# Performance Comparisons of Greedy Algorithms in Compressed Sensing

Jeffrey D. Blanchard[1]*and Jared Tanner[2]

[1] *Department of Mathematics and Statistics, Grinnell College, Grinnell, IA 50112*
[2] *Mathematics Institute, University of Oxford, 24-29 St Giles', Oxford OX1 3LB, UK*

SUMMARY

Compressed sensing has motivated the development of numerous *sparse approximation* algorithms designed to return a solution to an underdetermined system of linear equations where the solution has the fewest number of nonzeros possible, referred to as the sparsest solution. In the compressed sensing setting, *greedy* sparse approximation algorithms have been observed to be both able to recovery the sparsest solution for similar problem sizes as other algorithms and to be computationally efficient; however, little theory is known for their average case behavior. We conduct a large scale empirical investigation into the behavior of three of the state of the art greedy algorithms: NIHT, HTP, and CSMPSP. The investigation considers a variety of random classes of linear systems. The regions of the problem size in which each algorithm is able to reliably recovery the sparsest solution is accurately determined, and throughout this region additional performance characteristics are presented. Contrasting the recovery regions and average computational time for each algorithm we present *algorithm selection maps* which indicate, for each problem size, which algorithm is able to reliably recovery the sparsest vector in the least amount of time. Though no one algorithm is observed to be uniformly superior, NIHT is observed to have an advantageous balance of large recovery region, absolute recovery time, and robustness of these properties to additive noise and for a variety of problem classes. The algorithm selection maps presented here are the first of their kind for compressed sensing. Copyright © J.D. Blanchard and J. Tanner.

Submitted 31 March 2013

KEY WORDS: Greedy algorithm, compressed sensing, GPU computing, hard thresholding

## 1. INTRODUCTION

Compressed sensing [1, 2] is a technique where the prior knowledge that data is compressible allows for the data to be acquired with fewer measurements than would otherwise be necessary. The resulting reduced number of measurements is proportional to the desired end compression rate and can therefore result in dramatic savings when fully measuring the data set has some significant underlying cost. In its simplest form, compressed sensing can be expressed in terms of linear algebra. Let $x \in \mathbb{R}^n$ be a vector with $k$ nonzero entries (referred to as $k$ sparse) and let $A$ be an $m \times n$ matrix whose $m$ rows represent inner products used to acquire the measurements $y = Ax \in \mathbb{R}^m$. If one is willing to take a full set of $m = n$ measurements, the vector can be acquired by measuring the

*Correspondence to: J.D. Blanchard, Department of Mathematics and Statistics, Grinnell College, Grinnell, IA 50112. Email: jeff@math.grinnell.edu

Contract/grant sponsor: Leverhulm Trust and Nvidia Professor Partnership Program

Contract/grant sponsor: National Science Foundation; contract/grant number: DMS 1112612, OISE 0854991

entries in $x$ directly with the $n \times n$ identity matrix ($A = I$) so that $y = Ax = Ix = x$. In contrast, if one knew before hand the location of the $k$ nonzero entries one could simply measure those $k$ values similarly. While one often knows *a priori* that a vector has few nonzeros (or few large entries), one rarely knows the location of the nonzeros requiring more than $k$ measurements. Thus, the number of measurements, $m$, satisfies $k < m < n$.

When the location of the nonzeros is unknown one can formulate the recovery of the sparse vector $x$ from knowledge of $A$ and the measurements $y = Ax$ as the combinatorial optimization problem

$$\min_{z \in \mathbb{R}^n} \|z\|_0 \text{ subject to } y = Az \tag{1}$$

where $\|z\|_0$ denotes the number of nonzeros in the vector $z$. A naive method for solving (1) is to begin by checking if there is a single column of $A$ that can be used to represent $y$, failing this to check if there is any combination of two columns of $A$ that can be used to represent $y$, and increasing the number of columns under consideration until a solution is determined. As $A$ is underdetermined it is always possible to find a solution with $m$ or fewer nonzeros that will satisfy (1). Unfortunately this exhaustive search is computationally implausible for all but the smallest values of $n$.

Much of the development of compressed sensing has focused on the design and analysis of computationally efficient algorithms which can solve (1). An intensively studied technique is to replace (1) by the convex relaxation of the objective

$$\min_{z \in \mathbb{R}^n} \|z\|_1 \text{ subject to } y = Az \tag{2}$$

in which case (2) can be recast as a linear program and solved using well developed, off-the-shelf software [3, 4, 5] or algorithms more recently designed specifically for compressed sensing [6, 7, 8, 9]. The equivalence of when the solution to (2) will be identical to the solution to (1) has been fully characterized by Donoho in [10, 11] with precise sampling theorems derived. In particular for $A$ with entries drawn Gaussian i.i.d., precise values of $\rho(m/n)$ are derived so that for $k < m \cdot \rho(m/n)$ the solution to (2) will coincide with the solution to (1); moreover, when $k > m \cdot \rho(m/n)$ the solution to (2) is typically observed to be different from the solution to (1). The abrupt change in the probability of recovery for an algorithm is referred to as a *phase transition* and the curve $\rho(m/n)$ denoting the transition is referred to as a *phase transition curve*. Other notable examples of precise sampling theorems have been derived for similar formulations such as: robust variants of (1) and (2) by Xu and Hassibi [12, 13], inclusion of nonnegativity by Donoho and Tanner [14, 15, 16], uniqueness of solutions with bound constraints by Donoho and Tanner [17], recovery of low rank matrices by Recht, Xu and Hassibi [18], block sparse signals by Stojnic [19], and equivalence to the state evolution of message passing algorithms by Donoho, Maleki, and Montanari [20, 21]. Large-scale empirical testing has shown that many of these results hold for many more matrix ensembles than the Gaussian ensemble implied by the theory [22, 23].

A second line of compressed sensing algorithm development has considered algorithms which attempt to directly solve (1) or a noise resistant extension such as

$$\min_{z} \|y - Az\| \text{ subject to } \|z\|_0 \leq k. \tag{3}$$

Examples of such algorithms which have been observed to perform well in testing include Compressive Sampling Matching Pursuit (CoSaMP) [24], Subpace Pursuit (SP) [25], Iterative Hard Thresholding (IHT) [26], Normalized Iterative Hard Thresholding (NIHT) [27], and Hard Thresholding Pursuit (HTP) [28] though this list is far from exhaustive. Each of these algorithms has been analyzed using techniques such as the restricted isometry property [29] and have been proven to recover the solution to (1) for a wide class of sensing matrices $A$ at the optimal order of $m$ proportional to $k$. However, these sufficient conditions are extremely pessimistic with associated phase transitions well below observed performance [30, 31]. These algorithms employ projections via hard thresholding which sets selected coefficients to zero while leaving the remaining coefficients unchanged. We refer to this class of algorithms broadly as *hard thresholding algorithms.* Unfortunately there are no precise average case characterizations for when (1) is solved by hard thresholding algorithms, with the technical difficulty being that the hard thresholding projections are not Lipshitz continuous [32].

### *1.1. Relationship to Prior Work*

This paper provides an empirical average case analysis of the hard thresholding algorithms via large-scale testing of problems with realistic, application sized problems. The literature already has several examples of empirical testing of algorithms for compressed sensing, notably [33, 34, 23]. In [33] Donoho and Maleki studied optimal tuning parameters for various algorithms, including hard thresholding algorithms. In [34], Sturm studied 15 algorithms with 7 vector distributions focusing entirely on locations of the phase transition for Gaussian matrices. In [23], Monajemi et. al studied the behavior of linear programing (2) recovery probability for numerous deterministic matrices. The information provided in the previous work, such as the tuning parameters for IHT and CSMPSP from [33] have been incorporated into our work. This work is distinct from these predecessors in three main ways.

First, each of those empirical analyses focus on small problem dimensions of $n = 800$, $n = 400$, and $n \approx 1000$, in [33], [34], and [23] respectively. This present empirical analysis is based on problems with ambient dimension frequently set at $n = 2^{18}$ and $n = 2^{20}$. Empirical studies with more realistic, application sized problems offer better insight into the large-scale behavior and should bolster practitioners' confidence in the results. The ability to test large problem sizes is particularly important in evaluating the behavior of algorithms in the extreme undersampling regime of $m \ll n$. When $n$ is small as in previous empirical studies, significant undersampling quickly forces the sparsity level toward zero resulting in poor resolution of $k/m$. As compressed sensing is primarily concerned with extreme undersampling, this is the most critical region for empirical testing.

Second, none of the previous works consider the very competitive greedy algorithms NIHT or HTP. While [23] is focused exclusively on $\ell_1$ minimization, the other two projects examined greedy algorithms, including IHT, CoSaMP, and Subspace Pursuit. Whereas, [33] identified a tuned, fixed step size for IHT, Blumensath and Davies [27] updated IHT to NIHT by incorporating the optimal step size in the current supporting subspace, leading to considerable improvement in both recovery capability and time. Foucart [28] altered IHT adopting the projection ideas from CoSaMP and Subspace Pursuit and combining them with the iterative update step from IHT. These two greedy algorithms are highly competitive with other greedy algorithms and their practical behavior is extensively studied in this work.

Finally, the most significant delineating contribution of this paper is the focus on other behavior characteristics of the algorithms. While [33, 34, 23] are almost entirely focused on the location of the empirical, average case, recovery phase transition, this paper is focused on providing information about how an algorithm performs in relationship to the others, specifically in regions where more than one algorithm is successful. The location of the phase transition is a critical element in this analysis as it identifies the region in which an algorithm reliably recovers the measured vector. However, which algorithm should be selected when more than one algorithm is able to reliably recover the measured vector? This work not only provides empirical phase transitions for large, realistic sized problems, it also provides an algorithm selection map for each problem class, indicating which algorithm is able to reliably recover the measured vector in the least time. Information about the mean and variance for certain performance characteristics is also presented, including time for recovery, ratio of time to fastest algorithm, iterations, convergence rate, and fraction of support set recovered.

The increased scope of testing here is made possible by our development of a software package designed for such large scale, rapid testing using graphics processing units (GPUs). The software, *GPU Accelerated Greedy Algorithms for Compressed Sensing* (GAGA) [35, 36], includes a full testing suite which generates a problem on the GPU, solves the problem with one of five hard thresholding algorithms (also referred to as greedy algorithms), and returns performance characteristics to a text file. The software is also capable of solving a problem directly rather than generating a random problem and is therefore useful for applications; a detailed description of the software including timings and acceleration ratios is presented in [35]. The software is available for download at [36].

### *1.2. Preliminaries*

Our empirical investigation of the hard thresholding algorithms for compressed sensing considers the ability to recover the underlying signal, recovery time, and other performance characteristics of NIHT, HTP, and CSMPSP (a hybrid algorithm akin to Subspace Pursuit and CoSaMP). $A_S^\dagger$ is the pseudo inverse of $A_S$ where $A_S$ is the $m \times |S|$ submatrix consisting of the columns of $A$ indexed by $S$. In our implementation, the projection $A_S^\dagger y$ is performed via the conjugate gradient method restricted to the subspace defined by $S$. Thus, if $z = A_S^\dagger y$, then $z = z_S$ is a vector supported on the index set $S$. The pseudo code presented for the algorithms include the subroutines `DetectSupport`$(z)$, returning the $k$ largest magnitude entries in $z$, and `Threshold`$(z, S)$, the hard thresholding operator that leaves the values in $z_S$ unchanged and sets all other values of $z$ to zero. Details of the implementations of the algorithms and subroutines in GAGA are described in [35]. For conciseness the algorithm pseudo code omits the initialization and stopping criteria, which are stated in Alg. 6.

**Algorithm 1** NIHT (Normalized Iterative Hard Thresholding [27])

**Iteration:** During iteration $l$, **do**

1: $\omega_l = \frac{\|A^*_{T_{l-1}} r_{l-1}\|_2^2}{\|A_{T_{l-1}} A^*_{T_{l-1}} r_{l-1}\|_2^2}$ (optimal step size in the $k$-subspace $T_{l-1}$)
2: $x_l = x_{l-1} + \omega_l A^* r_{l-1}$ (steepest descent step)
3: $T_l = \texttt{DetectSupport}(x_l)$ (proxy to the support set)
4: $x_l = \texttt{Threshold}(x_l, T_l)$ (restriction to proxy support set $T_l$)
5: $r_l = y - Ax_l$ (update the residual)

**Algorithm 2** HTP (Hard Thresholding Pursuit [28])

**Iteration:** During iteration $l$, **do**

1: $\omega_l = \frac{\|A^*_{T_{l-1}} r_{l-1}\|_2^2}{\|A_{T_{l-1}} A^*_{T_{l-1}} r_{l-1}\|_2^2}$ (optimal step size in the $k$-subspace $T_{l-1}$)
2: $x_l = x_{l-1} + \omega_l A^* r_{l-1}$ (steepest descent step)
3: $T_l = \texttt{DetectSupport}(x_l)$ (proxy to the support set)
4: $x_l = A^\dagger_{T_l} y$ (projection onto the $k$-subspace $T_l$)
5: $r_l = y - Ax_l$ (update the residual)

**Algorithm 3** CSMPSP (CoSaMP [24], Subspace Pursuit [25])

**Iteration:** During iteration $l$, **do**

1: $S_l = \texttt{DetectSupport}(A^* r_{l-1})$ ($k$ columns most correlated with residual)
2: $\Lambda_l = T_{l-1} \cup S_l$ (form a larger proxy for the support set)
3: $x_l = A^\dagger_{\Lambda_l} y$ (projection onto the $2k$-subspace $\Lambda_l$)
4: $T_l = \texttt{DetectSupport}(x_l)$ (proxy to the support set)
5: $x_l = \texttt{Threshold}(x_l, T_l)$ (restriction to proxy support set $T_l$)
6: $r_l = y - Ax_l$ (update the residual)

**Problem Class** For each algorithm we conduct tests for random matrices, drawn from different ensembles, used to measure sparse vectors whose nonzeros are similarly drawn from a variety of distributions. We refer to the combination of a matrix ensemble and sparse vector ensemble as a *problem class* and denote it by $(Mat, vec)$. Each algorithm is evaluated for a specific problem class by repeatedly testing different problem instances at a variety of problem sizes.

**Problem Instance** Let $A$ and $x$ be drawn from a specific problem class $(Mat, vec)$ at size $(k, m, n)$ where the matrix $A$ is of size $m \times n$ and the vector $x$ has $k$ nonzeros. We refer to a problem instance without noise as recovery of $x$ from the pair $(A, y)$ where $y = Ax$, and a problem instance with noise as recovery of a $k$ sparse vector from the pair $(A, y)$ where $y = Ax + e$ with $e$ is a normally distributed additive noise vector. When a problem class is tested for problem instances that have additive noise we denote the problem class by $(Mat, vec_\epsilon)$ with $e$ drawn from $\mathcal{N}(0, \epsilon\|Ax\|_2)$. The following paragraphs detail the matrix and sparse vector ensembles investigated in this work.

**Matrix ensemble** The measurement matrices are drawn from one of three matrix ensembles denoted by $\mathcal{N}$ (normal/Gaussian), $\mathcal{S}_p$ (sparse), and DCT (discrete cosine transform):

- $\mathcal{N}$: Gaussian matrix normalized to have expected unit Euclidean length columns, i.e. entries drawn i.i.d. from $\mathcal{N}(0, m^{-1})$.
- $\mathcal{S}_p$: Sparse matrix with $p$ nonzeros per column with support set drawn uniformly at random and nonzero values drawn uniformly at random from plus or minus $p^{-1/2}$.
- $DCT$: Random subsampled rows of the discrete cosine transform matrix.

The sparse matrix ensemble $\mathcal{S}_p$ is tested primarily for $p = 4$ and $p = 7$, with $\mathcal{S}_7$ having nearly as high a recovery phase transition as for larger values of $p$ while retaining a lower computational cost. The results for $\mathcal{S}_4$ are shown to illustrate that such sparse matrices have substantially lower recovery phase transitions; whereas App. F shows limited increase in the phase transition region as $p$ is increased from 7 to 13.

**Sparse vector ensembles** Each vector has a support set chosen uniformly at random and the nonzero entries drawn i.i.d. from one of the following distributions denoted by $B$ (binary), $U$ (uniform), and $N$ (normal):

- $B$: Nonzeros drawn from plus and minus one, $\{-1, 1\}$, with equal probability;
- $U$: Nonzeros drawn uniformly from the unit interval, $\mathcal{U}(0, 1)$;
- $N$: Nonzeros drawn from the standard normal distribution, $\mathcal{N}(0, 1)$.

The main findings of our empirical tests are summarized in the following section, with the remainder of the manuscript presenting a distillation of the data generated. These findings by no means exhaust the information contained in the data, nor do they answer all questions a practitioner might have. More quantitative variants of these claims are found in the sections that present the supporting evidence. The software used to conduct these tests and process the data are available at [36], written with the aim that interested parties can easily generate different data and easily process it in order to address other questions.

### *1.3. Main Findings*

The following six claims summarize the more detailed observations in Sections 3 and 4. While Claims 1 and 4 are minor extensions and reinforcement for similar claims from previous work, the remaining claims offer new insight in the behavior of the algorithms across a wide range of problem instances.

