint function. We will also consider the *constraint graph* in which vertices i and j are connected by an edge if they occur together in a constraint. This graph is the so-called "skeleton" of the constraint hypergraph, and we will denote it by G'. The set $sat(G) \subset \{0,1\}^n$ is a set of assignments that satisfy all of the constraints in a given constraint hypergraph G. Any set $C \subset \{0,1\}^n$ that is defined as $C = sat(G)$ and that obeys the following two properties is also known as a *locally testable code*, or LTC [GS06].

- (G is robust:) For any $x \in \{0,1\}^n$, the fraction of constraints that are unsatisfied by x is a good approximation of the Hamming distance of x from the set $sat(G)$.
- ($sat(G)$ has distance:) The set $sat(G)$ has constant relative distance, i.e. $x \neq y \in sat(G)$ implies that x, y differ on a constant fraction of their coordinates.

*Weizmann Institute of Science and Microsoft New England. Email: irit.dinur@weizmann.ac.il. Research supported in part by the Israel Science Foundation and by the Binational Science Foundation and by an ERC grant.

†Bar-Ilan University, ISRAEL. Email: kaufmant@mit.edu. Supported by an Alon fellowship.

We defer the formal definition of LTCs to Section 2.

LTCs were first developed in the context of PCPs [**?**, GS06], and known constructions of LTCs are often based on previously studied PCPs. Several constructions of LTCs are known, see for example a survey by [**?**]. A major question is to find the best possible parameters of LTCs, most importantly the rate. The currently best known rate for an LTC [**?**, **?**, **?**] is significantly worse than the best possible rate of a general error correcting code. Local testability seems to be in tension with having large distance, and it is interesting to study what properties enable the two to coincide. One research direction explores the role of symmetry in locally testable codes. So far we know that certain types of symmetry imply testability while others do not [**?**, **?**]. In this work we explore another aspect of this interesting class of objects, namely the expansion properties of the constraints.

We show that if $C = sat(G)$ is a locally testable code then the structures of G' and $sat(G)$ are nicely coupled together.

Our Results

A naive conjecture regarding the structure of LTCs might be that the associated graph must be an expander (i.e., every set of vertices should have many outgoing edges). The intuition is that in an LTC a typical constraint looking at only a constant number of bits 'approximately knows' whether the global string is a valid codeword. How can a local view relate to the global picture unless the graph is sufficiently "mixing"?

We prove that this intuition is indeed correct in that the constraint graph must be a small set expander, in which all sets up to some linear size expand. However, the conjecture, as stated, is clearly false: if C_1, C_2 are two LTCs, then the code $C = C_1 \times C_2$ defined by putting them side by side is also an LTC, yet its graph is not even connected. We prove that this is essentially the only obstacle to expansion.

Namely, we prove that either the code is "irreducible" and then the graph must be an expander; or else both the graph and the code decompose according to the same partition of V.

Theorem 1.1 (Main Theorem - Informal)**.** *The constraint graph of every (strong) locally testable code $C \subset \{0,1\}^n$ is a small-set expander (i.e. all sets up to some linear size expand) . Its vertices can be decomposed into a constant number of parts $[n] = S_1 \cup \cdots \cup S_t$, such that*

- *Graph: The graph induced on each S_i is an expander, and there are relatively few edges between every S_i and S_j.*
- *Code: The original code is approximately the cartesian product of the codes on each S_i,*

$$C \approx C_{S_1} \times \cdots \times C_{S_t}$$

where C_{S_i} is the projection of codewords of C to the coordinates in S_i. C_{S_i} is itself approximately a locally testable code with parameters similar to those of C.

Our result suggests a subtle relationship between LTCs and expanders. On the one hand, it follows from the analysis of [BSHR05] that an LDPC code whose parity check matrix describes an odd-neighbor expander cannot be an LTC. Recall that an LDPC code is a code defined by local *linear* constraints, and the parity check matrix gives exactly the constraint hypergraph. On the other hand, we show here that the constraint graph of strong LTCs must have non-trivial expansion, i.e., every small set of vertices must have a constant fraction of its edges going out. We conclude that the

constraint graphs of LTCs must have some non-trivial expansion, yet their constraint hypergraphs cannot have too much expansion[1].

A variant of Theorem 1.1 holds also for weak LTCs (see Theorem 3.2 for exact formulation). For weak LTCs we can not claim that the constraint graph is a small set expander. However, the graph can be decomposed into constant many parts such that each $G(S_i)$ is an expander in an analogous 'weak' sense: only sets whose size is above γn are guaranteed to expand. Moreover, the codes induced on these parts are again approximately LTCs.

Our result gives a necessary criterion for codes being locally testable. A code that is defined by a local constraint hypergraph G *cannot* be an LTC unless the corresponding skeleton graph G' is a small set expander.

We mention that in [**?**] the second author and Sudan defined a notion of an LTC being "single-orbit", which essentially means that the code is defined by a single constraint and its permutations under some group acting on the coordinates. They showed that if the group is affine then any single-orbit code is locally testable (Theorem 89?? in [**?**]). They left open the question of whether the same holds if the group is cyclic. Our theorem implies a negative answer to this question. Indeed, any code defined by a single orbit under a cyclic group cannot be locally testable, since the constraint graph cannot be an expander, being contained in a Cayley graph of an cyclic group with a constant number of generators.

