Expander ℓ_0-decoding

Rodrigo Mendoza-Smith and Jared Tanner

***Abstract*—We introduce two new algorithms, Serial-ℓ_0 and Parallel-ℓ_0 for solving a large underdetermined linear system of equations $y = Ax \in \mathbb{R}^m$ when it is known that $x \in \mathbb{R}^n$ has at most $k < m$ nonzero entries and that A is the adjacency matrix of an unbalanced left d-regular expander graph. The matrices in this class are sparse and allow a highly efficient implementation. A number of algorithms have been designed to work exclusively under this setting, composing the branch of *combinatorial compressed-sensing* (CCS).**

Serial-ℓ_0 and Parallel-ℓ_0 iteratively minimise $\|y - A\hat{x}\|_0$ by successfully combining two desirable features of previous CCS algorithms: the information-preserving strategy of ER [1], and the parallel updating mechanism of SMP [2]. We are able to link these elements and guarantee convergence in $\mathcal{O}(dn \log k)$ operations by assuming that the signal is *dissociated*, meaning that all of the 2^k subset sums of the support of x are pairwise different. However, we observe empirically that the signal need not be exactly dissociated in practice. Moreover, we observe Serial-ℓ_0 and Parallel-ℓ_0 to be able to solve large scale problems with a larger fraction of nonzeros than other algorithms when the number of measurements is substantially less than the signal length; in particular, they are able to reliably solve for a k-sparse vector $x \in \mathbb{R}^n$ from m expander measurements with $n/m = 10^3$ and k/m up to four times greater than what is achievable by ℓ_1-regularization from dense Gaussian measurements. Additionally, due to their low computational complexity, Serial-ℓ_0 and Parallel-ℓ_0 are observed to be able to solve large problems sizes in substantially less time than other algorithms for compressed sensing. In particular, Parallel-ℓ_0 is structured to take advantage of massively parallel architectures.

I. Introduction

Compressed sensing [3, 4, 5, 6, 7, 8] considers the problem of sampling and efficiently reconstructing a compressible finite dimensional signal $x \in \mathbb{R}^n$ from far fewer measurements than what Nyquist and Shannon deemed possible [9, 10]. In its simplest form compressed sensing states that if $x \in \mathbb{R}^n$ has at most $k < n$ nonzero entries, then it can be sampled from m linear measurements $y = Ax \in \mathbb{R}^m$ and that x can be recovered from (y, A) with computationally efficient algorithms provided $m < n$ is sufficiently large, see [11].

The most widely studied sensing matrices A are from the classes of: a) Gaussian or uniformly drawn projections which are most amenable to precise analysis due to their spherical symmetry, and b) partial Fourier matrices which have important applications for tomography and have fast transforms allowing A and A^* to be applied in $\mathcal{O}(n \log n)$ operations. Unfortunately the partial Fourier matrices are not known to allow the asymptotically optimal order number of measurements of $m \sim k \sim n$, rather the best analysis ensures recovery for $m \sim k \log^5 n$ [11]. Sparse binary matrices with a fixed number of non-zeros per column offer the possibility of A and A^* being applied in $\mathcal{O}(n)$ time and for asymptotically optimal order number of measurements $m \sim k \sim n$ [12, 13]. When restricting to these matrices, compressed sensing is referred to as *combinatorial compressed sensing*, [13].

RMS and JT are with the Mathematical Institute, University of Oxford, Oxford, UK. RMS is supported by CONACyT. (email: {mendozasmith,tanner}@maths.ox.ac.uk)

Manuscript submitted August 2015.

A. Combinatorial compressed sensing

The problem of sparse recovery with compressed sensing resembles the problem of *linear sketching* in theoretical computer science. This area considers sketching high dimensional vectors $x \in \mathbb{R}^n$ using a sparse matrix $A \in \mathbb{R}^{m \times n}$ with the aim that Ax has lower dimensionality than x, but still preserves some of its properties with high probability. In an attempt to reconcile this area with the compressed-sensing paradigm, [13] proposed sensing $x \in \chi_k^n$ using an expander matrix, $i.e.$ the adjacency matrix an unbalanced bipartite graph with high connectivity properties[1]. We denote the $m \times n$ matrices in this class by $\mathbb{E}_{k,\varepsilon,d}^{m \times n}$, but abbreviate to $\mathbb{E}_{k,\varepsilon,d}$ when the size is understood by its context. Expander matrices $\mathbb{E}_{k,\varepsilon,d}^{m \times n}$ are sparse binary matrices with $d \ll m$ ones per column, but with their nonzeros distributed in such a way that any submatrix composed of k columns has at least $(1 - \varepsilon)kd$ rows which are nonzero [2]. This structure makes them suitable for sparse recovery, and also makes them low complexity in terms of storage, generation, and computation (see Table I). Additionally, some applications like the single-pixel camera [14] consider measurement devices with binary sensors that inherently correspond to binary and sparse inner products, and that unfortunately, fall outside the set of matrices for which the widely used restricted isometry techniques apply.

The authors of [13] showed that, although being sparse, expander matrices can sense elements in χ_k^n at the optimal measurement rate $\mathcal{O}(k \log(k/m))$, and that these can be recovered accurately and efficiently via ℓ_1-regularization. Following this result, a series of algorithms designed specifically to work with expander matrices was presented in [1, 2, 15, 16]. The analysis of these algorithms requires the use of techniques and ideas borrowed from combinatorics, which is why this branch was labeled by [13] as *combinatorial compressed sensing* (CCS). It is in this realm that we make our main contributions.

B. Main contributions

Our work is in the nexus of a series of papers [1, 2, 15, 16] proposing iterative greedy algorithms for combinatorial com-

[1] See Section II-B for details.

[2] Such expander matrices can be generated by drawing i.i.d. columns with the location of their nonzeros drawn uniformly from the $\binom{m}{d}$ support sets of cardinality d, [12]

	Storage	Generation	A^*y	m
Gaussian/Bernoulli	$\mathcal{O}(mn)$	$\mathcal{O}(mn)$	$\mathcal{O}(mn)$	$\mathcal{O}(k\log(n/k))$
Partial Fourier	$\mathcal{O}(m)$	$\mathcal{O}(m)$	$\mathcal{O}(n\log n)$	$\mathcal{O}(k\log^5(n))$
Expander	$\mathcal{O}(dn)$	$\mathcal{O}(dn)$	$\mathcal{O}(dn)$	$\mathcal{O}(k\log(n/k))$

TABLE I: Complexity of measurement operators.

pressed sensing. The algorithms put forward in the aforementioned sequence of papers recover the sparsest solution of a large underdetermined linear system of equations $y = Ax$ by iteratively refining an estimation $\hat{x}$ using information about the residual $r = y - A\hat{x}$. Though these algorithms have the same high-level perspective[3], their particulars are optimised to best tradeoff speed, robustness, and recovery region; see Table II for a summary. For instance, at each iteration, SMP [2] updates several entries of $\hat{x}$ in parallel, allowing it to provably recover an arbitrary $x \in \chi_k^n$ in $\mathcal{O}(\log \|x\|_1)$ iterations of complexity $\mathcal{O}(dn + n\log n)$. However, SMP is only able to recover the sparsest solution when the fraction of nonzeros in the signal is substantially less than other compressed sensing algorithms. On the other hand, at each iteration, LDDSR [16] and ER [1] update a single entry of $\hat{x}$ in such a way that a contraction of $\|y - A\hat{x}\|_0$ is guaranteed. This reduction in the residual's sparsity is achieved by exploiting an important property of expander graphs, which we call the *information-preserving* property (see Theorem II.4). Essentially, this property guarantees that most of the entries from x will appear repeatedly as entries in $y = Ax$. In other words, it guarantees that for most $i \in [m]$, we will have $y_i \in \{x_j : j \in \text{supp}(x)\}$. In [16] and [1], this property is used to give sufficient conditions for decrease of $\|y - A\hat{x}\|_0$ under the regime of single updating of $\hat{x}$. However, this regime of single updating in LDDSR and ER typically requires greater computational time than existing compressed-sensing algorithms. Our main contribution is in the design and analysis of an algorithmic model that successfully combines the information-preserving strategy of LDDSR and ER with the parallel updating scheme of SMP. This synthesis is made possible by assuming that the signal of interest is *dissociated*.

Definition I.1 (Dissociated signals)**.** A signal $x \in \mathbb{R}^n$ is dissociated if

$$\sum_{j\in T_1} x_j \neq \sum_{j\in T_2} x_j \qquad \forall\, T_1, T_2 \subset \text{supp}(x) \text{ s.t. } T_1 \neq T_2. \quad (1)$$

The name *dissociated* comes from the field of additive combinatorics (See Definition 4.32 in [17]), where a set S is called *dissociated* if the set of all sums of distinct elements of S has maximal cardinality. Even though the model (1) might seem restrictive, it need not be exactly fulfilled for our algorithm to work. In fact, it is fulfilled almost surely for isotropic signals, and more generally by any signal whose nonzeros can be modelled as being drawn from a continuous distribution. Moreover, it is discussed in Section IV-C3 that non-dissociated signals, such as integer or binary signals, can be recovered if instead the columns of A are scaled by dissociated values, and the nonzeros of x are drawn independently of A. Also, numerical experiments show that the algorithm recovery ability decreases gracefully as the dissociated property is lost by having a fraction of the nonzeros in x be equal, see Figure 10,.

With this assumption, our contributions are a form of *model-based compressed sensing* [18] in which apart from assuming $x \in \chi_k^n$, one also assumes special dependencies between the values of its nonzeros with the goal to improve the algorithms speed or recovery ability. Our contributions are Serial-ℓ_0 and Parallel-ℓ_0, Algorithms 1 and 2 respectively, and their convergence guarantees summarised in Theorem I.2.

Algorithm 1: Serial-ℓ_0

Data: $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$; $y \in \mathbb{R}^m$; $\alpha \in (1, d]$
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $y = A\hat{x}$
$\hat{x} \leftarrow 0$, $r \leftarrow y$;
while *not converged* **do**
  for $j \in [n]$ **do**
    $T \in \{\omega_j \in \mathbb{R} : \|r\|_0 - \|r - \omega_j a_j\|_0 \geq \alpha\}$;
    for $\omega_j \in T$ **do**
      $\hat{x}_j \leftarrow \hat{x}_j + \omega_j$;
    end
    $r \leftarrow y - A\hat{x}$;
  end
end

Algorithm 2: Parallel-ℓ_0

Data: $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$; $y \in \mathbb{R}^m$; $\alpha \in (1, d]$
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $y = A\hat{x}$
$\hat{x} \leftarrow 0$, $r \leftarrow y$;
while *not converged* **do**
  $T \leftarrow \{(j, \omega_j) \in [n] \times \mathbb{R} : \|r\|_0 - \|r - \omega_j a_j\|_0 \geq \alpha\}$;
  for $(j, \omega_j) \in T$ **do**
    $\hat{x}_j \leftarrow \hat{x}_j + \omega_j$;
  end
  $r \leftarrow y - A\hat{x}$;
end

Theorem I.2 (Convergence of Expander ℓ_0-Decoders)**.** Let $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ and $\varepsilon \leq 1/4$. and $x \in \chi_k^n$ be a dissociated signal. Then, Serial-ℓ_0 and Parallel-ℓ_0 with $\alpha = (1-2\varepsilon)d$ can recover x from $y = Ax \in \mathbb{R}^m$ in $\mathcal{O}(dn\log k)$ operations.

