Local Computation and Reducibility

Kenji Christopher Obata

Electrical Engineering and Computer Sciences
University of California at Berkeley

May 17, 2006

Acknowledgement

See body of thesis.

Local computation and reducibility

by

Kenji Christopher Obata

B.Sc. (Yale University) 1999

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Luca Trevisan, Chair
Professor Satish Rao
Professor David Aldous

Spring 2006

The dissertation of Kenji Christopher Obata is approved:

_____

Chair   Date

_____

Date

_____

Date

University of California, Berkeley

Spring 2006

Local computation and reducibility

Copyright 2006

by

Kenji Christopher Obata

Abstract

Local computation and reducibility

by

Kenji Christopher Obata

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Luca Trevisan, Chair

A large body of recent work has been concerned with algorithms requiring access to only a sublinear, or even constant, sample of input bits. We study fundamental limitations in this model of computation and relationships to classical problems in combinatorial optimization.

———————————————————————

Professor Luca Trevisan
Dissertation Committee Chair

For my parents,

Contents

Acknowledgments

I gratefully acknowledge the guidance and support of my academic advisor Luca Trevisan, whose generous supply of insight and patience was essential to the development of virtually all of the ideas contained in this work. I also thank my dissertation committee members Satish Rao and David Aldous for their valuable feedback and encouragement.

I enjoyed countless memorable experiences and enlightening discussions while studying at Berkeley. I must especially thank Andrej Bogdanov – variously cook, co-author, antagonist, therapist, and drinking partner – for his consistent friendship. Many thanks are due to my fellow graduate students, including Scott Aaronson, Kamalika Chaudhuri, Kevin Chen, Holly Fait, Chris Harrelson, Matthew Harren, Lawrence Ip, James Lee, David Molnar, David Ratajczak, Samantha Riesenfeld, Stephen Sorkin, Kunal Talwar, and Hoeteck Wee. I was fortunate as an undergraduate to have enjoyed the inspiration of many outstanding educators, of whom I particularly acknowledge Michael Fischer, Ravi Kannan, and László Lovász. I am indebted to the Antolin family for their warm hospitality and support. Finally, I thank my parents, to whom this dissertation is dedicated.

Chapter 1

Introduction

Sampling is a powerful technique for estimating properties of large systems. Motivated by the revolution in probabilistically checkable proofs, Goldreich, Goldwasser and Ron [GGR98] initiated a study of the application of sampling techniques to traditional algorithmic problems, such as graph coloring and connectivity. When available, such algorithms offer astonishing performance advantages over traditional computational methods.

In this work, we study fundamental limitations in this model of computation. In particular, we show that 3CNF satisfiability and related problems are uncomputable by sampling-based algorithms. The core of our proof is the construction of families of constraint problems for which global satisfiability cannot be estimated from local considerations. We then introduce a notion of local reducibility between combinatorial problems analogous to Turing reducibility in traditional complexity theory, and apply this to derive explicit combinatorial constructions of graphs for which 3-colorability is undecidable by local algorithms.

Finally, we consider two variations on the theme of local computability. We study error-correcting codes for which decoding can be performed by local algorithms, and prove exponential lower bounds on the size of such local coding schemes. We then explore relationships between our analysis of local coloring and classical problems in combinatorial optimization, yielding new approximation algorithms for integral multicommodity flow problems and related integral-flow/multicut duality relationships.

Chapter 2

Local computation

We begin by making precise our notions of local computability and approximation.

Throughout, we assume standard definitions and notations from computational complexity theory (Turing Machines, oracles, asymptotic notation, and so forth). The reader is referred to [Pap93] for these standard definitions. We work always with some finite alphabet Σ. For a given Σ, we denote by Σ^n the set of all ordered n-tuples of elements of Σ and by Σ^* the union $\bigcup_n \Sigma^n$. A *language* $\mathcal{L}$ on Σ is some particular subset $\mathcal{L} \subseteq \Sigma^*$ (the set of encodings of instances having some property of interest).

Roughly speaking, a language $\mathcal{L}$ is locally computable if there exists an algorithm which determines whether a given string $x \in \Sigma^n$ lies in $\mathcal{L}$ and which queries only a sublinear number of entries of x. For many non-trivial problems, it turns out that the number of queries may even be independent of the size of the input.

Clearly, for non-trivial problem domains $\mathcal{L}$, such algorithms must be randomized and allowed some probability of failure. Also, we can only demand that sublinear algorithms

distinguish between inputs in $\mathcal{L}$ and inputs which are *far* from $\mathcal{L}$ in the sense that a constant fraction of entries must be modified to carry the input into $\mathcal{L}$.

(The situation is closely analogous to that of classical statistical sampling – one is able to count the number of objects having a given property from among a large set by inspecting only a constant number of randomly selected objects. However, this tremendous efficiency comes at the expense of a small probability of failure and a small error in one's estimate of the number of objects.)

We shall use standard notions of distance between strings and sets.

Definition 1 (Distance) *For any alphabet Σ and $x, y \in \Sigma^n$ for $n \in \mathbf{N}$, we say that x and y are at* distance $d(x, y) = \varepsilon$ *if x and y disagree on εn entries. For sets $X, Y \subseteq \Sigma^n$, we define*

$$d(X, Y) = \min_{x \in X, y \in Y} d(x, y).$$

In particular, if $\mathcal{L}$ is a language on alphabet Σ and $x \in \Sigma^n$ for some $n \in \mathbf{N}$, then

$$d(x, \mathcal{L}) = \min_{y \in \mathcal{L}} d(x, y).$$

(In our applications, $\Sigma^n \bigcap \mathcal{L}$ will always be non-empty so this distance is well-defined.)

Combining this with an allowance for (one- or two-sided) error, we have the following natural definition for local computability.

Definition 2 (Local computability) *We say that a language $\mathcal{L}$ is locally computable with complexity $c(n)$ and distance parameter ε if there exists a probabilistic algorithm $A(n)$*

and constants $\delta_1, \delta_2, \varepsilon \geq 0$ such that, given access to an oracle O for input $x \in \Sigma^n$, $A^O(n)$

has the following properties:

  (i) $A^O(n)$ makes at most $c(n)$ invocations to the oracle O.

  (ii) For all $x \in \Sigma^n$, if $x \in \mathcal{L}$ then $A^O(n)$ accepts with probability at least $1 - \delta_1$.

  (iii) For all $x \in \Sigma^n$, if $d(x, \mathcal{L}) \geq \varepsilon$ then $A^O(n)$ rejects with probability at least $1 - \delta_2$.

(This definition was introduced by Goldreich, Goldwasser, and Ron [GGR98], who used the term *property testing* to describe this setting. We believe the term *local computability* is somewhat more general and descriptive, so we use this term throughout.)

In the special case where $\delta_1 = 0$, we say that A has *one-sided error*. Otherwise, A has *two-sided error*.

We are especially interested in languages $\mathcal{L}$ for which $c(n)$ is a *sublinear* function in n; that is, $c(n) = o(n)$. In this case, we say simply that $\mathcal{L}$ is *locally computable*. Goldreich and Ron [GR97, GR99] showed, for example, that in an appropriate representation, connectivity is locally computable with $c(n) = O(1)$ and bipartiteness with $c(n) = \tilde{O}(\sqrt{n})$.

Encodings and representation dependency

In the traditional computational setting, the particular choice of representation for the input is typically irrelevant to the computational complexity of the problem domain. For instance, whether a graph is represented in an adjacency list or an adjacency matrix representation is insignificant, since either representation can be converted to the other in polynomial time. In local computational models, however, we do not enjoy the luxury of

even linear time in which to perform this transformation, so one must be careful to specify the particular representation to be used.

Note also that our notion of distance is strongly dependent on the choice of representation – two objects which are close in one representation may be far in another. For example, consider the set of connected graphs: If represented as a bounded-degree adjacency list, the empty graph on n vertices is ε-far from the set of connected graphs for any $\varepsilon < 1$ and sufficiently large n; on the other hand, any graph is asymptotically ε-close to a connected graph when represented as an adjacency matrix.

Throughout this work, we will deal frequently with simple, bounded-degree graphs and $(3, k)$CNF formulas. We shall use always sparse representations of our graphs and formulas. That is, for graphs G with degree bound Δ, we encode G in an adjacency list representation. Similarly, for $(3, k)$CNF formulas, we encode the instance as a membership list which provides, for each variable, the indices of the clauses in which the variable appears, along with the list of clauses, encoded as the indices of each variable in the clause along with a bit indicating whether each variable appears negated.

For more information and results related to alternate representations, the reader is referred to [GR97], [AFKS99], and [KKR04].

Chapter 3

Lower bounds for CSPs

We consider the local computability of constraint satisfaction problems (CSPs). In particular, we prove that there cannot exist local algorithms to decide graph 3-colorability and E3LIN2 satisfiability. The results in this chapter are joint work with Andrej Bogdanov and Luca Trevisan [BOT02].

As a first step, we show that no local algorithm with one-sided error, given a degree Δ graph with n vertices, can view fewer than δn entries of the adjacency list representation of the graph, yet reject with constant probability graphs that are ε-far from 3-colorable. To this end, a simple observation is that a one-sided local algorithm must accept whenever its view of the graph is 3-colorable. In other words, it is sufficient to construct a graph G that is ε-far from 3-colorable, yet every one of its induced subgraphs on δn edges is 3-colorable.

In Section 3.1 we give a probabilistic construction of such graphs based on a technique due to Erdős [Erd62]. For every $\alpha > 0$, there exist constants $\Delta = O(1/\alpha^2)$ and $\delta > 0$ such that some Δ-regular graph on n vertices is $(1/3 - \alpha)$-far from 3-colorable, yet

every subgraph induced by up to δn edges is 3-colorable. As a consequence, we obtain:

Theorem 3 *For every $\alpha > 0$ there exist constants Δ and $\delta > 0$ such that if A is a local algorithm for distinguishing 3-colorability on degree Δ graphs with one-sided error and distance parameter $1/3 - \alpha$, then the query complexity of A is at least δn, where n is the number of vertices.*

Notice that no graph is more than 1/3-far from being 3-colorable, so our result applies to the full spectrum of gaps for which the problem is well defined. Furthermore, for small enough α, the distinguishing problem is solvable deterministically in polynomial time with the Frieze-Jerrum algorithm [FJ97]. This gives a separation between the distinguishing ability of polynomial time versus (one-sided error) sublinear time algorithms for a natural problem.

To prove a lower bound for two-sided error algorithms, by Yao's principle, it is enough to construct two distributions $\mathcal{G}_{3col}$ and $\mathcal{G}_{far}$ over bounded-degree graphs such that graphs in $\mathcal{G}_{3col}$ are always 3-colorable, graphs in $\mathcal{G}_{far}$ are typically far from being 3-colorable, and the two distributions are indistinguishable by algorithms with sublinear query complexity.

Towards this goal, we will first construct two distributions of instances of E3LIN2, $\mathcal{D}_{sat}$ and $\mathcal{D}_{far}$, such that instances in $\mathcal{D}_{sat}$ are always satisfiable and instances in $\mathcal{D}_{far}$ are typically far from satisfiable, yet the two distributions are indistinguishable by algorithms with sublinear query complexity. Using the notion of local reducibility introduced in Chapter 4, we will later reduce E3LIN2 to 3-coloring and argue that the transformation preserves, respectively, satisfiability and 3-colorability, as well as farness from satisfiabil-

ity and 3-colorability. Moreover, an oracle for a reduced instance can be implemented in constant time given the original instance.

In order to define $\mathcal{D}_{sat}$ and $\mathcal{D}_{far}$, we show that for every c there exists $\delta > 0$ such that there is an E3LIN2 instance I with n variables and cn equations such that any subset of δn equations are linearly independent. We do so using a probabilistic argument. We then define $\mathcal{D}_{sat}$ to be the distribution of instances obtained by first picking an assignment to the variables, and then setting the right-hand side of I to be consistent with the assignment. In $\mathcal{D}_{far}$ we set the right-hand side of I uniformly at random. For algorithms that look at less than a δ fraction of equations, the two distributions are identical. However instances in $\mathcal{D}_{sat}$ are always satisfiable while instances in $\mathcal{D}_{far}$ are about $(1/2 - O(1/\sqrt{c}))$-far from satisfiable, except with negligibly small probability. Thus we obtain:

Theorem 4 *There exist universal constants δ, ε such that if A is a local algorithm for distinguishing E3LIN2 satisfiability with distance parameter ε, then the query complexity of A is at least δn, where n is the number of variables.*

3.1 Probabilistic constructions

In this section we provide probabilistic constructions of combinatorial objects (graphs and 3-hypergraphs) that will be used to obtain problem instances for 3-colorability and E3LIN2 for which it is difficult to distinguish, respectively, colorability and satisfiability.

We shall make frequent use of standard constructs and techniques from probability theory such as martingales, tail inequalities, and so forth. The reader is referred to [MR95] for these definitions and proofs.

Graphs and hypergraphs with no small dense subgraph

For purposes of this section, it will be somewhat more convenient to work with multigraphs instead of graphs. We consider a distribution $\mathcal{G}$ on n-vertex multigraphs G (where n is even) obtained as follows:

Let $C_1, \ldots, C_\Delta$ be independent random perfect matchings on the vertices of G. The edge set of G is the multiset union of the C_i, so that the multiplicity of an edge equals the number of matchings C_i in which it appears. If $(u, v) \in C_i$, we say that v is the i-th neighbor of u in G.

We denote by $G|_S$ the restriction of multigraph G on vertex set $S \subseteq V(G)$. Let X_S be the number of edges in $G|_S$. Then $\mathrm{E}(X_S) = \Delta\binom{|S|}{2}\frac{1}{n-1}$. Fix a partition $\{S_1, S_2, S_3\}$ of $V(G)$. We are interested in bounding the probability that this partition is $1/3$-close to a valid coloring of G. Let $X = X_{S_1} + X_{S_2} + X_{S_3}$.

Lemma 5 *For every partition $\{S_1, S_2, S_3\}$ of $V(G)$ and every constant $\alpha > 0$,*

$$\Pr(X < (1/6 - \alpha)\Delta n) \leq \exp(-(\alpha - o(1))^2 \Delta n).$$

Proof Consider the random process $I_1, \ldots, I_{\Delta n/2}$ on G, which reveals the edges of G one by one. For a fixed partition $\{S_1, S_2, S_3\}$, the random variable X determines a Doob martingale with respect to this process. A simple computation shows that for $1 < j \leq \Delta n/2$,

$$|\mathrm{E}(X|I_1, \ldots, I_j) - \mathrm{E}(X|I_1, \ldots, I_{j-1})| \leq 1.$$

By convexity, $\mathrm{E}(X) \geq \frac{\Delta n}{6}\frac{n-3}{n-1}$ (this value is attained when $|S_1| = |S_2| = |S_3| = n/3$).