Finally, we remark that one very well-studied class of LTCs is that of polynomial codes or affine-invariant codes [KS08]. In such locally testable codes the resulting constraint graph is nothing but a clique. Nevertheless, our result is not vacuous even in this case, since one may still be interested in the structure of *sparse* testers for this code (these are testers obtained by taking a subset of the possible tests)[2]. Our result says that any sparse tester for these codes must have an associated *expanding* constraint graph.

Our Techniques

The main idea that we use for proving our main result is that LTCs decompose on sparse cuts. We then rely on the distance of the code to deduce that sparse cuts can occur only on large sets. We also show that a small-set expander can be decomposed into *constant* many expanders.

LTC decomposes on sparse cuts. For an LTC, the constraint graph can have a sparse cut $(S, \bar{S})$ only if the LTC decomposes on this cut, i.e., if there are two codes C_1, C_2 such that every word $w \in C$ is very close in Hamming distance to a word $w_1(S)w_2(\bar{S})$ where $w_i \in C_i$ and where notation $a(X)b(Y)$ means a string that equals a on coordinates in X and b on coordinates in Y. The assertion is true essentially because if a "hybrid" $w_1(S)w_2(\bar{S})$ were far from C, then it must have been rejected with proportional probability. However, the only edges that can reject this word must cross the sparse cut $(S, \bar{S})$, and there are too few such edges. One then has to work some more in order to prove that each C_i is approximately an LTC with the claimed parameters.

Decomposing a small-set expander into 'standard' expanders. A graph is a small-set expander if every set $S \subset V$ of size at most δn expands (i.e. has at least $\beta \cdot |S|$ outgoing edges).

[1] The constraint hypergraph can be described by a bipartite graph between vertices and constraints. The constraint graph is simply the square of this bipartite graph.

[2] In fact, a random linear number of tests was proved to give a good tester by [GS06]

We prove that every small set expander G can be decomposed into at most $1/\delta$ expanders. More accurately, denoting $E(S,T)$ the number of edges from S to T, we show

Theorem 1.2. *Let $R > 0$ and let $G = (V, E)$ be a graph in which for every set $S \subset V$ of size at most δn, $E(S, \bar{S}) \geq R \cdot |S|$. Then for every $\tau \leq R \cdot (\frac{\delta}{4})^{1/\delta}/2$ there is a partition $V = V_1 \cup \cdots \cup V_t$ into $t \leq 1/\delta$ parts such that*

1. *(Each part is an expander:) The graph $G(V_i)$ is a τ-expander for each i, (namely every set S has at least $\tau |S|$ outgoing edges).*
2. *(Large parts:) $|V_i| \geq \delta n$, where $n = |V|$.*
3. *(Few edges between parts:) $E(V_i, V \setminus V_i) \leq r' \cdot |V_i|$ for $r' = 2\tau \cdot (\frac{4}{\delta})^{1/\delta}$.*

Our theorem is proven by analyzing a (straightforward) recursive procedure that finds a sparsest cut and decomposes according to it. Clearly this is not an efficient procedure, but we only care about existence. Variants of this procedure have been analyzed previously (e.g., [KVV04, Tre05, GMR+11]), but in a more algorithmic context. Our setting is slightly different in that it requires decomposing a graph into expanders of *linear* size.

On the structure of general constraint satisfaction instances

Our work can be viewed within the context of a more general question of understanding the structure of CSPs, both through the structure of the constraint graph G and through the structure of the set $sat(G)$. One concrete question pertaining to the graph structure is the following:

For what graph structures is a CSP NP-hard to approximate?

Two 'separate' aspects of CSPs are their constraint type and their constraint graph. The type is defined by which constraint functions we allow (i.e. whether it is a $3SAT$ instance or a $3LIN$ instance, etc.). It is very well studied by now with dozens of papers on various types of constraints. In fact, a long line of works culminating in e.g. [Rag08, AM08] yields a very nice classification of the approximation behavior of different constraint types, under Khot's unique games conjecture [Kho02]. In contrast, the relationship between the *structure* of CSP instances and their hardness is much less studied.

Nevertheless, some works do exist. Arora et. al [AKK+08] prove that unique games are never hard-to-approximate on expanders. This is certainly not the case for non-unique CSPs. In the extreme, one can super-impose an expander with empty constraints on any constraint graph, thereby transforming it into an expander, without changing the set of satisfying assignments. In fact our intuition, which we prove for the case of LTCs, is that every NP-hard CSP has an underlying expanding structure.

A related line of work studies the structure of the set $sat(G)$ for random constraint graphs G. This is a very active field, particularly for random k-sat. While random CSPs may not be NP-hard to approximate, they are thought to be hard for example to refute. Feige [Fei02] proved interesting hardness-of-approximation consequences based on a related hypothesis. In his work the expanding structure of random instances plays an important role, and this demonstrates that understanding the structure of NP-hard instances may lead to new hardness-of-approximation results.

2 Preliminaries

2.1 Expanders and Constraint (Hyper-)Graph

Let $G = (V, E)$ be a graph. For a set $S \subset V$ we denote its complement in the graph by $\bar{S} = V \setminus S$. For vertex sets $A, B \subset V$ we denote by $E(A, B)$ the set of edges with one endpoint in A and the other in B.