The focus of this paper is on charting the development of Serial-ℓ_0 and Parallel-ℓ_0 and on proving Theorem I.2. In doing so, we contrast Serial-ℓ_0 and Parallel-ℓ_0 to the state-of-the-art algorithms for compressed-sensing and show that when the signal is dissociated, these are the fastest algorithms available when implemented, respectively, in a serial or a parallel architecture. We support these claims with a series of numerical experiments that additionally show that any loss in universality due to our signal model is traded off by

[3]See Section III and Table II

unusually high recovery regions when $\delta := m/n$ is small and substantially higher than those of previous CCS algorithms.

C. Outline

Section II gives the main background theory in expander graphs necessary for our discussion. Then, Section III reviews past advances in CCS, putting emphasis on deconstructing these into their essential ideas, and on pointing out common elements between them. Section IV contains our main contributions: Serial-ℓ_0 and Parallel-ℓ_0. We prove Theorem I.2 and point out some technical details regarding the implementation of Serial-ℓ_0 and Parallel-ℓ_0. We also discuss some connections of the dissociated model (1) with Information Theory. Finally, in Section V we evaluate the empirical performance of these algorithms with a series of numerical experiments.

II. Background

In this section, we present the basic notions of graph theory that are necessary for understanding our subsequent analyses, as well as the relevant previous work in combinatorial compressed sensing. We start by defining some notation.

A. Notation

For a subset $S \subset \Omega$, we let $|S|$ be its cardinality, and $\Omega \setminus S$ denote its complement. We adopt notation from combinatorics and use the shorthand $[n] := \{1, \dots, n\}$ for $n \in \mathbb{N}$. We also define $[n]^{(k)} = \{S \subset [n] : |S| = k\}$ and $[n]^{(\leq k)} = \{S \subset [n] : |S| \leq k\}$. As mentioned in the previous section, for $x \in \mathbb{R}^n$, we let $\text{supp}(x) = \{i : x_i \neq 0\}$ be its support, and $\text{argsupp}(x) = \{x_i : i \in \text{supp}(x)\}$ be the set of nonzero values in x. With this, we define $\|x\|_0 = |\text{supp}(x)|$, and $\chi_k^n = \{x \in \mathbb{R}^n : \|x\|_0 \leq k\}$; vectors in χ_k^n are said to be k-sparse. We let $H_k : \mathbb{R}^n \to \chi_k^n$ be the hard thresholding operator that sets to zero all but the k largest elements in x. Throughout this work, we implicitly assume that $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, and that $A \in \mathbb{R}^{m \times n}$ is a binary sparse matrix with d ones per column. It is also implicitly assumed that $m < n$ and that $\|x\|_0 < m$. For a given signal x, we will use k to refer to its sparsity, unless we specify otherwise.

B. Expander graphs

A *bipartite graph* is a 3-tuple $G = (U, V, E)$ such that $U \cap V = \emptyset$ and $E \subset U \times V$. Elements in $U \cup V$ are called *nodes*, while tuples in E are called *edges*. Under the assumption that $|U| = n$ and $|V| = m$, we abuse notation and let $U = [n]$ be the set of *left-nodes*, and $V = [m]$ be the set of *right-nodes*. A bipartite graph is said to be *left d-regular* if the number of edges emanating from each left node is identically d, and is said to be *unbalanced* if $m < n$. For $S \subset U \cup V$ we define $\mathcal{N}(S) \subset U \cup V$ to be the *neighbourhood* of S, *i.e.* the set of nodes in $U \cup V$ that are connected to S through an element of E. We note that for bipartite graphs, $\mathcal{N}(S) \subset V$ only if $S \subset U$, and $\mathcal{N}(S) \subset U$ only if $S \subset V$. An *expander graph* (Figure 1) is an unbalanced, left d-regular, bipartite graph that is *well-connected* in the sense of the following definition.

Definition II.1 (Expander graph)**.** An unbalanced, left d-regular, bipartite graph $G = ([n], [m], E)$ is a (k, ε, d)-expander if

$$|\mathcal{N}(S)| > (1 - \varepsilon) d |S| \;\; \forall \;\; S \in [n]^{(\leq k)}. \tag{2}$$

We call $\varepsilon \in (0, 1)$ the *expansion parameter* of the graph.

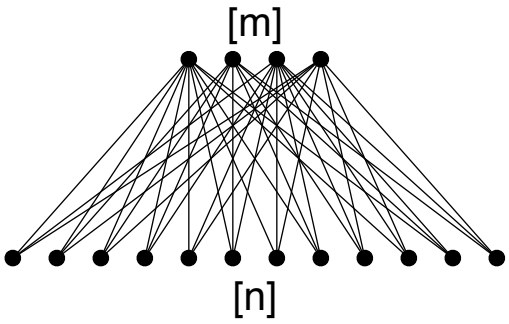

Fig. 1: **Schematic of an expander graph with $d = 3$.** Every left d-regular bipartite graph is an expander for some k and ε

Hence, the expander graphs that we consider can be thought of as tuples $G = ([n], [m], E)$ such that all subsets $S \in [n]^{(\leq k)}$ have at most $\varepsilon d |S|$ fewer neighbours than the number of edges emanating from S. It will be convenient to think of an expander in linear algebra terms, which can be done via its *adjacency matrix*.

Definition II.2 (Expander matrix $\mathbb{E}_{k,\varepsilon,d}^{m \times n}$)**.** The adjacency matrix of an unbalanced, left d-regular, bipartite graph $G = ([n], [m], E)$ is the binary sparse matrix $A \in \mathbb{R}^{m \times n}$ defined as

$$A_{ij} = \begin{cases} 1 & i \in \mathcal{N}(j) \subset [m] \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We let $\mathbb{E}_{k,\varepsilon,d}$ be a the set of adjacency matrices of (k, ε, d)-expander graphs.

We note that $A \in \mathbb{E}_{k,\varepsilon,d}^{m \times n}$ is a sparse binary matrix with exactly d ones per column, and also that any left d-regular bipartite graph will satisfy (2) for some k and ε. As mentioned previously, [13] showed that these matrices possess a bounded restricted isometry constant (RIC) in the ℓ_1 norm in the linear growth asymptotic where $k \sim m \sim n \to \infty$; making these matrices computationally highly attractive for compressed sensing. The existence of expander graphs with optimal measurement rate of $m = \mathcal{O}(k \log(n/k))$, is addressed in the following theorem.

Theorem II.3 (Existence of optimal expanders [19, 20])**.** For any $n/2 \geq k \geq 1$ and $\varepsilon > 0$, there is a (k, ε, d)-expander with

$$d = \mathcal{O}(\log(n/k)/\varepsilon), \quad \text{and} \quad m = \mathcal{O}(k \log(n/k)/\varepsilon^2). \tag{4}$$

Theorem II.3 also implies that in the linear growth asymptotic of $k \sim m \sim n \to \infty$ and for a fixed $\varepsilon > 0$, it holds that $d = \mathcal{O}(1)$; that is, the number of nonzeros per column does not increase with the problem size. Apart from this fact, expander matrices are of interest in compressed sensing because they are nearly information preserving, meaning that for $x \in \chi_k^n$ at

least $(1-2\varepsilon)kd$ entries of $y = Ax$ equal a nonzero value of x. This property is guaranteed by Lemma II.4.

Lemma II.4 (Information-preserving property)**.** Let $G = ([n], [m], E)$ be an unbalanced, left d-regular, bipartite graph, and $S \in [n]^{(\leq k)}$. Define,

$$\mathcal{N}_1(S) = \{i \in \mathcal{N}(S) : |\mathcal{N}(i) \cap S| = 1\}, \tag{5}$$

and

$$\mathcal{N}_{>1}(S) = \mathcal{N}(S) \setminus \mathcal{N}_1(S). \tag{6}$$

Then, G is a (k, ε, d)-expander graph if and only if

$$|\mathcal{N}_1(S)| > (1-2\varepsilon)d|S| \ \ \forall \ \ S \in [n]^{(\leq k)}. \tag{7}$$

Proof: See Appendix A. ∎

The information-preserving property is widely used in the analysis of CCS, and is a central piece in the analysis of our algorithms as it implies the lower ℓ_1-RIC bound [13]. Finally, we remark that adjacency matrices of expander graphs are not only useful for compressed-sensing, but also for a number of applications including linear sketching, data-stream computing, graph sketching, combinatorial group testing, network routing, error-correcting codes, fault-tolerance, and distributed storage [13, 20].

III. Overview of CCS prior art

Iterative greedy algorithms for compressed sensing seek the sparsest solution to a large underdetermined linear system of equations $y = Ax$ and typically do so by operating on the residual $r = y - A\hat{x}$, where $\hat{x}$ is an estimate of the sparsest solution. Algorithms for combinatorial compressed sensing differ by considering updating the j^{th} entry of the approximation, $\hat{x}_j$, based on a non inner product score $s_j \in \mathbb{R}$ dependent on $r_{\mathcal{N}(j)}$; that is, on the residual restricted to the support set of the j^{th} column of A. In order to standardise the convergence rate guarantees of previous CCS, we define the notion of an iteration as follows.

Definition III.1 (Iteration)**.** Let $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, and $y = Ax$. For an iterative greedy algorithm updating an estimation $\hat{x} \in \mathbb{R}^n$ of x from a residual $r = y - A\hat{x}$, an iteration is defined as the sequence of steps performed between two updates of r.

In the remainder of this section we deconstruct past CCS algorithms into their essential components so as to give a high-level overview of their shared characteristics.

A. Sparse Matching Pursuit (SMP)

SMP was proposed in [2] to decode $\hat{x}$ from $y = Ax$ with a voting-like mechanism in the spirit of the *count-median* algorithm from data-stream computing (see [21] for details). SMP can also be viewed as an *expander* adaptation of the Iterative Hard Thresholding algorithm (IHT) [22], which uses the line-search $\hat{x} \leftarrow H_k[\hat{x}+p]$ to minimise $\|y-A\hat{x}\|_2^2$ over χ_k^n, indeed it was rediscovered from this perspective in [11][pp. 452] where it is referred to as EIHT. Due to the structure of expander matrices, SMP chooses the direction $p = \mathcal{M}(y-A\hat{x})$ with $\mathcal{M} : \mathbb{R}^m \to \mathbb{R}^n$ defined as

$$[\mathcal{M}(r)]_j = \text{median}(r_{\mathcal{N}(j)}). \tag{8}$$

After thresholding, this choice yields the iteration,

$$\hat{x} \leftarrow H_k\left[\hat{x} + H_{2k}[\mathcal{M}(y - A\hat{x})]\right]. \tag{9}$$

SMP and its theoretical guarantees are stated in Algorithm 3 and Theorem III.2.