*Claim 1* (Recovery regions: sparse binary vectors without noise)
For problem classes $(Mat, B)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_4, \mathcal{S}_7, DCT\}$ and problem instances drawn without noise, the recovery phase transitions for NIHT, HTP, and CSMPSP increasingly approach one another for $m \ll n$, with the exception of CSMPSP for $(DCT, B)$ where a substantially lower phase transition is observed. For ratios of $m/n$ approaching one CSMPSP has a phase transition that is substantially higher for problem classes $(\mathcal{N}, B)$ and $(\mathcal{S}_7, B)$ and a modestly higher phase transition for $(DCT, B)$; NIHT and HTP are observed to have recovery regions similar to each other.

*Claim 2* (Algorithm selection: sparse binary vectors without noise)
NIHT is able to recover problems from the class $(DCT, B)$ near the observed recovery phase

transition in less time than can HTP or CSMPSP. NIHT is typically able to recover problems from the class $(\mathcal{N}, B)$ for $m \ll n$ in less time than HTP and CSMPSP; however, for $m/n$ approaching one CSMPSP is preferable due to a substantially higher phase transition. CSMPSP is able to recover problems from the class $(\mathcal{S}_p, B)$ for $p = 4$ and $p = 7$ in less time than can NIHT and HTP.

*Claim 3* (Algorithm variability: binary vectors without noise)
For each of NIHT, HTP, and CSMPSP, at any $(k, m, n)$ with $m$ and $n$ fixed, once $k/m$ decreases below the recovery phase transition curve, the following algorithm properties are highly concentrated about their mean: the fraction of true positives of the support set recovered, number of iterations to recovery, and asymptotic convergence rate.

*Claim 4* (Recovery regions: alternate vector ensembles)
For NIHT, HTP, and CSMPSP with $Mat \in \{\mathcal{N}, \mathcal{S}_4, \mathcal{S}_7, DCT\}$, the recovery phase transition for $(Mat, B)$ is below those of problem classes $(Mat, U)$ and $(Mat, N)$. This behavior extends to problem instances with noise, where the phase transitions for problem classes $(Mat, U_\epsilon)$ and $(Mat, N_\epsilon)$ are substantially higher than the phase transitions for the problem class $(Mat, B_\epsilon)$ for $m < n/2$.

*Claim 5* (Algorithm recovery: effects of moderate noise ($\epsilon = 1/10$))
For $m \ll n$, the recovery phase transitions for algorithms NIHT, HTP, and CSMPSP decrease insubstantially for moderate levels of noise, $\epsilon$, across all problem classes $(Mat, vec)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_4, \mathcal{S}_7, DCT\}$ and $vec \in \{B, U, N\}$. The phase transition curves for problem classes $(Mat, B_\epsilon)$ and $(Mat, B)$ are similar throughout the phase space. For $vec \in \{U, N\}$, the gap between the phase transition curves for $(Mat, vec_\epsilon)$ and $(Mat, vec)$ grows significantly as $m/n$ approaches one.

*Claim 6* (Algorithm selection: all vector ensembles with moderate noise ($\epsilon = 1/10$))
Across all problem classes contaminated by moderate noise ($\epsilon = 1/10$), $(Mat, vec_\epsilon)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_4, \mathcal{S}_7, DCT\}$ and $vec \in \{B, U, N\}$, NIHT recovers vectors in less time than HTP or CSMPSP throughout the overwhelming majority of the phase space. Moreover, when NIHT is not the fastest algorithm and $m < n/2$, NIHT never requires more than twice the time of HTP or CSMPSP.

Claims 1–6 are informed by the totality of the data generated by the testing including the formal observations, figures, and tables presented in Secs. 3–4, and Apps. A–F.

### *1.4. Outline*

The manuscript is organized as follows. Sec. 2 describes the experimental setup including: algorithm initialization, termination conditions, data fitting techniques, as well as the size and number of problem classes tested. Sec. 3 presents the data for the problem class $(Mat, B)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ for $p = 4$ and $p = 7$, with recovery phase transitions presented, including: regions of high probability of recovery, average time for recovery, algorithm selection maps, and the variance of these and other performance properties of NIHT, HTP, and CSMPSP. Sec. 4 presents a similar analysis as in Sec. 3, extended to show how the algorithm properties change with the introduction of moderate values of noise ($\epsilon = 1/10$), Sec. 4.1, and for vector ensembles $U$ and $N$, Sec. 4.2. Sec. 5 outlines future extensions.

Only a small portion of the algorithm performance data calculated as described in Sec. 2 is presented in the main body of the manuscript. Extensive appendices present additional aspects of the data, further bolstering the claims and observations from the main body of the manuscript. App. A presents recovery phase transition curves for $(Mat, B)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ for $p = 4$ and $p = 7$ at smaller problem sizes than presented in Sec. 3; this data shows the convergence of the recovery phase transition curves as $n$ increases. App. B presents algorithm selection maps for $(Mat, B)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_7, DCT\}$ for smaller values of $n$ than those presented in Sec. 3, as well as algorithm selection maps and relative timing information for $(\mathcal{S}_4, B)$. These results show that the algorithm selection maps presented in Sec. 3 are consistent across problem sizes and for $(\mathcal{S}_p, B)$ for smaller values of $p$. App. C presents the recovery phase transitions for $(Mat, vec)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ with $p = 4$ and $p = 7$ and $vec \in \{B, U, N, B_\epsilon, U_\epsilon, N_\epsilon\}$ with $\epsilon = 1/10$ where each plot considers a single problem class and each of the algorithms NIHT, HTP, and CSMPSP.

App. D presents the ratio of the minimum average time over the average time for each of the algorithms NIHT, HTP, and CSMPSP for problem classes $(Mat, vec)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_7, DCT\}$ and $vec \in \{U_\epsilon, N_\epsilon\}$ for $\epsilon = 1/10$. The plots in App. D can be interpreted as algorithm selection maps when the measurements have moderate levels of additive noise. App. E presents recovery phase transitions with each plot considering a single algorithm and matrix ensemble but three values of $vec \in \{B_\epsilon, U_\epsilon, N_\epsilon\}$; plots are shown for each combination of algorithms NIHT, HTP, and CSMPSP, matrix ensembles $Mat \in \{\mathcal{N}, \mathcal{S}_4, DCT\}$ and for both $\epsilon = 0$ and $\epsilon = 1/10$. App. F presents recovery phase transitions for $(\mathcal{S}_p, vec)$ for $p = 4$, 7, and 13, for each of $vec \in \{B, U, N, B_\epsilon, U_\epsilon, N_\epsilon\}$ with $\epsilon = 1/10$. These plots show that the gap between the recovery phase transition curves for $p = 13$ and $p = 7$ is much smaller than the gap between $p = 7$ and $p = 4$; this and other data, informed the focus in the main body of the manuscript on $p = 7$ for $Mat = \mathcal{S}_p$.

## 2. EXPERIMENTAL SETUP

This manuscript describes the average case behavior of five hard thresholding algorithms with an emphasis on the following three algorithms: NIHT, HTP, and CSMPSP whose pseudo code are stated in Alg. 1-3. For completeness, where appropriate, we include in our comparison the behavior of two even simpler, though often less efficient, algorithms: IHT [26], Alg. 4, which is the precursor to NIHT differing by the use of a fixed step size, and Hard Thresholding [37], Alg.5, which corresponds to one step of HTP initialized with a starting vector of all zeros.

**Algorithm 4** IHT (Iterative Hard Thresholding [26])

**Iteration:** During iteration $l$, **do**

1: $x_l = x_{l-1} + \omega A^* r_{l-1}$ (steepest descent step with fixed step size)
2: $T_l = \texttt{DetectSupport}(x_l)$ (proxy to the support set)
3: $x_l = \texttt{Threshold}(x_l, T_l)$ (restriction to proxy support set $T_l$)
4: $r_l = y - Ax_l$ (update the residual)

**Algorithm 5** Hard Thresholding

**Approximation:**

1: $\hat{x} = A^\dagger_{T_0} y$ (projection onto the $T_0$ subspace)

**Algorithm 6** Initialization Procedure

**Input:** $A$, $y$, $k$
**Output:** A $k$-sparse approximation $\hat{x}$ of the target signal $x$

**Initialization and Initial Support Detection**

1: $x_0 = A^* y$ (initial approximation)
2: $T_0 = \texttt{DetectSupport}(x_0)$ (proxy to the support set)
3: $x_0 = \texttt{Threshold}(x_0, T_0)$ (restriction to proxy support set $T_0$)
4: $r_0 = y - Ax_0$ (initialize residual)

Each of the five hard thresholding algorithms discussed take as their arguments the measurements, $y$, the matrix by which the measurements were acquired, $A$, and the sparsity level $k$. With the exception of Alg. 5, each algorithm is initialized using Alg. 6 and continues iterating until one of the following stopping criteria are met:

- the residual is small: $\|r_l\|_2 < .001 \cdot \frac{m}{n}$;
- a maximum number of iterations has been met: 5000 for Alg. 1 and Alg. 4 and a maximum number of 300 iterations for Alg. 2 and Alg. 3;

- the residual has failed to change significantly in 16 iterations:

$$\max_{j=1,\dots,15} \left| \|r_{l-j+1}\|_2 - \|r_{l-j}\|_2 \right| \leq 10^{-6};$$

- after many iterations, the convergence rate is close to one: let $c = 700$ for Alg. 1 and Alg. 4 and $c = 125$ for Alg. 2 and Alg. 3; the algorithm terminates

$$\text{if } l > c \text{ and } \left( \frac{\|r_{l-15}\|_2^2}{\|r_l\|_2^2} \right)^{\frac{1}{15}} > 0.999.$$

When any one of the stopping criteria are met at iteration $l$, the algorithm terminates and returns the $k$ sparse approximation $x_l$ which we denote by $\hat{x}$.

There are two distinct analysis frameworks used to interpret the data generated from the empirical testing. In Sec. 3, the analysis focuses on extensive testing of the problem classes $(Mat, B)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$. As stated in Claim 4 and in previous studies [33, 34], the random vector ensemble $vec = B$ has the lowest recovery phase transition for hard thresholding algorithms. In this noise free setting, we perform extensive testing for many values of $n$ in order to provide insight into dependence on the problem size $n$. This testing is detailed in Sec. 2.1. Sec. 4 provides an analysis of testing conducted on all problem classes $(Mat, vec)$ and $(Mat, vec_\epsilon)$; here a single value of $n$ is tested per problem class and there is a relaxation in the definition of successful recovery as described in Sec. 2.2. Details of all tests and the distinctions between the testing conducted in Secs. 3 and 4 are outlined in the following subsections.

### *2.1. Testing and analysis of problem class $(Mat, B)$ without noise*

The data supporting the three main findings for the algorithm behavior on problem class $(Mat, B)$ without noise, Claims 1–3, are generated from problem instances when the problem sizes $(k, m, n)$ tested are selected to best explore each of the properties under consideration. Determining the recovery phase transition behavior underlying Claim 1 is best resolved by focusing the tests near the phase transition. Determining the average time for each algorithm in the region where they are able to recover the measured sparse vector, which informs Claim 2, requires testing throughout their recovery region. The aforementioned tests can provide a wealth of other information about the tested algorithms throughout their recovery regions, such as the mean and standard deviation of: the fraction of true positives of the recovered vector, the number of iterations, and the asymptotic convergence rates. To more concisely convey the data establishing claim 3 we conduct greater numbers of tests for a few selective values of $(k, m, n)$ in the union of the algorithms' recovery region.

**Determination of average case phase transitions for various problem classes** The following tests are conducted to identify the average case recovery phase transition for various $(Mat, B)$ problem classes. The measured vector $x$ from a randomly drawn problem instance from the problem class is considered to be *successfully recovered* if the algorithm returns the approximation $\hat{x}$ which differs from $x$ by no more than $10^{-3}$ in any component:

$$\|x - \hat{x}\|_\infty < .001. \tag{4}$$

Tests are conducted for the values of $n$ that increase by a factor of 4 from: $2^{10}$ to $2^{20}$ for $Mat = DCT$, from $2^{10}$ to $2^{18}$ for $Mat = \mathcal{S}_p$ with $p = 4$ and 7, and from $2^{10}$ to $2^{14}$ for $Mat = \mathcal{N}$. For each value of $n$, tests were conducted for 30 values of $m$ selected as $m = \lceil \delta \cdot n \rceil$ for each

$$\delta \in \{.001, .002, .004, .006, .008, .01, .02, .04, .06, .08, 0.1, \dots, 0.99\} \tag{5}$$

with 18 additional, linearly spaced values of $\delta$ from 0.1 to 0.99. For each $m, n$ pair, problem instances are tested for values of $k$ selected as follows. First the bisection method is used, starting with $k = 1$ and $k = m - 1$, to determine values $k_{min}$ and $k_{max}$ such that the algorithm is observed to

have successfully recovered at least nine out of ten problem instances for those values of $k < k_{min}$ tested during the bisection method, and to have successfully recovered at most one out of ten problem instances for those values of $k > k_{max}$ tested during the bisection method. The interval $[k_{min}, k_{max}]$ is an estimate for the recovery phase transition region, which is then sampled by testing ten problem instances for each of 50 independent values of $k$ uniformly spaced in $[k_{min}, k_{max}]$, or if the interval width is less than 50, then ten problem instances are tested for each value of $k$ in the interval $[k_{min}, k_{max}]$.

The above experimental setup includes testing more than 15,000 problem instances for each algorithm, each problem class, and each value of $n$. To present this data as concisely as possible, for each $m, n$ pair, we compute a logistic regression of the data

$$\hat{\pi}(k/m) = \frac{1}{1 + \exp(-b_0(1 - b_1 k/m))} \tag{6}$$

where the fit parameters $b_0$ and $b_1$ are calculated to minimize

$$\sum_k \left| \hat{\pi}(k/m) - \frac{\#success(k, m, n)}{\#trials(k, m, n)} \right| \tag{7}$$

Fig. 1 displays the curve $\rho = \rho^{alg}_{(Mat,B)}(\delta)$ such that $\hat{\pi}(\rho) = 1/2$ for NIHT, HTP, and CSMPSP with $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ and $n = 2^{20}$ for $Mat = DCT$, $n = 2^{18}$ for $Mat = \mathcal{S}_p$ with $p = 4$ and $p = 7$, and $n = 2^{14}$ for $Mat = \mathcal{N}$. Figs. 16-19 display the logistic regression phase transitions for the other values of $n$ tested as well as the width of the transition regions given by $\rho_{0.1} - \rho_{0.9}$ where $\hat{\pi}(\rho_c) = c$. Unless otherwise stated, all phase transition curves presented are for the 50% recovery level curve defined by $\hat{\pi}(k/m) = 1/2$.

**Determination of algorithm timing and other behavior in recovery region** The behavior of the hard thresholding algorithms is explored over the entire $(k, m, n)$ region for the same algorithms, problem classes, and $m, n$ pairs as described for the phase transition evaluation. However, as opposed to the previous focus of $k$ near the phase transition, for each $(k, m, n)$ triple, ten problem instances are tested with $k = \lceil jm/50 \rceil$ beginning with $j = 1$ and increasing until the algorithm fails to successfully recover the measured $k$ sparse vector in each of the ten problem instances at that value of $(k, m, n)$. The algorithm run time data generated from these tests are used to determine an *algorithm selection map* where for each $(k, m, n)$ tested we indicate the algorithm which is able to reliably recover the measured sparse vectors with the least run time, see Fig. 2. The data generated includes: relative errors in multiple norms, average time per iteration, average total time, number of iterations, average convergence rate, and true/false support set discovery rates.

The focus of this manuscript is on providing practical information about which algorithm is able to reliably recover the measured sparse vector in the least amount of time, and to present this data throughout the $(k, m, n)$ parameter space. The average recovery time is presented for each $n$ using the axis of phase transition diagrams $m/n$ vs $k/m$. Fig. 2 presents algorithm selection maps for $(Mat, B)$ for $Mat = DCT$ with $n = 2^{20}$, $Mat = \mathcal{N}$ with $n = 2^{14}$, and $Mat = \mathcal{S}_7$ with $n = 2^{18}$ respectively, where the algorithm with the least average recovery time is indicated for each $(k, m, n)$, as well as a plot of the absolute times for the fastest algorithm; Figs. 3, 4, and 5 present accompanying ratios of the average recover time for each algorithm over the least average time. Figures 20, 21, and 24 present algorithm selection maps with smaller values of $n$ for the same problem classes and algorithms.