Definition 2.1 (Expander)**.** A graph $G = (V, E)$ is an *expander* with expansion parameter τ (or τ-*expander*) if for every set $S \subset V$ of at most half the vertices, $E(S, \bar{S}) \geq \tau |S|$.

Definition 2.2 (Small-Set Expander)**.** A graph $G = (V, E)$ is a *small-set expander* with expansion parameters τ, α (or (τ, α)-*small-set expander*) if for every set $S \subset V$, $|S| \leq \alpha |V|$, $E(S, \bar{S}) \geq \tau |S|$.

Definition 2.3 (Large-Set Expander)**.** A graph $G = (V, E)$ is a *large-set expander* with expansion parameters τ, γ (or (τ, γ)-*large-set expander*) if for every set $S \subset V$, $\gamma |V| \leq |S| \leq \frac{1}{2}|V|$, $E(S, \bar{S}) \geq \tau |S|$.

Definition 2.4 (Constraint hyper-graph)**.** A *constraint hyper-graph* is a hyper-graph $G = (V, E)$ where each hyper-edge $e \in E$ is labeled by some function $f_e : \{0,1\}^{|e|} \to \{0,1\}$, where f_e depends on all its coordinates and its image is not always 1 (i.e., f_e does not represent a dummy constraint) . An assignment to the constraint hyper-graph is a labeling of the vertices with 0/1 values, alternatively viewed as a string of n bits. We define $sat(G)$ to be the set of strings in $\{0,1\}^{|V|}$ that, when viewed as assignments, satisfy every constraint in G. We also define $rej(w)$ to be the fraction of constraints of G that reject an assignment w.

Definition 2.5 (Constraint Graph)**.** Let G be some constraint hyper-graph. The constraint graph of G, denoted G', is the graph over the same vertex set of G, and such that u, v are connected by an edge iff they are contained together in some hyper-edge of G.

We note that a constraint hypergraph can be described by a bipartite graph between vertices and constraints. The constraint graph is simply the square of this bipartite graph, i.e. two vertices are connected if there is a length two path between them in the bipartite graph.

2.2 Codes and Locally Testable Codes

A code is a set $C \subset \{0,1\}^n$. We define Hamming distance between $x, y \in \{0,1\}^n$ to be the number of coordinates on which they differ. The relative distance is the distance divided by n. The relative distance of a set $C \subset \{0,1\}^n$ is the smallest relative distance between a pair of distinct $x, y \in C$. We say that x, y are δ-close if their relative distance is at most δ. Let $w \in C \subset \{0,1\}^n$ and let $S \subset [n]$, then we denote by w_S the restriction of w to the coordinates in S, and C_S is the set of all w_S for all $w \in C$. For a partition of the coordinates $[n] = S_1 \cup \cdots S_t$ we write $C_{S_1} \times C_{S_2} \times \cdots C_{S_t}$ to mean the set of all words $w^1(S_1)w^2(S_2)\cdots w^t(S_t)$ where this notation denotes the string w that equals w^i on coordinates S_i.

Definition 2.6 (Codes approximating each other)**.** A code C is η-approximated by a code C' if for every $w \in C$ there is some $w' \in C$ that is η close to it. If C η-approximates C' and C' η-approximates C then we denote $C' \approx_\eta C$.

(In this case the code C can be seen as contained in a collection of η-balls around codewords of C': $C \subset \cup_{w' \in C'} B_\eta(w')$.)

Definition 2.7 (Locally testable code (strong definition))**.** A code $C \subset \{0,1\}^n$ together with a constraint hyper-graph $G = ([n], E)$ is a ρ-strong-LTC if $C = sat(G)$ and if for every $w \in \{0,1\}^n$,

$$rej(w) \geq \rho \cdot dist(w, C).$$

Definition 2.8 (Locally testable code (weak definition))**.** A code $C \subset \{0,1\}^n$ together with a constraint hyper-graph $G = ([n], E)$ is a (γ, ε)-LTC if $C = sat(G)$ and if for every $w \in \{0,1\}^n$

$$dist(w, C) > \gamma \qquad \Longrightarrow \qquad rej(w) > \varepsilon.$$

Definition 2.9 (Approximately locally testable code)**.** If codes C, C' are such that $C' \approx_\eta C$, and C' is a strong/weak-LTC, then C is *approximately-strong/weak-LTC*.

For the rest of the paper, if C is an LTC with some constraint graph G, then G is the constraint graph "associated" with C.

Definition 2.10 (Non-degenerate Code)**.** Let $C \subset \{0,1\}^n$ be a code. A set $A \subset [n]$ of coordinates is *degenerate* for the code, if every pair of words $x, y \in C_A$ have distance at most $|A|/3$. A code is *non-degenerate* if it has no non-empty degenerate set of coordinates.

Intuitively, on any subset of the coordinates of a code, one can find codewords that differ in many locations. For example, in a linear code (a code whose elements are a linear subspace of $\{0,1\}^n$) we have,

Claim 2.11. *If C is a linear code with no coordinate which is identically zero, and A is a degenerate subset of coordinates, then for every $w \in C$, $w_A = 0$. (proof omitted).* ■

In this extended abstract version we only deal with non-degenerate codes. (The more general structure statements also split off the degenerate part of the code).

3 Proof of the Main Theorem

3.1 Decomposition Lemma

The following lemma is a key to our proof. Essentially it says that for each cut (S_1, S_2), either many edges cross the cut, or else the LTC splits into two LTCs according to the cut.