Azuma's inequality yields

$$\Pr\left(X < \left(\frac{1}{6}\frac{n-3}{n-1} - \alpha'\right)\Delta n\right) \le \exp(-\alpha'^2 \Delta n).$$

The conclusion follows, with $\alpha = \alpha' + \frac{1}{3(n-1)}$. ∎

Denote by $\bar{G}$ the graph obtained by identifying every multiedge of G with an ordinary edge.

Lemma 6 *For any constant $\alpha > 0$ there exists a constant Δ such that, with probability $1 - o(1)$, any 3-coloring of the vertices of $\bar{G}$ has at least $(1/6 - \alpha)\Delta n$ violating edges.*

Proof First we show that the conclusion holds for G. The number of tripartitions of $V(G)$ is 3^n. By combining a union bound with the bound from Lemma 5, it follows that any such partition has $(1/6 - \alpha)\Delta n$ violating edges if $\Delta > \ln 3/\alpha^2$.

For any pair of vertices (u, v), let $M_{u,v}$ indicate the event that (u, v) is an edge of G with multiplicity two or more. Then $\Pr(M_{u,v} = 1) = O(\Delta/n^2)$. By Markov's inequality, the probability that there are $\Delta \log n$ or more pairs (u, v) with $M_{u,v} = 1$ is $o(1)$. Since no edge of G has multiplicity more than Δ, it follows that $|E(G)| - |E(\bar{G})| \le \Delta^2 \log n = o(n)$. Therefore, the conclusion of the lemma carries over to $\bar{G}$. ∎

Lemma 7 *For every $K > 1$ there exists $\delta > 0$ such that with probability $1 - o(1)$, all graphs $\bar{G}|_S$ with $|S| \le \delta n$ have at most $K|S|$ edges.*

Proof Suppose some set S of cardinality s contains Ks edges $(u_1, v_1), \ldots, (u_{Ks}, v_{Ks})$. Denote by $X_{i,k}, Y_{i,k}$ the vertices matched to u_i and v_i, respectively, in the matching C_k.

Then

$$\Pr\left(\exists k : X_{i,k} = v_i \wedge Y_{i,k} = u_i | X_{p,q}, Y_{p,q} : 1 \le p \le i-1, 1 \le q \le \Delta\right) \le \Delta/(n-2s),$$

since for any fixed q, the variables $X_{p,q}$ and $Y_{p,q}$ determine the neighbors of at most $2s$ vertices in matching C_k. It follows that

$$\Pr(\forall i, 1 \le i \le \Delta : \exists k : X_{i,k} = b_i \wedge Y_{i,k} = a_i)$$
$$\le \left(\frac{\Delta}{n-2s}\right)^{Ks}$$
$$< \left(\frac{\Delta}{(1-2\delta)n}\right)^{Ks}.$$

For fixed s, the set S can be chosen in $\binom{n}{s}$ ways, while the set $\{(u_1, v_1), \ldots, (u_{Ks}, v_{Ks})\}$ can be chosen in $\binom{\binom{s}{2}}{Ks}$ ways. Therefore, for some constant s_0,

$$\Pr(\exists S, s_0 \le |S| < \delta n : |E(G|_S)| K|S|)$$
$$\le \sum_{s=s_0}^{\delta n} \binom{n}{s} \binom{\binom{s}{2}}{Ks} \left(\frac{\Delta}{(1-2\delta)n}\right)^{Ks}$$
$$\le \sum_{s=s_0}^{\delta n} \left(\frac{ne}{s}\right)^s \left(\frac{s^2 e/2}{Ks}\right)^{Ks} \left(\frac{\Delta}{(1-2\delta)n}\right)^{Ks}$$
$$= \left(\frac{e^2 \Delta}{2} \left(\frac{e\Delta}{2K(1-2\delta)}\right)^K \left(\frac{s}{n}\right)^{K-1}\right)^s = o(1).$$

It is easy to see that the contribution of sets S of size less than s_0 is also $o(1)$. ∎

We define an analogous distribution $\mathcal{H}$ on 3-hypergraphs (hypergraphs with multiple hyperedges where each hyperedge has cardinality 3) with n vertices, where n is a

multiple of 3. To obtain a graph $H \sim \mathcal{H}$, we choose Δ independent uniformly random partitions of the vertex set $V(H)$ into 3-hyperedges (3-element subsets). With probability $1 - o(1)$, all hyperedges of H have multiplicity one. An argument similar to the proof of Lemma 7 shows the following.

Lemma 8 *For every $K > 1/2$ there exists $\delta > 0$ such that with probability $1 - o(1)$, all 3-hypergraphs $H|_S$ with $|S| \leq \delta n$ have at most $K|S|$ edges.*

Hard instances

We now show the existence of graphs that are almost 1/3-far from 3-colorable yet, for some $\delta > 0$, all vertex-induced subgraphs of size δn are 3-colorable.

Choose a multigraph G according to the distribution $\mathcal{G}$ of Section 3.1, and let $\bar{G}$ denote the graph obtained from G by ignoring multiplicities. We show that the graph $\bar{G}$ has the desired property. As in [Erd62], we use the fact that every vertex in any minimal non-3-colorable subgraph has degree at least three.

Theorem 9 *For every $\alpha > 0$ there exists $\delta > 0$ such that with probability $1 - o(1)$, the graph $\bar{G}$ is $(1/3 - \alpha)$-far from 3-colorable, yet all subgraphs $G|_S$ with $|S| < \delta n$ are 3-colorable.*

Proof By Lemma 6 with parameter $\alpha/2$, every tripartition of $V(\bar{G})$ has at least $(1/3 - \alpha)\Delta n/2$ violating edges, so $\bar{G}$ is 1/3-far from 3-colorable.

Suppose that there exists a set S of size $s < \delta n$ such that $\bar{G}|_S$ is not 3-colorable. We may assume that S is a minimal set with this property. Suppose that $\bar{G}|_S$ contains a vertex v of degree two or less (with respect to $\bar{G}|_S$). By the minimality of S, there is a 3-coloring of the graph $\bar{G}|_{S-\{v\}}$. However, this coloring extends to a 3-coloring of $\bar{G}|_S$, by

picking a color for v that does not match any of its neighbors. It follows that any vertex in $\bar{G}|_S$ must have degree at least 3. Therefore, $\bar{G}|_S$ must contain at least $3s/2$ edges. By Lemma 7 with $K = 3/2$, this is not possible. ∎

Using the 3-hypergraph construction, we prove the existence of certain matrices that will later be used as the left hand side of hard E3LIN2 instances.

Theorem 10 *For every $c > 0$ there exists a $\delta > 0$ such that for every n there exists a matrix $A \in \{0,1\}^{n \times cn}$ with n columns and cn rows, such that each row has exactly three non-zero entries, each column has exactly $3c$ non-zero entries, and every collection of δn rows is linearly independent.*

Proof By Lemma 8, there exists a $3c$-regular 3-hypergraph H on n vertices such that any $H|_S$ with $|S| \leq 3\delta n$ has strictly fewer than $2|S|/3$ edges. Let A be the incidence matrix of H: The columns of A correspond to vertices of H, the rows of A correspond to hyperedges of H, and $A_{ve} = 1$ if and only if $v \in e$. Suppose that there is a set R of δn rows of A (or hyperedges of H) that are linearly dependent. We may assume that R is a minimal set with this property. Let $S \subseteq V(H)$ denote the set of vertices incident to hyperedges in R, so that $|S| \leq 3\delta n$. By minimality of R, every element of S must appear in at least two rows of R. Therefore, R contains at least $2|S|/3$ hyperedges. Contradiction. ∎

3.2 Explicit constructions

In this section, we give an explicit construction of an infinite family of CSPs on n variables and $m = O(n)$ clauses over a fixed boolean predicate such that any subset of δm

clauses is satisfiable, but no assignment satisfies more than $(1 - \varepsilon)m$ clauses. By applying the gap-preserving local reductions presented in Chapter 4, we shall achieve an explicit construction of an infinite family of bounded degree graphs G on n vertices and m edges such that every subgraph induced by δm edges is 3-colorable, but any 3-coloring of G has at least εm monochromatic edges.

(In the previous section we used the probabilistic method to prove only the existence of such graphs.)

Since we will deal frequently with partially satisfiable constraint satisfaction problems, we introduce the following notation.

Definition 11 $((\delta, 1 - \varepsilon)$-satisfiability) *A constraint satisfaction problem on m clauses is $(\delta, 1 - \varepsilon)$-satisfiable if any subset of at most δm constraints is satisfiable, but no assignment satisfies more than $(1 - \varepsilon)m$ constraints.*

For a fixed Δ, we will consider 2Δ-ary constraints of the form

$$h : \{0, 1\}^{\Delta} \times \{0, 1\}^{\Delta} \to \{0, 1\},$$

where $h(x_1, \ldots, x_\Delta, y_1, \ldots, y_\Delta)$ is satisfied exactly when $\sum_{i=1}^{\Delta} x_i = \sum_{i=1}^{\Delta} y_i + 1$, and we identify the boolean $\{0, 1\}$ inputs with the integers 0 and 1 in the obvious way.

Let $G(V, E)$ be an undirected multigraph. We write $\Gamma(v)$ for the neighbor set of vertex $v \in V$, $\Gamma(v, i)$ for the i-th neighbor of v (where we index $\Gamma(v)$ in an arbitary way), and $\Gamma(S)$ for the neighbor set of a vertex-subset $S \subseteq V$.

Our constructions shall make extensive use of *expander graphs*, particularly graphs with the following strong expansion property.

Definition 12 ((n, Δ)-Expander) *A multigraph G is an (n, Δ)-expander if it is Δ-regular and if, for every subset $S \subset V$ with $|S| \leq \frac{1}{2}|V|$, $|\Gamma(S)| \geq |S|$.*

Explicit constructions of (n, Δ)-expanders are known [Mar73, GG81], and we assume that we have an infinite family of (n, Δ)-expanders for some universal constant Δ.

Define the constraint satisfaction problem f_n on Δn variables and n clauses over h as follows: Let $G(V, E)$ be an (n, Δ)-expander. Begin by converting G into a directed multigraph $G'(V, E')$ by replacing each undirected edge $(i, j) \in E$ with two directed edges $(i, j), (j, i) \in E'$. Each edge $(i, j) \in E'$ is identified with a boolean variable $x_{i,j}$ in f_n. One constraint h is introduced for each $v \in V$, with the predicate variables mapped to the edges incident to v:

$$f_n = \bigwedge_{v \in V} h(x_{v,\Gamma(v,1)}, \ldots, x_{v,\Gamma(v,\Delta)}, x_{\Gamma(v,1),v}, \ldots, x_{\Gamma(v,\Delta),v})$$

Theorem 13 *There exist constants $\delta, \varepsilon > 0$ such that the CSP formulas f_n are $(\delta, 1 - \varepsilon)$-satisfiable.*

Proof   We begin by finding ε such that no subset of more than $(1 - \varepsilon)n$ constraints can be satisfied. Suppose there is an assignment satisfying some subset S of constraints with $|S| > (1 - \varepsilon)n$. Then the following network flow problem is solvable: Contract the vertices corresponding to $\overline{S}$ into a single sink vertex t, create a source vertex s with unit capacity edges from s to every vertex in S, and interpret the remaining edges of G as unit capacity

edges. The assignment can then be interpreted as an (s,t)-flow of weight greater than $(1-\varepsilon)n$ on this network. However, the cut $(t, G\backslash t)$ has weight at most $\Delta\varepsilon n$, so this is impossible if we choose $\varepsilon < \frac{1}{\Delta+1}$.

On the other hand, for $\delta \leq \frac{1}{2}$, any subset S of constraints with $|S| = \delta n$ can be satisfied. To see this, we define the following network flow problem: Contract the vertices of G corresponding to the $(1-\delta)n$ constraints in $\overline{S}$ to a sink vertex t, create a source vertex s with unit capacity edges from s to every node in S, and interpret the remaining edges of G as unit capacity edges. We claim that there is a flow of weight at least δn in this system. By the max-flow/min-cut theorem, it is enough to show that there is no (s,t)-cut with weight less than δn (the cut $(s, G\backslash s)$ has weight δn). Let C be an arbitrary (s,t)-cut, and denote by C_s, C_t the vertices of S in the partitions containing s and t respectively. Each node in C_t incurs a cut cost of weight one due to the unit constraint edges we added from s. By the strong expansion property, $|\Gamma(C_s)| \geq |C_s|$, and each of the edges connecting C_s to $\Gamma(C_s)$ also incurs a cut cost of weight one. Summing up, $|C| \geq |C_s| + |C_t| = \delta n$, so there must exist an flow of weight δn in this system. Furthermore, the *integrality property* of flows implies that we can assume the flow solution is $(0,1)$-valued. Assigning this flow to the edge variables gives a satisfying assignment to the constraints in S. ∎

3.3 Lower bounds

We are now prepared to prove Theorems 3 and 4.

Proof of Theorem 3

To prove Theorem 3, we observe that any local algorithm with one-sided error must accept whenever the subgraph it has queried is 3-colorable. In particular, when presented with the graphs constructed in Theorem 9, any algorithm with query complexity at most δn must accept with probability 1. However, these graphs are $(1/3 - \alpha)$-far from 3-colorable, so we cannot have a local algorithm for 3-colorability with parameter $1/3 - \alpha$.

Proof of Theorem 4

Our families of hard instances for two-sided error algorithm are derived from the matrix A constructed in Theorem 10.

We consider the following two families of distributions on instances of E3LIN2 with n variables, cn equations, and each variable appearing in exactly $3c$ equations:

(i) Distribution $\mathcal{D}_{far}$ consists of instances $Ax = b$, where $b \in \{0, 1\}^{cn}$ is chosen uniformly at random.

(ii) Distribution $\mathcal{D}_{sat}$ consists of instances $Ax = Az$, where $z \in \{0, 1\}^n$ is chosen uniformly at random.

By construction, every instance in $\mathcal{D}_{sat}$ is satisfiable. On the other hand, instances in $\mathcal{D}_{far}$ are far from satisfiable.

Lemma 14 *For every $\alpha > 0$, there exists c such that, with probability $1 - o(1)$, an instance sampled from $\mathcal{D}_{far}$ is $(1/2 - \alpha)$-far from satisfiable.*

Proof For a fixed assignment x, the vector $Ax - b$ is uniformly distributed in $\{0, 1\}^{cn}$.