**Variance of algorithm behavior** The data generated from the testing described in the prior paragraph can be used to display numerous properties of an algorithm's behavior. For conciseness we limit our display of the non-timing data for a few representative problem sizes $(k, m, n)$. For each problem class $(Mat, B)$ with $Mat \in (\mathcal{N}, \mathcal{S}_7, DCT)$ we conduct tests for NIHT, HTP. and CSMPSP with $n$ selected as $2^{12}$ for $Mat = \mathcal{N}$, $n = 2^{16}$ for $Mat = \mathcal{S}_7$, and $n = 2^{18}$ for $Mat = DCT$. In each case five values of $m$ are selected so that $m/n$ is the value of (5) that is

nearest to $\{1/100, 1/20, 1/10, 3/10, 3/5\}$. For each $m, n$ pair, approximately one thousand problem instances were tested for each of three values of $k$ selected so that $k = c \cdot \rho^{\star}_{(Mat,B)}(m/n)$ for $c = \{1/2, 3/4, 9/10\}$ where $\rho^{\star}_{(Mat,B)}(m/n)$ is the maximum of the 50% level curves of the logistic regression fit of the recovery probability from the three algorithm's previously generated data. For each of NIHT, HTP, and CSMPSP the following data is presented in Tabs. I, II, and III for $Mat = \mathcal{N}$, $Mat = DCT$, and $Mat = \mathcal{S}_7$ respectively: $\#success$, $\#trials$, as well as the average and standard deviation of the fraction of the true support set contained in the support set of the recovered vector, the recovery time, number of iterations, and convergence rate.

### *2.2. Testing and analysis of problem classes both with and without noise*

While the discussion in Sec. 2.1 and the data presented in Sec. 3 focus on the binary vector ensemble $B$ at many values of $n$, the data generated for Sec. 4 considers the problem classes $(Mat, vec)$ for $vec \in \{B, U, N, B_\epsilon, U_\epsilon, N_\epsilon\}$ for a single value of $n$ per problem class. For $Mat \in \{S_4, S_7, DCT\}$ we fix $n = 2^{17}$ and for $Mat = \mathcal{N}$ we fix $n = 2^{13}$. While the analysis remains the same as described in the Sec. 2.1, the main change is an alternate definition of *successful recovery*. For $vec \in \{U, N\}$, there is a positive probability nonzero entries of the vector $x$ will take values below the success criteria defined by (4). Similarly for problem classes $(Mat, vec_\epsilon)$, even when the support set of the vector $x$ is known *a priori* the noise will introduce errors proportional to the noise level $\epsilon$. Therefore, the measured vector $x$ from a randomly drawn problem instance from the problem classes $(Mat, vec)$ with $vec \in \{B, U, N, B_\epsilon, U_\epsilon, N_\epsilon\}$ is considered to be *successfully recovered* if the relative $\ell_2$ error is no greater than $10^{-3}$ plus a constant multiple of the noise level $\epsilon$:

$$\frac{\|x - \hat{x}\|_2}{\|x\|_2} < .001 + 2\epsilon, \tag{8}$$

where $2\epsilon$ was observed to most accurately identify correct support set identification for $(Mat, B)$. Notice that this analysis uses the relative $\ell_2$ norm even in the noise free case where $\epsilon = 0$. All data displayed in Sec. 4 utilize (8). When comparing figures and data from Secs. 3 and 4 it is clear that defining success via (4) or (8) results in very similar phase transition curves $\rho^{alg}_{(Mat,B)}(m/n)$. It is important to note that this difference in the definition of success has no impact on algorithms during the recovery process; determining success is solely an aspect of the analysis and display of the output from the algorithms.

For ease of comparison, a second change to the presentation of results in Sec. 4 is the determination of problem instances $(k, m, n)$ for the tabular data detailing the variance of the algorithms; the dimensions of the problem size, $(k, m, n)$, are selected to match those in Tabs. I–III. In this way, one can directly compare the impact of noise on the algorithms for a fixed set of problems. The only change to the information in the table is again the definition of successful recovery where (8) is used in Tabs. IV–VI while (4) is used in Tabs. I–III.

## 3. ALGORITHM PERFORMANCE FOR PROBLEM CLASS $(Mat, B)$

This section presents and analyzes the main features of the data from the testing of problem classes $(Mat, B)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_7, DCT\}$ and the five hard thresholding algorithms Thresholding, IHT, NIHT, HTP, and CSMPSP. Focus on the sparse signed vector ensemble $B$ is due to both it being the vector ensemble with the lowest recovery phase transition, see [33, 34] and Claim 4, and due to the ease by which one can certify identification of the support set. The majority of the results shown are given as figures in order to provide as concise as possible a presentation of the large data set collected. The plots all have the undersampling ratio $m/n$ as the horizontal axis and the oversampling ratio $k/m$ as the vertical axis; this is consistent with earlier presentation of similar compressed sensing studies [38, 30, 22, 33, 34] and with theoretical sampling theorems [39].

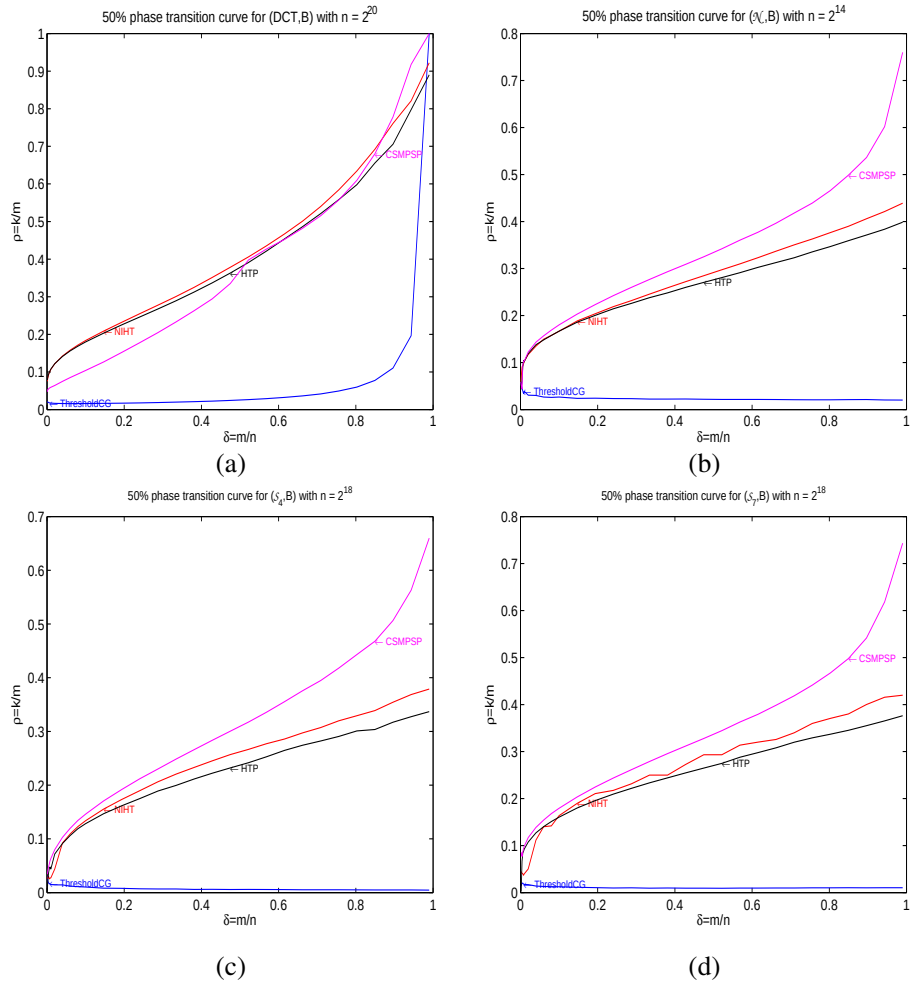



Figure 1. 50% recovery probability logistic regression curves for $vec = B$ and algorithms NIHT, HTP, CSMPSP, and Threshold: (a) $Mat = DCT$ with $n = 2^{20}$, (b) $Mat = \mathcal{N}$ with $n = 2^{14}$, (c) $Mat = \mathcal{S}_4$ with $n = 2^{18}$, and (d) $Mat = \mathcal{S}_7$ with $n = 2^{18}$.

### *3.1. Regions of high probability of recovery for* $(Mat, B)$

Rather than presenting the success probability directly for relatively few values of $(k, m, n)$, testing is conducted for 30 values of $\delta$ in (5), and for each $m, n$ pair, ten tests are performed for each of approximately 50 values of $k$ across the success phase transition region. The diversity of the values of $(k, m, n)$ tested give a broad sampling of the parameter space, but the resulting probability has less resolution for each $(k, m, n)$ tested resulting in a noisy appearance if displayed directly. For each $m, n$ pair the success probability is fit using logistic regression (6) and level curves of the logit fit are shown. Fig. 1 presents the 50% recovery level curves for each of the problem classes $(Mat, B)$ at the largest value of $n$ tested; curves are shown for algorithms Thresholding, NIHT, HTP, and CSMPSP. The recovery curves for IHT are observed to be at or below those for NIHT in all instances and are not displayed in Fig. 1. Figs. 16-19 present the 50% logistic regression fit for the smaller values of $n$ tested, as well as the gap between the 10% and 90% logistic regression.

Thresholding is observed to have a substantially lower recovery phase transition than do the iterative hard thresholding algorithms; this feature is exacerbated as $n$ is increased. In fact, for any fixed $\delta \in (0, 1)$, the phase transition for hard thresholding converges pointwise to zero at the rate $1/\log(n)$ [37]. In contrast the phase transitions for IHT, NIHT, HTP, and CSMPSP concentrate with increasing $n$ to values near those shown in Fig. 1; see Figs. 16-19.

*Observation 3.1.1*
For the problem class $(DCT, B)$ with $n = 2^{18}$ and $2^{20}$, NIHT and HTP have recovery phase transitions within $0.01$ of each other for $m/n < 1/4$ and approach one another as $m/n$ decreases to zero; whereas CSMPSP is observed to have a substantially lower phase transition as $m/n$ decreases below $1/2$. For $m/n > 1/2$ the phase transitions of NIHT remains slightly above that of HTP, with CSMPSP comparable near $m/n = 1/2$ and increasing to have the highest phase transition as $m/n$ approaches one. (See Fig. 1.)

*Observation 3.1.2*
For the problem classes $(\mathcal{N}, B)$ and $(\mathcal{S}_p, B)$ for $p = 4$ and $p = 7$, CSMPSP has a recovery phase transition that is above NIHT, whose phase transition is above HTP. Within each problem class, as $m/n$ approaches zero the phase transitions of these three algorithms approach one another. As $m/n$ approaches one the phase transitions for CSMPSP becomes more than twice those of NIHT and HTP. (See Fig. 1.)

Similar recovery phase transition studies to those shown in Fig. 1 were conducted in [33, 34, 23]. Quantifying and comparing the recovery phase transition curves for algorithms is of primary importance for their application as it indicates the problem sizes $(k, m, n)$ where the algorithm will return the measured vector. Obs. 3.1.1 and 3.1.2 highlight instances where the phase transitions are substantially different, indicating problem sizes where only one algorithm is able to reliably recovery the measured vector. However, these observations also highlight problem sizes in which multiple algorithms are able to recover the measured sparse vector, in particular, as $m/n$ approaches zero, the region of greatest practical interest for compressed sensing as this region corresponds to the most substantial undersampling. In regions where more than one algorithm is able to reliably recover the measured vector, further information is needed in order to determine which algorithm to select. To provide this additional information, this manuscript focuses on the average recovery time of the algorithms.

### *3.2. Average time for recovery and algorithm selection maps*

Figs. 2 and 20-24 present timing information for IHT, NIHT, HTP, and CSMPSP for problem classes $(Mat, B)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ for $p = 4$ and $p = 7$. The average recovery time for each algorithm is recorded for each $(k, m, n)$ where an algorithm's average error satisfies $\|\hat{x} - x\|_\infty < 1/100$; for each such $(k, m, n)$ the minimum of these average times is displayed, see Figs. 2 and 3-5. Accompanying each such minimum time plot is an *algorithm selection map* which indicates the algorithm whose minimum recovery time is shown in the plot at a given $(k, m, n)$; consequently, the algorithm selection map indicates which algorithm is recommended at that particular value of $(k, m, n)$. Fig. 2 displays the algorithm selection maps for the largest value of $n$ tested for each of the problem classes $(Mat, B)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_7, DCT\}$; algorithm selection maps for these problem classes and smaller values of $n$ are displayed in Figs. 20-24, including also $(\mathcal{S}_4, B)$.

For problem classes $(DCT, B)$, $(\mathcal{N}, B)$, and $(\mathcal{S}_7, B)$, Figs. 3-5 respectively display the accompanying ratio of the average recovery time for each algorithm divided by the minimum recovery time. These plots highlight how much slower algorithms are that did not have the minimum time, in some instances showing that multiple algorithms have very comparable execution times and regions where some algorithms are substantially slower than the fastest algorithm.

*Observation 3.2.1*
For the problem class $(DCT, B)$, NIHT is able to reliably successfully recover the measured vector in less time than HTP and CSMPSP for approximately $\frac{1}{2}\rho^{niht}_{(DCT,B)}(\delta) \lessapprox k/m \lessapprox \rho^{niht}_{(DCT,B)}(\delta)$ and $\delta \lessapprox 4/5$. HTP has a smaller recovery time for lower values of values of $k/m$, and CSMPSP has an even lower recovery time as $m/n$ approaches one or $k/m$ approaches zero. (See Fig. 2 Panel (a) and Fig. 20.) Throughout the recovery region the algorithms have recovery times that are typically within a multiple of five of each other, and often much closer. (See Fig. 3.)

Obs. 3.2.1 encourages the use of NIHT for the problem class $(DCT, B)$, with the computational time scaling approximately proportional to $n$. NIHT has its greatest recovery time at its phase

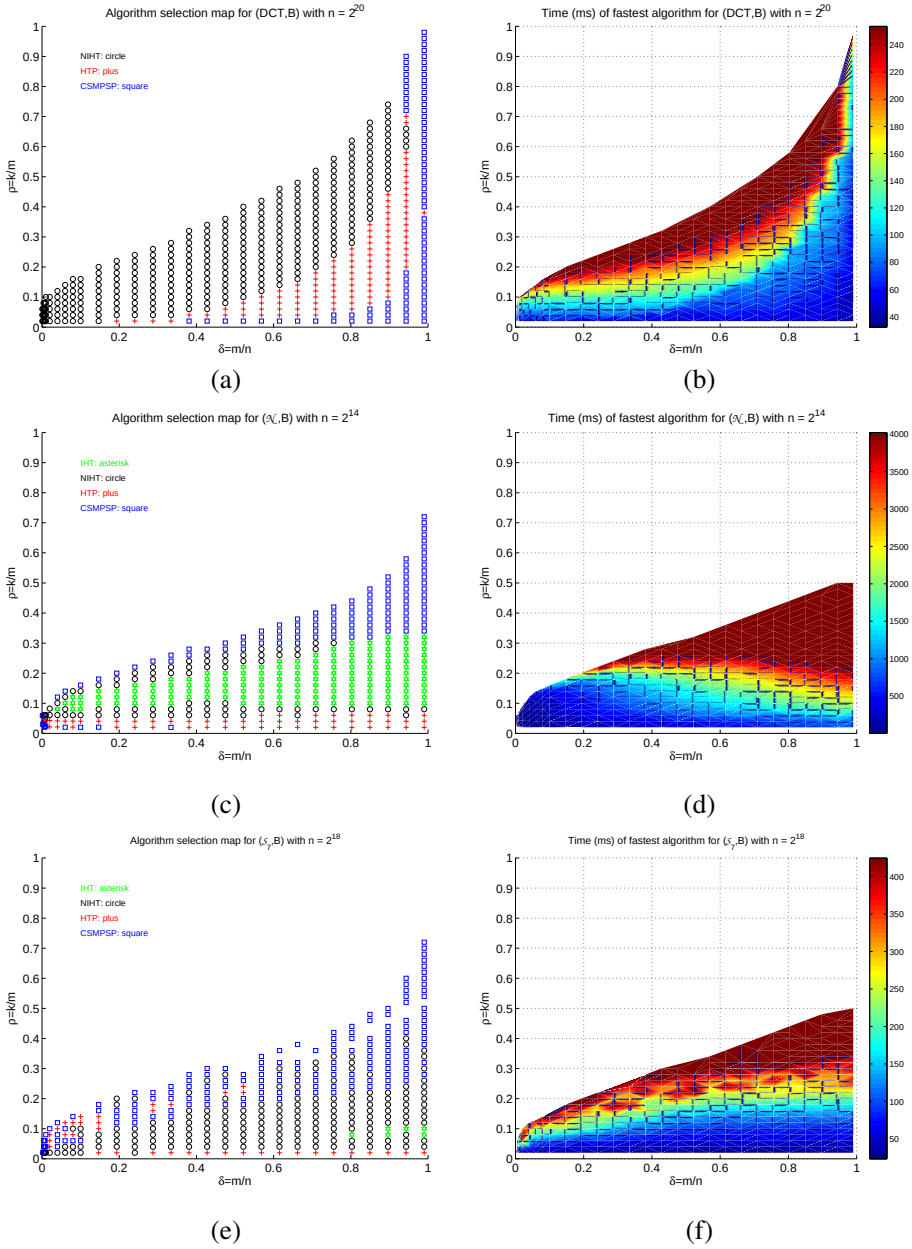



Figure 2. Left panels: Algorithm selection maps. Right panels: Average time for the fastest algorithm. Panels (a-b) $(DCT, B)$ with $n = 2^{20}$, (c-d) $(\mathcal{N}, B)$ with $n = 2^{14}$, and (e-f) $(\mathcal{S}_7, B)$ with $n = 2^{18}$.

transition, but even there NIHT is able to reliably recover the measured vector with $n = 2^{20}$ in, on average, under 250ms.