Algorithm 3: SMP [2]

Data: $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$; $y \in \mathbb{R}^m$
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $\|x - \hat{x}\|_1 = \mathcal{O}(\|y - Ax\|_1/d)$
$\hat{x} \leftarrow 0$, $r \leftarrow y$;
while *not converged* **do**
  $\hat{x} \leftarrow H_k\left[\hat{x} + H_{2k}[\mathcal{M}(r)]\right]$;
  $r \leftarrow y - A\hat{x}$;
end

Theorem III.2 (SMP [2])**.** Let $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ and let $y = Ax+\eta$ for $x \in \chi_k^n$. Then, there exists an $\varepsilon \ll 1$ such that SMP recovers $\hat{x} \in \mathbb{R}^n$ such that $\|x - \hat{x}\|_1 = \mathcal{O}(\|\eta\|_1/d)$. The algorithm terminates in $\mathcal{O}(\log(d\|x\|_1/\|\eta\|_1))$ iterations with complexity $\mathcal{O}(nd + n\log n)$.

B. Sequential Sparse Matching Pursuit (SSMP)

It was observed in [15] that SMP typically failed to converge to the sought sparsest solution when the problem parameters fall outside the region of theoretical guarantees. Though SMP updates each entry in x to individually reduce the ℓ_1 norm of the residual, by updating multiple values of x in parallel causes SMP to diverge even for moderately small ratios of k/m. To overcome these limitations, the authors proposed SSMP, which updates $\hat{x}$ sequentially rather than in parallel. That is, at each iteration, SSMP will look for a single node $j \in [n]$ and an update $\omega \in \mathbb{R}$ that minimise $\|r - \omega a_j\|_1$, which can be found by computing $\arg\max_{j\in[n]} \mathcal{M}(r)$, see the discussion in Section III-E2. This approach results in a strict decrease in $\|r\|_1$, but the sequential update results in an overall increase in computational complexity, see Table II. SSMP and its theoretical guarantees are stated in Algorithm 4 and Theorem III.3.

Algorithm 4: SSMP [15]

Data: $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$; $y \in \mathbb{R}^m$; $c > 1$;
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $\|x - \hat{x}\|_1 = \mathcal{O}(\|y - Ax\|_1/d)$
$\hat{x} \leftarrow 0$, $r \leftarrow y$;
while *not converged* **do**
  Find $(j, \omega) \in [n] \times \mathbb{R}$ s.t. $\|r - \omega a_j\|_1$ is minimized;
  $\hat{x}_j \leftarrow \hat{x}_j + \omega$;
  Perform $\hat{x} \leftarrow H_k[\hat{x}]$ every $(c-1)k$ iterations;
  $r \leftarrow y - A\hat{x}$;
end

Theorem III.3 (SSMP [15])**.** Let $A \in \mathbb{E}_{(c+1)k,\varepsilon,d}$ and let $y = Ax + \eta$ for $x \in \chi_k^n$. Then, there exists an $\varepsilon \ll 1$ such that SSMP with fixed $c > 1$ recovers $\hat{x} \in \mathbb{R}^n$ such that $\|x - \hat{x}\|_1 = \mathcal{O}(\|\eta\|_1)$. The algorithm terminates in $\mathcal{O}(k)$ iterations of complexity $\mathcal{O}\left(\frac{d^3n}{m} + n + \left(\frac{n}{k}\log n\right)\log\|x\|_1\right)$.

C. Left Degree Dependent Signal Recovery (LDDSR)

LDDSR was proposed in [16] and decodes by exploiting the information preserving property given in Lemma II.4. The main insight is that one can lower bound the number of elements in $\{i \in [m] : y_i \in \operatorname{argsupp}(x)\}$, and use the structure of A to find a $j \in [n]$ and a nonzero value $\omega \in \mathbb{R}$ that appears more than $d/2$ times in $r_{\mathcal{N}(j)} \in \mathbb{R}^d$. It is shown in [16] that updating $\hat{x}_j \leftarrow \hat{x}_j + \omega$ guarantees a decrease in $\|r\|_0$ when $\varepsilon = 1/4$. LDDSR and its theoretical guarantees are stated in Algorithm 5 and Theorem III.4.

Algorithm 5: LDDSR [16]

Data: $A \in \mathbb{E}^{m\times n}_{k,\varepsilon,d}$; $y \in \mathbb{R}^m$
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $y = A\hat{x}$
$\hat{x} \leftarrow 0$, $r \leftarrow y$;
while *not converged* **do**
  Find $(j,\omega) \in [n] \times \mathbb{R} \setminus \{0\}$ s.t.
  $|\{i \in \mathcal{N}(j) : r_i = \omega\}| > \frac{d}{2}$;
  $\hat{x}_j \leftarrow \hat{x}_j + \omega$;
  $r \leftarrow y - A\hat{x}$;
end

Theorem III.4 (LDDSR [16])**.** Let $A \in \mathbb{E}^{m\times n}_{k,\varepsilon,d}$ with $\varepsilon = 1/4$ and $x \in \chi^n_k$. Given $y = Ax$, LDDSR recovers x in at most $\mathcal{O}(dk)$ iterations with complexity $\mathcal{O}(\frac{d^3 n}{m} + n)$.

D. Expander Recovery (ER)

ER [1] differs from LDDSR by considering $\varepsilon \leq 1/4$ and suitably adapting the set of indices from which an entry in $\hat{x}$ may be updated. This modification allows the number of iterations guaranteed to be improved, see Theorem III.4. In particular, ER gurantees convergence in $\mathcal{O}(k)$ iterations of complexity $\mathcal{O}(nd)$. ER and its theoretical guarantees are stated in Algorithm 6 and Theorem III.5.

Algorithm 6: ER [1]

Data: $A \in \mathbb{E}^{m\times n}_{k,\varepsilon,d}$; $y \in \mathbb{R}^m$
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $y = A\hat{x}$
$\hat{x} \leftarrow 0$, $r \leftarrow y$;
while *not converged* **do**
  Find $(j,\omega) \in [n] \times \mathbb{R} \setminus \{0\}$ s.t.
  $|\{i \in \mathcal{N}(j) : r_i = \omega\}| \geq (1-2\varepsilon)d.$;
  $\hat{x}_j \leftarrow \hat{x}_j + \omega$;
  $r \leftarrow y - A\hat{x}$;
end

Theorem III.5 (ER [1])**.** Let $A \in \mathbb{E}_{2k,\varepsilon,d}$ with $\varepsilon \leq 1/4$ and $m = \mathcal{O}(k\log(n/k))$. Then, for any $x \in \chi^n_k$, given $y = Ax$, ER recovers x in at most $\mathcal{O}(k)$ iterations of complexity $\mathcal{O}(\frac{d^3 n}{m} + n)$.

Though ER seemingly requires knowledge of ε to implement, which is NP-hard to compute, knowledge of ε can be circumvented by selecting the node to update by

$$\arg\max_{j\in[n]} \left|\{i \in \mathcal{N}(j) : r_i = \operatorname{mode}(r_{\mathcal{N}(j)})\}\right|. \tag{10}$$

E. Discussion

Having introduced these algorithms, we now point out some important commonalities between them.

1) Iterative greedy algorithms: The CCS algorithms we have presented share the structure of Algorithm 7.

Algorithm 7: Iterative greedy CCS algorithms

Data: $A \in \mathbb{R}^{m\times n}$; $y \in \mathbb{R}^m$
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $y = A\hat{x}$
$\hat{x} \leftarrow 0$, $r \leftarrow y$;
while *not converged* **do**
  Compute a *score* s_j and an *update* u_j $\forall$ $j \in [n]$;
  Select $S \subset [n]$ based on a rule on s_j;
  $\hat{x}_j \leftarrow \hat{x}_j + u_j$ for $j \in S$;
  k-threshold $\hat{x}$;
  $r \leftarrow y - A\hat{x}$;
end

The dominant computational cost in CCS greedy algorithms is concentrated in computing s_j and u_j, and in selecting the set S of nodes that will be updated. At each step of these algorithms, a subset $S \subset [n]$ is selected. In SMP, we have $S = [n]$ which makes it of sublinear complexity in $\|x\|_1$, but typically diverges for even moderate values of $\rho := k/m$. All other algorithms update a single entry of $\hat{x}$ per iteration; that is, they choose $S \subset [n]$ with $|S| = 1$. This brings benefits in terms of convergence and recovery region, but compromises the computational complexity of the algorithms. A summary of these properties is given in Table II.

2) Median minimises $\|r\|_1$: The operation $\operatorname{median}(r_{\mathcal{N}(j)})$ can be recast as the problem of finding the scalar $\omega \in \mathbb{R}$ that minimises $\|r - \omega a_j\|_1$. To see this, note that the function

$$\|r - \omega a_j\|_1 = \sum_{i\in\mathcal{N}(j)} |r_i - \omega| + \text{constant} \tag{11}$$

is at a minimum when $|\{i \in \mathcal{N}(j) : r_i - \omega > 0\}| = |\{i \in \mathcal{N}(j) : r_i - \omega < 0\}|$. Then, by definition of the median,

$$\arg\min_{\omega\in\mathbb{R}} \|r - \omega a_j\|_1 = \operatorname{median}(r_{\mathcal{N}(j)}) \tag{12}$$

This is independent of the expansion parameter ε.

3) Mode does not minimise $\|r\|_0$: In [1, 16], it is shown that Algorithms 5 and 6 use Lemma II.4 to find a pair (j,ω) such that

$$\|y - A(\hat{x} + \omega e_j)\|_0 < \|y - A\hat{x}\|_0 - (1-4\varepsilon)d. \tag{13}$$

However, when $(y - A\hat{x})_{\mathcal{N}(j)}$ does not contain any zeros, we can guarantee that

$$\|y - A(\hat{x} + \omega e_j)\|_0 < \|y - A\hat{x}\|_0 - (1-2\varepsilon)d. \tag{14}$$

For dissociated signals, where $\sum_{j\in\operatorname{supp}(x)} x_j \neq 0$, we can always ensure that the greater contraction rate will be achieved.