By a Chernoff bound, with probability $1 - \exp(-\Omega(\alpha^2 cn))$, $Ax - b$ has Hamming weight at least $(1/2 - \alpha)cn$. A union bound over all 2^n possible assignments for x yields the desired result, as long as $c = \Theta(1/\alpha^2)$. ∎

Lemma 15 *For every $\alpha > 0$ there exist constants c and $\delta > 0$ such that every local algorithm for E3LIN2 satisfiability on n variables and at most c occurrences with distance parameter $1/2 - \alpha$ must have query complexity at least δn.*

Proof   Consider an instance $Ax = b$ of cn E3LIN2 equations. Obtain a subinstance $A'x' = b'$ by choosing *any* subset of δn equations. By Theorem 10, the rows of A' are linearly independent. Therefore, for a uniformly random $z' \in \{0, 1\}^n$, $A'z'$ is uniformly distributed in $\{0, 1\}^{\delta n}$. It follows that the instances $A'x' = b'$ and $A'x' = A'z'$ are generated with the same probability; that is, $\Pr_{\mathcal{D}_{far}}(A'x' = b') = \Pr_{\mathcal{D}_{sat}}(A'x' = b')$.

Let D be any algorithm of query complexity less than δn. If D can decide whether a given instance $Ax = b$ is satisfiable with any constant probability, then D has an advantage at distinguishing instances sampled from $\mathcal{D}_{sat}$ (which are always satisfiable) from instances sampled from $\mathcal{D}_{far}$ (which are $(1/2-\alpha)$-far from satisfiable with high probability). However, the queries of D only reveal a subinstance $A'x' = b'$ of at most δn equations, and the two distributions are statistically indistinguishable on such a subinstance. ∎

Theorem 4 follows immediately.

3.4 Approximation algorithms

We note some applications of our constructions to approximation algorithms for coloring. The following theorem follows directly from Lemma 15.

Theorem 16 *For every $\varepsilon > 0$, every $(1/2+\varepsilon)$-approximate algorithm for Max-E3LIN2 and every $(7/8+\varepsilon)$-approximate algorithm for Max E3SAT has query complexity $\Omega(n+m)$, where n is the number of variables and m is the number of equations/clauses. The theorem applies to the special case where every variable occurs in $O(1)$ equations/clauses and $m = O(n)$.*

Indeed, Lemma 15 is the unconditional version for sublinear time algorithms of the hardness of approximation proved in [Hås97] for Max-E3LIN2. Håstad [Hås97] then uses approximation preserving reductions to show that the hardness of Max-E3LIN2 implies hardness of approximation results for other problems. Since the reductions used in [Hås97] preserve the existence of sublinear time algorithms (for proper instance representation), we also have unconditional inapproximability results for other problems, with respect to sublinear-time algorithms.

The standard [FGL+96] reduction from Max-E3LIN2 to Vertex Cover is such that if every variable occurs in $O(1)$ equations in the E3LIN2 instance, then the graph produced by the reduction has constant degree. Therefore, the following result also follows from Lemma 15 (see [Hås97] for a calculation of the inapproximability factor).

Theorem 17 *For every $\varepsilon > 0$, there are constants d, δ such that every $(7/6+\varepsilon)$-approximate algorithm for Minimum Vertex Cover in graphs of degree $\leq \delta$ has query complexity at least δn.*

Similarly, we have a linear query complexity lower bound for every $(21/22 + \varepsilon)$-approximate algorithm for Max-2SAT, even for the restricted case where every variable occurs in $O(1)$ clauses.

Regarding Max-CUT, the reduction used in [Hås97] does not create a bounded-degree graph, even if in the original E3LIN2 instance every variable occurred in a bounded number of equations. However the reduction in [Tre01] can be used to show that every $(16/17 + \varepsilon)$-approximate algorithm for Max-CUT in bounded-degree graphs has linear query complexity.

Chapter 4

Local reducibility

In this chapter, we define a notion of reducibility between constraint satisfaction problems which preserves, up to modification of constants, the property that a decision problem has a sublinear algorithm, and exhibit such a reduction from $(3, k)$SAT to 3-colorability in bounded degree graphs.

By applying this reduction to the construction of Section 3.2, we achieve an explicit construction of an infinite family of bounded degree graphs G on n vertices and m edges such that every subgraph induced by δm edges is 3-colorable, but any 3-coloring of G has at least εm monochromatic edges. (In the proof of Theorem 9 we used the probabilistic method to prove only the existence of such graphs.) While natural in our context, the use of reductions in the explicit construction of combinatorial objects is a novel approach which seems interesting in its own right.

Similarly, applying this reduction to the constructions of Lemma 15 yields a generalization of Theorem 3 to local algorithms with two-sided error.

4.1   Local reducibility

For our purposes, the following notion of reduction will be appropriate.

Definition 18 (Gap-preserving local reduction) *Let $\mathcal{L}_1, \mathcal{L}_2$ be two languages. We say that a mapping φ is a gap-preserving local reduction from $\mathcal{L}_1$ to $\mathcal{L}_2$ if there exist universal constants $c_1, c_2 > 0$ such that the following properties hold:*

(i) *If $x \in \mathcal{L}_1$, then $\varphi(x) \in \mathcal{L}_2$.*

(ii) *If $d(x, \mathcal{L}_1) \geq \varepsilon$ then $d(\varphi(x), \mathcal{L}_2) \geq \varepsilon/c_1$.*

(iii) *The answer to an oracle query to $\varphi(x)$ can be computed by making at most c_2 oracle queries to x.*

We note three easy lemmas, which will allow us to move between various CSP formulations.

Lemma 19 *Let H be an arbitrary fixed set of boolean predicates on a finite number of variables. There exists a gap-preserving local reduction from CSPs defined on H which carries an instance f with n variables and m clauses into a 3CNF formula with $O(n + m)$ variables and $O(m)$ clauses.*

Proof   It is a basic fact that an arbitrary boolean predicate on a finite number of variables can be expressed as a 3CNF formula, possibly with introduction of a constant number of auxiliary variables. It is easy to check that applying this transformation to each clause of f gives a reduction which has the claimed properties. ∎

Lemma 20 *Gap-preserving local reductions are closed under composition.*

Proof   Trivially, if φ, φ' are gap-preserving local reductions with distortion constants c_1, c_2 and c_1', c_2' respectively, then $\varphi \circ \varphi'$ is a gap-preserving local reduction with distortion constants $c_1 c_1', c_2 c_2'$. ∎

Lemma 21 *If $\varphi : \mathcal{L}_1 \to \mathcal{L}_2$ is a gap-preserving local reduction with distortion constants c_1, c_2 and f is a $(\delta, 1 - \varepsilon)$-satisfiable CSP, then $\varphi(f)$ is a $(\delta/c_2, 1 - \varepsilon/c_1)$-satisfiable CSP.*

Proof   Let f_A be a $(\delta, 1 - \varepsilon)$-satisfiable instance of A, and $f_B = \varphi(f_A)$. That the problem f_B is $\frac{\varepsilon}{c_1}$-far from satisfiable is immediate from the definition of a gap-preserving local reduction. Now, let m be the number of clauses in problem f_B and consider any subset $C_1', \ldots, C_{k'}'$ of $\frac{\delta}{c_2}m$ of these clauses. By the locality property, these clauses are a function of some set of clauses $C_1, \ldots, C_k$ of f_A with $k \leq c_2 \frac{\delta}{c_2} m = \delta m$. Since f_A is $(\delta, 1 - \varepsilon)$-satisfiable, the clauses $C_1, \ldots, C_k$ are satisfiable, and we can extend these clauses to a new, satisfiable instance f_A' of A by setting every clause other than $C_1, \ldots, C_k$ to a satisfiable clause on fresh variables. φ must send f_A' into a satisfiable instance, and this instance contains clauses $C_1', \ldots, C_{k'}'$. In particular, the clauses $C_1', \ldots, C_{k'}'$ must be satisfiable. ∎

4.2   Local 3-colorability

We now exhibit a gap-preserving local reduction φ from $(3, k)$SAT to 3-coloring in bounded degree graphs. We comment that a reduction with essentially the same properties was given by Petrank in [Pet94]. However, Petrank's construction does not yield a bounded
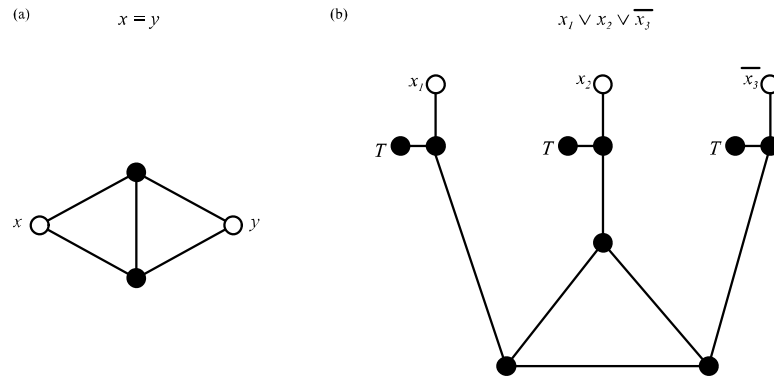
Figure 4.1: Gadgets for Theorem 22

degree graph, which is essential in our context. Also, our construction is somewhat simpler to describe and analyze.

Construction: Let f be the $(3, k)$CNF formula on n variables and m clauses to be mapped. First, we introduce a large set of nodes which are independent of the clauses of f which we label D_i, T_i, and F_i for $i = 1, \ldots, 2kn$. The nodes D_i will all assume the color corresponding to the "dummy" color (this color is used as in the standard 3-coloring reduction), T_i to the "true" color, and F_i to the "false" color. To assure that nodes in a given *color class* are the same color, we introduce equality gadgets (Figure 4.1.a) between nodes D_i and D_j for all $(i, j) \in E_{2kn}$ where $G_{2kn}(V_{2kn}, E_{2kn})$ is a $(2kn, \Delta)$-expander as in Lemma 2 (similarly for the classes T and F). To assure that nodes in distinct color classes have distinct colors, for $i = 1, \ldots, 2kn$ we introduce triangles $\{(D_i, T_i), (D_i, F_i), (T_i, F_i)\}$.

For each variable x_i in f, we introduce $2k$ literal nodes $x_i^1, \ldots, x_i^k, \overline{x_i^1}, \ldots, \overline{x_i^k}$. Literal nodes for a particular variable and sign should be colored identically, so we introduce equality constraints between x_i^j and $x_i^{j'}$ for all $1 \leq i, j \leq k$ with $i \neq j$ (similarly for $\overline{x_i^j}$ and $\overline{x_i^{j'}}$). We fix some one-to-one correspondence between the literal nodes and the color class

nodes for each color class (we can do so since we have $2kn$ nodes in each color class). Since literal nodes should be colored only with "true" or "false", every literal node is connected to its corresponding node D_i. Since only one of $x_i, \overline{x_i}$ can be true, we introduce edges $(x_i^j, \overline{x_i^j})$ for all i, j. Finally, for each clause in f, we introduce a clause gadget (Figure 4.1.b) on the literals appearing in the clause. We can do so in such a way that each literal node is used in at most one clause gadget since we have k literal nodes for each literal, and each variable appears in at most k clauses. Similarly, we can have each T node used in at most one clause gadget, since the gadgets consume at most $kn < 2kn$ T nodes. The clause gadget allows any coloring of the literal nodes with "true" or "false" other than the coloring which corresponds to an assignment where all literals are false (and the clause goes unsatisfied).

Theorem 22 *The mapping φ is a gap-preserving local reduction from $(3, k)SAT$ to 3-coloring in bounded degree graphs. In particular, if f is a $(\delta, 1 - \varepsilon)$-satisfiable $(3, k)CNF$ formula, then the graph $\varphi(f)$ has degree bounded by some universal constant b and the 3-coloring CSP of $\varphi(f)$ is $(\delta/bc, 1 - \varepsilon/8)$-satisfiable.*

Proof   It is clear by observation that the mapping φ always produces graphs bounded by some constant degree b, and that there exists a constant c such that φ converts a $(3, k)$CNF formula on n variables to a graph on at most cn nodes. Furthermore, one can answer a query for an edge of $\varphi(f)$ making at most one query into f, namely, for the clause in which the queried edge is a part (if any). Write n' for the number of nodes in $\varphi(f)$, and $m' < bn' \leq bcn$ for the number of edges.

Suppose that the original $(3, k)$CNF formula is $(\delta, 1 - \varepsilon)$-satisfiable. Clearly any subgraph of $\varphi(f)$ induced by δn edges is 3-colorable – such a subgraph contains nodes

participating in at most δn clause gadgets, where we say that a node participates in a clause gadget if it is contained in the clause gadget, or is a color class node corresponding to a literal node contained in the clause gadget. By definition, there exists a boolean assignment satisfying these δn clauses of f. The coloring which sets all color classes to their intended colors and colors the literal nodes "true" or "false" as in this assignment satisfies these $\delta n > \frac{\delta}{bc} m'$ 3-coloring constraints.

Note that if we delete γt edges from the expander graph G_t with $\gamma \leq \frac{1}{2}$, then there must remain a connected component of size at least $(1 - \gamma)t$, for disconnecting a set S of nodes with $|S| \leq \frac{1}{2}t$ requires at least $|\Gamma(S)|$ edge deletions which, by the expansion property, is at least $|S|$. Applying this to the equality gadgets between color class nodes, we see that deletion of $\gamma(2kn)$ edges leaves each color class with at least $(1 - \gamma)(2kn)$ color class nodes in a connected component with equality constraints intact. Therefore, it leaves at least $(1 - 3\gamma)(2kn)$ triples $\{D_i, T_i, F_i\}_{i \in S}$ such that the D_i must be colored the same as D_j for $i, j \in S$ (similarly for T_i and F_i). The disconnected triples $\overline{S}$ participate in at most $2 \cdot 3\gamma(2kn)$ clause gadgets. Furthermore, deleting $\gamma(2kn)$ edges modifies constraints about nodes participating in at most $2 \cdot \gamma(2kn)$ clauses of f. Summing up, deletion of $\gamma(2kn)$ edges leaves the 3-coloring construction for at least $m - (2 \cdot 3\gamma(2kn) + 2 \cdot \gamma(2kn)) = m - 16\gamma kn$ clauses of f intact. If f is $(\delta, 1 - \varepsilon)$-satisfiable, then no coloring of the remaining graph can be valid if $m - 16\gamma kn > (1 - \varepsilon)m$ or, equivalently, $\gamma < \frac{\varepsilon}{16k}$. Changing notation so that $\gamma' m' = \gamma(2kn)$ (ie we have deleted a fraction γ' of the edges of $\varphi(f)$ in the above discussion) and noting that $m' > n$, we get that $\frac{\varepsilon}{16k} > \gamma = \frac{\gamma' m'}{2kn} > \frac{\gamma'}{2k}$ or $\gamma' < \varepsilon/8$.