Lemma 3.1 (Decomposition Lemma)**.** *Let C be a (γ, ε)-LTC with relative distance Δ. Let $G = (V, E)$ be the associated constraint graph, and denote by $d = \frac{|E|}{|V|}$ the average degree of G. If for some cut (S_1, S_2) with $\alpha := \frac{|S_1|}{|V|} \geq 3\gamma$ we have $E(S_1, S_2) < \frac{\tau d}{\alpha} \cdot |S_1|$ for some $\tau \leq \varepsilon$, then*

$$C \approx_\gamma C_{S_1} \times C_{S_2}$$

and there are $(2\gamma/\alpha, \varepsilon - \tau)$-LTCs $C'_i \subset C_{S_i}$, such that $C'_i \approx_\gamma C_{S_i}$, and the relative distance of C'_i is at least $\frac{\Delta - \gamma}{\alpha}$, and $\alpha \geq \Delta - \gamma$. In particular, $C \approx_{2\gamma} C'_1 \times C'_2$.

One way to interpret the lemma is that if the LTC does not decompose on a cut $(S, \bar{S})$, and if $|S| \geq \gamma n$ then the number of edges exiting S is at least $\varepsilon |E| = \frac{\varepsilon d}{\gamma} \cdot |S|$ where d is the average degree.

Proof. We divide the proof into a few steps.

1. We first prove that
$$C \approx_\gamma C_{S_1} \times C_{S_2}.$$
Clearly $C \subseteq C_{S_1} \times C_{S_2}$, so it remains to prove that for any $(w_1, w_2) \in C_{S_1} \times C_{S_2}$ there is a nearby word in C. By definition, there are words $w^1, w^2 \in C$ such that $(w^i)_{S_i} = w_i$. Since no test rejects on w^1 or w^2, the only tests that can reject (w_1, w_2) are those crossing the cut (S_1, S_2). By assumption there are fewer than $\frac{\tau d}{\alpha} |S_1| = \tau |E| \leq \varepsilon |E|$ such edges. Since the code is an (γ, ε)-*LTC*, the distance of (w_1, w_2) from C must be at most γ times n.

2. Let C_i' be constructed from C_{S_i} greedily by repeating the following process: add a word $x \in C_{S_i}$ into C_i' and remove from C_{S_i} the entire ball around x of radius up to γn. Denoting $n_i = |S_i|$, and letting $\alpha = n_1/n$ it is easy to see that C_i' γ/α-approximates C_{S_i}, and that every distinct pair of words in C_i' are at least $\gamma/\alpha \cdot n_1$ bits apart.

   It follows from the previous argument that every word in $C_1' \times C_2'$ is γ-close to a word in C. To see that every word in $w \in C$ is approximated by some $w_1 w_2$ just note that each w_{S_i} is at most γn away from some word in C_i' so we get a $2\gamma n$ approximation,
$$C \approx_{2\gamma} C_1' \times C_2'.$$

3. We proceed to argue about the distance of C_i'. Since C is non-degenerate[3], C_1' must have at least two distinct words $a, a' \in C_1'$ (since $n_1 := |S_1| \geq 3\gamma n$ there must be two words a, a' that differ on more than $n_1/3$ coordinates). Let b be such that $w = ab \in C$ (such b exists by construction of C_1'). Let w' be the closest word in C to $a'b$. We claim that $w \neq w'$. Otherwise, $dist(a'b, C) = dist(a'b, w) = dist(a', a) \geq \gamma n$ but then the LTC condition implies that at least $\varepsilon |E|$ edges should reject $a'b$. These can only be edges crossing the (S_1, S_2)-but there are too few of those. Using $w \neq w'$ and the triangle inequality we deduce
$$dist(a, a') = dist(ab, a'b) \geq dist(w, w') - \gamma n \geq (\Delta - \gamma)n = \frac{\Delta - \gamma}{\alpha} n_1$$
which means that the relative distance of the code C_1' is at least $\frac{\Delta - \gamma}{\alpha}$. It also means that $n_1 \geq (\Delta - \gamma)n$.

4. Next, we prove that C_i' is an $(2\gamma/\alpha, \varepsilon - \tau)$-LTC. Let w be a word whose distance from C_i' is at least $2\gamma/\alpha \cdot n_1 = 2\gamma n$ bits. Then by construction its distance from C_{S_i} is at least γn bits. Thus, the best possible continuation $\tilde{w}$ of w to a codeword in C is still at least γn away, and must be rejected by at least $\varepsilon |E|$ edges. All of these edges must touch the set S_i, but some can go from S_i out of S_i. Since there are relatively few cut edges, this leaves at least $(\varepsilon - \tau) |E|$ edges that must reject w.

■

3.2 Proof of the Main Theorem for strong LTCs

In this section we state and prove the main theorem, Theorem 1.1, that applies for strong LTCs.