4) Updating s_j and u_j: Algorithms 4, 5, 6 need to compute a score $s_j = s_j(r_{\mathcal{N}(j)})$ for each $j \in [n]$, which can be done at cost $\mathcal{O}(dn)$. It is important to note that they do not need to recompute all the scores at each iteration. A common strategy is to compute each of the scores once and store them with their corresponding node $j \in [n]$ in some data structure (like priority queues [15] or red-black trees [1]). Then, at each iteration, we can efficiently request the node $j \in [n]$ that maximises the score (median, mode, etc.) and use it to update $\hat{x}_j$. This update will affect $d = |\mathcal{N}(j)|$ entries of the residual, so we only need to recompute the scores corresponding to $|\bigcup_{i\in\mathcal{N}(j)} \mathcal{N}(i)| = \mathcal{O}(d^2 n/m)$ right nodes.

IV. Main contributions: Iterative ℓ_0-minimisation

Our main contributions, Serial-ℓ_0 and Parallel-ℓ_0, advance combinatorial compressed sensing by having comparatively high phase transitions while retaining the low computational complexity of SMP and the parallel implementation of LDDSR. In particular, Parallel-ℓ_0 is observed to typically recover the sparsest solution of underdetermined systems of equations in less time than any other compressed sensing algorithm when the signal is dissociated and the sensing matrix is an expander graph.

Serial-ℓ_0 and Parallel-ℓ_0 look for a solution by identifying nodes which if updated sequentially would strictly reduce the $\|r\|_0$ by at least α. That is, they will choose a coordinate j of x, and an update value ω such that,

$$(j,\omega) \in [n] \times \mathbb{R} \quad \text{s.t.} \quad \|r\|_0 - \|r - \omega a_j\|_0 \geq \alpha, \tag{15}$$

for some $\alpha \in (1, d]$. By selecting a pair (j,ω) satisfying (15), Serial-ℓ_0 yields a decrease in $\|r\|_0$ at every update, and is guaranteed to converge in $\mathcal{O}(n \log k)$ iterations of computational complexity $\mathcal{O}(d)$ if the signal is dissociated. Parallel-ℓ_0 is designed similarly, but adapted to be able to take full advantage of modern massively parallel computational resources. Indeed, Parallel-ℓ_0 selects and update all pairs (j,ω) satisfying (15) and updates these values in x in parallel. Under this updating scheme, a strict contraction in $\|r\|_0$ is guaranteed at every iteration when the signal is dissociated and $\alpha = (1-2\varepsilon)d$ with $\varepsilon \leq 1/4$, though we show in Section V that one can fix $\alpha = 2$ and get high phase transitions and exceptional speed.

Section IV-A presents the key technical lemmas that explain the behaviour of an iteration of Serial-ℓ_0 and Parallel-ℓ_0. In particular, technical lemmas are stated to show how often values in Ax appear when $x \in \chi_k^n$ and $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$, and that when a value in Ax appears sufficiently often it must be a value from x at a specified location. This property ensures the algorithm updates its approximation $\hat{x}$ with values x_j in the j^{th} entry, that is with the exact values from x at the correct locations. The dissociated signal model, Definition I.1, is an essential component in the analysis presented in Section IV-A, though we will observe that the algorithms' recovery region degrade gracefully as the fraction of duplicate entries in x increases. The convergence rate of Serial-ℓ_0 and Parallel-ℓ_0 are presented in Section IV-B, and together they establish Theorem I.2.

A. Technical lemmas

Lemma IV.1 (Properties of dissociated signals)**.** Let $x \in \chi_k^n$ be dissociated. Then,

(i) $x_i \neq x_j \ \ \forall\, i,j \in \text{supp}(x), \ \ i \neq j$.
(ii) $\sum_{j\in T} x_j \neq 0 \ \ \forall\, \emptyset \neq T \subset \text{supp}(x)$.

Proof: The result follows from (1). For (i) we set $T_1 = \{i\}$ and $T_2 = \{j\}$, and for (ii) we let $T_2 = \emptyset$. ∎

Lemma IV.2 (Bounded frequency of values in expander measurements of dissociated signals)**.** Let $x \in \chi_k^n$ be dissociated, $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$, and ω a nonzero value in Ax. Then, there is a unique set $T \subset \text{supp}(x)$ such that $\omega = \sum_{j\in T} x_j$ and the value ω occurs in y at most d times,

$$|\{i \in [m] : y_i = \omega\}| \leq d \quad \forall\, \omega \neq 0. \tag{16}$$

Proof: The uniqueness of the set $T \subset \text{supp}(x)$ such that $\omega = \sum_{j\in T} x_j$ follows by the definition of dissociated. Since $|\mathcal{N}(j)| = d$ for all $j \in [n]$, we have that,

$$|\{i \in [m] : y_i = \omega\}| = \left|\bigcap_{j\in T} \mathcal{N}(j)\right| \leq |\mathcal{N}(j_0)| = d \tag{17}$$

for any $j_0 \in T$. ∎

Lemma IV.3 (Pairwise column overlap)**.** Let $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$. If $\varepsilon \leq 1/4$, every pair of columns of A intersect in less than $(1-2\varepsilon)d$ rows, that is, for all $j_1, j_2 \in [n]$ with $j_1 \neq j_2$

$$\left|\mathcal{N}(j_1) \bigcap \mathcal{N}(j_2)\right| < (1-2\varepsilon)d. \tag{18}$$

Proof: Let $S \subset [n]$ be such that $|S| = 2$ then

$$|\mathcal{N}(S)| > 2(1-\varepsilon)d \geq 2d - (1-2\varepsilon)d, \tag{19}$$

where the first inequality is Definition II.1 and the second inequality follows from $\varepsilon \leq 1/4$. However, $|\mathcal{N}(S)|$ can be rewritten as

$$|\mathcal{N}(S)| = |\mathcal{N}(j_1)| + |\mathcal{N}(j_2)| - \left|\mathcal{N}(j_1) \bigcap \mathcal{N}(j_2)\right|, \tag{20}$$

for some $j_1, j_2 \in [n]$. Coupling (20) with (19) gives (18). ∎

Lemma IV.4 (Progress)**.** Let $y = Ax$ for dissociated $x \in \chi_k^n$ and $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ with $\varepsilon \leq 1/4$. There is a pair $(j,\omega) \in [n] \times \mathbb{R}$ such that

$$|\{i \in \mathcal{N}(j) : y_i = \omega\}| \geq (1-2\varepsilon)d. \tag{21}$$

Proof: Let $S = \text{supp}(x)$, then by the information-preserving property (7) it holds that $|\mathcal{N}_1(S)| > (1-2\varepsilon)d|S|$, where $\mathcal{N}_1(S)$ is defined in (5), or alternatively, by $\mathcal{N}_1(S) = \{i \in [m] : y_i = x_j, j \in S\}$ in the context of dissociated signals. Given the lower bound in $|\mathcal{N}_1(S)| > (1-2\varepsilon)d|S|$, if $|S| \neq 0$, at least one $j \in S$ must have at least $(1-2\varepsilon)d$ neighbours in y with identical nonzero entries. Letting ω take the value of such repeated nonzeros in y gives the required pair $(j,\omega) \in [n] \times \mathbb{R}$. ∎

Lemma IV.5 (Support identification)**.** Let $y = Ax$ for dissociated $x \in \chi_k^n$ and $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ with $\varepsilon \leq 1/4$. Let $\omega \neq 0$ be such that (21) and $\omega = x_j$.

		Objective	Score	Signal	Concurrency	Number of iterations	Iteration cost
Prior art	SMP [2]	ℓ_1	median	any	parallel	$\mathcal{O}(\log \|x\|_1)$	$\mathcal{O}(nd + n\log n)$
	SSMP [15]	ℓ_1	median	any	serial	$\mathcal{O}(k)$	$\mathcal{O}(d^3 n/m + n + (\frac{n}{k}\log n)\log \|x\|_1)$
	LDDSR [16]	ℓ_0	mode	any	serial	$\mathcal{O}(dk)$	$\mathcal{O}(\frac{d^3 n}{m} + n)$
	parallel-LDDSR	ℓ_0	mode	dissociated	parallel	$\mathcal{O}(\log k)$	$\mathcal{O}(nd)$
	ER [1]	ℓ_0	mode	any	serial	$\mathcal{O}(k)$	$\mathcal{O}(\frac{d^3 n}{m} + n)$
Contributions	serial-ℓ_0	ℓ_0	ℓ_0-decrease	dissociated	serial	$\mathcal{O}(n\log k)$	$\mathcal{O}(d)$
	parallel-ℓ_0	ℓ_0	ℓ_0-decrease	dissociated	parallel	$\mathcal{O}(\log k)$	$\mathcal{O}(nd)$

TABLE II: Summary of prior art in combinatorial compressed-sensing.

Proof: Our claim is that for any ω which is a nonzero value from y, if the cardinality condition (21) is satisfied then the value $\omega = \sum_{j\in T} x_j$ occurs for the set T being a singleton, $|T| = 1$. Lemma IV.2 states that T is unique and that

$$|\{i \in \mathcal{N}(j) : y_i = \omega\}| = \left|\bigcap_{j\in T} \mathcal{N}(j)\right|. \tag{22}$$

If $|T| > 1$ then the above is not more than the cardinality of the intersection of any two of the sets $\mathcal{N}(j_1)$ and $\mathcal{N}(j_2)$, and by (18) in Lemma IV.3 that is less than $(1-2\varepsilon)d$ which contradicts the cardinality condition (21) and consequently $|T| \leq 1$. However, Lemma IV.4 guarantees that $|T| > 0$, so $|T| = 1$ and $\omega = x_j$. ∎

Equipped with Lemmas IV.1 - IV.5 we prove Theorem I.2 considering Serial-ℓ_0 and Parallel-ℓ_0 separately, beginning with the later. Note that since $x \in \chi_k^n$ and the algorithm only sets entries in $\hat{x}$ to the correct values of x, then $x - \hat{x} \in \chi_k^n$, and Lemmas IV.4 and IV.5 hold with y replaced by $r = y - A(x - \hat{x})$.

B. Proof of Theorem I.2

Theorem IV.6 (Convergence of Parallel-ℓ_0)**.** Let $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ and let $\varepsilon \leq 1/4$, and $x \in \chi_k^n$ be dissociated. Then, Parallel-ℓ_0 with $\alpha = (1-2\varepsilon)d$ can recover x from $y = Ax \in \mathbb{R}^m$ in $\mathcal{O}(\log k)$ iterations of complexity $\mathcal{O}(dn)$.