Combining the conclusions of the previous two paragraphs, we see that the graph

3-coloring problem $\varphi(f)$ is $(\delta/bc, 1 - \varepsilon/8)$-satisfiable. ∎

As immediate corollaries, we obtain the promised constructions for families of 3-coloring instances hard for one- and two-sided local algorithms.

Corollary 23 *Let φ_{3CNF} be the gap-preserving local reduction of Lemma 19, and φ_{3Col} that of Theorem 22. The (explictly constructed) set $\{\varphi_{3Col}(\varphi_{3CNF}(f_n))\}_n$ is an infinite family of bounded-degree graphs G_n on m_n edges such that, for universal constants $\delta, \varepsilon > 0$, every subgraph induced by δm_n edges is 3-colorable, but every 3-coloring of G_n has at least εm_n monochromatic edges.*

Proof We need only note that the 3CNF formulas $\{\varphi_{3CNF}(f_n)\}_n$ are in fact $(3, k)$CNF formulas. This is because the variable $x_{i,j}$ corresponding to edge (i, j) appears only in the constraints around vertices i and j. In particular, if l is the number of clauses in a 3CNF representation of the predicate h, then $x_{i,j}$ can appear in at most $2l$ clauses. The claim then follows from Lemmas 20 and 21. ∎

Corollary 24 *There exist universal constants $\delta, \varepsilon, \Delta$ such that if A is a local algorithm for deciding 3-colorability of degree Δ graphs with distance parameter ε, then the query complexity of A is at least δn, where n is the number of vertices.*

Proof The canonical reduction from E3LIN2 to E3SAT is a gap-preserving local reduction with $c_1 = c_2 = 4$. Apply this to the construction in Lemma 15 and compose with φ_{3Col}. ∎

Chapter 5

Locally decodable codes

In this chapter, we study linear codes $\mathbf{C} : \{0,1\}^n \rightarrow \{0,1\}^m$ which have the property that, for constants $\delta, \varepsilon > 0$, any bit of the message can be recovered with probability $\frac{1}{2}+\varepsilon$ by an algorithm reading only 2 bits of a codeword corrupted in up to δm positions. Such codes are known to be applicable to, among other things, the construction of scalable, fault-tolerant data storage systems and the analysis of information-theoretically secure private information retrieval schemes.

Unfortunately, we show in this work that such coding schemes cannot be space-efficient. In particular, m must be at least $\exp(\Omega\left(\delta n/(1-2\varepsilon)\right))$.

The following construction, due to Trevisan, shows that this lower bound is optimal within a constant factor in the exponent: Recall that the *Hadamard code* on $x \in \{0,1\}^n$ is given by $y_i = a_i \cdot x$ where a_i runs through all 2^n vectors in $\{0,1\}^n$. Hadamard codes are locally decodable with 2 queries as, for any $i \in \{1, \ldots, n\}$ and $r \in \{0,1\}^n$,

$$x_i = r \cdot x + (r + e_i) \cdot x = e_i \cdot x$$

where e_i is the ith canonical unit vector in $\{0, 1\}^n$. It is easy to see that the queries of this decoder form perfect matchings on the n-dimensional hypercube, and the code has recovery parameter $\varepsilon = \frac{1}{2} - 2\delta$.

For given δ, ε, let $c = \frac{1-2\varepsilon}{4\delta}$. It can be shown that for feasible values of δ, ε, $1 - 2\varepsilon \geq 4\delta$ so that $c \geq 1$. We divide the input bits into c blocks of $\frac{n}{c}$ bits, and encode each block with the Hadamard code on $\{0, 1\}^{\frac{n}{c}}$. The resulting code has length $\frac{1-2\varepsilon}{4\delta} 2^{\frac{4\delta}{1-2\varepsilon}n}$ which is, say, less than $2^{4.01\frac{\delta}{1-2\varepsilon}n}$ for sufficiently large n. Finally, since each code block has at most a fraction $c\delta$ of corrupt entries, the code achieves recovery parameter

$$\frac{1}{2} - 2c\delta = \frac{1}{2} - 2\left(\frac{1 - 2\varepsilon}{4\delta}\right)\delta = \varepsilon$$

as required.

5.1 Background

Our results extend work by Goldreich, Karloff, Schulman, and Trevisan [GKST02], who show that m must be at least $\exp(\Omega(\varepsilon\delta n))$. Note that the prior bound does not grow arbitrarily large as the error probability of the decoder goes to zero ($\varepsilon \to \frac{1}{2}$), as intuitively it should; the results presented here have the correct qualitative behavior.

The key to our improved bounds is an analysis which bypasses an intermediate reduction used in both prior works. The resulting improvement in the efficiency of the overall analysis is sufficient to achieve a lower bound optimal within a constant factor in the exponent.

Subsequent to the publication of this work in [Oba02], Kerenidis and de Wolf [KdW03] applied quantum information techniques to obtain exponential lower bounds for *arbitrary* locally decodable codes. We note that virtually all of the techniques presented here can be carried over into the general case, the essential component in [KdW03] being the use of the sub-additivity property of Von Neumann entropy to lower bound the dimension of the space of codewords.

It remains an open problem to obtain strong lower bounds for locally decodable codes with 3 or more queries. Such an analysis appears well beyond currently known techniques.

Our work is structured as follows: In the remainder of this section, we briefly review the definitions and techniques employed in [KT00] and [GKST02]. In Section 5.2, we establish a relationship between the probability that an edge of a graph sampled from any distribution intersects any vertex-subset of a given size, and the size of a maximum matching in the graph. The analysis in this result seems independently interesting, and may be applicable in other contexts. In Section 5.3, we show how the combination of this result with the techniques of [GKST02] establishes lower bounds for this class of locally decodable codes.

Locally decodable and smooth codes

Let Σ_1, Σ_2 be arbitrary finite alphabets. The following definition was introduced in [KT00].

Definition 25 (Locally decodable code) *For fixed constants δ, ε, q, a mapping*

$$\mathbf{C} : \Sigma_1^n \to \Sigma_2^m$$

is a (q, δ, ε)-locally decodable code if there exists a probabilistic oracle machine A such that:

- *A makes at most q queries (without loss of generality, A makes exactly q queries).*

- *For every $x \in \Sigma_1^n$, $y \in \Sigma_2^m$ with $d(y, \mathbf{C}(x)) \leq \delta$, and $i \in \{1, \ldots, n\}$,*

$$\Pr\left(A^y(i) = x_i\right) \geq \frac{1}{|\Sigma_1|} + \varepsilon$$

where the probability is over the randomness of A.

In this work, we consider codes $\mathbf{C}$ satisfying the above properties where, in addition, Σ_1, Σ_2 are fields and $\mathbf{C}$ is a *linear* mapping from $\Sigma_1^n \to \Sigma_2^m$. While all of our results are applicable to finite fields in general, and some to non-linear codes, we will for simplicity narrow our current discussion to linear codes on $\mathbf{Z}_2$. Also, while we have observed that our results are equally applicable to reconstruction algorithms making queries adaptively, we limit our comments in this abstract to algorithms making non-adaptive queries.

We begin by reviewing the techniques of [KT00] and [GKST02], which our results build upon.

It was observed in [KT00] that a locally decodable code should have the property that a decoding algorithm A reads from each location in the code word with roughly uniform probability. This motivates the following definition.

Definition 26 (Smooth code) *For fixed constants c, ε, q, a mapping*

$$\mathbf{C} : \Sigma_1^n \to \Sigma_2^m$$

is a (q, c, ε)-smooth code if there exists a probabilistic oracle machine A such that:

- *A makes at most q queries (without loss of generality, A makes exactly q queries).*

- *For every $x \in \Sigma_1^n$ and $i \in \{1, \ldots, n\}$,*

$$\Pr\left(A^{\mathbf{C}(x)}(i) = x_i\right) \geq \frac{1}{|\Sigma_1|} + \varepsilon.$$

- *For every $i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}$, the probability that on input i machine A queries index j is at most c/m.*

Intuitively, if a code is insufficiently smooth, so that a particular small subset of indices is queried with too high a probability, then corrupting that subset causes the decoding algorithm to fail with too high a probability. Thus, a locally decodable code must have a certain smoothness. Specifically, [KT00] proved:

Theorem 27 *If $\mathbf{C} : \Sigma_1^n \to \Sigma_2^m$ is a (q, δ, ε)-locally decodable code, then $\mathbf{C}$ is also a $(q, q/\delta, \varepsilon)$-smooth code.*

The lower bounds for linear locally decodable codes in [GKST02] are proved by establishing lower bounds for smooth codes. The result for locally decodable codes follows by application of Theorem 27.

Smooth codes are closely related to the concept of information-theoretically secure private information retrieval schemes introduced in [CGKS98]. Briefly, the idea in these constructions is to allow a user to retrieve a value stored in a database in such a way that the database server does not learn significant information about what value was queried. It is easy to see that, in the information-theoretic setting, achieving privacy in this sense with a single database server requires essentially that the entire database be transferred to the user on any query. [CGKS98] showed, however, that by using 2 (non-colluding) servers, one can achieve privacy in this sense with a single round of queries and communication complexity $O(n^{1/3})$. [KT00] observed that if one interprets the query bits sent to the databases as indexes into a 2-query decodable code, then the smoothness parameter of a code can be interpreted as a statistical indistinguishability condition in the corresponding retrieval scheme. In this way, one can construct and analyze smooth codes, and therefore locally decodable codes, from private information retrieval schemes and vice versa. We refer the reader to [GKST02] for a detailed discussion.

The basic technique for proving lower bounds for smooth codes introduced in [KT00] and extended in [GKST02] is to study, for each $i \in \{1, \ldots, n\}$, the *recovery graph* G_i defined on vertex set $\{1, \ldots, m\}$ where (q_1, q_2) is an edge of G_i iff for all $x \in \{0,1\}^n$,

$$\Pr\left(A^{\mathbf{C}(x)}(i) = x_i | A \text{ queries } (q_1, q_2)\right) > \frac{1}{2}.$$

Such edges are called *good* edges. Then, one shows a lower bound on the size of a maximum matching in the recovery graphs G_i which is a function of the smoothness parameter of $\mathbf{C}$:

Lemma 28 ([KT00], [GKST02]) *If* $\mathbf{C}$ *is a* $(2, c, \varepsilon)$*-smooth code with recovery graphs* $\{G_i\}_i$ *then, for every* i, G_i *has a matching of size at least* $\varepsilon m/c$.

For *linear* smooth codes, it is easy to see that an edge (q_1, q_2) can be good for x_i iff x_i is a linear combination of q_1, q_2. To simplify matters, one narrows the analysis to codes in which these linear combinations are non-trivial:

Definition 29 (Normal code) *A linear code* $\mathbf{C}$ *is* normal *if none of the entries in the range of* $\mathbf{C}$ *is a scalar multiple of an input entry.*

We can assume normality in smooth codes with only a constant factor modification in length and recovery parameters:

Theorem 30 ([GKST02]) *For* $n > 4c/\varepsilon$, *let* $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ *be a* (q, c, ε)*-smooth code. Then there exists a* $(q, c, \varepsilon/2)$*-smooth code* $\mathbf{C}' : \{0,1\}^{n'} \to \{0,1\}^{m'}$ *with* $n' \geq n/2, m' \leq m$ *in which for all* $i \in \{1, \ldots, n'\}, j \in \{1, \ldots, m'\}$, *the* j*-th bit of* $\mathbf{C}'(x)$ *is not a scalar multiple of* x_i.

Putting the pieces together, we have that a normal $(2, c, \varepsilon)$-smooth code has for every $i \in \{1, \ldots, n\}$ a recovery graph G_i containing a matching of size at least $\varepsilon m/c$, and for each of the edges (q_1, q_2) in this matching, x_i is in the span of q_1, q_2, but is not a scalar multiple of q_1 or q_2. Thus, the preconditions for the following key result of [GKST02] are satisfied:

Lemma 31 *Let* $q_1, \ldots, q_m$ *be linear functions on* $x_1, \ldots, x_n \in \{0,1\}^n$ *such that for every* $i \in \{1, \ldots, n\}$ *there is a set* M_i *of at least* γm *disjoint pairs of indices* j_1, j_2 *such that* $x_i = q_{j_1} + q_{j_2}$. *Then* $m \geq 2^{\gamma n}$.

Composing this with the normal reduction of Theorem 30, we have:

Theorem 32 ([GKST02]) *Let* $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ *be a* $(2, c, \varepsilon)$*-smooth linear code.* *Then* $m \geq 2^{\frac{\varepsilon n}{4c}}$.

Finally, composing this with the locally decodable to smooth reduction, this says:

Theorem 33 ([GKST02]) *Let* $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ *be a* $(2, \delta, \varepsilon)$*-locally decodable linear* *code. Then* $m \geq 2^{\frac{\varepsilon \delta n}{8}}$.

Note that Lemma 31 yields a lower bound which is exponential in the fraction of vertices in $\{0,1\}^m$ covered by a matching in every recovery graph of the code. Thus, if we can prove a tighter lower bound on the size of these matchings, then we get a corresponding improvement in the exponent in the final lower bound. This is exactly the method used in our work. In particular, we achieve an optimized bound on the size of the matchings in the recovery graphs by bypassing the reduction to smooth codes and instead arguing directly about locally decodable codes. The resulting direct reduction is strong enough to yield a tight final lower bound.

5.2   Blocking game and matchings

In this section, we prove a combinatorial theorem regarding the relationship between the probability that an edge of a graph sampled from any distribution intersects any vertex-subset of a given size, and the size of a maximum matching in the graph. We will later see that this game captures completely the interplay between the encoder, decoder, and channel adversary.

Let $G(V, E)$ be an undirected graph on n vertices, $w : E \to \mathbf{R}^+$ a probability distribution on the edges of G, $\mathcal{W}$ the set of all such distributions, and S a subset of V. Our concern in this section is to establish a bound on the following parameter of G based on the size of a maximum matching in G:

Definition 34 (Blocking probability) *Let X^w denote a random edge of G sampled according to distribution w. Define the* blocking probability $\beta_\delta(G)$ *as*

$$\beta_\delta(G) = \min_{w \in \mathcal{W}} \left(\max_{S \subseteq V, |S| \leq \delta n} \Pr\left(X^w \cap S \neq \emptyset\right) \right).$$

One can think of $\beta_\delta(G)$ as the value of a game in which the goal of the first player (the decoding algorithm) is to sample an edge from G which avoids a vertex in a δn-set selected by the second player (the channel adversary), whose goal is to maximize the probability of blocking the edge selected by the first player.