*Observation 3.2.2*
For the problem class $(\mathcal{N}, B)$, the algorithms show a clear layering in $k/m$ of minimum compute time. CSMPSP has a substantially higher phase transition than the other algorithms, particularly as $m/n$ approaches one, with the algorithm selection map marking CSMPSP in that region. Once entering the region where the other algorithms are also able to reliably recover the measured vector, we observe NIHT or IHT to require the least time for moderate values of $k/m$, and HTP taking the least time as $k/m$ approaches zero. (See Fig. 2 Panel (c) and Fig. 21) In the region where IHT

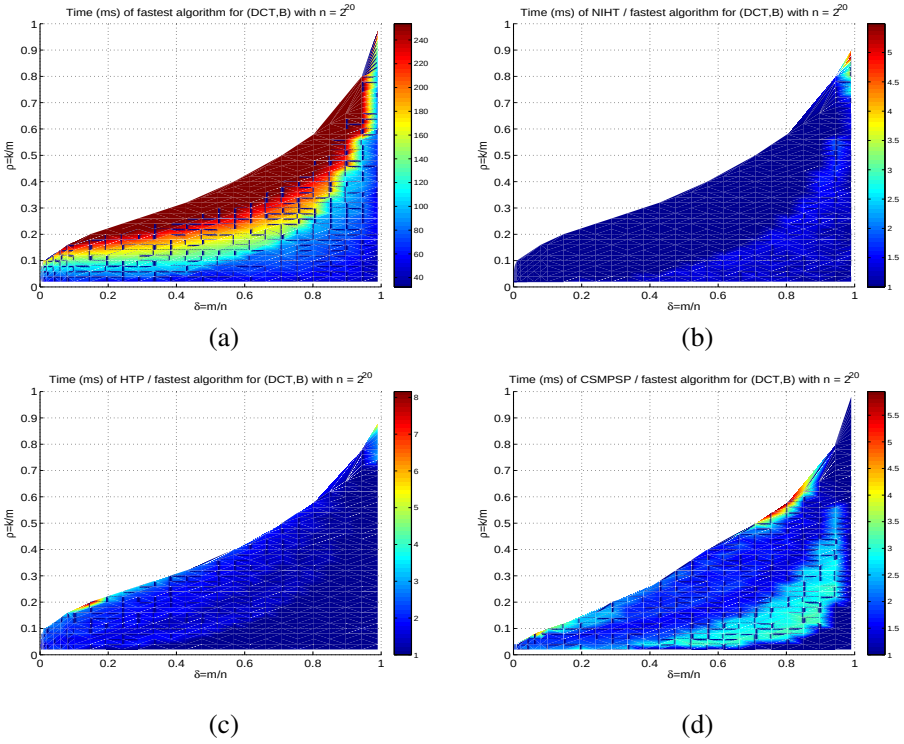



Figure 3. Panel (a), time for fastest algorithm for $(DCT, B)$ with $n = 2^{20}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

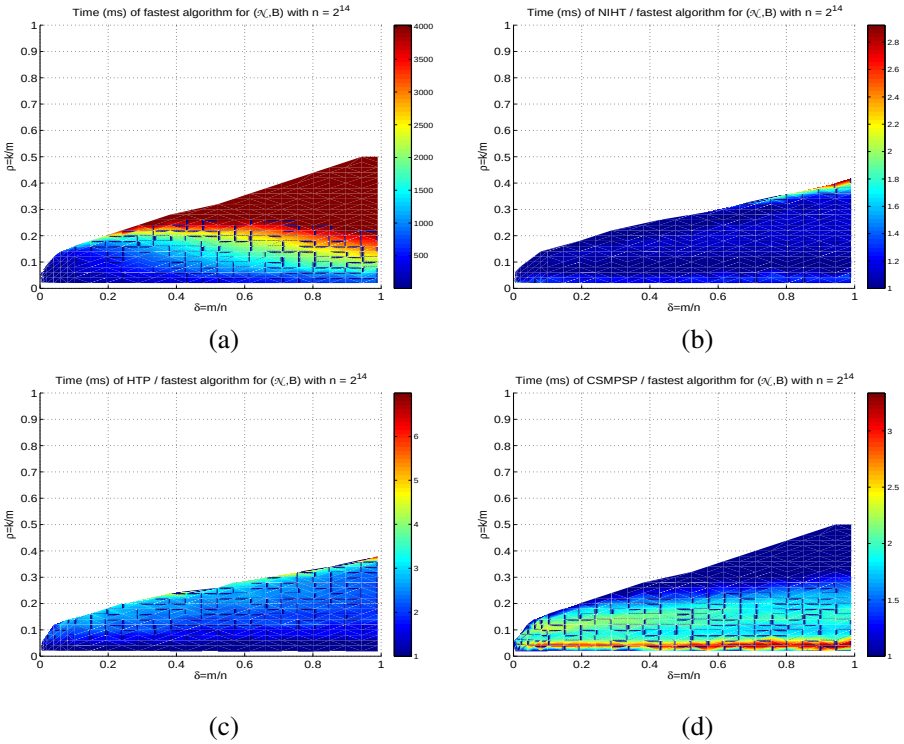



Figure 4. Panel (a), time for fastest algorithm for $(\mathcal{N}, B)$ with $n = 2^{14}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

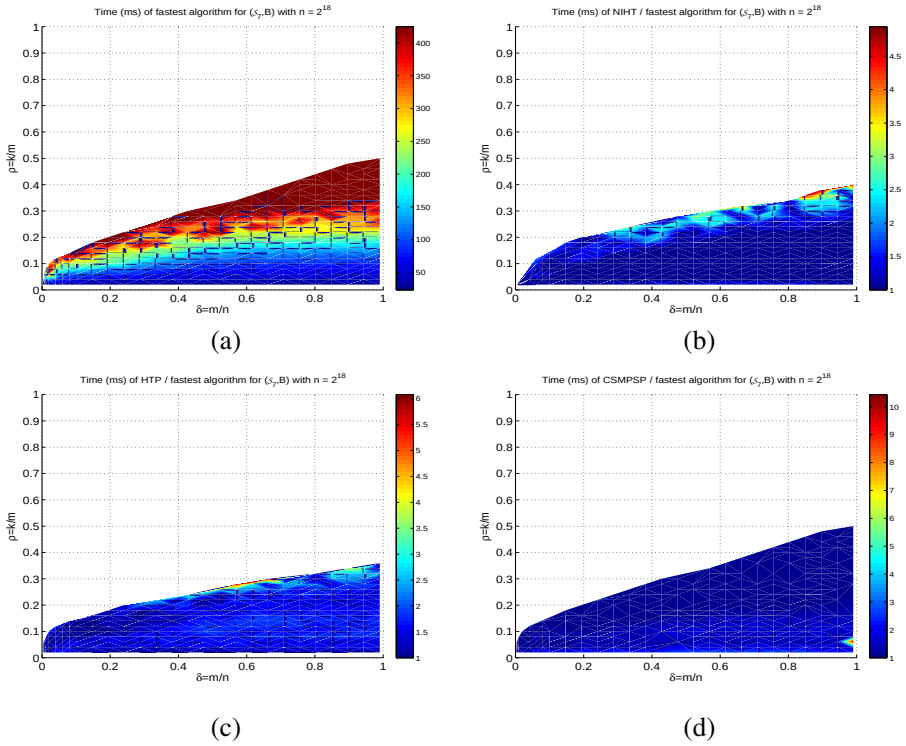



Figure 5. Panel (a), time for fastest algorithm for $(\mathcal{S}_7, B)$ with $n = 2^{14}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

is marked, NIHT typically requires about 40% more time due to calculating the step size, whereas IHT is using the fixed step size $\mu = 0.65$ advocated in [33]. HTP takes from about 3 times as long down to approximately the same time as $k/m$ decreases to zero. (See Fig. 2 Panel (d) and Fig. 4.)

Obs. 3.2.2 encourages the use of CSMPSP for the problem class $(\mathcal{N}, B)$ and the largest values of $k/m$ recoverable, most notably as $m/n$ approaches one. For $m/n \lessapprox 1/2$ the recovery time for NIHT is either the least or within about 40% of the fastest; HTP and CSMPSP take modestly longer. For $m/n < 1/10$ the absolute recovery time for problems of size $n = 2^{14}$ in less than 500ms in regions where these algorithms are reliably able to recover the measured vector.

*Observation 3.2.3*
For the problem class $(\mathcal{S}_p, B)$ with $p = 4$ and $p = 7$ the algorithms show two recovery regions. CSMPSP has a substantially higher phase transition than the other algorithms, particularly as $m/n$ approaches one. CSMPSP has the least recovery time, or within a small multiple, throughout its recovery region. NIHT has a modestly smaller recovery time than CSMPSP in the majority of its recovery region, with HTP having the least recovery time for the smallest value of $k/m$. (See Fig. 2 Panel (e)-(f) and Figs. 5, 22-24.)

Obs. 3.2.3 encourages the use of CSMPSP for the problem class $(\mathcal{S}_p, B)$ for $p = 4$ and $p = 7$; with average recovery times for problems of size $n = 2^{18}$ in under 400ms. Increasing $p$ from 4 to 7 causes a notable rise in the recovery region for NIHT, in which it has approximately the same recovery time as CSMPSP for $(k, m, n)$ modestly below the recovery phase transition of NIHT.

### *3.3. Variance of algorithm recovery behavior*

Within each problem class $(Mat, B)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$, both the region of recovery and the average time for recovery of NIHT, HTP, and CSMPSP show little variability for $(k, m, n)$ fixed once $n$ is large. The low variability is most readily apparent by testing the algorithms for

| $m$ | $k$ | Algorithm | succ/test | true pos fraction | time (ms) | iterations | convergence rate |
|---|---|---|---|---|---|---|---|
| 246 | 19 | NIHT | 892/892 | 1(± 0) | 19.23(± 2) | 14.27(± 2) | 0.4699(± 0.03) |
| | | HTP | 896/896 | 1(± 0) | 30.22(± 8) | 1.371(± 0.5) | 0.02743(± 0.1) |
| | | CSMPSP | 907/907 | 1(± 0) | 29.59(± 5) | 0.1213(± 0.3) | 0.8793(± 0.3) |
| | 28 | NIHT | 890/900 | 0.9947(± 0.05) | 31.38(± 5e+01) | 24.38(± 3e+01) | 0.5589(± 0.06) |
| | | HTP | 898/913 | 0.992(± 0.06) | 76.6(± 9e+01) | 4.179(± 6) | 0.1035(± 0.1) |
| | | CSMPSP | 904/906 | 0.9991(± 0.02) | 50.45(± 2e+01) | 1.396(± 1) | 0.04757(± 0.2) |
| | 34 | NIHT | 634/938 | 0.8481(± 0.2) | 73.15(± 2e+02) | 55.09(± 1e+02) | 0.7403(± 0.2) |
| | | HTP | 644/919 | 0.8597(± 0.2) | 234.9(± 2e+02) | 14.09(± 1e+01) | 0.4621(± 0.4) |
| | | CSMPSP | 808/945 | 0.927(± 0.2) | 224.2(± 4e+02) | 13.33(± 3e+01) | 0.2533(± 0.3) |
| 410 | 37 | NIHT | 909/909 | 1(± 0) | 25.97(± 2) | 16(± 1) | 0.5099(± 0.02) |
| | | HTP | 895/895 | 1(± 0) | 46.89(± 1e+01) | 1.706(± 0.5) | 0.02567(± 0.01) |
| | | CSMPSP | 884/884 | 1(± 0) | 43.16(± 9) | 0.3077(± 0.5) | 0.6944(± 0.5) |
| | 56 | NIHT | 972/979 | 0.9968(± 0.04) | 41.54(± 7) | 26.48(± 4) | 0.6177(± 0.04) |
| | | HTP | 969/976 | 0.9971(± 0.03) | 108.1(± 6e+01) | 4.589(± 3) | 0.143(± 0.1) |
| | | CSMPSP | 973/974 | 0.9995(± 0.01) | 73.1(± 4e+01) | 2.476(± 2) | 0.05187(± 0.05) |
| | 67 | NIHT | 639/1022 | 0.8419(± 0.2) | 71.21(± 5e+01) | 44.57(± 3e+01) | 0.7928(± 0.2) |
| | | HTP | 605/1017 | 0.8308(± 0.2) | 383(± 3e+02) | 17.76(± 1e+01) | 0.5923(± 0.3) |
| | | CSMPSP | 938/1022 | 0.9634(± 0.1) | 308.1(± 6e+02) | 14.26(± 3e+01) | 0.233(± 0.2) |
| 1178 | 153 | NIHT | 902/902 | 1(± 0) | 68.01(± 3) | 20.76(± 0.9) | 0.6097(± 0.02) |
| | | HTP | 896/896 | 1(± 0) | 136.5(± 2e+01) | 2.252(± 0.4) | 0.05033(± 0.02) |
| | | CSMPSP | 900/900 | 1(± 0) | 131.3(± 9) | 1.002(± 0.05) | 0.03883(± 0.1) |
| | 230 | NIHT | 913/913 | 1(± 0) | 124.4(± 1e+01) | 38.6(± 4) | 0.7276(± 0.01) |
| | | HTP | 905/907 | 0.9993(± 0.01) | 344.2(± 1e+02) | 6.529(± 2) | 0.2566(± 0.08) |
| | | CSMPSP | 894/894 | 1(± 0) | 194(± 2e+01) | 2.851(± 0.7) | 0.07642(± 0.03) |
| | 275 | NIHT | 305/930 | 0.7925(± 0.2) | 222.9(± 5e+01) | 68.8(± 1e+01) | 0.9262(± 0.1) |
| | | HTP | 150/911 | 0.7333(± 0.1) | 1436(± 5e+02) | 29.34(± 1e+01) | 0.9197(± 0.2) |
| | | CSMPSP | 892/902 | 0.9965(± 0.03) | 383.7(± 6e+02) | 6.91(± 1e+01) | 0.209(± 0.1) |
| 2521 | 477 | NIHT | 894/894 | 1(± 0) | 177.4(± 5) | 28.89(± 0.9) | 0.7176(± 0.007) |
| | | HTP | 894/894 | 1(± 0) | 360.6(± 3e+01) | 3.128(± 0.3) | 0.09977(± 0.02) |
| | | CSMPSP | 905/905 | 1(± 0) | 304.2(± 8) | 2.253(± 0.5) | 0.0652(± 0.03) |
| | 716 | NIHT | 890/892 | 0.9995(± 0.01) | 429.3(± 6e+01) | 71.35(± 9) | 0.8254(± 0.009) |
| | | HTP | 886/926 | 0.9916(± 0.04) | 1628(± 7e+02) | 16.48(± 8) | 0.532(± 0.1) |
| | | CSMPSP | 883/883 | 1(± 0) | 444.6(± 2e+01) | 3.677(± 0.6) | 0.1295(± 0.03) |
| | 859 | NIHT | 19/926 | 0.7727(± 0.05) | 660(± 1e+02) | 108.5(± 2e+01) | 0.9973(± 0.02) |
| | | HTP | 0/912 | 0.7158(± 0.03) | 3099(± 8e+02) | 32.03(± 8) | 1(± 0) |
| | | CSMPSP | 919/919 | 1(± 0) | 762.6(± 2e+02) | 8.607(± 2) | 0.3564(± 0.06) |

Table I. Performance of NIHT, HTP, and CSMPSP for $(\mathcal{N}, B)$ with n=4096

large numbers of problem instances for a given problem size $(k, m, n)$. For problem classes $(\mathcal{N}, B)$, $(DCT, B)$, and $(\mathcal{S}_7, B)$ Tabs. I, II, and III respectively present the average and standard deviation of algorithm characteristics: support set true positive fraction, recovery time, number of iterations, and convergence rates.

For $Mat \in \{\mathcal{S}_7, DCT\}$, five values of $m$ are tested for a single value of $n$ and for $Mat = \mathcal{N}$ four values of $m$ are tested with $n = 2^{12}$. For each $m, n$ pair approximately one thousand problem instances were tested for each of three values of $k$ selected so that $k = c \cdot \rho^{\star}_{(Mat,B)}(m/n)$ for $c = \{1/2, 3/4, 9/10\}$ where $\rho^{\star}_{(Mat,B)}(m/n)$ is the maximum of the 50% level curves of the logistic regression fit of the recovery probability from the previously generated data. These three values of $k$ are selected as illustrative of the algorithm behavior in regions where at least one of the algorithms is able to reliably recover the measured vector. In particular, for $m, n$ fixed, variability decreases as $k$ decreases.