Proof: Let $\hat{x} = 0$ be our initial approximation to $x \in \chi_k^n$. During the ℓ^{th} iteration of Parallel-ℓ_0, let $S_\ell = \text{supp}(x - \hat{x})$ and include a subscript on the identification set $T = T_\ell \subset [n]$. As $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ and $\varepsilon \leq 1/4$, by Lemma IV.5 and the required entry-wise reduction in the residual by at least $\alpha = (1-2\varepsilon)d$, it follows that Parallel-ℓ_0 only sets entries in $\hat{x}$ to the correct values of x and as a result $\|x - \hat{x}\|_0 \leq \|x\|_0 = k$ for every iteration. Moreover, by Lemma IV.4, the set $T_\ell \neq \emptyset$ as long as $x \neq \hat{x}$, so the algorithm eventually converges.

In fact, we show that the rate of reduction of $\|x - \hat{x}\|_0$ per iteration is by at least a fixed fraction $\frac{2\varepsilon d}{1+\lfloor 2\varepsilon d\rfloor}$. As $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ has d nonzeros per column, the reduction in the cardinality of the residual, say $\|r^\ell\|_0 - \|r^{\ell+1}\|_0$, can be at most $d|T_\ell|$. That is,

$$\|r^\ell\|_0 - \|r^{\ell+1}\|_0 \leq d|T_\ell|. \tag{23}$$

To establish a fractional decrease in $|S_{\ell+1}|$ we develop a lower bound on $\|r^\ell\|_0 - \|r^{\ell+1}\|_0$. For $Q \subset S_\ell$ define the set $\mathcal{N}_1^{S_\ell}(Q)$ to be the set of nodes in $\mathcal{N}_1(S_\ell)$ and such that $i \in \mathcal{N}(j)$ for some $j \in Q$, *i.e.*

$$\mathcal{N}_1^{S_\ell}(Q) = \{i \in \mathcal{N}_1(S_\ell) : i \in \mathcal{N}(j), j \in Q\}. \tag{24}$$

Consider the partition $S_\ell = T_\ell \cup (S_\ell \setminus T_\ell)$ and rewrite $\mathcal{N}_1(S_\ell)$ as the disjoint union

$$\mathcal{N}_1(S_\ell) = \mathcal{N}_1^{S_\ell}(T_\ell) \cup \mathcal{N}_1^{S_\ell}(S_\ell \setminus T_\ell). \tag{25}$$

Note that $\mathcal{N}_1^{S_\ell}(T_\ell) \neq \mathcal{N}_1(T_\ell)$, and that by (24) and the dissociated signal model, $\mathcal{N}_1^{S_\ell}(T_\ell) \subset [m]$ is the set of indices in r^ℓ that are identical to a nonzero in x and that have a frequency of at least $\alpha = (1-2\varepsilon)d$, so

$$\|r^\ell\|_0 - \|r^{\ell+1}\|_0 \geq |\mathcal{N}_1^{S_\ell}(T_\ell)|. \tag{26}$$

At iteration ℓ, if $T_\ell = S_\ell$, the full support of x is correctly identified, so $x = \hat{x}$ after updating $\hat{x}$. Otherwise, $T_\ell \neq S_\ell$ and the set $S_\ell \setminus T_\ell$ is not identified by the algorithm at this iteration. We derive a lower bound on $|\mathcal{N}_1^{S_\ell}(T_\ell)|$ by considering two cases: $\alpha \in \mathbb{N}$ and $\alpha \notin \mathbb{N}$.

If $\alpha \in \mathbb{N}$, then each node in $S_\ell \setminus T_\ell$ has at most $\alpha - 1$ duplicates in r^ℓ, so

$$\left|\mathcal{N}_1^{S_\ell}(S_\ell \setminus T_\ell)\right| \leq (\alpha - 1)\,|S_\ell \setminus T_\ell|. \tag{27}$$

Using the the information-preserving property (7) and the identity given in (24) it follows that

$$\begin{aligned} &|\mathcal{N}_1^{S_\ell}(T_\ell)| + |\mathcal{N}_1^{S_\ell}(S_\ell \setminus T_\ell)| \\ &> (1-2\varepsilon)d\,(|T_\ell| + |S_\ell \setminus T_\ell|) \\ &= (1-2\varepsilon)d|T_\ell| + |S_\ell \setminus T_\ell| + (\alpha-1)|S_\ell \setminus T_\ell|. \end{aligned} \tag{28}$$

Now, using (27) to lower bound (28), and solving for $|\mathcal{N}_1^{S_\ell}(T_\ell)|$ gives

$$|\mathcal{N}_1^{S_\ell}(T_\ell)| \geq (1-2\varepsilon)d|T_\ell| + |S_\ell \setminus T_\ell|. \tag{29}$$

By coupling (29), (26), and (23) into a chain of inequalities it is seen that

$$(1-2\varepsilon)d|T_\ell| + (|S_\ell| - |T_\ell|) \leq d|T_\ell|, \tag{30}$$

which simplifies to

$$|T_\ell| \geq \frac{1}{1+2\varepsilon d}|S_\ell|. \tag{31}$$

If $\alpha \notin \mathbb{N}$, then each node in $S_\ell \setminus T_\ell$ has at most $\lfloor \alpha \rfloor$ duplicates in r^ℓ, so

$$\left|\mathcal{N}_1^{S_\ell}(S_\ell \setminus T_\ell)\right| \leq \lfloor \alpha \rfloor |S_\ell \setminus T_\ell|. \tag{32}$$

Similarly as in the former case, using (24) and the the information-preserving property (5), we obtain

$$\begin{aligned} &|\mathcal{N}_1^{S_\ell}(T_\ell)| + |\mathcal{N}_1^{S_\ell}(S_\ell \setminus T_\ell)| \\ &> (1-2\varepsilon)d\left(|T_\ell| + |S_\ell \setminus T_\ell|\right) \\ &= (1-2\varepsilon)d|T_\ell| + (\alpha - \lfloor \alpha \rfloor)|S_\ell \setminus T_\ell| + \lfloor \alpha \rfloor |S_\ell \setminus T_\ell|. \end{aligned} \tag{33}$$

Just as in the previous case, (33) is bounded from below using (32), and the resulting inequality is used to get

$$|\mathcal{N}_1^{S_\ell}(T_\ell)| \geq (1-2\varepsilon)d|T_\ell| + (\alpha - \lfloor \alpha \rfloor)|S_\ell \setminus T_\ell|. \tag{34}$$

Inequalities (34), (26), and (23) are then used to derive

$$\alpha|T_\ell| + (\alpha - \lfloor \alpha \rfloor)(|S_\ell| - |T_\ell|) \leq d|T_\ell|. \tag{35}$$

It follows from $\alpha = (1-2\varepsilon)d \notin \mathbb{N}$ and the properties of step functions that (35) is equivalent to

$$|T_\ell| \geq \frac{1 - 2\varepsilon d + \lfloor 2\varepsilon d \rfloor}{1 + \lfloor 2\varepsilon d \rfloor}|S_\ell|. \tag{36}$$

Finally, note that (36) reduces to (31) when $\alpha \in \mathbb{N}$, so using $S_{\ell+1} = S_\ell \setminus T_\ell$ and (36), we conclude that

$$|S_{\ell+1}| \leq \frac{2\varepsilon d}{1 + \lfloor 2\varepsilon d \rfloor}|S_\ell|. \tag{37}$$

Since $|S_0| = k$ it follows that Parallel-ℓ_0 will have converged after ℓ^* iterations when $k(2\varepsilon d/(1+\lfloor 2\varepsilon d \rfloor))^{\ell^*} < 1$, which is achieved for

$$\ell^* \geq \left(\log^{-1}\left(\frac{1+\lfloor 2\varepsilon d \rfloor}{2\varepsilon d}\right)\right)\log k. \tag{38}$$

Each iteration of Parallel-ℓ_0 involves computing (21) for each $j \in [n]$, which is equivalent to n instances of finding the mode of a vector of length d which can be solved in $\mathcal{O}(d)$ complexity provided $\alpha > \lfloor d/2 \rfloor$[23]. ∎

Theorem IV.7 (Convergence of Serial-ℓ_0)**.** Let $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ and let $\varepsilon \leq 1/4$, and $x \in \chi_k^n$ be a dissociated signal. Then, Serial-ℓ_0 with $\alpha = (1-2\varepsilon)d$ can recover x from $y = Ax \in \mathbb{R}^m$ in $\mathcal{O}(n \log k)$ iterations with complexity $\mathcal{O}(d)$.

Proof: The loop over $j \in [n]$ for Serial-ℓ_0 identifies singletons T to update values in $\hat{x}$ in serial. The union of the singletons for $j \in [n]$ includes the set of all nodes for which the residual would be reduced by at least α if one were to forgo the serial update in $\hat{x}$. For $\alpha = (1-2\varepsilon)d$, the proof of convergence for Theorem IV.6 establishes that this results in a reduction of the cardinality of $\text{supp}(x - \hat{x})$ by at least a fraction $2\varepsilon d/(1+\lfloor 2\varepsilon d \rfloor)$. That is, for p an integer, Serial-ℓ_0 satisfies

$$|\text{supp}(x - \hat{x})| \leq k\left(\frac{2\varepsilon d}{1+\lfloor 2\varepsilon d \rfloor}\right)^p \tag{39}$$

after $\ell = pn$ iterations, and converges to $\hat{x} = x$ after at most $p^* > \log(k)/\log((1+\lfloor 2\varepsilon d \rfloor)/(2\varepsilon d))$ for convergence after

$$\ell^* \geq n\left(\log^{-1}\left(\frac{1+\lfloor 2\varepsilon d \rfloor}{2\varepsilon d}\right)\right)\log k. \tag{40}$$

iterations. Each iteration of Serial-ℓ_0 involves computing the mode of a vector of length d and updating d entries in the residual. Since we are interested in knowing the mode of $r_{\mathcal{N}(j)}$ only when the most frequent element occurs more than $d/2$ times, this value can be found at cost $\mathcal{O}(d)$ [23]. ∎

C. Discussion

1) The computational cost of computing a mode can be improved if d is small: Evaluating (21) for a given column $j \in \text{supp}(x)$ is equivalent to finding the mode of $r_{\mathcal{N}(j)}$. This can be done at cost $\mathcal{O}(d)$ using the Boyer-Moore Majority vote algorithm [23]. However, this algorithm requires that an element of the array occurs more than $\lfloor d/2 \rfloor$ times, so it might fail when we set $\alpha \in [\lfloor d/2 \rfloor]$. Our numerical experiments (Section V) show that best recovery regions are obtained for $\alpha = 2$, so we prefer to have an algorithm with $\mathcal{O}(d)$ per-iteration cost for all $\alpha \in [d]$.

Our approach is presented in Algorithm 8. Instead of looking for an $\omega \in \mathbb{R}$ satisfying (21) for each $j \in [n]$, at the ℓ^{th} iteration we consider the reduction caused by ω_j, defined as the $\ell \pmod d$-th element in $r_{\mathcal{N}(j)}$. When using this shifting strategy we compromise the final number of iterations, but we also keep a fixed cost of d complexity per iteration for any $\alpha \in [d]$. The convergence guarantees of our algorithms when using this shifting strategy are presented in Theorem IV.8.