Our goal is the following theorem:

Theorem 35 *Let G be a graph with $m(G) = (1 - \alpha)n$. Then*

$$\beta_\delta(G) \geq \min\left(\frac{\delta}{1 - \alpha}, 1\right).$$

The key in the analysis is the following special family of graphs: For each n, α, define the graph $K(n, \alpha)$ with vertex set

$$K_1(n, \alpha) \cup K_2(n, \alpha), |K_1(n, \alpha)| = \alpha n, |K_2(n, \alpha)| = (1 - \alpha)n,$$

such that the edge set of $K(n, \alpha)$ is the union of the edge set of the complete bipartite graph with bipartition $(K_1(n, \alpha), K_2(n, \alpha))$ and the $(1 - \alpha)n$-clique on $K_2(n, \alpha)$.

Probabilistic proof of Theorem 35

We begin with a probabilistic proof of Theorem 35. This argument does not characterize the optimal strategies for the blocking game, but is sufficient to prove our ultimate result. In the next section, we give a derandomized analysis which explicitly describes the optimal strategies.

Fix an arbitrary edge-distribution w on $K(n, \alpha)$ and, for $\delta < 1 - \alpha$, select a subset S of δn vertices of $K_2(n, \alpha)$ uniformly at random. The resulting blocking probability β can be written as a sum $\beta = \sum_e \beta_e$ over edges e, where β_e is a random variable with value w_e if S intersects e, or 0 otherwise. By linearity of expectation,

$$E(\beta) = \sum_e E(\beta_e) = \sum_e w_e \Pr\left(S \cap e \neq \emptyset\right)$$

where the randomness is over the selection of the subset S. Clearly, S intersects each edge e with probability at least $\frac{\delta n}{(1-\alpha)n} = \frac{\delta}{1-\alpha}$, so this expectation is at least

$$\sum_e w_e \frac{\delta}{1 - \alpha} = \frac{\delta}{1 - \alpha} \sum_e w_e = \frac{\delta}{1 - \alpha}.$$

In particular, there must exist some subset S achieving this expectation, proving the theorem.

Explicit proof of Theorem 35

We now give a longer, but explicit, proof of Theorem 35 which describes the optimal strategy for any decoder subject to a given lower bound on the matching number of G.

For a graph G, the *independence number* $\alpha(G)$ of G is the size of a maximum independent set of vertices in G, and the *matching number* $m(G)$ of G is the number of vertices in a maximum matching in G (note that this definition differs from the standard one by a factor of 2). We begin our analysis by observing that $n - m(G)$ is a lower bound on $\alpha(G)$. We then define a relaxation of the optimization problem for the blocking probability on graphs with a given independence number. For this relaxed problem, we define a special family of distributions and show that some distribution in this family optimizes the blocking probability. Finally, we exhibit a lower bound on the blocking probability of a particular set of δn vertices with respect to any distribution in this family of distributions.

Lemma 36 *Let G be a graph with $m(G) = (1 - \alpha)n$. Then*

$$\beta_\delta(G) \geq \beta_\delta(K(n, \alpha)).$$

Proof We begin by noting that $\alpha(G) \geq n - m(g)$. To see this, fix any maximum matching in G. The $n - m(g)$ vertices left uncovered in this matching must be an independent set, for an edge between any of these vertices would allow us to increase the size of the matching by at least one.

By the assumption on $m(G)$, then, we have $\alpha(G) \geq \alpha n$. With a labeling of vertices

of $K(n, \alpha)$ which sets $K_1(n, \alpha)$ to an arbitrary αn-subset of S, it is easy to see that the edge set of $K(n, \alpha)$ contains the edge set of G. Therefore, the optimization of w on $K(n, \alpha)$ is a relaxation of the optimization of w on G (a distribution w on G can be expressed as a distribution w' on $K(n, \alpha)$ in which any edge of $K(n, \alpha)$ not in G has probability 0). The claim follows. $\blacksquare$

We will focus on the following special class of distributions on $K(n, \alpha)$ and show that the blocking probability of $K(n, \alpha)$ is always optimized by some distribution in this class:

Definition 37 ((λ_1, λ_2)-Symmetric Distribution) *An edge distribution w on the graph $K(n, \alpha)$ is (λ_1, λ_2)-symmetric if for every edge $e \in (K_1(n, \alpha), K_2(n, \alpha))$, $w(e) = \lambda_1$, and for every edge $e \in (K_2(n, \alpha), K_2(n, \alpha))$, $w(e) = \lambda_2$.*

Lemma 38 *Let $w_1, \ldots, w_k$ be edge distributions on G such that*

$$\max_{S \subseteq V, |S| \leq \delta n} \Pr\left(X^{w_i} \cap S \neq \emptyset\right) = \beta_\delta(G).$$

Then for any convex combination of the distributions $w = \sum_i \gamma_i w_i$,

$$\max_{S \subseteq V, |S| \leq \delta n} \Pr\left(X^w \cap S \neq \emptyset\right) = \beta_\delta(G).$$

Proof For every $S \subseteq V$,

$$\Pr\left(X^w \cap S \neq \emptyset\right) = \sum_i \gamma_i \Pr\left(X^{w_i} \cap S \neq \emptyset\right)$$

since this is simply the sum over edge weights of edges of G incident to S. By the condition on the w_i, for any subset S with $|S| \leq \delta n$,

$$
\begin{aligned}
\Pr\left(X^w \cap S \neq \emptyset\right) &\leq \sum_i \gamma_i \beta_\delta(G) \\
&= \beta_\delta(G) \sum_i \gamma_i \\
&= \beta_\delta(G).
\end{aligned}
$$

Therefore,

$$
\max_{S \subseteq V, |S| \leq \delta n} \Pr\left(X^w \cap S \neq \emptyset\right) \leq \beta_\delta(G).
$$

However, by definition of $\beta_\delta(G)$, this must be at least $\beta_\delta(G)$. Therefore,

$$
\max_{S \subseteq V, |S| \leq \delta n} \Pr\left(X^w \cap S \neq \emptyset\right) = \beta_\delta(G).
$$

∎

Recall that the *automorphism group* of a graph G is the set of permutations π on the vertices of G such that $(\pi(i), \pi(j)) \in E \iff (i, j) \in E$. Let Γ be the automorphism group of $K(n, \alpha)$.

Lemma 39 *There exists a (λ_1, λ_2)-symmetric distribution w such that*

$$
\max_{S \subseteq V, |S| \leq \delta n} \Pr\left(X^w \cap S \neq \emptyset\right) = \beta_\delta(K(n, \alpha)).
$$

Proof Let w' be any distribution which optimizes the blocking probability of $K(n, \alpha)$. It is obvious that if w' is such a distribution, then so is $\pi(w')$ for $\pi \in \Gamma$ (where we extend the

action of Γ to the edges of G in the natural way). By Lemma 38, the distribution

$$w = \frac{1}{|\Gamma|} \sum_{\pi \in \Gamma} \pi(w')$$

optimizes the blocking probability of $K(n, \alpha)$. We claim that w is a (λ_1, λ_2)-symmetric distribution: For any edge $e \in E$ and $\sigma \in \Gamma$,

$$
\begin{aligned}
w(e) &= \frac{1}{|\Gamma|} \sum_{\pi \in \Gamma} w'(\pi(e)) \\
&= \frac{1}{|\Gamma|} \sum_{\pi \in \Gamma} w'(\pi\sigma(e)) \\
&= w(\sigma(e))
\end{aligned}
$$

where the second step is the usual group-theoretic trick of permuting terms in summations over Γ. Therefore, if $e, e' \in E$ are in the same orbit under the action of Γ, $w(e) = w(e')$. It is easy to verify that Γ is the direct product of the group of permutations of the vertices of $K_1(n, \alpha)$ and $K_2(n, \alpha)$, and so there are exactly two edge-orbits of $K(n, \alpha)$ under Γ, one consisting of the edges $(K_1(n, \alpha), K_2(n, \alpha))$ and the other $(K_2(n, \alpha), K_2(n, \alpha))$. This is exactly the condition for a (λ_1, λ_2)-symmetric distribution. ∎

Finally, we need to compute a lower bound on the blocking probability for a (λ_1, λ_2)-symmetric distribution:

Lemma 40 *Let w be a (λ_1, λ_2)-symmetric distribution on $K(n, \alpha)$. Then there exists a*

subset $S \subseteq V$ with $|S| \leq \delta n$ such that

$$\Pr\left(X^w \cap S \neq \emptyset\right) \geq \min\left(\frac{\delta}{1-\alpha}, 1\right).$$

Proof   We will study a blocking set which selects any δn vertices of $K_2(n, \alpha)$. Note that, by (λ_1, λ_2)-symmetry, it does not matter which δn vertices we select. Further, we can assume that $\delta < 1 - \alpha$, for if $\delta \geq 1 - \alpha$ we can cover all of $K_2(n, \alpha)$ and thereby achieve blocking probability 1.

Placing a blocking set in this manner and summing up over edges and weights, we achieve blocking probability

$$(\delta n)(\alpha n)\lambda_1 + \frac{1}{2}(\delta n)(\delta n - 1)\lambda_2 + (\delta n)(1 - \alpha - \delta)n\lambda_2.$$

Since w is a probability distribution, we must have

$$(\alpha n)(1 - \alpha)n\lambda_1 + \frac{1}{2}(1 - \alpha)n\left((1 - \alpha)n - 1\right)\lambda_2 = 1.$$

Using this to eliminate λ_1 from the first expression, we obtain blocking probability

$$\delta\left(\frac{1}{1 - \alpha} + \frac{1}{2}n^2(1 - \alpha - \delta)\lambda_2\right).$$

Since $\delta < 1 - \alpha$, the second term in the sum is positive (and, obviously, optimized when $\lambda_2 = 0$), so the blocking probability must be at least $\frac{\delta}{1-\alpha}$. ∎

It is now easy to prove Theorem 35.

By Lemma 36, $\beta_\delta(G) \geq \beta_\delta(K(n,\alpha))$. By Lemma 39, the blocking probability of $K(n,\alpha)$ is optimized by some (λ_1, λ_2)-symmetric distribution. By Lemma 40, there exists a subset of δn vertices which blocks any such distribution with probability at least $\min\left(\frac{\delta}{1-\alpha}, 1\right)$. Therefore,

$$\beta_\delta(G) \geq \beta_\delta(K(n,\alpha)) \geq \min\left(\frac{\delta}{1-\alpha}, 1\right).$$

5.3 Lower bounds

In this section, we apply Theorem 35 to our original problem of finding lower bounds for locally decodable linear codes.

To simplify our analysis, we would again like to put the decoder into a canonical normal form in which the decoder simply outputs the sum of two distinct input bits. The following shows that we may make this assumption without loss of generality.

Lemma 41 *Let $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ be a $(2, \delta, \varepsilon)$-locally decodable linear code where n is large enough so that $\frac{2(n+1)}{2^n} \leq \delta/201$. Then there exists a normal $(2, \delta/2.01, \varepsilon)$-locally decodable linear code $\mathbf{C}' : \{0,1\}^n \to \{0,1\}^{2m}$.*

Proof Write the ith entry y_i of the codeword as $y_i = a_i \cdot x$ where $a_i \in \{0,1\}^n$. A straightforward probabilistic argument shows that there exists a vector $r \in \{0,1\}^n$ such that the Hamming weights of r and $a_i + r$ are at least 2 for a fraction at least $\left(1 - \frac{2(n+1)}{2^n}\right)$ of the y_i. Let S be the set of y_i satisfying this property. Note that for $y_i \in S$, $(a_i + r) \cdot x$ is a not a scalar multiple of an input entry. We form a normal code $\mathbf{C}' : \{0,1\}^n \to \{0,1\}^{2m}$ from $\mathbf{C}$ by setting $y_i' = (a_i + r) \cdot x$ for $y_i \in S$, $y_i' = (1, \ldots, 1) \cdot x$ for all other indices, and

adding a set of m codeword bits $y_i'' = r \cdot x$ for all $i \in \{1, \ldots, m\}$.

We claim that $\mathbf{C}'$ is a $(2, \delta/2.01, \varepsilon)$-locally decodable code. Let A be a recovery algorithm for $\mathbf{C}$, and recall that an edge for A can be good only if the answer of A is a linear combination of the entries it queries. Without loss of generality, we can assume that A only queries good edges (otherwise, we can ignore the answers to the queries and output a random coin flip). We implement a recovery algorithm A' for $\mathbf{C}'$ as follows: If A takes a non-trivial linear combination of queries y_i, y_j, then A' simulates A but executes queries y_i', y_j'; if A is the identity on a query y_i, then A' makes queries y_i', y_i'' and takes the (non-trivial) linear combination $y_i' + y_i''$, which for $i \in S$ equals $(a_i + r) \cdot x + r \cdot x = a_i \cdot x = y_i$. Finally, we note that if at most $\delta/2.01$ entries of a codeword of $\mathbf{C}'$ are corrupted, then A' exactly simulates the behavior of A when interacting with some code word with at most

$$\frac{\delta}{2.01}2m + |\overline{S}| \leq \frac{200}{201}\delta m + \frac{2(n+1)}{2^n}m \leq \frac{200}{201}\delta m + \frac{1}{201}\delta m = \delta m$$

corrupt entries. By the decoding condition on A, A' succeeds with probability at least $\frac{1}{2} + \varepsilon$.

∎

Therefore, we can essentially assume that we are working with a normal decoder. We are now prepared to prove our lower bound.

Theorem 42 *Let $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ be a (q, δ, ε)-locally decodable linear code for $0 < \delta, \varepsilon < \frac{1}{2}$. Then for sufficiently large n, $m \geq 2^{\frac{1}{4.03}\frac{\delta}{1-2\varepsilon}n}$.*

Proof By Lemma 41, for sufficiently large n, the existence of $\mathbf{C}$ implies the existence of a normal $(2, \delta/2.01, \varepsilon)$-locally decodable code $\mathbf{C}' : \{0,1\}^n \to \{0,1\}^{2m}$. As before, we can

assume that the recovery algorithm A' for $\mathbf{C}'$ only queries good edges. On one hand, for all $i \in \{1, \ldots, n\}$ and $y \in \{0, 1\}^{2m}$ such that $d(y, \mathbf{C}'(x)) \leq \frac{\delta}{2.01}(2m)$,

$$\Pr\left(A'^y(i) \neq x_i\right) \leq \frac{1}{2} - \varepsilon.$$

On the other, if $(1 - \alpha)(2m)$ is the maximum over all $i \in \{1, \ldots, n\}$ of the matching number of the recovery graph G_i of A', then

$$\Pr\left(A'^y(i) \neq x_i\right) \geq \frac{1}{2} \frac{\delta/2.01}{1 - \alpha}$$

for, by Theorem 35, there exists a fraction $\delta/2.01$ of vertices S such that an adversary which sets the values of S to random coin flips causes A' to read a blocked edge, and therefore have probability $\frac{1}{2}$ of outputting an incorrect response, with probability at least $\frac{\delta/2.01}{1-\alpha}$. Therefore,

$$\frac{1}{2} \frac{\delta/2.01}{1 - \alpha} \leq \frac{1}{2} - \varepsilon \implies \alpha \leq 1 - \frac{\delta/2.01}{1 - 2\varepsilon}$$

which is equivalent to saying that there exists for all $i \in \{1, \ldots, n\}$ a set of at least $\frac{1}{2} \frac{\delta/2.01}{1-2\varepsilon}(2m)$ disjoint pairs of indices j_1, j_2 such that $x_i = q_{j_1} + q_{j_2}$. Then by Lemma 31, $2m \geq 2^{\frac{1}{2} \frac{\delta/2.01}{1-2\varepsilon} n}$ or $m \geq 2^{\frac{1}{4.02} \frac{\delta}{1-2\varepsilon} n - 1}$ which is at least, say, $2^{\frac{1}{4.03} \frac{\delta}{1-2\varepsilon} n}$ for sufficiently large n.