Theorem 1.1 (Main Theorem). *Let $C \subset \{0,1\}^n$ be an ρ-strong LTC with relative distance Δ. Let $G = (V, E)$ be the constraint graph of C whose average degree is $d = |E|/|V|$. Then G is a $(\frac{d\rho}{3}, \frac{3\Delta}{4})$-small-set expander, i.e., all sets $S \subset V$, $|S| \leq \frac{3\Delta n}{4}$ have $E(S, \bar{S}) \geq \frac{d\rho}{3}|S|$. V can be decomposed into $t \leq \frac{2}{\Delta}$ parts $V = S_1 \cup \cdots \cup S_t$, where each $|S_i| \geq \frac{3\Delta n}{4}$ such that for any $\gamma < \Delta/8$ and $\beta < \frac{d\rho\gamma}{3\Delta} \cdot (3\Delta/16)^{4/3\Delta}$:*

[3]See Definition 2.10 and recall that in this preliminary version we assume all codes are non-degenerate.

- *Graph: The graph induced on each S_i is a β-expander, and $E(S_i, V/S_i) \leq 2\beta(16/3\Delta)^{4/3\Delta}|S_i|$.*
- *Code: $C \approx_{t\gamma} C'_1 \times \cdots \times C'_t$, where $C'_i \subset C_{S_i}$ has distance $\Delta - \gamma$, $C'_i \approx_\gamma C_{S_i}$, and C'_i is a $(\frac{8\gamma}{3\Delta}, \frac{\rho\gamma}{2})$-LTC.*

Proof. The proof has three steps.

1. We first prove that G must be a small set expander: Every set S of size up to $\frac{3\Delta n}{4}$ has at least $R \cdot |S|$ outgoing edges (where $d = \frac{|E|}{|V|}$ is the average degree), for $R = \frac{d\rho}{3}$.

   Any ρ-strong LTC is also a $(\gamma, \rho\gamma)$-LTC for every $\gamma > 0$. Let $(S, \bar{S})$ be a cut and let $3\gamma = |S|/|V| \leq 1/2$. We will prove that if $E(S, \bar{S}) < \frac{d\rho}{3}|S|$, then $\gamma \geq \frac{\Delta}{4}$. The fraction of edges leaving S out of the total number $|E|$ of edges is less than $\frac{\rho d|S|}{3|E|} = \rho\gamma$. We can invoke Lemma 3.1 and get $3\gamma = \frac{|S|}{|V|} \geq \Delta - \gamma$. In other words $\gamma \geq \frac{\Delta}{4}$ as claimed, and thus $|S| \geq \frac{3\Delta n}{4}$.

2. We invoke the small set expander decomposition, Theorem 1.2, with parameter $\delta = \frac{3\Delta}{4}$ and $\beta < R \cdot (\delta/4)^{1/\delta}/2 = \frac{d\rho}{3} \cdot (\delta/4)^{1/\delta}/2$ to be chosen below, to get a partition of the vertices into $S_1 \cup \cdots \cup S_t$ such that $t \leq \frac{4}{3\Delta}$, and

   (a) $G(S_i)$ is a β-expander for each i.

   (b) $E(S_i, V \setminus S_i) \leq 2\beta(4/\delta)^{1/\delta} \cdot |S_i|$ for each i.

   (c) $|S_i| \geq \frac{3\Delta}{4}|V|$.

3. We can now view C as a $(\gamma, \rho\gamma)$-LTC with distance Δ, for some $\gamma \leq \Delta/8$. We want to apply Lemma 3.1 on the cut $(S_i, V \setminus S_i)$ in the associated structure graph. This cut has at most $r' = 2\beta(4/\delta)^{1/\delta}$ times $|S_i|$ edges. Set τ arbitrarily to $\rho\gamma/2$, and observe that as long as r' is smaller than $\rho\gamma/2 \cdot d/\alpha \leq \frac{2\rho\gamma d}{3\Delta}$, we get that C_{S_i} is γ-approximated by C'_i and that C'_i is an $(2\gamma/\alpha, \rho\gamma/2)$-LTC (i.e., $(\frac{8\gamma}{3\Delta}, \frac{\rho\gamma}{2})$-LTC) with distance $\Delta - \gamma$. The choice of γ is still somewhat flexible as long as $\gamma < \Delta/8$. The smaller we make γ, the smaller must β be, which is a weaker expansion guarantee on each $G(S_i)$.

■

It would be nice to improve this theorem so as to get sub-codes in the decomposition that are strong rather than weak LTCs.

3.3 Proof of a variation to the Main Theorem for weak LTCs

In this section we show that Theorem 1.1 holds with some variations also for weak LTCs. See Theorem 3.2 below for exact formulation. For weak LTCs we can not claim that the constraint graph is a small set expander. However, the graph can be decomposed into constant many parts such that each $G(S_i)$ is an expander in an analogous 'weak' sense: only sets whose size is above γn are guaranteed to expand. Moreover, the codes induced on these parts are again approximately LTCs.