Algorithm 8: Computation of score for serial-ℓ_0 and parallel-ℓ_0.

Data: $j \in [n]$; $r \in \mathbb{R}^m$; $\omega \in \mathbb{N}$
Result: $s_j \leftarrow |\{i \in \mathcal{N}(j) : r_i = \omega\}|$

Theorem IV.8 (Convergence of Shifted Parallel-ℓ_0)**.** Let $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ with $\varepsilon \leq 1/4$, and $x \in \chi_k^n$ be dissociated. Then, the shifted versions of Serial-ℓ_0 and Parallel-ℓ_0 with $\alpha = (1-2\varepsilon)d$ can recover x from $y = Ax \in \mathbb{R}^m$ in an average of $\mathcal{O}(dn \log k)$ operations.

Proof: Let $\hat{x} = 0$ be the initial approximation to $x \in \chi_k^n$, and $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ with $\varepsilon \leq 1/4$. At ℓ^{th} iteration, let $T = T_\ell$ be the set satisfying (21), that is, the one that Parallel-ℓ_0 has marked for update. For $j \in T$, let ω_j be the most frequent element in $r_{\mathcal{N}(j)}$. In shifted-parallel-ℓ_0, ω_j is not directly computed. Instead, at iteration ℓ, the frequency of the $\ell \pmod d$-th value in $r_{\mathcal{N}(j)}$ is computed using Algorithm 8 and tested against the imposed threshold α. In the worst case, this increases the number of iterations by a factor $\mathcal{O}(d)$. However, on average, this is not the case, and convergence in $\mathcal{O}(\log k)$ iterations is guaranteed.

To see this, let $j \in T$ and let ω be drawn at random from $r_{\mathcal{N}(j)}$. Then, $\Pr(\omega = \omega_j) \geq 1 - 2\varepsilon$, so on average at iteration ℓ we will identify $|T_\ell|(1-2\varepsilon)$ correct entries in $\text{supp}(x - \hat{x})$. Given the bound for $|T_\ell|$ (36) in the proof of parallel-ℓ_0, we have that at each iteration we identify at least $(1-2\varepsilon)\frac{1-2\varepsilon d+\lfloor 2\varepsilon d \rfloor}{1+\lfloor 2\varepsilon d \rfloor}|S_\ell|$. Therefore

$$|S_{\ell+1}| \leq \left(\frac{(1-2\varepsilon)(1-2\varepsilon d + \lfloor 2\varepsilon d \rfloor)}{1+\lfloor 2\varepsilon d \rfloor} \right) |S_\ell|. \tag{41}$$

■

2) Our theoretical guarantees immediately apply to LDDSR: When $\varepsilon = 1/4$, we have that $(1-2\varepsilon)d = d/2$, so we recover a parallel version of LDDSR (Algorithm 5) for dissociated signals. We call this algorithm Parallel-LDDSR, and we test its performance in Section V.

3) Non-dissociated signals can be recovered with a dissociated A: There are many signals models in which the dissociated condition does not hold. For instance, if x is a binary signal or has integer-valued nonzeros. In this case, the sensing matrix A can be modified to make the nonzero elements of x identifiable by our algorithms. In particular, scaling each column of the matrix by $i.i.d.$ random numbers coming from a continuous distribution introduces enough information in y for our algorithms to correctly identify $\text{supp}(x)$.

4) Expander matrices preserve information of dissociated signals: We now discuss the concept of dissociated signals under an Information Theory viewpoint. To do this, suppose that $(X_1, \ldots, X_k)$ is a vector of k random variables associated with $\{x_1, \ldots, x_k\} = \text{supp}(x)$ and that $(X_1, \ldots, X_k) \sim p$ for some distribution p supported on a finite set. Note that condition (iii) in Definition I.1 implies that,

$$x_{i_1} + \cdots + x_{i_\ell} \neq x_{j_1} + \cdots + x_{j_\ell} \qquad \text{for} \qquad i_1 \neq j_1, \ldots, i_\ell \neq j_\ell. \tag{42}$$

Now, consider the following Shannon-entropy inequalities,

Lemma IV.9 (Entropy inequalities)**.** For a random variable $X \sim p$, let $H(\cdot)$ be its Shannon entropy. Now, let $X_1, \cdots, X_k$ be a set of random variables with joint distribution $(X_1, \ldots, X_k) \sim p$. Assume that the random variable X_i is supported on $\{(x_i)_1, \ldots, (x_i)_\ell\}$. Then,

$$H(X_1 + \cdots + X_k) \leq H(X_1, \ldots, X_k) \leq H(X_1) + \cdots H(H_k) \tag{43}$$

With equality on the left if and only if $(x_1)_{i_1} + \cdots + (x_k)_{i_k} \neq (x_1)_{j_1} + \cdots + (x_k)_{j_k}$ for $i_l \neq j_l$, and equality on the right if and only if $X_i \perp X_j$ for $i \neq j$.

Proof: See [24] and [25] for a proof. ■

In the case of discretely supported distributions, a dissociated signal can be understood as one in which the entries on $\text{supp}(x)$ are drawn according to a distribution p fulfilling,

(i) $\Pr[X_i = \omega \mid X_j = \omega] = 0 \ \forall \ i \neq j \ \forall \ \omega \neq 0$.
(ii) $\Pr[\sum_{j \in T} X_j = 0] = 0 \ \forall \ T \subset [k]$
(iii) $\Pr[\sum_{j \in T_1} X_j = \sum_{j \in T_2} X_j] = 0 \ \forall \ T_1, T_2 \subset [k]$ with $T_1 \neq T_2$.

Property (iii) above, together with Lemma (IV.9) say that probability distribution on the support of dissociated signals imply

$$H\left(\sum_{j \in T} X_j\right) = H(X_1, \ldots, X_k) \qquad \forall \qquad T \subset [k] \tag{44}$$

And since the value of each entry in $y = Ax$ is distributed according to $\sum_{j \in T} X_j$ for some $T \subset [k]$, we get that when computing y with a $A \in \mathbb{E}^{m \times n}_{k,\varepsilon,d}$ having $\varepsilon \leq 1/4$ and a dissociated signal x, (44) will hold. This implies that linear transformations with expander matrices preserve the information in x.

V. Numerical Experiments

In this section we perform a series of numerical experiments to compare Parallel-ℓ_0 and Serial-ℓ_0 with state-of-the-art compressed sensing algorithms. These comparisons are done by adding Parallel-ℓ_0 and Serial-ℓ_0 to the GAGA software package [26] which includes `CUDA-C` implementations of a number of compressed sensing algorithms as well as a testing environment to rapidly generate synthetic problem instances. This approach allows us to solve hundreds of thousands of randomly generated problems and to solve problems with n in the millions.

Unless otherwise stated, all tests were performed with the nonzeros of x drawn from a standard normal distribution $\mathcal{N}(0,1)$ and the parameter α in Serial-ℓ_0 and Parallel-ℓ_0 was set to 2.

Figures 2-8 were computed using a Linux machine with Intel Xeon E5-2643 CPUs @ 3.30 GHz, NVIDIA Tesla K10 GPUs, and executed from Matlab R2015a. Figures 9-11 were computed using a Linux machine with Intel Xeon E5-2667 v2 CPUs @ 3.30GHz, NVIDIA Tesla K40 GPUs, and executed from Matlab R2015a.

A. Substantially higher phase transitions

The phase transition of a compressed-sensing algorithm [27] is the largest value k/m, which we denote $\rho^*(m/n)$ noting its dependence on m/n, for which the algorithm is typically (say greater than half of the instances) able recovery all k sparse vectors with $k < m\rho^*(m/n)$. The value $\rho^*(m/n)$ often converges to a fixed value as n is increased with m/n being a fixed fraction. Figure 2 shows the phase transition curve for each of the CCS algorithms stated in Section III, as well as Parallel-ℓ_0 and Serial-ℓ_0. To facilitate comparison with non-CCS algorithms, Figure 2 also includes the theoretical phase transition curve for ℓ_1-regularization for A drawn Gaussian [28, 29], which is observed to be consistent [30] with ℓ_1-regularization for $A \in \mathbb{E}^{m \times n}_{k,\varepsilon,d}$. The curves were computed by setting $n = 2^{18}$, $d = 7$, and a tolerance of 10^{-6}. The testing is done at $m = \delta_p n$ for

$$\delta_p \in \{0.02p : p \in [4]\} \cup \left\{ 0.1 + \frac{89}{1900}(p-1) : p \in [20] \right\}.$$

For each δ_p, we set $\rho = 0.01$ and generate 10 synthetic problems to be applied to the algorithms, with x having independent and identically distributed normal Gaussian entries. With this restrictions, our signals are dissociated. If at least

one such problem was recovered successfully, we increase ρ by 0.01 and repeat the experiment. The recovery data is then fitted using a logistic function in the spirit of [31] and the 50% recovery transition of the logistic function is computed and shown in Figure 2.

Note the low phase-transition curve of SMP and the substantially higher phase-transition curve of Parallel-ℓ_0 and Serial-ℓ_0. As mentioned previously, the multiple updating mechanism of SMP gives it sublinear convergence guarantees, but greatly compromises its region of recovery. We emphasise that the phase transition curves for Serial-ℓ_0 and Parallel-ℓ_0 are higher than those for SMP, SSMP, ER, and parallel-LDDSR. In particular, they are even higher than ℓ_1-regularisation for $\delta \lessapprox 0.4$.

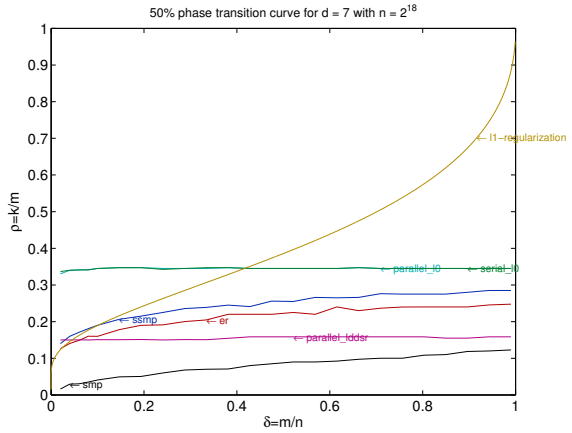

Fig. 2: 50% recovery probability logistic regression curves for $\mathbb{E}_{\varepsilon,7,k}$ and $n = 2^{18}$. The curve for ℓ_1-regularisation is the theoretical curve for dense Gaussian ensembles, and is shown for reference.