∎

Chapter 6

Integral flows

We saw in Chapter 3 that integral flow systems can be applied to establish lower bounds for local computations. We now approach this relationship from the opposite direction; that is, starting with the notion of approximation used in local computational models, we derive new algorithms for multicommodity integral flow and max-integral-flow/min-multicut duality relationships. This yields new approaches to classical combinatorial optimization problems and insights into the behavior of the classic Edmonds-Karp algorithm in the multicommodiy setting.

The results in this chapter appeared previously in [Oba04].

6.1 Background

A classic theorem of Ford and Fulkerson asserts that the maximum flow between two vertices in a graph equals the weight of a minimum cut separating the two vertices [FF56]. This flow/cut duality further enjoys a beautiful *integrality* property: If the capacities

of the edges of G are integers, then there exists an integral flow achieving the weight of the minimum cut. Integrality is often a necessary or desirable property in applications. In this chapter, we consider corresponding integrality results in the setting of *multicommodity* flows.

Unfortunately, the analogous equality does not generally hold in the multicommodity setting. However, in a body of work initiated by Leighton and Rao ([LR99, PT93, GVY93, KPR93, AR98, LLR95] and others), a number of *approximate* max-flow/min-cut relations have been established. All of these results differ in a fundamental way from the Ford-Fulkerson theorem in that they are based on dual rounding procedures. This approach loses the guarantee of primal integrality and, indeed, the integrality gap for multicommodity flows can be $\Omega(k)$, where k is the number of commodities.

In this chapter, we establish several approximate max-*integral*-flow/min-multicut theorems. While in general this ratio can be trivially large, we prove strong approximation ratios in the case where the min-multicut is a constant fraction ε of the total capacity of the graph. This setting is motivated by several combinatorial and algorithmic applications. Prior to this work, a general max-integral-flow/min-multicut bound was known only for the special case where the graph is a tree [GVY97].

Our proofs are constructive in the sense that we give efficient algorithms which compute either an integral flow achieving the claimed approximation ratios, or a witness that the precondition is violated.

The main goal in this chapter shall be the following max-integral-flow/min-multicut approximation theorem.

Theorem 43 *Let $G(V, E)$ be a graph with capacity function $c : E \to \mathbf{Z}^+$, and K be a demand graph on a k-element subset of V such that the weight of any multicut separating all pairs of vertices $(k_1, k_2) \in K$ is at least εC, where $C = \sum_{e \in E} c(e)$. Then*

    (i) The max-integral-flow/min-multicut ratio is $O(\varepsilon^{-1} \log k)$. If k^* is the vertex cover number of K, the ratio is improved to $O(\varepsilon^{-1} \log k^*)$.

    (ii) If, for some constant r, G does not contain $K_{r,r}$ as a minor (for instance, if G is planar), then the ratio is improved to $O(1/\varepsilon)$.

    (iii) If $c \in \{0, 1\}^E$ and G is δ-dense for some $\delta > 0$, then the ratio is improved to $O(1/\sqrt{\varepsilon \delta})$.

The proof of Theorem 43, as in previous work in this area, depends on efficient low-radius decompositions in a graph metric. Prior to this work, the traditional approach has been to decompose G using the metric induced by the minimum solution to the dual of the maximum flow problem; the approximation ratio then follows by the strong duality theorem. The fundamental difference here is that G is decomposed according to the *unweighted* distance metric on G, and this is used to construct a large *primal* solution; the ratio follows by the promised lower bound on the minimum multicut. Two of the decompositions we use are standard in the multicommodity flow literature; the third is a "new" one. The details are discussed in Section 6.2.

The approximate max-integral-flow algorithm works roughly as follows: For a given family of graphs (general, planar, dense, etc.), we define a "greed" function $g(t)$ which is an increasing function of time t. At time t, the algorithm greedily increases the primal value by pushing a unit of flow along a path of length at most $g(t)$. If this can repeated for sufficiently

many iterations then we are done. Otherwise, the algorithm constructs a witness multicut

by combining both primal and dual information. The details are presented in Section 6.3.

Section 6.4 describes some algorithmic and combinatorial applications of Theorem 43.

Definitions and notation

Throughout this chapter, $G(V, E)$ denotes a simple, undirected graph, $c : E \to$ $\mathbf{Z}^+$ denotes a $\mathbf{Z}^+$-valued capacity function on the edges of G, and K denotes a simple, undirected, unweighted graph defined on a vertex-subset of V. A K-*path* in G is a simple path in G between k_1, k_2 where $(k_1, k_2) \in E(K)$, a K-*cut* is an edge-subset $F \subseteq E$ such that $G \setminus F$ has no K-path, and a K-*partition* is a disjoint system of vertex-subsets $(S_1, \dots, S_t)$ such that each $k \in K$ is in some S_i.

By the *characteristic vector* of a simple path P, we mean the vector in $\mathbf{Z}^E$ which is 1 on edges $e \in P$ and 0 otherwise. A set $\{p_1, \dots, p_t\}$ of characteristic vectors of K-paths along with positive weights $w_1, \dots, w_t$ is called a K-*flow* with respect to a capacity function c when $v = \sum_i w_i p_i$ is such that $v(e) \leq c(e)$ for all edges e. The *weight* of a K-flow is the sum $\sum w_i$ of all path weights. By a *maximum flow*, we mean a K-flow with maximum weight among all K-flows. A flow is *integral* when the all path weights w_i are integer-valued. A *maximum integral flow* is a flow with maximum weight among all integral K-flows.

Often, we identify c with the graph formed by including edges for which c is non-zero. When $G(V, E)$ is understood, we write (S_1, S_2) to denote the subset of edges with one endpoint in S_1 and the other in S_2, where $S_i \subseteq V$. For $F \subseteq E$, we define $c(F) = \sum_{e \in F} c(e)$, and for $W \subseteq V, c(W) = c((W, W))$. If $S \subseteq V$, we write $\nabla(S) = (S, V \setminus S)$ and $\nabla(S_1, \dots, S_t) = \bigcup_i \nabla(S_i)$. For a capacity function c, $\nabla_c(S)$ denotes the capacity

function which equals c on $\nabla(S) \cap \{e : c(e) > 0\}$ and is zero elsewhere. By the norm $\|c\|$ of c we mean $\|c\|_1 = \sum_e c(e)$.

We use $d(v_1, v_2)$ to denote the distance between vertices v_1 and v_2, $B(v, \rho) = \{w \in V \mid d(v, w) \leq \rho\}$ to denote the closed ball of radius ρ around v and $B^\circ(v, \rho) = \{w \in V \mid d(v, w) = \rho\}$ to denote its boundary. The diameter $\mathrm{diam}(S) = \max_{v_i, v_j \in S} d(v_i, v_j)$ and the radius $\rho(S) = \frac{1}{2}\mathrm{diam}(S)$. The radius of a K-partition $(S_1, \ldots, S_t)$ is the maximum over i of $\rho(S_i)$.

6.2 Three graph decompositions

The proof of Theorem 43 depends critically on low-radius decompositions of graphs. In this section, we discuss three such decompositions, each of which correspond to the three cases described in Theorem 43. The first is generic but gives the weakest bounds; the latter two apply only to specialized cases, but give much stronger results. The first two have been used previously to establish max-flow/min-multicut approximation ratios, while the third one is "new" (the algorithm was used earlier, but we re-cast it in the more general setting of multicommodity flows and provide a new analysis).

For all $\rho \geq 0$ and families $\mathcal{G}$ of $\mathbf{Z}^+$-valued edge-functions on graphs, define

$$f_{\mathcal{G},k}(\rho) = \max_{\substack{c \in \mathcal{G}, C > 0 \\ K \subseteq V, |K| \leq k}} \frac{1}{C} \left(\min_{(S_1, \ldots, S_t) \in \mathcal{P}_K^\rho} c\left(\nabla(S_1, \ldots, S_t)\right) \right)$$

where $\mathcal{P}_K^\rho$ denotes the set of all K-partitionings of V of radius ρ. For brevity, we will omit the parameter k. Note that, for any $\mathcal{G}$, $f_{\mathcal{G}}(\rho)$ is monotonically decreasing in ρ since, for $\rho' > \rho$, a radius ρ decomposition is also a radius ρ' decomposition. Each of the following

decompositions gives an algorithmic upper bound on $f_{\mathcal{G}}(\rho)$ for particular values of $\mathcal{G}$.

General graphs

We begin with the most generic decomposition algorithm, which is applicable to arbitrary graphs. The essential idea behind this decomposition was introduced by Leighton and Rao in their original work [LR99]. The version which follows is the special case of a subsequent refinement by Garg, Vazirani, and Yannakakis [GVY93] in which the dual variables are fixed to $(1, \ldots, 1)$.

Garg-Vazirani-Yannakakis Decomposition: Let $\alpha > 0$ be a parameter, to be selected later. While there exists any terminal vertex v in G repeat the following: Set $v \leftarrow$ an arbitrary terminal vertex in G, $t \leftarrow 0$; while $c(\nabla(B(v,t))) + \frac{C}{k} > \alpha c(B(v,t))$, set $t \leftarrow t + 1$; output $B(v,t)$ and set $G \leftarrow G \setminus B(v,t)$. The set of output subsets gives a K-partition of V.

Lemma 44 *Let $S_1, \ldots, S_t$ denote the K-partition produced by the Garg-Vazirani-Yannakakis algorithm with parameter α. Then*

$$\rho(S_i) \leq \frac{\ln(k+1)}{\alpha} \quad and \quad c(\nabla(S_1, \ldots, S_t)) \leq 2\alpha C.$$

Proof Apply Lemmas 4.1 and 4.2 from [GVY93] in the special case where the dual variables are set to $(1, \ldots, 1)$. ∎

Corollary 45 *Let $\mathcal{G}^*$ denote the set of all $\mathbf{Z}^+$-valued capacity functions. Then*

$$f_{\mathcal{G}^*}(\rho) \le \frac{2\ln(k+1)}{\rho}.$$

Proof Set $\alpha = \frac{\ln(k+1)}{\rho}$. Then $\rho(S_i) \le \rho$ and $c(\nabla(S_1, \ldots, S_t)) \le \frac{2\ln(k+1)}{\rho}C$. ∎

Graphs excluding $K_{r,r}$

We now consider the special case of graphs excluding a fixed minor. (For instance, by the weak direction of Kuratowski's Theorem, this includes the family of planar graphs.) An ingenuous decomposition algorithm for this setting was designed by Klein, Plotkin, and Rao [KPR93]. The version below is a variant in which, as before, we fix the input dual variables to $(1, \ldots, 1)$. We also scan along possible cut points to assure the cut weight on each iteration is at most a fixed fraction of the total capacity.

Klein-Plotkin-Rao Decomposition: Let $\alpha > 0$ be a parameter, to be selected later, and let r be such that $K_{r,r}$ does not occur as a minor of G. While G is non-empty, repeat the following: Set $v \leftarrow$ an arbitrary vertex in G and $t \leftarrow$ a value from $\{0, \ldots, \alpha - 1\}$ to be selected later; for $i = 1, \ldots, n$ let $G_i = \bigcup_{q \in R_i} B^\circ(v, q)$ where R_i is the ith set in the sequence $[0, t), [t, t + \alpha), [t + \alpha, t + 2\alpha), \ldots$, and recurse on G_i to a maximum depth r. For each level of the recursion, select a value t such that the weight of the edges cut at that level is $\le C/\alpha$ (such a value must exist because the cuts induced by each possible selection of t partition E into α classes). Interpret the set of regions at the bottom of the recursion as a vertex-partition of G, disregarding empty regions.

Lemma 46 *Suppose that G does not contain $K_{r,r}$ as a minor and let $S_1, \ldots, S_t$ denote the partition produced by the Klein-Plotkin-Rao algorithm. Then*

$$\rho(S_i) \leq 2r^2\alpha \ \text{and} \ c\left(\nabla(S_1, \ldots, S_t)\right) \leq \frac{r}{\alpha}C.$$

Proof   In [KPR93], it is proved that $\rho(S_i) \leq 2r^2\alpha$. By the selection of t, we introduce a cut of weight at most C/α at each level of the recursion. Since the recursion has depth at most r, the resulting cut must have weight at most $\frac{r}{\alpha}C$. ■

Corollary 47 *Let $\mathcal{G}^{r,r}$ denote the set of $\mathbf{Z}^+$-valued capacity functions on graphs excluding $K_{r,r}$. Then*

$$f_{\mathcal{G}^{r,r}}(\rho) \leq \frac{2r^3}{\rho}.$$

Proof   Set $\alpha = \frac{\rho}{2r^2}$. Then $\rho(S_i) \leq \rho$ and $c\left(\nabla(S_1, \ldots, S_t)\right) \leq \frac{2r^3}{\rho}C$. ■

Dense graphs

For a fixed constant $0 < \delta \leq 1$, an unweighted graph is called δ-*dense* if $|E| \geq \delta n^2$. In other words, a graph is dense when a constant fraction of all possible edges are included in the graph. We achieve our tightest bounds when the capacity function c is 0-1-valued and G is δ-dense.

We use a technique due to Komlós which was originally introduced to prove tight bounds on the size of a minimum edge-set intersecting all odd cycles in a graph. We observe that this method is applicable in the more general context of multicommodity flows. We also provide a new and simpler proof of Komlós' result.

Komlós Decomposition: Let $\alpha > 0$ be a parameter, to be selected later. While G is non-empty, repeat the following: Set $v \leftarrow$ an arbitrary vertex in G, $t \leftarrow 0$; while $|B^\circ(v, t)||B^\circ(v, t+1)| > \alpha|B(v, \infty)||B(v, t)|$ set $t \leftarrow t + 1$; output $B(v, t)$ and set $G \leftarrow G \setminus B(v, t)$. The set of output sets gives a partitioning of V.