*Observation 3.3.1*
When successfully recovering a vector from the problem class $(Mat, B)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$, the following performance characteristics for each algorithm exhibit small variation about the mean: fraction of support set identified, time for recovery, iterations, and convergence rate. (See Tabs. I–III.)

The low variance of the algorithm characteristics in regions where they recover the measured vector allow for highly reliable conclusions to be drawn on the performance of the tested hard thresholding algorithms in their regions of applicability. This consistent behavior is an important aspect of the reliability of the claims and observations in this manuscript. Moreover, they inform a practitioner that the algorithm can be expected to perform consistently for problem instances from the same class and similar problem sizes.

| $m$ | $k$ | Algorithm | succ/test | true pos fraction | time (ms) | iterations | convergence rate |
|---|---|---|---|---|---|---|---|
| 2622 | 141 | NIHT | 1799/1799 | 1(± 0) | 27.45(± 0.8) | 13.03(± 0.4) | 0.4216(± 0.009) |
| | | HTP | 1808/1808 | 1(± 0) | 52.2(± 1e+01) | 1.314(± 0.5) | 0.01257(± 0.01) |
| | | CSMPSP | 1775/1787 | 1(± 0.0006) | 52.99(± 4e+01) | 0.1589(± 2) | 0.996(± 0.06) |
| | 211 | NIHT | 1809/1809 | 1(± 0) | 39.67(± 2) | 18.97(± 0.8) | 0.4942(± 0.006) |
| | | HTP | 1820/1820 | 1(± 0) | 99.35(± 8) | 2.913(± 0.3) | 0.05225(± 0.01) |
| | | CSMPSP | 0/1812 | 0.8563(± 0.04) | 695.3(± 8e+01) | 23.23(± 3) | 1(± 0) |
| | 253 | NIHT | 1812/1814 | 0.9996(± 0.01) | 54.68(± 6) | 26.48(± 3) | 0.5377(± 0.02) |
| | | HTP | 1802/1802 | 1(± 0) | 158(± 3e+01) | 4.967(± 1) | 0.1415(± 0.04) |
| | | CSMPSP | 0/1785 | 0.626(± 0.05) | 740.6(± 8e+01) | 24.86(± 3) | 1(± 0) |
| 15729 | 1239 | NIHT | 1942/1942 | 1(± 0) | 33.87(± 0.8) | 15.8(± 0.4) | 0.4776(± 0.003) |
| | | HTP | 1951/1951 | 1(± 0) | 74.22(± 0.5) | 2(± 0) | 0.03196(± 0.0007) |
| | | CSMPSP | 1954/1967 | 1(± 0.0003) | 118.9(± 7e+01) | 2.438(± 2) | 0.5167(± 0.4) |
| | 1858 | NIHT | 1967/1967 | 1(± 0) | 49.53(± 1) | 23.6(± 0.5) | 0.5857(± 0.004) |
| | | HTP | 1966/1966 | 1(± 0) | 109.5(± 8) | 3.108(± 0.3) | 0.06403(± 0.02) |
| | | CSMPSP | 0/1633 | 0.8551(± 0.01) | 572.8(± 3e+01) | 18.22(± 1) | 1(± 0) |
| | 2229 | NIHT | 1000/1000 | 1(± 0) | 67.92(± 2) | 32.4(± 1) | 0.6358(± 0.003) |
| | | HTP | 1000/1000 | 1(± 0) | 198.3(± 1e+01) | 6.137(± 0.5) | 0.1984(± 0.02) |
| | | CSMPSP | 0/998 | 0.6846(± 0.02) | 644.2(± 1e+02) | 20.65(± 4) | 1(± 0) |
| 26215 | 2401 | NIHT | 992/992 | 1(± 0) | 36.4(± 0.8) | 16.84(± 0.4) | 0.4915(± 0.002) |
| | | HTP | 994/994 | 1(± 0) | 76.21(± 0.9) | 2(± 0) | 0.03263(± 0.006) |
| | | CSMPSP | 998/998 | 1(± 0) | 128.7(± 2e+01) | 2.805(± 0.7) | 0.1602(± 0.2) |
| | 3601 | NIHT | 987/987 | 1(± 0) | 54.13(± 1) | 25.4(± 0.5) | 0.6164(± 0.003) |
| | | HTP | 966/966 | 1(± 0) | 129.8(± 1e+01) | 3.757(± 0.4) | 0.09367(± 0.02) |
| | | CSMPSP | 0/975 | 0.8665(± 0.008) | 601.3(± 2e+02) | 18.81(± 5) | 1(± 3e-07) |
| | 4321 | NIHT | 980/980 | 1(± 0) | 75.85(± 2) | 35.41(± 0.9) | 0.6673(± 0.002) |
| | | HTP | 978/978 | 1(± 0) | 222.3(± 1e+01) | 6.782(± 0.5) | 0.2258(± 0.02) |
| | | CSMPSP | 0/979 | 0.7107(± 0.01) | 657.6(± 2e+02) | 20.74(± 6) | 1(± 1e-06) |
| 75332 | 10451 | NIHT | 1000/1000 | 1(± 0) | 46.54(± 0.8) | 20.16(± 0.4) | 0.5589(± 0.003) |
| | | HTP | 999/999 | 1(± 0) | 83.88(± 0.4) | 2(± 0) | 0.03451(± 0.0003) |
| | | CSMPSP | 1000/1000 | 1(± 0) | 101.7(± 1e+01) | 1.68(± 0.5) | 0.03724(± 0.03) |
| | 15676 | NIHT | 1000/1000 | 1(± 0) | 72.99(± 0.6) | 31.98(± 0.1) | 0.6809(± 0.001) |
| | | HTP | 1000/1000 | 1(± 0) | 147.5(± 7) | 3.938(± 0.2) | 0.1036(± 0.01) |
| | | CSMPSP | 8/1001 | 0.9291(± 0.007) | 1407(± 1e+03) | 42.75(± 4e+01) | 0.9925(± 0.08) |
| | 18811 | NIHT | 1000/1000 | 1(± 0) | 102.8(± 1) | 44.71(± 0.6) | 0.7325(± 0.001) |
| | | HTP | 1000/1000 | 1(± 0) | 255.3(± 9) | 7.117(± 0.3) | 0.2429(± 0.02) |
| | | CSMPSP | 0/1000 | 0.8067(± 0.007) | 2475(± 1e+03) | 76.76(± 5e+01) | 1(± 4e-07) |
| 161288 | 37721 | NIHT | 991/991 | 1(± 0) | 54.11(± 0.3) | 21(± 0) | 0.5707(± 0.0007) |
| | | HTP | 998/998 | 1(± 0) | 61.18(± 0.08) | 1(± 0) | 0.003893(± 3e-05) |
| | | CSMPSP | 993/993 | 1(± 0) | 73.4(± 1e+01) | 0.2951(± 0.5) | 0.711(± 0.4) |
| | 56581 | NIHT | 994/994 | 1(± 0) | 86.26(± 0.6) | 34.07(± 0.3) | 0.7081(± 0.001) |
| | | HTP | 999/999 | 1(± 0) | 105.9(± 6) | 2.056(± 0.2) | 0.02632(± 0.008) |
| | | CSMPSP | 997/997 | 1(± 0) | 120.1(± 3) | 2.22(± 0.4) | 0.03593(± 0.01) |
| | 67897 | NIHT | 992/992 | 1(± 0) | 122.8(± 1) | 48.24(± 0.4) | 0.7714(± 0.001) |
| | | HTP | 998/998 | 1(± 0) | 214.3(± 5) | 5.007(± 0.1) | 0.1505(± 0.008) |
| | | CSMPSP | 994/994 | 1(± 0) | 167.9(± 3e+01) | 3.642(± 1) | 0.08627(± 0.06) |

Table II. Performance of NIHT, HTP, and CSMPSP for $(DCT, B)$ with n=262144

## 4. ADDITIONAL PERFORMANCE ANALYSES

The superiority of NIHT, HTP, and CSMPSP when compared to IHT and Hard Thresholding, for the problem class $(Mat, B)$, is evident in Sec. 3. This section focuses exclusively on these three algorithms and expands the problem classes from $(Mat, B)$ to $(Mat, vec)$ with $vec \in \{B, U, N, B_\epsilon, U_\epsilon, N_\epsilon\}$. Sec. 4.1 continues the focus on the vector ensemble $B$ as in Sec. 3 while considering the effects of noise by studying problem classes $(Mat, B)$ and $(Mat, B_\epsilon)$. Sec. 4.2 extends the analysis further to the vector ensembles $\{U, N, U_\epsilon, N_\epsilon\}$.

Recall from Sec. 2.2 that the analysis throughout this section considers a slightly relaxed definition of success than was considered in the previous section. For $Mat \in \{\mathcal{S}_p, DCT\}$, the tests are conducted with $n = 2^{17}$ while $n = 2^{13}$ for $Mat = \mathcal{N}$. While various other noise levels were tested, this manuscript studies the moderate noise level $\epsilon = 0.1$ as representative of the impact on the algorithms of non adversarial, additive noise.

### *4.1. Performance Analysis in the Presence of Noise*

In this section we study the effect of noise on the greedy algorithms by studying the problem classes $(Mat, B)$ and $(Mat, B_\epsilon)$ for all matrix ensembles. As detailed in Sec. 2.2 the algorithms attempt to find the vector $x$ from the information $y = Ax + e$ where $e$ is drawn from $\mathcal{N}(0, \epsilon\|Ax\|_2)$. For vector

| $m$ | $k$ | Algorithm | succ/test | true pos fraction | time (ms) | iterations | convergence rate |
|---|---|---|---|---|---|---|---|
| 656 | 32 | NIHT | 283/376 | 0.8396(± 0.3) | 266.9(± 4e+02) | 203.5(± 3e+02) | 0.5789(± 0.2) |
| | | HTP | 327/327 | 1(± 0) | 41.85(± 9) | 2.043(± 0.6) | 0.01716(± 0.01) |
| | | CSMPSP | 367/367 | 1(± 0) | 39.21(± 5) | 1.27(± 0.5) | 0.2327(± 0.2) |
| | 48 | NIHT | 1/329 | 0.2098(± 0.1) | 978.4(± 5e+01) | 748.9(± 4e+01) | 0.9986(± 0.03) |
| | | HTP | 335/337 | 0.9974(± 0.04) | 101.3(± 3e+01) | 5.733(± 2) | 0.1383(± 0.09) |
| | | CSMPSP | 353/353 | 1(± 0) | 60.72(± 9) | 2.541(± 0.7) | 0.03072(± 0.02) |
| | 58 | NIHT | 0/481 | 0.1559(± 0.1) | 983.6(± 1e+01) | 751(± 0.05) | 1(± 1e-05) |
| | | HTP | 362/510 | 0.878(± 0.2) | 257.6(± 1e+02) | 15.05(± 9) | 0.4933(± 0.3) |
| | | CSMPSP | 441/485 | 0.9587(± 0.1) | 192.8(± 4e+02) | 10.9(± 2e+01) | 0.1785(± 0.3) |
| 3933 | 305 | NIHT | 896/902 | 0.9982(± 0.02) | 95.77(± 8e+01) | 73.22(± 6e+01) | 0.5434(± 0.05) |
| | | HTP | 892/892 | 1(± 0) | 73.69(± 2e+01) | 3.998(± 1) | 0.0878(± 0.04) |
| | | CSMPSP | 897/897 | 1(± 0) | 57.81(± 9) | 2.344(± 1) | 0.0633(± 0.06) |
| | 458 | NIHT | 907/976 | 0.9744(± 0.09) | 195.3(± 2e+02) | 148.3(± 2e+02) | 0.668(± 0.09) |
| | | HTP | 962/962 | 1(± 0) | 119.7(± 2e+01) | 6.765(± 1) | 0.2073(± 0.04) |
| | | CSMPSP | 955/961 | 0.9942(± 0.07) | 100.8(± 2e+01) | 8.445(± 5) | 0.2875(± 0.2) |
| | 549 | NIHT | 451/967 | 0.793(± 0.2) | 247.4(± 2e+02) | 184.6(± 2e+02) | 0.8542(± 0.2) |
| | | HTP | 513/999 | 0.8373(± 0.2) | 415.8(± 2e+02) | 25.02(± 9) | 0.7193(± 0.3) |
| | | CSMPSP | 975/975 | 1(± 0) | 143.4(± 3e+01) | 11.8(± 7) | 0.3983(± 0.3) |
| 6554 | 589 | NIHT | 896/896 | 1(± 0) | 58.91(± 8) | 45.65(± 6) | 0.5786(± 0.02) |
| | | HTP | 899/899 | 1(± 0) | 65.68(± 1e+01) | 3.514(± 0.7) | 0.07501(± 0.03) |
| | | CSMPSP | 901/901 | 1(± 0) | 77.21(± 2e+01) | 6.3(± 5) | 0.2432(± 0.2) |
| | 883 | NIHT | 873/876 | 0.9992(± 0.01) | 139.1(± 7e+01) | 106.1(± 5e+01) | 0.6812(± 0.02) |
| | | HTP | 891/891 | 1(± 0) | 125(± 1e+01) | 7.103(± 0.9) | 0.2297(± 0.04) |
| | | CSMPSP | 895/895 | 1(± 0) | 115(± 2e+01) | 10.88(± 6) | 0.4(± 0.3) |
| | 1060 | NIHT | 459/916 | 0.8333(± 0.2) | 213.9(± 1e+02) | 159.1(± 1e+02) | 0.8623(± 0.1) |
| | | HTP | 274/924 | 0.7826(± 0.1) | 507(± 2e+02) | 30.68(± 1e+01) | 0.8448(± 0.2) |
| | | CSMPSP | 874/878 | 0.9956(± 0.06) | 158.3(± 3e+01) | 14.23(± 7) | 0.493(± 0.3) |
| 18833 | 2460 | NIHT | 994/994 | 1(± 0) | 73.32(± 8) | 53.84(± 6) | 0.6731(± 0.01) |
| | | HTP | 996/996 | 1(± 0) | 78.79(± 4) | 4.012(± 0.3) | 0.1145(± 0.02) |
| | | CSMPSP | 990/993 | 0.9971(± 0.05) | 69.91(± 7) | 3.482(± 2) | 0.1125(± 0.1) |
| | 3690 | NIHT | 991/997 | 0.994(± 0.08) | 140.9(± 2e+01) | 103.8(± 2e+01) | 0.7722(± 0.02) |
| | | HTP | 990/993 | 0.997(± 0.05) | 181.2(± 2e+01) | 9.972(± 0.9) | 0.3624(± 0.04) |
| | | CSMPSP | 994/994 | 1(± 0) | 99.65(± 5) | 6.038(± 1) | 0.201(± 0.06) |
| | 4428 | NIHT | 154/1008 | 0.7779(± 0.1) | 333.2(± 2e+02) | 234.6(± 1e+02) | 0.9714(± 0.07) |
| | | HTP | 0/995 | 0.6872(± 0.02) | 1190(± 7e+02) | 69.57(± 4e+01) | 1(± 0) |
| | | CSMPSP | 1002/1004 | 1(± 1e-05) | 153.7(± 1e+02) | 8.323(± 5) | 0.2805(± 0.06) |
| 40322 | 7656 | NIHT | 996/996 | 1(± 0) | 75.67(± 8) | 53.56(± 6) | 0.7551(± 0.01) |
| | | HTP | 999/999 | 1(± 0) | 103.5(± 3) | 5.018(± 0.1) | 0.1685(± 0.01) |
| | | CSMPSP | 983/992 | 0.9912(± 0.09) | 77.66(± 8) | 2.293(± 0.6) | 0.05613(± 0.09) |
| | 11484 | NIHT | 983/998 | 0.985(± 0.1) | 240.6(± 5e+01) | 153.3(± 3e+01) | 0.8528(± 0.02) |
| | | HTP | 988/988 | 1(± 0) | 418.1(± 6e+01) | 20.01(± 3) | 0.5146(± 0.005) |
| | | CSMPSP | 995/995 | 1(± 0) | 136.8(± 4) | 5.447(± 0.6) | 0.186(± 0.03) |
| | 13780 | NIHT | 4/993 | 0.7933(± 0.03) | 624(± 3e+02) | 362.3(± 2e+02) | 0.9996(± 0.007) |
| | | HTP | 0/995 | 0.7181(± 0.01) | 1882(± 9e+02) | 88.44(± 4e+01) | 1(± 0) |
| | | CSMPSP | 990/990 | 1(± 0) | 233.8(± 2e+01) | 9.882(± 2) | 0.36(± 0.06) |

Table III. Performance of NIHT, HTP, and CSMPSP for $(\mathcal{S}_7, B)$ with n=65536

ensemble $B$ we set $\epsilon = 0$ to eliminate additive noise and for vector ensemble $B_\epsilon$ we set $\epsilon = 1/10$. In other words, the vector ensemble $B_\epsilon$ consists of noisy, binary signals with a signal-to-noise ratio equal to 10. From (8), we consider a problem instance a successful recovery when

$$\frac{\|x - \hat{x}\|_2}{\|x\|_2} \leq .001 + 2\epsilon = \begin{cases} .001 & \text{for } vec = B \\ .201 & \text{for } vec = B_\epsilon \end{cases}.$$

This section performs a parallel analysis to that in Sec. 3 by describing the effect of noise on the region of successful recovery, the effect of noise on selecting an algorithm in the recovery region, and the variance of descriptive performance characteristics for the algorithms.