Theorem 3.2. *Let $C \subset \{0,1\}^n$ be an (γ, ε)-LTC with relative distance Δ. Let $G = (V, E)$ be the constraint graph of C whose average degree is $d = |E|/|V|$. Then V can be decomposed into $t \leq \frac{2}{\Delta}$ parts $V = S_1 \cup \cdots \cup S_t$, where each $|S_i| \geq \frac{\Delta n}{2}$ such that for any $\gamma < \frac{\Delta^2}{8} \cdot 2^{-2/\Delta}$ and $0 < \tau < \epsilon\Delta/2$:*

- *Graph: The graph induced on each S_i is a large-set-expander, i.e., every $S \subset S_i$, $3\gamma'|S_i| \leq |S| \leq \frac{1}{2}|S_i|$ has $E(S, S_i \setminus S) \geq \tau d_i/3\gamma'|S|$, where $\gamma' = \frac{2 \cdot 2^{2/\Delta}\gamma}{\Delta}$ and d_i is the average degree of $G(S_i)$. Moreover, for every i, $E(S_i, V \setminus S_i) \leq t\tau|E|$.*
- *Code: $C \approx_{t\gamma} C'_1 \times \cdots \times C'_t$, where $C'_i \subset C_{S_i}$ has distance at least $\Delta/2$, $C'_i \approx_\gamma C_{S_i}$, and C'_i is a $(2\gamma 2^{2/\Delta}/\Delta, \epsilon - t\tau)$-LTC.*

Proof. We repeatedly apply the main lemma to C. In the first step, either the graph has no large 'sparse' cut (S_1, S_2), which means that it is a $(\tau d/3\gamma, 3\gamma)$-large set expander: every set S_1 of size at least $3\gamma n$ must have at least $(\tau d/3\gamma) \cdot |S_1|$ outgoing edges. Else, the graph has a cut (S_1, S_2) on which it decomposes, and we consider separately the LTCs C'_1 and C'_2 and their associated graphs $G(S_1)$ and $G(S_2)$, and try to find large sparse cuts on which to decompose, or else we declare that they are large-set expanders. Let T_j be a set that resulted from j decomposing steps, $T_j \subset T_{j-1} \subset \cdots \subset T_1 \subset T_0 = V$. Let us compute, by induction, the parameters of the graph $G(T_j)$ and the code C'_j associated with this graph. Denoting by $\alpha_j = |T_j| \,/\, |T_{j-1}|$,

1. C'_j is a $(\gamma_j, \varepsilon - j\tau)$-LTC where $\gamma_j \leq \frac{2^j}{\alpha_1 \cdot \alpha_2 \cdots \alpha_j}\gamma = \frac{2^j|T_0|}{|T_j|} \leq \frac{2 \cdot 2^{2/\Delta}}{\Delta}\gamma$ (since $j \leq \frac{2}{\Delta}$ and $\frac{|T_j|}{|T_0|} \leq \frac{\Delta}{2}$). In addition, $C'_j \approx_{\gamma_j} C_{T_j}$.
2. The distance of C'_j is at least $(\Delta - (\gamma_1 + \gamma_2 + \cdots + \gamma_j)) \cdot n \geq (\Delta - 2\gamma_j)n \geq (\Delta - 2\frac{2 \cdot 2^{2/\Delta}}{\Delta}\gamma)n \geq \frac{\Delta n}{2}$ (since $\gamma < \frac{\Delta^2}{8} \cdot 2^{-2/\Delta}$).
3. Either in $G(T_j)$ every set of size at least $3\gamma_j |T_j|$ expands by $\tau d_j/3\gamma_j$ or else we will split T_j in a future step.
4. $E(T_j, V \setminus T_j) \leq j\tau\,|E|$.
5. $|T_j| \geq (\Delta - (\gamma_1 + \gamma_2 + \cdots + \gamma_j)) \cdot n \geq \frac{\Delta n}{2}$.

Clearly for $j = 1$ the parameters above simply follow from the main lemma. Assuming correctness for $j - 1$, it remains to plug in one more invocation of the main lemma, to obtain these values. It remains to see that if γ is such that $\gamma < \frac{\Delta^2}{8} \cdot 2^{-2/\Delta}$, then the number t of distinct sets S_i is $t \leq 2/\Delta$. This holds since after $2/\Delta$ steps, there are at least $2/\Delta$ sets S_i's each (by the selection of γ) is of size at least $\Delta n/2$, so there cannot be any more steps. ■

4 Decomposing Small Set Expanders

In this section we prove that if a graph is a small set expander, i.e., every set of size at most δn expands, then there is a way to decompose the graph by partitioning the vertices into at most $1/\delta$ sets, such that there are few edges between the parts, and each part is an expander in the usual sense.

We remark that similar decompositions have been analyzed previously (e.g., [KVV04, Tre05, GMR+11]), but we require a decomposition that is guaranteed to have a *constant* number of parts.

Theorem 1.2. *Let $R > 0$ and let $G = (V, E)$ be a graph in which for every set $S \subset V$ of size at most δn, $E(S, \bar{S}) \geq R \cdot |S|$. Then for every $\tau \leq R \cdot (\frac{\delta}{4})^{1/\delta}/2$ there is a partition $V = V_1 \cup \cdots \cup V_t$ into $t \leq 1/\delta$ parts such that*

1. *(Each part is an expander:) The graph $G(V_i)$ is a τ-expander for each i, (namely every set S has at least $\tau\,|S|$ outgoing edges).*

2. *(Large parts:)* $|V_i| \geq \delta n$, *where* $n = |V|$.

3. *(Few edges between parts:)* $E(V_i, V \setminus V_i) \leq r' \cdot |V_i|$ *for* $r' = 2\tau \cdot (\frac{4}{\delta})^{1/\delta}$.

Our approach is straightforward, we begin with G and iteratively partition it according to a small cut. At every step we take a part that's not yet an expander and split it again. However, to make this work it seems much more convenient to split on the so-called 'sparsest' cut, motivating the following definition.