Lemma 48 *Let G be a δ-dense graph, and let $S_1, \ldots, S_t$ denote the vertex-partition produced by the Komlós algorithm. If $c \in \{0, 1\}^E$, then*

$$\rho(S_i) \leq \frac{12}{\sqrt{\alpha}} \text{ and } c\left(\nabla(S_1, \ldots, S_t)\right) \leq \frac{\alpha C}{\delta}.$$

(In [Kom97], Komlós achieves the better constant $\sqrt{2e}$ in place of 12, although our constant is not optimized.)

Proof For brevity, set $b_i = |B^\circ(v, i)|$ and $B_i = |B(v, i)|$, so that $B_i = \sum_{0 \leq j \leq i} b_j$. The stopping rule requires that the sequence $\{b_i\}$ satisfy $b_i b_{i+1} > \alpha n B_i$. Then, for each i, at least one of b_i, b_{i+1} must be at least $\sqrt{\alpha n B_i}$. So, consider alternating entries of $\{B_i\}$, $B'_j = B_{2j+1}$. Then the sequence $\{B'_j\}$ satisfies the recurrence

$$B'_0 \geq \alpha n \; ; \; B'_{j+1} \geq B'_j + \sqrt{\alpha n B'_j}.$$

We claim that $B'_j \geq \frac{\alpha n}{9} j^2$. This is verified by observation for $j = 0, 1$. For larger j, induction shows that

$$B'_{j+1} \geq \frac{\alpha n}{9} j^2 + \sqrt{\alpha n \frac{\alpha n}{9} j^2} = \frac{\alpha n}{9} j^2 + \frac{\alpha n}{3} j \geq \frac{\alpha n}{9} (j + 1)^2$$

when $j \geq 1$. Then for odd i,

$$B_i \geq \frac{\alpha n}{9} \left(\frac{i-1}{2}\right)^2$$

so for $i \geq 2$,

$$B_i \geq \frac{\alpha n}{9} \left(\frac{i-2}{2}\right)^2 = \frac{\alpha n}{36}(i-2)^2$$

and (crudely), since $i - 2 \geq \frac{i}{2}$ for $i \geq 4$ and by inspection for $i = 0, \ldots, 3$, for any i,

$$B_i \geq \frac{\alpha n}{36} \left(\frac{i}{2}\right)^2 = \frac{\alpha n}{144} i^2.$$

But $B_i \leq n$ which implies that the maximum possible index i in such a sequence, which is also the maximum possible radius in a region, is at most $\sqrt{\frac{144}{\alpha}} = \frac{12}{\sqrt{\alpha}}$.

For the second part of the lemma, notice that the stopping rule implies that $c(\nabla(S_i)) \leq \alpha |S_i| n$. Then

$$c(\nabla(S_1, \ldots, S_t)) \leq \sum_i c(\nabla(S_i)) \leq \sum_i \alpha |S_i| n \leq \alpha n^2 \leq \alpha C / \delta.$$

∎

Corollary 49 *Let $\mathcal{G}^\delta$ denote the set of δ-dense graphs with capacity functions $c \in \{0, 1\}^E$.*

Then

$$f_{\mathcal{G}^\delta}(\rho) \leq \frac{144}{\delta \rho^2}.$$

Proof Set $\alpha = \frac{144}{\rho^2}$. Then $\rho(S_i) \leq \rho$ and $c(\nabla(S_1, \ldots, S_t)) \leq \frac{144}{\delta \rho^2} C$. ∎

6.3 Max-integral-flow/min-multicut approximation ratios

In this section, we apply the low-radius decompositions of Section 6.2 to prove our approximate max-integral-flow/min-multicut theorems. In fact, we show that there exist efficient algorithms which, given a lower bound on the weight of a min-multicut, either construct an integral flow achieving the claimed approximation ratio, or output a proof that the promise is violated in the form of a multicut with weight less than the asserted lower bound.

Proof of Theorem 43

Suppose we have an efficiently computable upper bound $f^*_{\mathcal{G}}(\rho) \geq f_{\mathcal{G}}(\rho)$ on f (from this point, we omit the subscript $\mathcal{G}$). Without loss of generality, we may assume f^* is monotonically decreasing in ρ. For all $\varepsilon > 0$, define

$$g(\varepsilon) = \min_{f^*(\rho) < \varepsilon} 2\rho$$

where $g(\varepsilon)$ is defined as ∞ if no such ρ exists. Note that $g(\varepsilon)$ is also monotonically decreasing in ε and that h can be efficiently constructed using $O(n)$ invocations of f^*.

Theorem 43 is proved using the following integral flow algorithm.

Approximate Max-Integral-Flow:

  set $\varepsilon_0 \leftarrow \varepsilon, v_0 \leftarrow (0, \ldots, 0), c_0 \leftarrow c$

  for $t = 0, \ldots, \infty$

   if $\varepsilon_t \leq 0$ **set** $F^* \leftarrow t$ and **break**

   set $\varepsilon_{t+1} \leftarrow \varepsilon_t - g(\varepsilon_t)/C$

  for $t = 0, \ldots, \infty$

   if $\exists\, k_j \in B_{c_t}(k_i, g(\varepsilon_t))$ for some $(k_i, k_j) \in K$

    set $p_t \leftarrow$ characteristic vector of any path of length $\leq g(\varepsilon_t)$ from $k_i \rightarrow k_j$

     set $v_{t+1} \leftarrow v_t + p_t, c_{t+1} \leftarrow c_t - p_t$

   else break

  if $t \geq F^*$

   output v_t and **accept**

  else

   set $m \leftarrow \nabla_{c_t}(S_1, \ldots, S_l)$ s.t. $\rho_{c_t}(S_i) \leq \frac{1}{2}g(\varepsilon_t)$ and $c_t(\nabla(S_1, \ldots, S_l)) < \varepsilon_t C$

   output $W = \{e \in E \mid [v_t + m](e) = c(e), c(e) > 0\}$ and **reject**

We claim that if the weight of a minimum K-cut of G is εC, then the algorithm produces a flow of weight F^* on input ε.

Assuming this for a moment, it is immediate that the max-integral-flow/min-multicut ratio is bounded by $\varepsilon C/F^*$ and so, to complete the proof of Theorem 43, we need only compute estimates of F^* given the functions $f_{\mathcal{G}^*}^*, f_{\mathcal{G}^{r,r}}^*, f_{\mathcal{G}^\delta}^*$ from Section 6.2. The following lemma states appropriate estimates.

Lemma 50 *Let $F^*_{\mathcal{G}}(\varepsilon)$ denote the value of F^* computed using function $f^*_{\mathcal{G}}$ for density ε. Then we have the following bounds:*

(i) $F^*_{\mathcal{G}^*}(\varepsilon) \geq \frac{\varepsilon^2}{16\ln(k+1)}C$

(ii) $F^*_{\mathcal{G}^{r,r}}(\varepsilon) \geq \frac{\varepsilon^2}{16r^2}C$

(iii) $F^*_{\mathcal{G}^\delta}(\varepsilon) \geq \frac{\varepsilon^{3/2}\delta^{1/2}}{48\sqrt{2}}C$

Proof   In each case, the bound follows easily by applying the estimates in Section 6.2 to upper bound the radius ρ^* of a decomposition inducing a multicut of weight at most $\varepsilon C/2$. The number of iterations F^* must then be at least $\varepsilon C/4\rho^*$.

(A factor of 2 improvement is possible by iterating this procedure over radii of at all orders of magnitude.) ∎

The asserted approximation ratios follow immediately from Lemma 50. We now prove the claim. Because the inner loop of the algorithm is an augmenting path process, the algorithm clearly produces an integral flow of weight t^*, where t^* is the last value of t reached in the main loop. Therefore, it is enough to show that, when the algorithm rejects, the algorithm outputs a K-cut of weight less than εC.

We begin by showing that W is a K-cut; that is, W intersects every K-path. There are two cases. First, we observe that every K-path in c_{t^*} intersects m. By the loop condition, there is no K-path in c_{t^*} of length $\leq g(\varepsilon_{t^*})$. Let $(S_1, \ldots, S_t)$ be any K-partitioning of V with c_{t^*}-radius $\leq \frac{1}{2}g(\varepsilon_{t^*})$. If p is a K-path in c_{t^*} which does not intersect $\nabla_{c_{t^*}}(S_1, \ldots, S_t)$ then the entire path must lie in the same partition S_i. But $\mathrm{diam}_{c_{t^*}}(S_i) \leq 2\rho_{c_{t^*}}(S_i) \leq g(\varepsilon_{t^*})$. Contradiction. Further, each edge $e \in m$ is also in W since $m(e) = c_{t^*}(e)$ implies that

$[m + v_{t^*}](e) = [c_{t^*} + v_{t^*}](e) = c(e)$. Second, we consider a K-path p such that p intersects an edge $e \in c$ but $\notin c_{t^*}$. Then $c(e) = c_{t^*}(e) + v_{t^*}(e) = v_{t^*}(e)$, so $e \in W$ and p intersects W.

We next need to bound $c(W)$. Clearly, $c(W) \leq \|m\| + \|v_{t^*}\|$. When $t^* < F^*$, $\varepsilon_{t^*} > 0$ and, since $\|p_i\| \leq g(\varepsilon_i)$ for all i, $\|v_{t^*}\| = \sum_{j=0,\ldots,t^*-1} \|p_i\| \leq \sum_{j=0,\ldots,t^*-1} g(\varepsilon_j) = (\varepsilon - \varepsilon_{t^*})C$. By definition of f^* and g, the K-cut $\nabla_{c_{t^*}}(S_1, \ldots, S_t)$ giving m exists (and, as in our case when the bounds on f^* are algorithmic, can be efficiently computed) and has $\|m\| = c_{t^*}(\nabla(S_1, \ldots, S_t)) \leq f^*(\frac{1}{2}g(\varepsilon_{t^*}))\|c_{t^*}\| < \varepsilon_{t^*}\|c_{t^*}\| \leq \varepsilon_{t^*}C$. Summing these, $c(W) < (\varepsilon - \varepsilon_{t^*})C + \varepsilon_{t^*}C = \varepsilon C$.

To prove the claim concerning vertex covers in case (i), note that, as observed by Günlük [Gün02] for the case of the fractional max-flow/min-multicut approximation ratio, it is sufficient to seed the partition selection step in the Garg-Vazirani-Yannakakis algorithm using a vertex cover K^* of K rather than the entire vertex set of K. The proof of the flow/cut approximation bound carries through in this case as well, since a K^*-partitioning is sufficient to intersect every sufficiently long shortest path between vertex-pairs in K. Thus, if k^* is the size of a minimum vertex cover of K, the approximation ratio is improved to $O(\varepsilon^{-1} \log k^*)$. In an extreme case, for instance a star graph, $k = O(n)$ while setting K^* to the center vertex of the star gives $k^* = 1$.

It still remains to remove the assumption that we give the algorithm as input the weight εC of a min-multicut of G. Since $g(\varepsilon_t)$ is an increasing function in t, we can remove the computation of the sequence $\{\varepsilon_i\}$ altogether and replace the augmentation loop with a pure greedy algorithm; that is, while there exists a path between any two vertex-pairs of K in c_t, push one unit of flow along the *shortest* such path. This variant clearly produces at

least as large a flow as the original algorithm on any input ε, in particular the true value of ε. However, this version, as with the original, runs in time only weakly polynomial in the input size. This is easily corrected by modifying the greedy algorithm to push $\min_{e \in p} c_t(e)$ units of flow along the path p selected on each iteration. Then clearly there can be at most $O(|E|)$ augmentation steps.

6.4 Related applications

In this section, we observe that the proof of Theorem 43 yields efficient approximation algorithms for maximum integral and fractional multicommodity flow and related problems. We also observe some natural but less obvious combinatorial applications.

Approximation algorithms for flow and edge-disjoint path problems

In traditional applications of multicommodity max-flow/min-cut inequalities, the maximum flow problem is polynomial-time computable (via polynomial-time linear programming algorithms) while the corresponding cut problem is NP-hard. However, in the integral case, both the flow [GVY97] and cut [DJP$^+$94] problems are NP-hard. In the case where the min-multicut has weight at least εC, the approximate maximum integral flow algorithm used to prove Theorem 43 and the weak duality relations give the following.

Corollary 51 *Let G be a graph with $\mathbf{Z}^+$-valued capacity function c, and K be a demand graph on a k-element subset of V such that the weight of any multicut separating all pairs of vertices $(k_1, k_2) \in K$ is at least εC. Then*

(i) There is a polynomial-time algorithm which constructs an integral flow within a factor

$O(\varepsilon^{-1}\log k)$ *of the optimum. If k^* is the vertex cover number of K, the ratio is improved to $O(\varepsilon^{-1}\log k^*)$.*

(ii) If G excludes $K_{r,r}$ for some constant r, the ratio is improved to $O(1/\varepsilon)$.

(iii) If $c \in \{0,1\}^E$ and G is δ-dense, the ratio is improved to $O(1/\sqrt{\varepsilon\delta})$.

A natural variation on the above is to apply scaling methods to recast the general (fractional) multicommodity flow problem as an integral flow problem. That is, fix some large integer Q and, for a non-negative *real*-valued capacity function c, let $c'(e) = \lfloor c(e)Q \rfloor$. If v' is an integral flow in c', then v'/Q is a feasible (possibly fractional) flow in c. Further, the scaling distorts the relative weight ε of the minimum cut and the original capacity function by a factor which goes to 1 as $Q \to \infty$.

Given a fractional capacity function c, consider the following *greedy heuristic* for the maximum multicommodity flow problem: Select a *shortest K-path* in G and push as much flow as possible along the path until it is saturated; repeat until there are no K-paths remaining.

Corollary 52 *Let G be a graph with a non-negative real-valued capacity function c, and let ε, k^* be as above. Then the greedy heuristic constructs a flow of weight within a factor $O(\varepsilon^{-1}\log k^*)$ of the optimum. If G excludes $K_{r,r}$ for some constant r, then the factor is improved to $O(\varepsilon^{-1})$.*

Proof   It is easy to see that the greedy heuristic corresponds to the pure greedy variant of the approximate max-integral-flow algorithm above when Q is an arbitrarily large integer.

∎

Note that we cannot apply scaling techniques to the family of δ-dense graphs. Unconditional versions of these approximation ratios were already known [GVY93, Gün02, KPR93], but it is interesting that the greedy heuristic constructs a flow achieving these ratios when the min-multicut has constant density.

Integral flows are closely related to edge-disjoint paths in unweighted graphs. The problem of connecting a maximum number of endpoints in K along edge-disjoint paths was one of the original NP-hard problems [Kar72]. The following consequence of Theorem 43 will be used later.