*Observation 4.1.1*
With $\epsilon = 1/10$, $alg \in \{$NIHT,HTP,CSMPSP$\}$, and $Mat \in \{\mathcal{N}, \mathcal{S}_7, DCT\}$ the phase transition curves $\rho^{alg}_{(Mat,B_\epsilon)}(\delta)$ and $\rho^{alg}_{(Mat,B)}(\delta)$ are within 0.01 of each other for $\delta < 1/3$. (See Figs. 6–7.)

Observation 4.1.1 describes what is commonly referred to as a "stability to noise" of the recovery phase transition. Each of these algorithms has a theoretical sufficient condition based on the restricted isometry property which guarantees the recovery error is bounded by a factor scaling with the noise level $\epsilon$. While these sufficient conditions are incredibly pessimistic [30, 31], Obs. 4.1.1 demonstrates all three algorithms' recovery errors are proportional to the noise level $\epsilon$ in the majority of problem instances within the noiseless recovery region.

*Observation 4.1.2*
For $\epsilon = 0.1$ and $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$, the relationships between the phase transition curves for

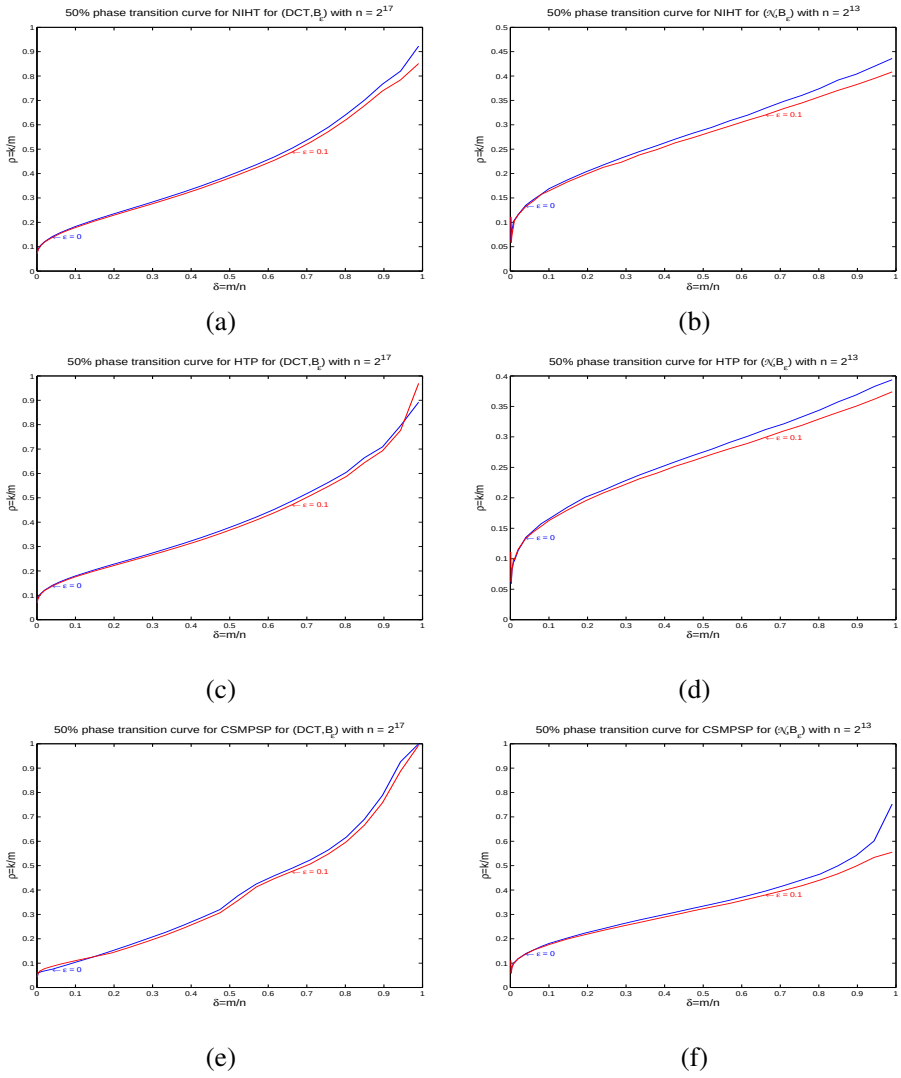



Figure 6. 50% recovery probability logistic regression curves for $vec = B_\epsilon$ with $\epsilon = 0$ and $\epsilon = 1/10$ for $Mat = DCT$ with $n = 2^{17}$ in the left panels and for $Mat = \mathcal{N}$ with $n = 2^{13}$ in the right panels. Algorithms NIHT, HTP, and CSMPSP in Panels (a)-(b), (c)-(d), and (e)-(f) respectively.

NIHT, HTP, and CSMPSP for problem class $(Mat, B_\epsilon)$ are consistent with Observations 3.1.1 and 3.1.2. (See Fig. 8.)

As the recovery phase transition curves for $(Mat, B_\epsilon)$ track closely with the related recovery phase transition from $(Mat, B)$, the relative positioning of the moderate noise recovery phase transitions remain the same as the noise free case. Observation 4.1.2 states that the regions of successful recovery share the same inclusion relationships both with and without noise. This is the type of recovery phase transition information has been reported in previous work. Here we observe that though the relationships between the recovery regions change little with the inclusion of moderate noise, the noise has a profound effect on the relationships of other performance characteristics of these algorithms. Observation 4.1.3 states that, in terms of average recovery time, the projection based methods HTP and CSMPSP are more significantly degraded than is the steepest descent based method NIHT. The ratios of recovery times from Figs. 10–12 show ratios of HTP and CSMPSP to NIHT regularly exceeding 10.

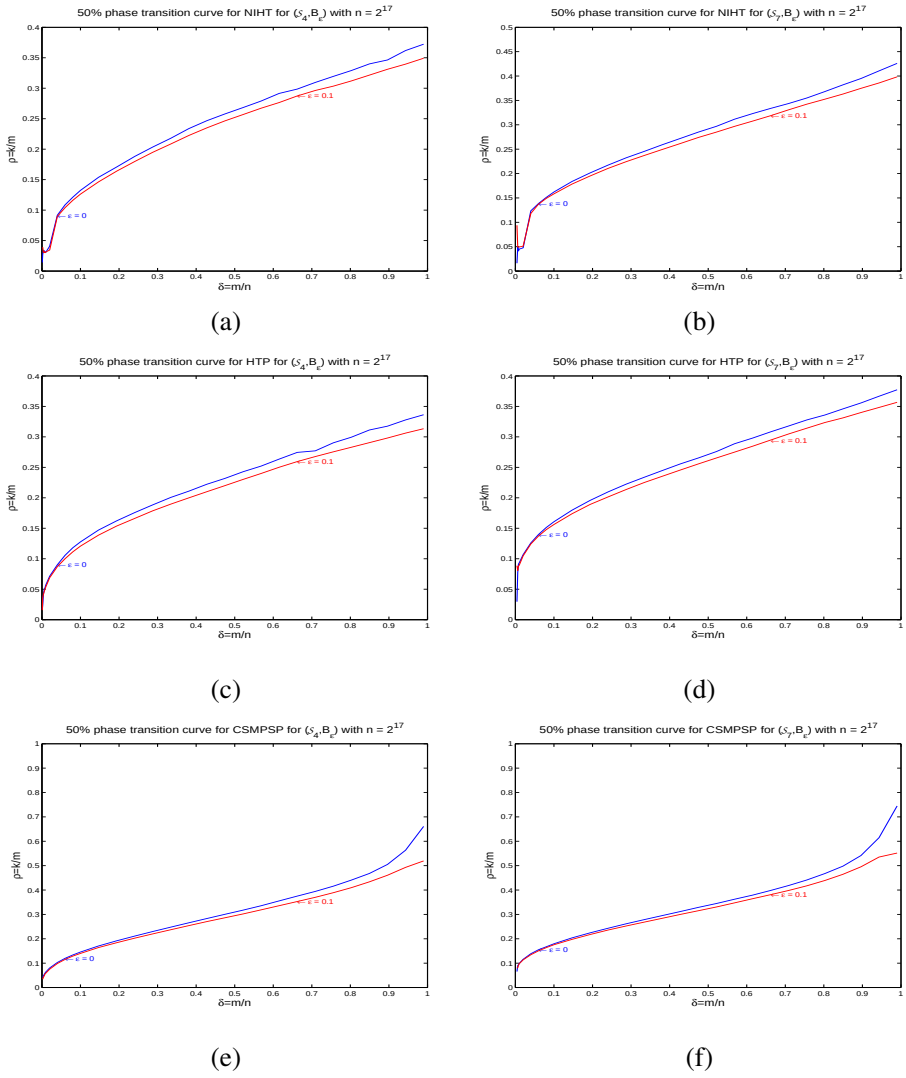



Figure 7. 50% recovery probability logistic regression curves for $n = 2^{17}$ and $vec = B_\epsilon$ with $\epsilon = 0$ and $\epsilon = 1/10$ for $Mat = \mathcal{S}_4$ in the left panels and for $Mat = \mathcal{S}_7$ in the right panels. Algorithms NIHT, HTP, and CSMPSP in Panels (a)-(b), (c)-(d), and (e)-(f) respectively.

*Observation 4.1.3*
For the problem class $(Mat, B_\epsilon)$ with $\epsilon = 1/10$ and $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$, NIHT is able to reliably successfully recover the measured vector in less time than HTP and CSMPSP for approximately $k/m \lessapprox \rho^{niht}_{(Mat,B_\epsilon)}(\delta)$ and essentially all of $\delta \in (0, 1)$. Moreover, NIHT is often 10 to 40 times faster than HTP and CSMPSP, and for $m/n < 1/2$ the recovery time for NIHT is never more than 2 times that of the fastest algorithm. For $\rho^{niht}_{(Mat,B_\epsilon)}(\delta) \lessapprox k/m \lessapprox \rho^{csmpsp}_{(Mat,B_\epsilon)}(\delta)$, CSMPSP is the only algorithm reliably able to successfully recover the measured vector and therefore has the lowest recovery time. For the problem class $(\mathcal{S}_7, B_\epsilon)$, as $m/n$ approaches zero HTP recovers the vector in the least time. (See Figs. 9–12.)

While the noise effects the magnitudes of the timings, the algorithms continue to exhibit a predictable, repeated behavior for successful recovery for a specific problem instance $(Mat, vec)$ and $(k, m, n)$. For example, considering the problem instance with $(k, m, n) = (10451, 75332, 262144)$, while the presence of noise in $(DCT, B_\epsilon)$ causes increases in time for recovery over $(DCT, B)$ for NIHT, HTP, and CSMPSP by factors of 1.4, 6.5, and 5.9, respectively,

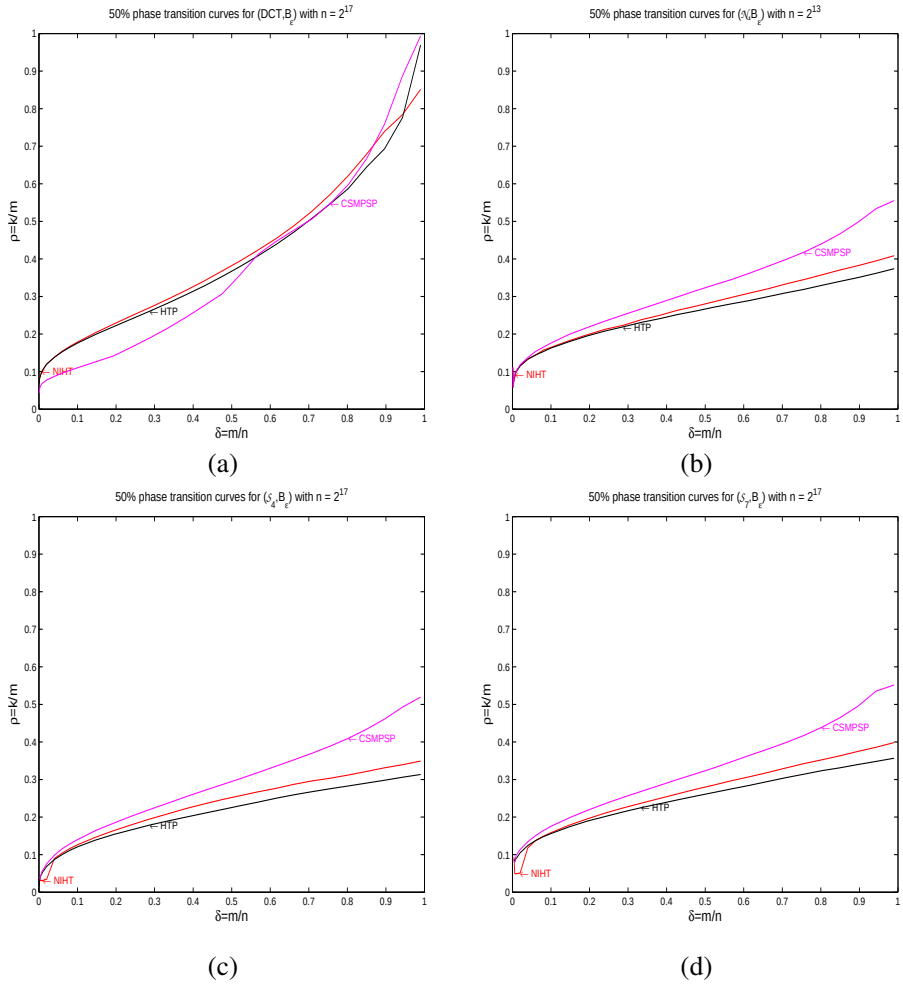



Figure 8. 50% recovery probability logistic regression curves for $vec = B_\epsilon$ with $\epsilon = 1/10$ and algorithms NIHT, HTP, and CSMPSP: (a) $Mat = DCT$ with $n = 2^{17}$, (b) $Mat = \mathcal{N}$ with $n = 2^{13}$, (c) $Mat = \mathcal{S}_4$ with $n = 2^{17}$, and (d) $Mat = \mathcal{S}_7$ with $n = 2^{17}$.

the recovery times have a variance less than 3.4% of the mean recovery time. Similar observations, such as iterations for the same problem instance, can be drawn from Tables. IV–VI. This is summarized in the following observation.

*Observation 4.1.4*
When successfully recovering a vector from the problem class $(Mat, B_\epsilon)$ with $\epsilon = 1/10$ and $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$, the following performance characteristics for each algorithm exhibit small variation about the mean: fraction of support set identified, time for recovery, iterations, and convergence rate. (See Tabs. IV–VI.)