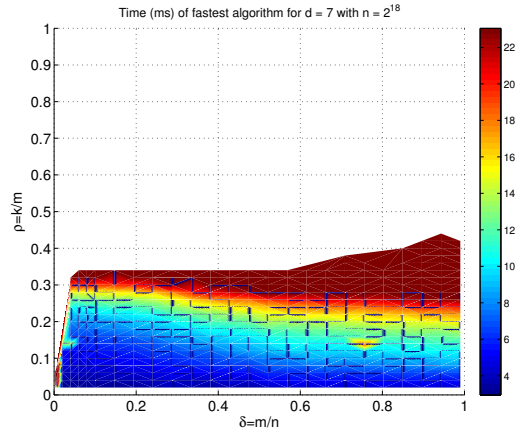

Fig. 3: Average recovery time (ms) of the fastest algorithm at each (δ, ρ) for $\mathbb{E}_{k,\varepsilon,7}$ and $n = 2^{18}$.

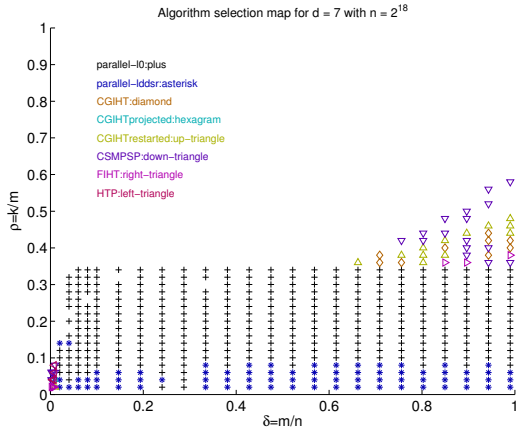

Fig. 4: Selection map of the fastest algorithm at each (δ, ρ) for $\mathbb{E}_{k,\varepsilon,7}$ and $n = 2^{18}$.

B. Fastest compressed sensing algorithm

When the signal is dissociated, Parallel-ℓ_0 is generally the fastest algorithm for matrices $A \in \mathbb{E}^{m\times n}_{k,\varepsilon,d}$. We show this numerically by computing the phase transitions of

Serial-ℓ_0, Parallel-ℓ_0, parallel-LDDSR, ALPS, CGIHT, CSMPSP, ER, FIHT, HTP, NIHT, SMP, SSMP;

and comparing their average time to convergence at each point of (δ, ρ). The phase transitions are computed similarly to those in Figure 2, with problem parameters of $n = 2^{18}$ and $d = 7$. In particular, Parallel-ℓ_0 is also used with $\alpha = 2$. The results are shown in Figures 3 and 4. Specifically, Figure 3 shows the time in milliseconds that the fastest algorithm takes to converge when the problem parameters are located at (δ, ρ). The fastest algorithm is in turn identified in Figure 4, where we can see that Parallel-ℓ_0 is consistently the fastest algorithm within its phase transition, except for $\rho \ll 1$ where parallel-LDDSR takes less time. However, we note that the convergence guarantees of parallel-LDDSR come as a byproduct of our analysis the domain in which it is faster than Parallel-ℓ_0 is the region of least importance for applications as it indicates more than three fold more measurements were taken than would have been necessary if Parallel-ℓ_0 were used.

C. Parallelisation brings important speedups: examples with $m \ll n$

As shown in Algorithm 7, the speed of Algorithms 3-6 can be improved if the scores s_j and updates u_j are computed in parallel for each $j \in [n]$. However, implementing this parallelisation is not enough to cut down an algorithm's complexity to that of the state-of-the-art's. Figures 5-6 show the average time to exact convergence for each of the combinatorial compressed sensing algorithms. It can be seen in addition to Serial-ℓ_0 and Parallel-ℓ_0 having higher phase transition than ER and SSMP, they are also substantially faster to converge to the true solution for $n = 2^{20}$ and either $\delta = 0.01$ or $\delta = 0.1$. It is interesting to note that for this problem size Serial-ℓ_0 is substantially faster than ER and SSMP, even when the two latter are implemented in parallel and run on a modern high performance computing GPU.

D. Convergence in $\mathcal{O}(\log k)$ iterations

The theoretical guarantees of Serial-ℓ_0 and Parallel-ℓ_0 state that convergence can be achieved in $\mathcal{O}(nd \log k)$ operations. The number of operations per iteration can be verified simply by counting operations in the algorithm, which is $\mathcal{O}(d)$ for

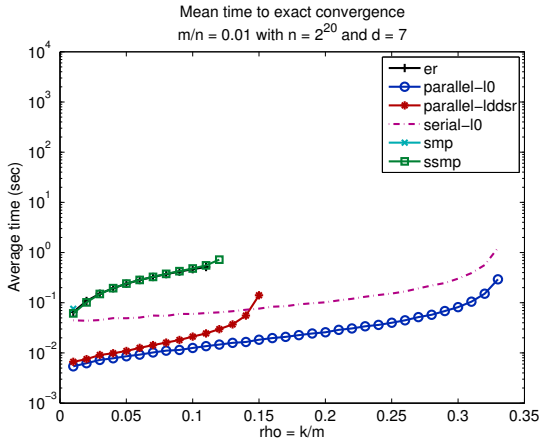

Fig. 5: Average recovery time (sec) with dependence on ρ for $\delta = 0.01$ and $\mathbb{E}_{k,\varepsilon,7}$ with $n = 2^{20}$.

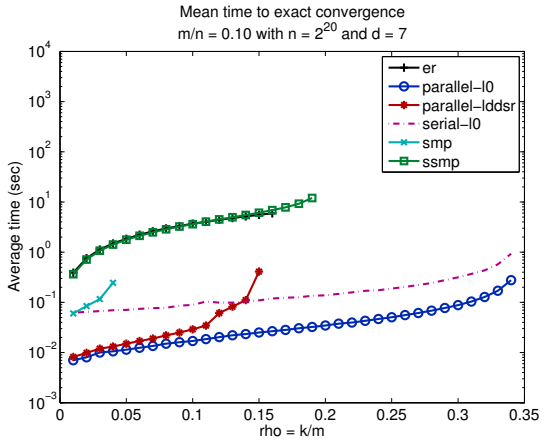

Fig. 6: Average recovery time (sec) with dependence on ρ for $\delta = 0.1$ and $\mathbb{E}_{k,\varepsilon,7}$ with $n = 2^{20}$.

Serial-ℓ_0 and $\mathcal{O}(nd)$ for Parallel-ℓ_0 and recording the number of iterations. Figure 7 shows that the number of iterations to convergence for Serial-ℓ_0, Parallel-ℓ_0, and parallel-LDDSR. The tests were performed by fixing $n = 2^{20}$, $\delta = 0.1$, and $d = 7$, and considering signals with sparsity ranging from $\rho = 0.05$ to $\rho = 0.1$. It can be seen in Figure 7 that the number of iterations to convergence is bounded by the curve $f(k) = \log k$, thus verifying our claims. We also make clear that by Definition III.1, Serial-ℓ_0 is shown to converge in $\mathcal{O}(n \log k)$ iterations, but for the sake of this experiment, we normalise the final number of iterations for Serial-ℓ_0 by a factor of n. Note the lower number of iteration by Serial-ℓ_0 due to its serial implementation with residual updates revealing more entries that satisfy the reduction of the residual by α. Now, to give a point of comparison, we also compute the number of iterations for ER and SSMP, which take $\mathcal{O}(k)$ iterations to converge. The results are shown in Figure 8, where the same parameters as in Figure 7 have been used. In particular, we can see that for a problem with $k/m = 0.8$, Parallel-ℓ_0 takes 5 iterations, while ER and SSMP take about 8000 iterations to solve the same problem.

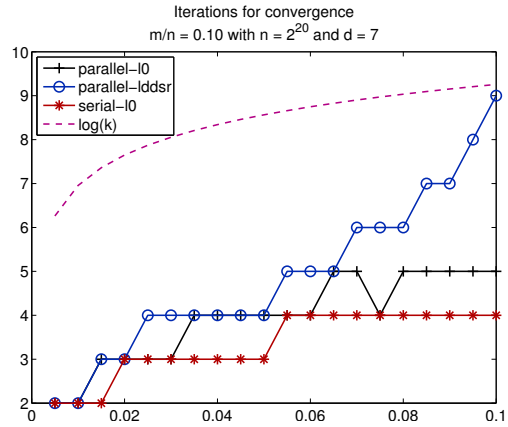

Fig. 7: Number of iterations to convergence for Parallel-ℓ_0, Serial-ℓ_0, and parallel-LDDSR at $\delta = 0.1$ with $\mathbb{E}_{k,\varepsilon,7}$ and $n = 2^{20}$. The number of iterations of Serial-ℓ_0 has been normalised by n to showcase its $\mathcal{O}(n \log k)$ guarantee in the number of iterations.

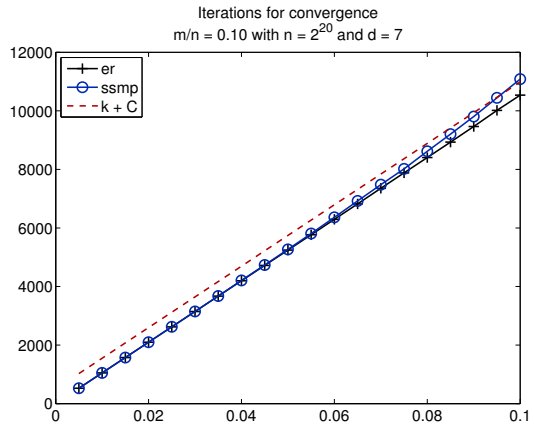

Fig. 8: Number of iterations to convergence for ER and SSMP at $\delta = 0.1$ with $\mathbb{E}_{k,\varepsilon,7}$ and $n = 2^{20}$.

E. Increasing phase transition as $\delta \to 0$ and $n \to \infty$

It is shown in Figure 2 that Serial-ℓ_0 and Parallel-ℓ_0 have a very high phase transition of just over 0.3 even for very small values of δ. We hypothesise that this high phase transition persists for any fixed $\delta \in (0, 1)$ provided n is sufficiently large. We provide numerical support of this claim in Figure 9, where for fixed $\delta = 10^{-3}$ and $d = 7$, we have plotted the average time to convergence for Parallel-ℓ_0 as ρ increases. The experiment was repeated for each $n \in \{2^{22}, 2^{24}, 2^{26}\}$, by initialising $\rho = 0.01$ and generating 30 problems at each ρ. If at least 50% of the problems converge we average out the time to convergence for successful cases, and perform the update $\rho \leftarrow \rho + 0.01$; otherwise, we stop. Our results in Figure 9 show that for $\delta = 10^{-3}$, the phase transition of the algorithm increases with n to just over 0.3.

Finally, in Table III we show the average timing depicted in Figure 9 for $\rho = 0.05$ which shows the approximate increase in the average computation time being proportional to n.