Corollary 53 *Let G be an unweighted graph with m edges and maximum degree Δ such that at least εm edges must be removed to separate all vertex-pairs in K. Then*

(i) $\Omega\left(\varepsilon^2 m/\Delta \log k\right)$ vertex-pairs in K can be connected along mutually edge-disjoint paths, and these paths can be computed in polynomial time.

(ii) If G excludes $K_{r,r}$ for some constant r, then the same is true for $\Omega\left(\varepsilon^2 m/\Delta\right)$ pairs.

(iii) If G is δ-dense, then the same is true for $\Omega\left(\varepsilon^{3/2}\delta^{1/2}m/\Delta\right)$ pairs.

Proof   Edge-disjoint paths in an unweighted graph are equivalent to the special case of integral flow where edge capacities are 0-1-valued. If G has maximum degree Δ, then at most Δ routed paths between a pair (k_1, k_2) in a K-flow can correspond to a single terminal pair. ∎

Intersecting odd cycles

Let G be an unweighted graph, and let $og(G)$ denote the *odd girth* of G, that is, the length of the shortest odd cycle in G. In [BESS78], Bollobás, Erdös, Simonovits, and Szemerédi considered the problem of determining the minimum cardinality of an edge-subset $F \subset E$ such that F intersects every odd cycle in G. They showed that $|F| = O\left(n^2/og(G)\right)$ and conjectured that $|F| = \Theta\left(n^2/og^2(G)\right)$. This conjecture was proved by Komlós [Kom97] using the eponymous decomposition in Section 6.2.

It turns out that odd cycles in a graph have a very natural formulation in terms of integral multicommodity flows. The relation is summarized in the following lemma.

Lemma 54 *Let $G(V, E)$ be an unweighted graph, and let (V_0, V_1) be a bipartition of V such that $|(V_0, V_0)| + |(V_1, V_1)|$ is minimized. Let $E = E_0 \cup E_1$ where $E_0 = (V_0, V_0) \cup (V_1, V_1)$ and $E_1 = (V_0, V_1)$, and set $K = E_0$. Then*

(i) *Every K-path in $G_1 = G(V, E_1)$ can be extended by an edge in E_0 into an odd cycle in G.*

(ii) *For all $S \subset V$, $|\nabla_{G_1}(S)|/|\nabla_K(S)| \geq 1$.*

(iii) *Any K-cut of G_1 has size at least $|E(K)|$.*

Proof   Let p be a K-path in G_1, say, between $(k_1, k_2) \in E_0$. By definition, G_1 is bipartite, so p has even length. Then $p \cup (k_1, k_2)$ is an odd cycle in G. Next, suppose there exists $S \subset V$ such that $|\nabla_{G_1}(S)| < |\nabla_K(S)|$. Then setting $V_i' = V_i - (V_i \cap S) + (V_{\bar{i}} \cap S)$ gives

$$|(V_0', V_0')| + |(V_1', V_1')| = |(V_0, V_0)| + |(V_1, V_1)| + |\nabla_{G_1}(S)| - |\nabla_K(S)| < |(V_0, V_0)| + |(V_1, V_1)|,$$

contradicting the minimality in the choice of (V_0, V_1). Finally, let M be a K-cut of G_1, and

let $S_1, \ldots, S_t$ denote the connected components of $G_1 \setminus M$. Then $\sum_i \nabla_K(S_i) = 2|E(K)| \leq$

$\sum_i \nabla_{G_1}(S_i) \leq 2|M|$. ∎

Theorem 55 *If $G \in \mathcal{G}$ then $|F| = O\left(m \log n / og(G)\right)$ and $|F| = O\left(n^2 / og^2(G)\right)$. If G*

excludes $K_{r,r}$ for some constant r, then $|F| = O\left(m / og(G)\right)$.

Proof By definition of $f_{\mathcal{G}}$, there exists a vertex-partition $(S_1, \ldots, S_t)$ with

$$|\nabla(S_1, \ldots, S_t)| \leq f_{\mathcal{G}}(\frac{1}{2}(og(G) - 3))m$$

such that $\rho(S_i) < \frac{1}{2}(og(G) - 1)$. It is easy to see that such a multicut intersects every odd

cycle of length at least $og(G)$. The claims follow by applying the upper bounds f^* proved

in Section 6.2. ∎

Each of these bounds can also be interpreted as an upper bound on $og(G)$ in terms

of $|F|$. Note that the first bound is stronger when G is a sparse graph; the second is Komlós'

result; the third is a stronger bound for sparse graphs in the case of graphs excluding $K_{r,r}$.

Local 2-colorability

We note an application to local decidability of bipartiteness. That is, we would

like to find an algorithm which, given a graph which is ε-far from bipartite, locates an odd

cycle in G. Such an algorithm obviously has one-sided error, and the probability of error is

exactly the probability that it fails to find an odd cycle in a far-from-bipartite graph. The

following lemma states that a graph which is ε-far from bipartite is dense in small witnesses

to this fact (odd cycles).

Lemma 56 *Let G be a graph with constant maximum degree which is ε-far from bipartite. Then G contains $\Omega(\varepsilon^2 m / \log n)$ edge-disjoint odd cycles of length $O(\varepsilon^{-2} \log n)$. If G excludes $K_{r,r}$ for some constant r, then it contains $\Omega(\varepsilon^2 m)$ edge-disjoint odd cycles of length $O(\varepsilon^{-2})$.*

Proof   With notation as earlier, $|F| \geq \varepsilon m$. Apply Corollary 53 to the construction in Lemma 54. That a constant fraction of these cycles have at most the given length follows by Markov's inequality.   ∎

Lemma 56 implies an efficient local algorithm for bipartiteness in the special case of graphs excluding $K_{r,r}$:

Theorem 57 *Let G be a graph with constant maximum degree Δ such that G excludes $K_{r,r}$ for some constant r and G is ε-far from bipartite. Then there exists an algorithm which locates an odd cycle in G with probability $1 - \delta$ using $\exp(O(\varepsilon^{-2})) \log(1/\delta)$ queries; in particular, the algorithm requires a number of queries which is independent of n.*

Proof   In the case of graphs excluding $K_{r,r}$, Lemma 56 implies that we can locate an odd cycle with constant probability by sampling $O(\varepsilon^{-2})$ random vertices and doing a breadth-first search about each vertex to radius $O(\varepsilon^{-2})$. Repeating this $O(\log(1/\delta))$ times reduces the failure probability to $1 - \delta$. The overall query complexity of this procedure is $\Delta^{O(\varepsilon^{-2})} \log(1/\delta)$.   ∎

On the other hand, Goldreich and Ron [GR97] showed that locating an odd cycle requires $\Omega(\sqrt{n})$ queries in general. A slightly modified argument shows that a graph with $|F| \geq \varepsilon n^2$ must contain $\Omega(\varepsilon^{3/2} n^2)$ edge-disjoint odd cycles. Alternatively, one can argue directly from Lemma 46 that bipartiteness as well as other properties on bounded-degree

graphs, including 3-colorability (which we saw earlier is not computable with a sublinear number of queries), are distinguishable in constant time on graphs excluding a fixed minor.

6.5 Tightness of bounds

With respect to the usual parameter k, the bounds of Theorem 43 are optimal. For general graphs, this follows from the lower bound in [GVY93], which is similar to the lower bound construction used in [LR99]: G is a Δ-regular expander graph on n vertices, all edges have unit capacity, and K is the set of all vertex-pairs (k_1, k_2) such that $d(k_1, k_2) \geq \log_\Delta(n/2)$. It is easy to check that the maximum (fractional) flow is $O(n/\log n)$ while the minimum multicut has weight $\Omega(n) = \Omega(m)$. Fractional flow is a relaxation of integral flow, so the bound holds in our case as well, even though the min-multicut has constant density. For the other families considered, the assertion is trivial.

It is not clear, however, whether Theorem 43 always captures correctly the optimal dependence of the max-integral-flow/min-multicut ratio on ε. For planar graphs, the grid example used in [GVY97] to establish the $\Omega(k)$ integrality gap shows that our dependence on ε for planar graphs is optimal. More generally, a potential problem in our approach is illustrated by the following example, which is the construction of Lemma 54 applied to the lower bound construction used in [BESS78]. Let G' be a path on $l = O(1/\sqrt{\beta})$ vertices $v_1', \ldots, v_l'$ and let G be the "blow-up" of G' by $t = O(\sqrt{\beta}n)$ vertices; that is, replace each vertex v_i' of G' by t vertices v_i^j for $j = 1, \ldots, t$, and let G be the graph formed by connecting all pairs $(v_j^{i_1}, v_{j+1}^{i_2})$ for all i_1, i_2, j. Set K to the set of pairs $(v_1^{i_1}, v_l^{i_2})$ for all i_1, i_2. Then G is δ-dense, with $\delta = \sqrt{\beta}$, and it is easy to see that any K-cut has weight $\Omega(\sqrt{\beta}C)$.

Applying Theorem 43 with $\varepsilon = \sqrt{\beta}$ gives that there exists a flow of weight $O(\beta C)$ when, in fact, there exists a flow with weight equal to the min-multicut – the optimum flow is achieved by pushing a unit flow along all paths $\mathcal{P} = \{(a_1, \ldots, a_l)\}$ where $\mathcal{P}$ is a pairwise independent family of vectors on $\{1, \ldots, t\}^l$ of size t^2 (this flow strategy was observed by Luca Trevisan). The problem is that, when we augment along a path, it may be the case that the path intersects a min-multicut at a single edge, while the approximate max-integral-flow algorithm reduces its lower bound on the weight of the unknown min-multicut by the length of the path. In this example, the worst case occurs on every augmentation, leading to a loss of a factor $l = O(1/\sqrt{\beta})$. On the other hand, the same example shows that our bound on $f_{\mathcal{G}^\delta}$ in this case is optimal within a constant factor.

Bibliography

[AFKS99] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. In *IEEE Symposium on Foundations of Computer Science*, pages 656–666, 1999.

[AFNS99] N. Alon, E. Fischer, I. Newton, and M. Szegedy. Regular languages are testable with a constant number of queries. In *IEEE Symposium on Foundations of Computer Science*, pages 645–655, 1999.

[AK02] N. Alon and M. Krivelevich. Testing k-colorability. *SIAM Journal on Discrete Mathematics*, 15:211–227, 2002.

[AR98] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.

[BESS78] B. Bollobás, P. Erdös, M. Simonovits, and E. Szemerédi. Extremal graphs without large forbidden subgraphs. *Annals of Discrete Mathematics*, 3:29–41, 1978.

[BOT02] A. Bogdanov, K. Obata, and L. Trevisan. A lower bound for testing 3-

colorability. In *IEEE Symposium on Foundations of Computer Science*, pages 93–102, 2002.

[BR00]   M. Bender and D. Ron. Testing acyclicity of directed graphs in sublinear time. In *International Conference on Automata, Languages, and Programming*, 2000.

[CGKS98]   B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan.   Private information retrieval. *Journal of the ACM*, 45(6):965–982, 1998.

[CRT00]   B. Chazelle, R. Rubinfeld, and L. Trevisan. Approximating the minimum spanning tree weight in sublinear time. In *International Conference on Automata, Languages, and Programming*, 2000.

[DJP⁺94]   E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[Erd62]   P. Erdős. On circuits and subgraphs of chromatic graphs. *Mathematika*, 9:170–175, 1962.

[FF56]   L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8, 1956.

[FGL⁺96]   U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy.   Approximating clique is almost np-complete. *Journal of the ACM*, 43:268–292, 1996.

[FJ97]   A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k CUT and MAX BISECTION. *Algorithmica*, 18:61–77, 1997.

[FK99]   A. Frieze and S. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19:175–220, 1999.

[GG81]   O. Gabber and Z. Galil. Explicit construction of linear sized superconcentrators. *Journal of Computer and System Sciences*, 22:407–425, 1981.

[GGR98]   O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connections to learning and approximation. *Journal of the ACM*, 45:653–750, 1998.

[GKST02] O. Goldreich, H. Karloff, L. Schulman, and L. Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *IEEE Conference on Computational Complexity*, 2002.

[GR97]   O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *ACM Symposium on Theory of Computing*, pages 406–415, 1997.

[GR99]   O. Goldreich and D. Ron. A sublinear bipartitness tester for bounded degree graphs. volume 19, pages 335–373, 1999.

[Gün02]   O. Günlük. A new min-cut max-flow ratio for multicommodity flows. In *Integer Programming and Combinatorial Optimization*, pages 54–66, 2002.

[GVY93]   N. Garg, V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *ACM Symposium on Theory of Computing*, pages 698–707, 1993.

[GVY97]   N. Garg, V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18:3–20, 1997.

[Hås97]   J. Håstad. Some optimal inapproximability results. In *ACM Symposium on Theory of Computing*, pages 1–10, 1997.

[Kar72]   R. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, New York, NY, 1972. Plenum Press.

[KdW03] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes. In *ACM Symposium on Theory of Computing*, 2003.

[KKR04] T. Kaufman, M. Krivelevich, and D. Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, pages 1441–1483, 2004.

[Kom97] J. Komlós. Covering odd cycles. *Combinatorica*, pages 393–400, 1997.

[KPR93] P. Klein, S. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *ACM Symposium on Theory of Computing*, pages 682–690, 1993.

[KT00]   J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *ACM Symposium on Theory of Computing*, 2000.

[LLR95]   N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

[LR99]   F. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multi-commodity flow problems with applications to approximation algorithms. *Journal of the ACM*, 46(6), 1999.

[Mar73]   G. Margulis. Explicit construction of a concentrator. *Problems of Information Transmission*, 9:71–80, 1973.

[MR95]   R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Oba02]   K. Obata. Optimal lower bound for 2-query locally decodable linear codes. In *6th International Workshop on Randomization and Approximation Techniques*, pages 39–50, 2002.

[Oba04]   K. Obata. Approximate max-integral-flow/min-multicut theorems. In *ACM Symposium on Theory of Computing*, pages 539–545, 2004.

[Pap93]   C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1993.

[Pet94]   E. Petrank. The hardness of approximations: Gap location. *Computational Complexity*, 4:133–157, 1994.

[PS94]   A. Polishchuk and D. Spielman. Nearly-linear size holographic proofs. In *ACM Symposium on Theory of Computing*, pages 194–203, 1994.

[PT93]   S. Plotkin and É. Tardos. Improved bounds on the max-flow min-cut ratio for multicommodity flows. In *ACM Symposium on Theory of Computing*, pages 691–697, 1993.

[PY91]   C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[Tre01]   L. Trevisan. Non-approximability results for optimization problems on bounded

degree instances. In *ACM Symposium on Theory of Computing*, pages 453–461,

2001.