### *4.2. Performance Analysis on Other Vector Distributions*

The empirical investigation now expands to include the sparse vector ensembles $\{B, U, N, B_\epsilon, U_\epsilon, N_\epsilon\}$. For this set of vector ensembles, an analysis similar to that performed

| $m$ | $k$ | Algorithm | succ/test | true pos fraction | time (ms) | iterations | convergence rate |
|---|---|---|---|---|---|---|---|
| 2622 | 141 | NIHT | 905/905 | 1(± 0) | 44.93(± 0.7) | 25.83(± 0.4) | 1(± 4e-08) |
| | | HTP | 896/896 | 1(± 0) | 458.1(± 1e+01) | 18.46(± 0.5) | 1(± 0) |
| | | CSMPSP | 899/899 | 0.9999(± 0.0007) | 462.2(± 1e+01) | 19.03(± 0.4) | 1(± 0) |
| | 211 | NIHT | 906/906 | 1(± 0) | 53.71(± 1) | 30.52(± 0.7) | 1(± 4e-08) |
| | | HTP | 900/900 | 1(± 0) | 496.3(± 5) | 19.99(± 0.2) | 1(± 0) |
| | | CSMPSP | 0/895 | 0.8388(± 0.04) | 616.4(± 7e+01) | 25.33(± 3) | 1(± 0) |
| | 253 | NIHT | 901/909 | 0.9961(± 0.04) | 68.24(± 8) | 38.39(± 4) | 1(± 4e-08) |
| | | HTP | 881/887 | 0.9971(± 0.04) | 560.9(± 5e+01) | 22.57(± 2) | 1(± 0) |
| | | CSMPSP | 0/899 | 0.6043(± 0.05) | 653.2(± 4e+01) | 26.83(± 2) | 1(± 0) |
| 15729 | 1239 | NIHT | 902/902 | 1(± 0) | 50.89(± 0.8) | 28.34(± 0.5) | 1(± 6e-09) |
| | | HTP | 909/909 | 1(± 0) | 486.9(± 0.2) | 19(± 0) | 1(± 0) |
| | | CSMPSP | 890/890 | 0.9987(± 0.001) | 495.9(± 1e+01) | 19.76(± 0.4) | 1(± 0) |
| | 1858 | NIHT | 885/885 | 1(± 0) | 63.18(± 0.6) | 34.88(± 0.4) | 1(± 2e-08) |
| | | HTP | 892/892 | 1(± 0) | 532.6(± 1e+01) | 20.75(± 0.4) | 1(± 0) |
| | | CSMPSP | 0/895 | 0.8415(± 0.01) | 521.3(± 9e+01) | 20.75(± 4) | 1(± 0) |
| | 2229 | NIHT | 997/997 | 1(± 0) | 80.65(± 2) | 43.93(± 1) | 1(± 3e-08) |
| | | HTP | 997/997 | 1(± 0) | 623.4(± 2e+01) | 24.27(± 0.8) | 1(± 0) |
| | | CSMPSP | 0/998 | 0.6641(± 0.02) | 571.2(± 8e+01) | 22.72(± 3) | 1(± 0) |
| 26215 | 2401 | NIHT | 994/994 | 1(± 0) | 53.73(± 0.8) | 29.24(± 0.4) | 1(± 0) |
| | | HTP | 996/996 | 1(± 0) | 498.5(± 0.2) | 19(± 0) | 1(± 0) |
| | | CSMPSP | 995/995 | 0.998(± 0.001) | 513.5(± 4) | 19.98(± 0.1) | 1(± 0) |
| | 3601 | NIHT | 999/999 | 1(± 0) | 67.87(± 0.9) | 36.48(± 0.5) | 1(± 1e-08) |
| | | HTP | 996/996 | 1(± 0) | 551.7(± 0.2) | 21(± 0) | 1(± 0) |
| | | CSMPSP | 0/994 | 0.8536(± 0.008) | 547.4(± 1e+02) | 21.28(± 4) | 1(± 5e-07) |
| | 4321 | NIHT | 992/992 | 1(± 0) | 88.77(± 2) | 46.89(± 1) | 1(± 1e-08) |
| | | HTP | 993/993 | 1(± 0) | 663.3(± 2e+01) | 25.22(± 0.8) | 1(± 0) |
| | | CSMPSP | 0/992 | 0.6897(± 0.01) | 583.8(± 2e+02) | 22.68(± 6) | 1(± 5e-07) |
| 75332 | 10451 | NIHT | 991/991 | 1(± 0) | 63.62(± 2) | 31.3(± 1) | 1(± 0) |
| | | HTP | 996/996 | 1(± 0) | 549.4(± 1e+01) | 19.02(± 0.4) | 1(± 0) |
| | | CSMPSP | 996/996 | 1(± 0) | 595.3(± 2e+01) | 21.03(± 0.7) | 1(± 0) |
| | 15676 | NIHT | 992/992 | 1(± 0) | 83.05(± 4) | 40.16(± 2) | 1(± 0) |
| | | HTP | 997/997 | 1(± 0) | 622(± 2e+02) | 21.5(± 6) | 1(± 0) |
| | | CSMPSP | 0/998 | 0.9188(± 0.003) | 1812(± 1e+03) | 63.94(± 5e+01) | 1(± 2e-07) |
| | 18811 | NIHT | 996/996 | 1(± 0) | 114.3(± 6) | 53.92(± 3) | 1(± 2e-09) |
| | | HTP | 995/995 | 1(± 0) | 788.1(± 2e+02) | 27.2(± 7) | 1(± 0) |
| | | CSMPSP | 0/991 | 0.7794(± 0.007) | 2207(± 1e+03) | 77.84(± 5e+01) | 1(± 3e-07) |
| 161288 | 37721 | NIHT | 997/997 | 1(± 0) | 212.6(± 4e+02) | 90.42(± 2e+02) | 1(± 0) |
| | | HTP | 998/998 | 1(± 0) | 1230(± 1e+03) | 36.68(± 4e+01) | 1(± 0) |
| | | CSMPSP | 995/995 | 1(± 0) | 1240(± 1e+03) | 37.66(± 4e+01) | 1(± 0) |
| | 56581 | NIHT | 995/995 | 1(± 0) | 328.2(± 5e+02) | 135.4(± 2e+02) | 1(± 0) |
| | | HTP | 992/992 | 1(± 0) | 1400(± 1e+03) | 41.65(± 4e+01) | 1(± 0) |
| | | CSMPSP | 992/992 | 1(± 2e-06) | 2559(± 2e+03) | 77.47(± 5e+01) | 1(± 0) |
| | 67897 | NIHT | 998/998 | 1(± 0) | 410.2(± 6e+02) | 166.4(± 2e+02) | 1(± 0) |
| | | HTP | 997/997 | 1(± 0) | 2040(± 2e+03) | 60.65(± 5e+01) | 1(± 0) |
| | | CSMPSP | 996/996 | 0.9997(± 0.0001) | 3134(± 1e+03) | 94.56(± 4e+01) | 1(± 3e-06) |

Table IV. Performance of NIHT, HTP, and CSMPSP for $(DCT, B_\epsilon)$ with n=262144

| $m$ | $k$ | Algorithm | succ/test | true pos fraction | time (ms) | iterations | convergence rate |
|---|---|---|---|---|---|---|---|
| 246 | 19 | NIHT | 892/892 | 1(± 0) | 21.44(± 1) | 27.75(± 1) | 1(± 3e-08) |
| | | HTP | 901/901 | 1(± 0) | 162.2(± 5) | 18.42(± 0.6) | 1(± 0) |
| | | CSMPSP | 892/892 | 1(± 0) | 166.4(± 1e+01) | 19.3(± 2) | 1(± 0) |
| | 28 | NIHT | 899/909 | 0.9947(± 0.05) | 28.03(± 2e+01) | 35.64(± 2e+01) | 1(± 3e-08) |
| | | HTP | 877/897 | 0.9904(± 0.07) | 185.6(± 3e+01) | 21.05(± 3) | 1(± 0) |
| | | CSMPSP | 894/896 | 0.9988(± 0.02) | 194.4(± 6e+01) | 22.53(± 7) | 1(± 5e-07) |
| | 34 | NIHT | 554/902 | 0.8094(± 0.3) | 48.63(± 8e+01) | 58.88(± 1e+02) | 1(± 1e-05) |
| | | HTP | 570/907 | 0.8266(± 0.2) | 233.8(± 7e+01) | 26.41(± 7) | 1(± 0) |
| | | CSMPSP | 706/890 | 0.9014(± 0.2) | 336.1(± 3e+02) | 38.95(± 3e+01) | 1(± 6e-06) |
| 410 | 37 | NIHT | 900/900 | 1(± 0) | 28.82(± 1) | 29.46(± 1) | 1(± 2e-08) |
| | | HTP | 908/908 | 1(± 0) | 229.2(± 6) | 18.78(± 0.5) | 1(± 0) |
| | | CSMPSP | 882/882 | 1(± 0) | 244.3(± 7e+01) | 20.46(± 6) | 1(± 9e-08) |
| | 56 | NIHT | 875/887 | 0.9944(± 0.05) | 39.26(± 5) | 39.54(± 4) | 1(± 3e-08) |
| | | HTP | 861/878 | 0.9928(± 0.05) | 273.3(± 4e+01) | 22.37(± 4) | 1(± 0) |
| | | CSMPSP | 910/910 | 1(± 0) | 348.1(± 2e+02) | 29.14(± 2e+01) | 1(± 5e-06) |
| | 67 | NIHT | 425/886 | 0.7834(± 0.2) | 54.31(± 4e+01) | 52.59(± 3e+01) | 1(± 3e-08) |
| | | HTP | 376/867 | 0.7653(± 0.2) | 367.6(± 1e+02) | 29.99(± 8) | 1(± 0) |
| | | CSMPSP | 742/896 | 0.9265(± 0.2) | 1014(± 6e+02) | 84.81(± 5e+01) | 1(± 2e-05) |
| 1178 | 153 | NIHT | 886/886 | 1(± 0) | 68.01(± 2) | 34.1(± 0.8) | 1(± 2e-08) |
| | | HTP | 895/895 | 1(± 0) | 562.8(± 1e+01) | 19.42(± 0.5) | 1(± 0) |
| | | CSMPSP | 885/885 | 1(± 0) | 1098(± 1e+03) | 38.74(± 3e+01) | 1(± 3e-06) |
| | 230 | NIHT | 877/881 | 0.9987(± 0.02) | 102.8(± 9) | 51.2(± 5) | 1(± 3e-08) |
| | | HTP | 890/898 | 0.9976(± 0.03) | 715.8(± 9e+01) | 24.69(± 3) | 1(± 0) |
| | | CSMPSP | 893/893 | 1(± 0) | 3080(± 1e+03) | 108.6(± 4e+01) | 1(± 2e-05) |
| | 275 | NIHT | 135/904 | 0.7259(± 0.1) | 142.3(± 3e+01) | 69.12(± 2e+01) | 1(± 2e-08) |
| | | HTP | 51/897 | 0.6837(± 0.1) | 917.7(± 3e+02) | 31.6(± 9) | 1(± 0) |
| | | CSMPSP | 871/899 | 0.9902(± 0.06) | 3561(± 3e+02) | 125.5(± 1e+01) | 1(± 5e-05) |
| 2521 | 477 | NIHT | 897/897 | 1(± 0) | 157.1(± 3) | 41.78(± 0.9) | 1(± 3e-08) |
| | | HTP | 891/891 | 1(± 0) | 1188(± 3e+01) | 20.43(± 0.5) | 1(± 0) |
| | | CSMPSP | 901/901 | 1(± 0) | 6969(± 1e+03) | 122.5(± 2e+01) | 1(± 1e-05) |
| | 716 | NIHT | 887/904 | 0.9965(± 0.03) | 343.5(± 6e+01) | 90.94(± 1e+01) | 1(± 2e-08) |
| | | HTP | 638/910 | 0.9434(± 0.09) | 2257(± 5e+02) | 38.81(± 9) | 1(± 0) |
| | | CSMPSP | 899/899 | 1(± 0) | 7210(± 3e+02) | 126.8(± 5) | 1(± 7e-05) |
| | 859 | NIHT | 1/880 | 0.7468(± 0.03) | 404.8(± 8e+01) | 106.1(± 2e+01) | 1(± 2e-08) |
| | | HTP | 0/922 | 0.6984(± 0.02) | 1941(± 5e+02) | 33.37(± 9) | 1(± 0) |
| | | CSMPSP | 882/893 | 0.9974(± 0.02) | 7225(± 4e+01) | 127(± 0.7) | 0.9999(± 0.0002) |

Table V. Performance of NIHT, HTP, and CSMPSP for $(\mathcal{N}, B_\epsilon)$ with n=4096

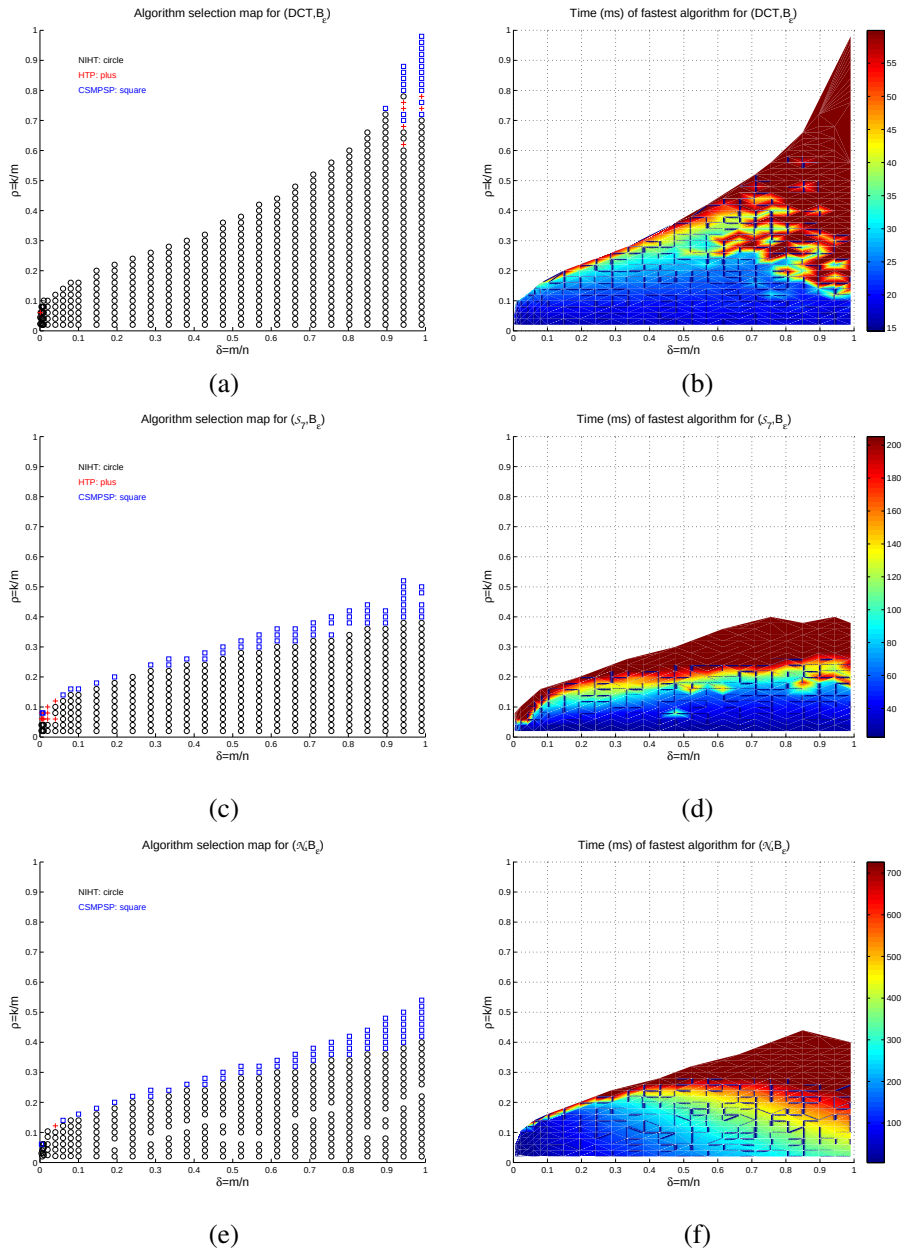



Figure 9. Left panels: Algorithm selection maps with moderate noise $\epsilon = 1/10$. Right panels: Average time for the fastest algorithm. Panels (a-b) $(DCT, B_\epsilon)$ with $n = 2^{17}$, (c-d) $(\mathcal{S}_7, B_\epsilon)$ with $n = 2^{17}$, and (e-f) $(\mathcal{N}, B_\epsilon)$ with $n = 2^{13}$.

in Secs. 3 and 4.1 will produce recovery phase transitions, algorithm selection maps, ratios of recovery time to the fastest algorithm, and information about the concentration of performance characteristics for the algorithms. The total combinations result in more information than can be readily presented in a single article. Here we present seven observations which highlight several findings which may be significant when selecting a recovery algorithm.

The recovery regions of NIHT, HTP, and CSMPSP for the problem classes $(Mat, vec)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ and $vec \in \{U, N, U_\epsilon, N_\epsilon\}$ are identified in Obs. 4.2.1–4.2.3 and App. C. Figures 26–29 identify the recovery regions; similar relationships between the recovery regions for

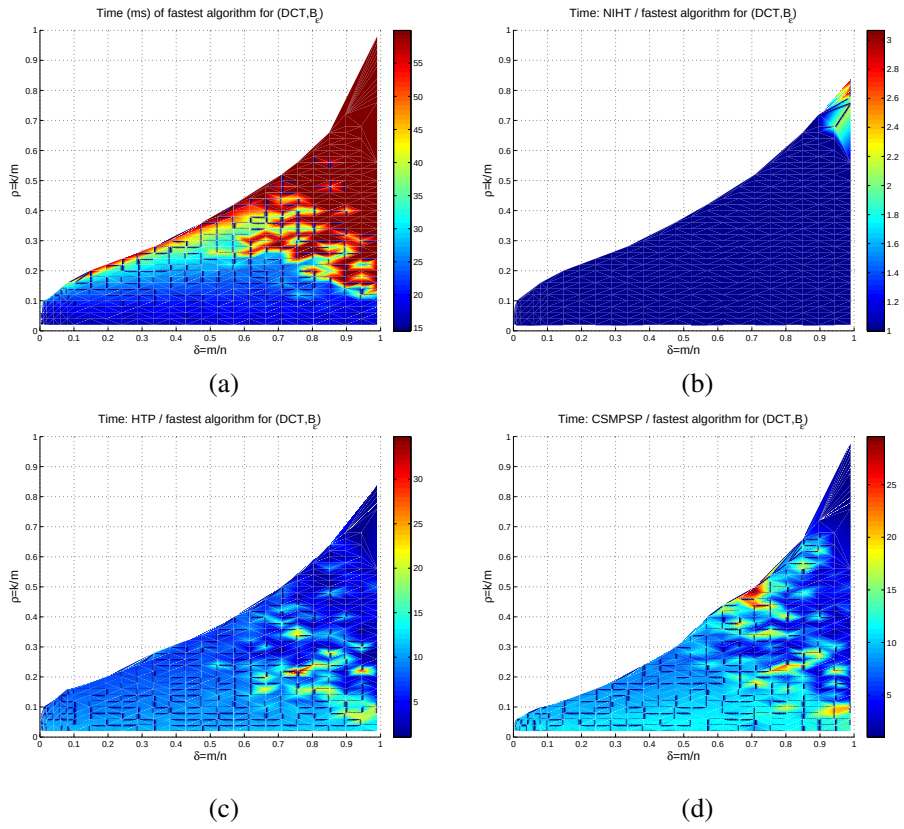



Figure 10. Panel (a), time for fastest algorithm for $(DCT, B_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{17}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

the algorithms to those depicted in Fig. 1 persist for $vec \in \{U, N, U_\epsilon, N_\epsilon\}$. Notable exceptions are identified by the following observations.