Definition 4.1. A cut (A, B) in a graph G is a partition of the vertices into two sets A, B. The cut has sparsity r if $r(A, B) := \frac{|E(A,B)|}{|A||B|} \leq r$.

Note that the sparsity of the cut is roughly equal to n times the expansion of the cut (defined as $\frac{|E(S,\bar{S})|}{\min(|S|,|\bar{S}|)}$, since, letting $|S| \leq |\bar{S}|$,

$$\frac{n}{2} \cdot r(S, \bar{S}) \leq \frac{E(S, \bar{S})}{|S|} = |\bar{S}| \cdot r(S, \bar{S}) \leq n \cdot r(S, \bar{S}).$$

The non-trivial part in our proof is to show that the number of components in the decomposition is bounded independently of n. We show that iterative process never splits a set into parts smaller than δn, which means it ends after $\leq 1/\delta$ steps. Suppose that at step i the process splits a subset S into A and $B = S \setminus A$. There is no a priori guarantee that $|A| \geq \delta n$, since the cut between A and $V \setminus A$ is not necessarily sparse. The next lemma allows us to prove inductively that this does hold. It is probably known or folklore but we weren't able to find a reference.

Lemma 4.2. *Let* $G = (V, E)$ *be a graph, let* $(S, \bar{S})$ *be a sparsest cut of sparsity* $r = r(S, \bar{S})$*, and assume that* $\delta \leq \frac{|S|}{|V|} \leq 1 - \delta$. *Consider* $G' = G(\bar{S})$ *and let* A, B *be a cut in* G'*, of sparsity* r'. *Then*

$$r_A := \frac{|E(A, V \setminus A)|}{|A|\,|V \setminus A|} \leq 2(r + r')/\delta$$

We defer the proof of the lemma and proceed to prove the theorem.

Proof. (of Theorem 1.2) We perform an iterative process of refining partitions of the vertices. Let $\tau^* = \tau \cdot \frac{2}{n}$. At each step the process takes a set S from the current partition finds a τ^*-cut in $G(S)$ and splits S according to this cut. The process terminates when for every set in the partition $G(S_i)$ has no τ^*-sparse cut. This implies that $G(S_i)$ is a τ-expander, since

$$\forall T \subset S_i,\ |T| \leq |S_i|/2, \qquad E(T, S_i \setminus T) \geq \tau^* |T|\,|S_i \setminus T| \geq \tau^* \frac{n}{2} |T| = \tau |T|.$$

This process has a tree-like structure, with the set V at the root, and where at each step a leaf S is either already a τ-expander, or is split into A, B, which become its two offspring in the tree. We prove by induction that each new set generated by the process has few edges going out of it to the rest of the graph. More accurately,

Claim 4.3. *Let* V' *be a set generated by the process at a certain step, such that its depth in the tree is* i*, then* $i \leq 1/\delta$ *and any partition* (A, B) *of* V' *such that* $r(A, B) \leq r^*$ *implies that* $r(A, V \setminus A) \leq (4/\delta)^i \cdot \max(\tau^*, r^*)$.

Proof. (of claim) Let $V' = V_1, V_2, \ldots, V_{i+1} = root$ be the path from V' to the root. We prove this claim by induction on the depth i. For $i = 0$ there is nothing to prove. Assume that the claim holds for $i-1$. We first prove that all sets S sitting in the tree at depth at most i have size at least δn. Indeed we apply the inductive hypothesis on the parent of S (whose depth is $i-1$) to deduce

$$r(S, V \setminus S) \leq \tau^* \cdot (4/\delta)^{i-1} \leq \tau^* \cdot (4/\delta)^{1/\delta},$$

where τ^* is a bound on the sparsity of the cut between S and $P \setminus S$ where P is the parent of S. This implies

$$E(S, V \setminus S) \leq \frac{|S|\,|V \setminus S|}{n} 2\tau(4/\delta)^{1/\delta} \leq \min(|S|, |V \setminus S|) \cdot 2\tau(4/\delta)^{1/\delta} \leq R \cdot \min(|S|, |V \setminus S|)$$

yet this contradicts the small set expansion property unless $\min(|S|, |V \setminus S|) \geq \delta n$.

We now invoke Lemma 4.2 on the graph $G = G(V_2)$, setting $\bar{S} = V_1$, $S = V_2 \setminus V_1$, and A, B are a partition of V_1. Note that the cut between $S, \bar{S}$ is indeed a sparsest cut, and that both $|S|$ and $\left|\bar{S}\right|$ have size at least δn. We deduce that

$$r(A, V_2 \setminus A) \leq \frac{2}{\delta} \cdot \max(r^*, \tau^*).$$

We now apply the inductive hypothesis with $V' = V_2$, whose depth is $i-1$, and on the partition $(A, V_2 \setminus A)$ to deduce that

$$r(A, V \setminus A) \leq (4/\delta)^{i-1} \cdot \max(\tau^*, 4/\delta \max(\tau^*, r^*)) \leq (4/\delta)^i \max(\tau^*, r^*).$$

■

This completes the entire proof since by applying the claim on each set S in the process (with $r^* = \tau^*$) we have that the sparsity of the cut defined by this set is at most $(4/\delta)^{1/\delta} \cdot \tau^*$ which means that the number of edges going out of S is at most $2\tau(4/\delta)^{1/\delta} \cdot \min(|S|, |V \setminus S|)$. ■

We now turn to prove Lemma 4.2.