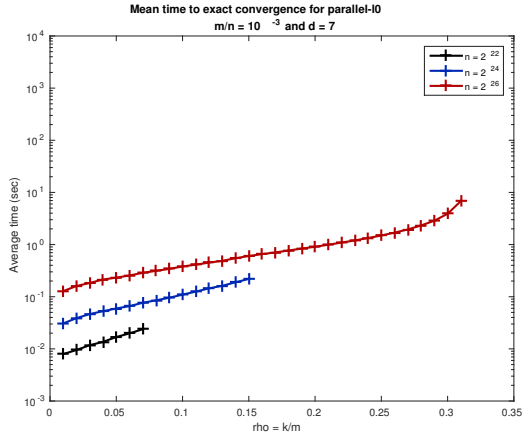

Fig. 9: Average recovery time (sec) for Parallel-ℓ_0, with dependence on ρ for $\delta = 0.001$ and $\mathbb{E}_{k,\varepsilon,7}$ with $n \in \{2^{22}, 2^{24}, 2^{26}\}$.

n	time t_n	ratio t_{4n}/t_n
2^{22}	0.0167	3.338
2^{24}	0.0557	4.163
2^{26}	0.2319	-

TABLE III: Average recovery time (sec) for Parallel-ℓ_0 at $\rho = 0.05$ and $\delta = 10^{-3}$ for $n \in \{2^{22}, 2^{24}, 2^{26}\}$.

F. Almost dissociated signals

The analysis of Parallel-ℓ_0 and Serial-ℓ_0 relied on the model of dissociated signals (1). We explore the effect on recovery ability of Parallel-ℓ_0 and Serial-ℓ_0 as the signal model is no longer dissociated, with a fixed fraction of the values in x being equal. To do this, we consider signals $x \in \chi_k^n$ with nonzero values composed of two bands: one in which *all* entries are equal to a fixed value drawn at random from a standard normal distribution $\mathcal{N}(0,1)$, and another one in which *each* entry is drawn independently of each other from $\mathcal{N}(0,1)$. Our results are shown in Figure 10, where we can see that as the fraction of values which are equal increases (shown in the figure by the parameter *band*), the phase transitions gracefully decrease from the flat shape observed for perfectly dissociated signals to an increasing log-shaped curve when *band* $= 0.9$. Note that the overall phase transition decreases, with the greatest decrease for $\delta \ll 1$.

G. d should be small, but not too small

Selection of the number of nonzeros per column, d, has not been adressed. In our numerical experiments we have consistently chosen $d = 7$ as the left-degree of our expander. Our choice of $d = 7$ for our problem size's order of magnitude is justified by Figure 11, where we have computed the phase transitions for Parallel-ℓ_0 for all odd values of d between 5 and 19. For $d = 5$, the phase transition of the algorithm is very low, thus signalling expanders of bad quality. For $d = 7$ the phase transition is substantially greater than when $d = 5$, and gradually decreases for values of d greater than seven. Note

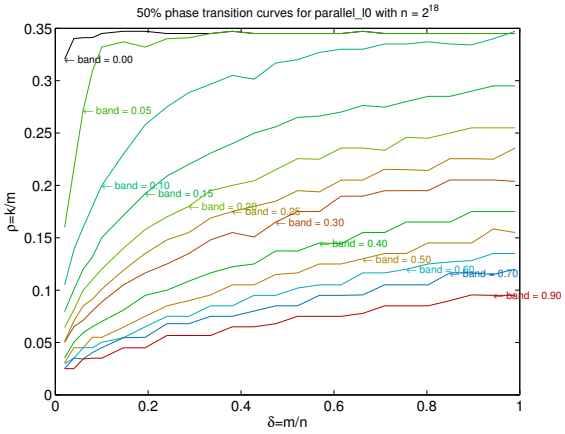

Fig. 10: 50% recovery probability logistic regression curves for Parallel-ℓ_0 with $\mathbb{E}_{k,\varepsilon,7}$ and $n = 2^{18}$, with signals having a fixed proportion, *band*, of identical nonzero elements in its support.

that the expander condition implies $(1-\varepsilon)dk < m$ which encourages small values of d in order that m/k can be as large as possible.

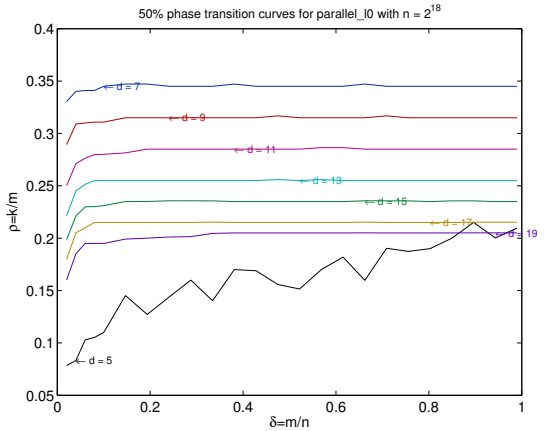

Fig. 11: 50% recovery probability logistic regression curves for Parallel-ℓ_0 with $\mathbb{E}_{k,\varepsilon,d}$ and $n = 2^{18}$ for $d \in \{5, 7, 9, 11, 13, 15, 17, 19\}$.

VI. Conclusions and Future Work

We have proposed two algorithms for combinatorial compressed sensing with provable convergence guarantees in $\mathcal{O}(dn \log k)$ operations and very high phase transitions when the signal x is dissociated. In particular, Parallel-ℓ_0 is observed to be empirically the fastest algorithm in compressed sensing when the signal is dissociated. We have used the dissociated signal model in the convergence proofs, but that in practice one can relax this assumption and still get reasonably high phase transitions.

As future work it remains to address the case of noisy observations, and to extend the scope of the algorithms to more general signal models. The proofs presented in this paper should extend trivially to noise which is bounded to be

less than half the minimal distance between obtainable values $\sum_{i\in T} x_i$ by introducing an equivalence class. A variant which is robust to Gaussian noise is scope for future work.

APPENDIX

For completeness, we give a proof of Lemma II.4

Proof: For any unbalanced, left d-regular, bipartite graph it holds that:

$$|\mathcal{N}_1(S)| + |\mathcal{N}_{>1}(S)| = |\mathcal{N}(S)|, \tag{45}$$

$$|\mathcal{N}_1(S)| + 2|\mathcal{N}_{>1}(S)| \leq d|S|. \tag{46}$$

Where (45) follows from the definition of $\mathcal{N}_{>1}(S)$, and (46) by double-counting the edges emanating from S to $\mathcal{N}(S)$. Now, to prove that (7) is necessary, assume that G is a (k, ε, d)-expander graph. Then, for $S \in [n]^{(\leq k)}$ we have that

$$|\mathcal{N}(S)| > (1-\varepsilon)d|S|. \tag{47}$$

Combining (45), (46) and (47) we get the chain of inequalities

$$d|S| - |\mathcal{N}_{>1}(S)| \geq |\mathcal{N}(S)| > (1-\varepsilon)d|S|, \tag{48}$$

which yield

$$|\mathcal{N}_{>1}(S)| < \varepsilon d|S|. \tag{49}$$

Plugging (49) into (45) and using (47) we obtain

$$|\mathcal{N}_1(S)| > (1-2\varepsilon)d|S|. \tag{50}$$

To prove the sufficiency of (7) for graph expansion, we couple it with (46) into the system

$$(1-2\varepsilon)d|S| < |\mathcal{N}_1(S)| \leq d|S| - 2|\mathcal{N}_{>1}(S)|, \tag{51}$$

and use the left and right hand sides recover (49). Now, using (7) and (45) we obtain

$$|\mathcal{N}(S)| - |\mathcal{N}_{>1}(S)| > (1-2\varepsilon)d|S|. \tag{52}$$

And using (49) in (52) allows us to recover (47), implying that G is a (k, ε, d)-expander graph. ∎

REFERENCES

[1] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank, "Efficient and robust compressed sensing using optimized expander graphs," *Information Theory, IEEE Transactions on*, vol. 55, no. 9, pp. 4299–4308, 2009.

[2] R. Berinde, P. Indyk, and M. Ruzic, "Practical near-optimal sparse recovery in the l1 norm," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 198–205.

[3] E. J. Candès and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.

[4] D. L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[5] E. J. Candès and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Foundations of Computational Mathematics*, vol. 6, no. 2, pp. 227–254, 2006.

[6] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

[7] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[8] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, 2006.

[9] H. Nyquist, "Certain topics in telegraph transmission theory," *American Institute of Electrical Engineers, Transactions of the*, vol. 47, no. 2, pp. 617–644, 1928.

[10] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[11] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Springer, 2013.

[12] B. BAH and J. TANNER, "Vanishingly sparse matrices and expander graphs, with application to compressed sensing," *IEEE transactions on information theory*, vol. 59, no. 11, pp. 7491–7508, 2013.

[13] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 798–805.

[14] R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, 2008.

[15] R. Berinde and P. Indyk, "Sequential sparse matching pursuit," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 36–43.

[16] W. Xu and B. Hassibi, "Efficient compressive sensing with deterministic guarantees using expander graphs," in *Information Theory Workshop, 2007. ITW'07. IEEE*. IEEE, 2007, pp. 414–419.

[17] T. Tao and V. H. Vu, *Additive combinatorics*. Cambridge University Press, 2006, vol. 105.

[18] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *Information Theory, IEEE Transactions on*, vol. 56, no. 4, pp. 1982–2001, 2010.

[19] L. A. Bassalygo and M. S. Pinsker, "Complexity of an optimum nonblocking switching network without reconnections," *Problemy Peredachi Informatsii*, vol. 9, no. 1, pp. 84–87, 1973.

[20] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, "Randomness conductors and constant-degree lossless expanders," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 2002, pp. 659–668.

[21] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *Journal of Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[22] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[23] R. S. Boyer and J. S. Moore, *MJRTYa fast majority vote algorithm*. Springer, 1991.

[24] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[25] M. Kelbert and Y. M. Suhov, *Information theory and coding by example*. Cambridge University Press, 2013.

[26] J. Blanchard and J. Tanner, "Gpu accelerated greedy algorithms for compressed sensing," *Preprint*, 2012.

[27] D. L. Donoho and J. Tanner, "Precise undersampling theorems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 913–924, 2010.

[28] D. L. Donoho, "High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension," *Discrete & Computational Geometry*, vol. 35, no. 4, pp. 617–652, 2006.

[29] ——, "Neighborly polytopes and sparse solution of underdetermined linear equations," 2005.

[30] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, vol. 367, no. 1906, pp. 4273–4293, 2009, with electronic supplementary materials available online. [Online]. Available: http://dx.doi.org/10.1098/rsta.2009.0152

[31] J. D. Blanchard and J. Tanner, "Performance comparisons of greedy algorithms in compressed sensing," 2013.