Proof. (of Lemma 4.2) The (natural) idea is to prove that if A has too many edges outgoing into S then the cut $(S \cup A, B)$ would be sparser than the cut $(S, A \cup B)$, thereby contradicting it being sparsest.

Note first that it is enough to bound $r(A, S)$ appropriately since

$$r_A = \frac{|E(A,B)| + |E(A,S)|}{|A|\,|B \cup S|} \leq \frac{|E(A,B)|}{|A|\,|B|} + \frac{|E(A,S)|}{|A|\,|S|} = r' + r(A,S).$$

We have $\bar{S} = A \cup B$ and so $\left|E(\bar{S}, S)\right| = |E(A,S)| + |E(B,S)|$. Dividing by $|S|\,(|A| + |B|)$, it follows that

$$r = \frac{|A|}{|A|+|B|} \cdot \frac{|E(A,S)|}{|A| \cdot |S|} + \frac{|B|}{|A|+|B|} \cdot \frac{|E(B,S)|}{|B| \cdot |S|},$$

so r is the weighted average of the cuts between S and B and between S and A:

$$r = \frac{|A|}{|A|+|B|} \cdot r(A,S) + \frac{|B|}{|A|+|B|} \cdot r(B,S). \tag{1}$$

Since the case $r(A,S) \leq r$ is done above we can safely assume that $r(A,S) > r$ which means, because of (1), that $r(B,S) < r$.

Next, note that if $|A| / |A \cup B| \geq \delta$ we are done, since

$$r(A,S) = \frac{|E(A,S)|}{|A|\,|S|} \leq \frac{|E(A,S)| + |E(B,S)|}{|A \cup B|\,|S|} \cdot \frac{|A \cup B|}{|A|} = r(A \cup B, S) \cdot \frac{|A \cup B|}{|A|} \leq r'/\delta.$$

So from now on we assume that $|A| / |A \cup B| < \delta$. Now, since the cut $(S, A \cup B)$ is a sparsest cut, we have

$$\frac{|E(S, A \cup B)|}{|S|\,(|A| + |B|)} = r(S, A \cup B) \leq r(S \cup A, B) = \frac{|E(S \cup A, B)|}{(|S| + |A|)\,|B|} \tag{2}$$

Let e_{AB} denote the number of edges between A and B. Similarly define e_{AS}, e_{BS}. Multiplying the above by $|S|\,(|A| + |B|)$ we get

$$e_{AS} \leq e_{AB} \cdot \frac{|S|\,(|A| + |B|)}{(|S| + |A|)\,|B|} + e_{BS} \cdot \left(\frac{|S|\,(|A| + |B|)}{(|S| + |A|)\,|B|} - 1\right) \tag{3}$$

To end the proof we will bound the factors multiplying e_{BS} and e_{AB},

1. $\frac{|S|(|A|+|B|)}{(|S|+|A|)|B|} - 1 = \frac{|A|(|S|-|B|)}{|B|(|S|+|A|)} \leq \frac{|A|}{|B|}$
2. $\frac{|S|(|A|+|B|)}{(|S|+|A|)|B|} \leq \frac{|S|}{|B|} \cdot \frac{1}{\delta}$.

Finally, dividing both sides of (3) by $|A|\,|S|$ we get the required bound on $r(A,S)$:

$$r(A,S) \leq \frac{1}{\delta} \cdot r(A,B) + r(B,S) \leq r'/\delta + r.$$

■

5 Acknowledgement

We thank Oded Goldreich for many things.

References

[ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Proc. 51st IEEE Symp. on Foundations of Computer Science*, pages 563–572, 2010.

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy: extended abstract. In *Proc. 40th ACM Symp. on Theory of Computing*, pages 21–28, 2008.

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[AM08] P. Austrin and E. Mossel. Approximation resistant predicates from pairwise independence. In *23rd Annual IEEE Conference on Computational Complexity*, pages 249–258. IEEE Computer Society, 2008.

[ARV09] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2), 2009.

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[BGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. *SIAM Journal on Computing*, 36(4):889–974, 2006. In special issue on Randomness and Computation.

[BSHR05] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF properties are hard to test. *SIAM J. Comput.*, 35(1):1–21, 2005.

[Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *IEEE Conference on Computational Complexity*, page 5, 2002.

[GKS09] Elena Grigorescu, Tali Kaufman, and Madhu Sudan. Succinct representation of codes with applications to testing. In *RANDOM-APPROX*, pages 534–547, 2009.

[GMR+11] Venkatesan Guruswami, Yury Makarychev, Prasad Raghavendra, David Steurer, and Yuan Zhou. Finding almost-perfect graph bisections. In *ICS*, pages 321–337, 2011.

[GS06] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *J. of the ACM*, 53(4):558–655, 2006.

[Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775. ACM Press, 2002.

[KS08] Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *Proc. 40th ACM Symp. on Theory of Computing*, pages 403–412, 2008.

[KVV04] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proc. 40th ACM Symp. on Theory of Computing*, pages 245–254, 2008.

[RS10] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proc. 42nd ACM Symp. on Theory of Computing*, 2010.

[Tre05] Luca Trevisan. Approximation algorithms for unique games. In *Proc. 46th IEEE Symp. on Foundations of Computer Science*, pages 197–205, 2005.