*Observation 4.2.1*
For problem classes $(Mat, U)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_7, DCT\}$, and $\delta \in [1/20, 1/2]$, the recovery phase transitions for NIHT and HTP have phase transition curves within $0.02$ of each other. For $(DCT, U)$ CSMPSP has the lowest PT curve but has the highest phase transition curve for $(\mathcal{N}, U)$ and $(\mathcal{S}_p, U)$. (See Fig. 26.)

*Observation 4.2.2*
For moderate noise $\epsilon = 1/10$ and problem classes $(Mat, U_\epsilon)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_7, DCT\}$, and $\delta \in [1/20, 1/2]$, the recovery phase transitions for NIHT and HTP have phase transition curves within $0.02$ of each other. For $(DCT, U_\epsilon)$ the phase transition curves for all algorithms have similar relationships to those in Obs. 4.2.1. Moreover, for $(\mathcal{N}, U_\epsilon)$ and $(\mathcal{S}_7, U_\epsilon)$ and $\delta \in [1/20, 1/4]$, the recovery phase transitions for NIHT, HTP, and CSMPSP have phase transition curves within $0.02$ of each other, but as $\delta$ exceeds approximately $1/2$ the recovery phase transition for CSMPSP increasingly falls below those of NIHT and HTP. (See Fig. 27.)

*Observation 4.2.3*
For problem classes $(Mat, N)$ and $(Mat, N_\epsilon)$, the recovery phase transition curves for the algorithms have qualitatively similar behavior to the associated phase transition curves for vector ensembles $U$ and $U_\epsilon$ as detailed in Obs. 4.2.1 and 4.2.2. The notable exceptions are

(i) $\rho^{niht}_{(\mathcal{N},N)}(\delta)$ which suffers from a large phase transition region when both the matrix ensemble and vector ensemble are normally distributed (Obs. C.1);

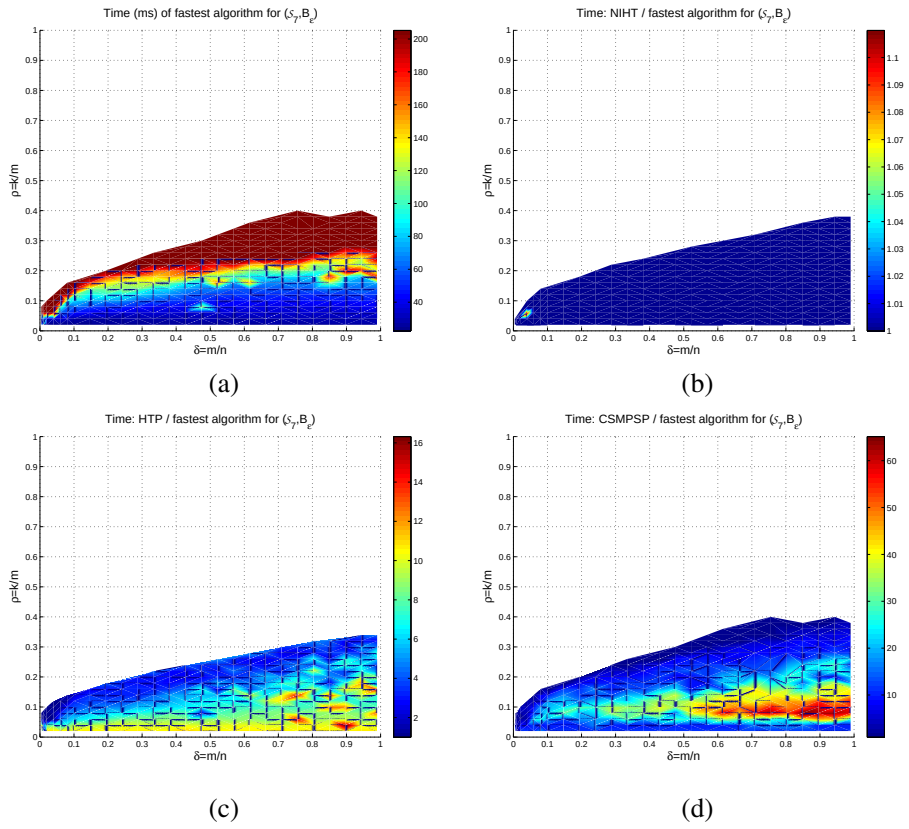



Figure 11. Panel (a), time for fastest algorithm for $(\mathcal{S}_7, B_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{17}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

(ii) HTP has a noticeably higher phase transition curve than NIHT for problem classes $(Mat, N)$, $(DCT, N_\epsilon)$, and $(\mathcal{N}, N)$. (See Fig. 28–29.)

Important information omitted from the recovery phase transitions is encapsulated in identifying which algorithm boasts the smallest time for successful recovery. Summary information regarding algorithm selection is given by Obs. 4.2.4, 4.2.5, and 4.2.6 for each matrix ensemble $DCT$, $\mathcal{S}_7$, and $\mathcal{N}$, respectively. The algorithm selection maps appear in Figs. 13 and 14 while the ratio of algorithm times to the fastest algorithm appear in Figs. 30–35.

*Observation 4.2.4*
For the problem classes $(DCT, U_\epsilon)$ or $(DCT, N_\epsilon)$ with $\epsilon = 1/10$, NIHT reliably recovers the measured vector in the least amount of time for $k/m \lessapprox \rho^{niht}_{(DCT,vec_\epsilon)}(\delta)$ with $\delta \lessapprox \frac{1}{2}$. In this region, HTP and CSMPSP have mean recovery times 5 to 10 times larger than the average recovery time for NIHT. For $k/m \lessapprox \rho^{niht}_{(DCT,vec_\epsilon)}(\delta)$ with $\delta \in (0,1)$, NIHT never requires more than 3 times the fastest recovery time. (See Figs. 13–14, 30, and 33.)

*Observation 4.2.5*
For the problem classes $(\mathcal{S}_7, U_\epsilon)$ or $(\mathcal{S}_7, N_\epsilon)$ with $\epsilon = 1/10$, HTP reliably recovers the measured vector in the least amount of time for $k/m \lessapprox \rho^{htp}_{(\mathcal{S}_7,vec_\epsilon)}(\delta)$ with $\delta \lessapprox 0.1$. NIHT reliably recovers the measured vector in the least time for $\delta \gtrapprox 0.1$ and never requires more than 2 times the fastest recovery time for $\delta \in (0,1)$. (See Figs. 13–14, 31, and 34.)

*Observation 4.2.6*
For the problem classes $(\mathcal{N}, U_\epsilon)$ or $(\mathcal{N}, N_\epsilon)$ with $\epsilon = 1/10$, NIHT reliably recovers the measured vector in less time than HTP or CSMPSP for $k/m \lessapprox \rho^{niht}_{(\mathcal{N},vec_\epsilon)}(\delta)$ with $.002 \lessapprox \delta \lessapprox 1$. As $k/m$

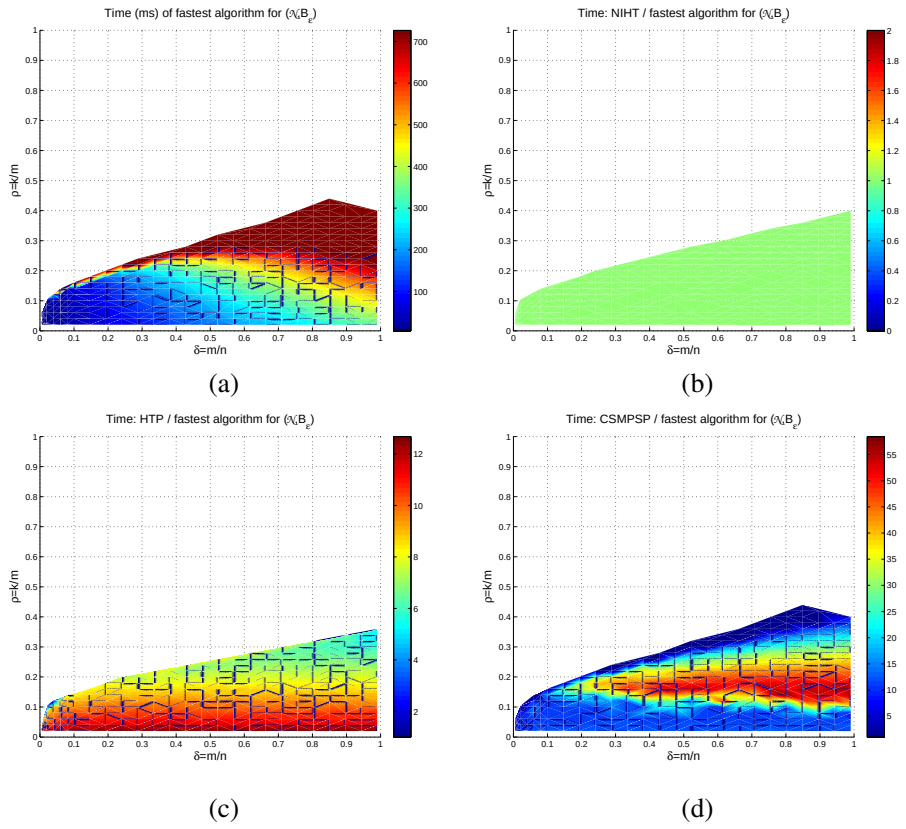



Figure 12. Panel (a), time for fastest algorithm for $(\mathcal{N}, B_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{13}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

increases to $\rho^{htp}_{(\mathcal{N},vec_\epsilon)}(\delta)$, HTP has the lowest recovery time. In the most extreme undersampling situation, $\delta \lessapprox .002$, CSMPSP recovers the vector in the least time. For $k/m \lessapprox \rho^{niht}_{(\mathcal{N},vec_\epsilon)}(\delta)$ with $\delta \in (0, 1)$, NIHT never requires more than 1.4 times the fastest recovery time. (See Figs. 13–14, 32, and 35.)

Observations 4.2.4–4.2.6 coupled with Obs. 4.1.3 provide convincing evidence that when recovering vectors from measurements contaminated with a moderate level of noise, NIHT is the algorithm of choice. Restricting attention to the region of the phase space with at least two times undersampling, $m/n < 1/2$, NIHT generally recovers a vector from noise contaminated measurements faster than the other two algorithms. Importantly, the exceptions to this rule are clearly identified in Obs. 4.2.4–4.2.6 and 4.1.3.

*Observation 4.2.7*
Across all problem classes $(Mat, vec)$, the recovery phase transitions for the the sparse vector ensemble $B$ is typically lower than for $U$ and $N$ (See Figs. 15 and 36-38); specifically:

(i) For NIHT, HTP, and CSMPSP, $\rho^{alg}_{(Mat,U)}(\delta) > \rho^{alg}_{(Mat,B)}(\delta)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$;

(ii) For NIHT, HTP, and CSMPSP, $\rho^{alg}_{(Mat,N)}(\delta) > \rho^{alg}_{(Mat,B)}(\delta)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ and with the lone exception of $\rho^{niht}_{(\mathcal{N},N)}(\delta) < \rho^{niht}_{(Mat,B)}(\delta)$ as remarked in Obs. C.1;

(iii) For NIHT and HTP, $alg \in \{\text{NIHT,HTP}\}$, $\rho^{alg}_{(Mat,U_\epsilon)}(\delta) > \rho^{alg}_{(Mat,B_\epsilon)}(\delta)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ and for CSMPSP with $\delta < \frac{1}{2}$;

| $m$ | $k$ | Algorithm | succ/test | true pos fraction | time (ms) | iterations | convergence rate |
|---|---|---|---|---|---|---|---|
| 656 | 32 | NIHT | 488/510 | 0.9708(± 0.1) | 50.52(± 1e+02) | 50.63(± 1e+02) | 0.9843(± 0.1) |
| | | HTP | 509/514 | 0.9903(± 0.1) | 228.3(± 2e+01) | 18.76(± 2) | 0.9903(± 0.1) |
| | | CSMPSP | 496/500 | 0.992(± 0.09) | 257(± 1e+02) | 21.43(± 1e+01) | 0.992(± 0.09) |
| | 48 | NIHT | 0/496 | 0.3603(± 0.09) | 764(± 1e+02) | 739.8(± 9e+01) | 0.9839(± 0.1) |
| | | HTP | 472/478 | 0.9885(± 0.1) | 261.5(± 3e+01) | 21.53(± 3) | 0.9895(± 0.1) |
| | | CSMPSP | 511/520 | 0.9827(± 0.1) | 371.9(± 3e+02) | 31.2(± 3e+01) | 0.9827(± 0.1) |
| | 58 | NIHT | 0/484 | 0.2509(± 0.07) | 759.8(± 1e+02) | 736.5(± 1e+02) | 0.9793(± 0.1) |
| | | HTP | 314/497 | 0.8389(± 0.2) | 319.1(± 5e+01) | 26.25(± 4) | 0.9879(± 0.1) |
| | | CSMPSP | 396/475 | 0.9141(± 0.2) | 577.9(± 5e+02) | 48.65(± 4e+01) | 0.9789(± 0.1) |
| 3933 | 305 | NIHT | 851/886 | 0.9789(± 0.1) | 105(± 1e+02) | 103.4(± 1e+02) | 0.9865(± 0.1) |
| | | HTP | 899/913 | 0.9847(± 0.1) | 249.8(± 3e+01) | 20.73(± 3) | 0.9847(± 0.1) |
| | | CSMPSP | 855/872 | 0.9804(± 0.1) | 742.6(± 6e+02) | 61.13(± 5e+01) | 0.9805(± 0.1) |
| | 458 | NIHT | 769/910 | 0.9359(± 0.2) | 214.3(± 2e+02) | 206.7(± 2e+02) | 0.9901(± 0.1) |
| | | HTP | 876/885 | 0.9898(± 0.1) | 292.5(± 3e+01) | 24.03(± 3) | 0.9898(± 0.1) |
| | | CSMPSP | 895/904 | 0.99(± 0.1) | 1387(± 4e+02) | 112.5(± 4e+01) | 0.99(± 0.1) |
| | 549 | NIHT | 219/895 | 0.6919(± 0.2) | 217(± 2e+02) | 205.4(± 2e+02) | 0.981(± 0.1) |
| | | HTP | 178/898 | 0.7125(± 0.2) | 397.9(± 1e+02) | 32.4(± 1e+01) | 0.9844(± 0.1) |
| | | CSMPSP | 863/874 | 0.9888(± 0.1) | 1499(± 3e+02) | 120.4(± 3e+01) | 0.9897(± 0.1) |
| 6554 | 589 | NIHT | 894/894 | 1(± 0) | 59.6(± 1e+01) | 60.34(± 1e+01) | 1(± 2e-09) |
| | | HTP | 883/883 | 1(± 0) | 245.3(± 9) | 20.66(± 0.7) | 1(± 0) |
| | | CSMPSP | 878/878 | 1(± 0) | 939.9(± 6e+02) | 77.47(± 5e+01) | 1(± 1e-05) |
| | 883 | NIHT | 877/896 | 0.993(± 0.05) | 133.7(± 1e+02) | 130.3(± 1e+02) | 1(± 2e-05) |
| | | HTP | 898/898 | 1(± 0) | 302(± 1e+01) | 25(± 1) | 1(± 0) |
| | | CSMPSP | 889/898 | 0.9901(± 0.1) | 1466(± 4e+02) | 118.3(± 3e+01) | 1(± 2e-05) |
| | 1060 | NIHT | 139/902 | 0.6925(± 0.2) | 169.4(± 1e+02) | 159.2(± 1e+02) | 1(± 3e-05) |
| | | HTP | 37/903 | 0.6625(± 0.1) | 398.8(± 2e+02) | 32.53(± 1e+01) | 0.9779(± 0.1) |
| | | CSMPSP | 899/899 | 0.9999(± 0.0003) | 1543(± 2e+02) | 123.6(± 2e+01) | 1(± 2e-05) |
| 18833 | 2460 | NIHT | 996/996 | 1(± 0) | 71.12(± 8) | 67.39(± 7) | 1(± 0) |
| | | HTP | 990/990 | 1(± 0) | 280.9(± 6) | 21.72(± 0.5) | 1(± 0) |
| | | CSMPSP | 995/995 | 1(± 0) | 1513(± 4e+02) | 116.3(± 3e+01) | 1(± 2e-06) |
| | 3690 | NIHT | 996/996 | 1(± 0) | 131(± 2e+01) | 120.3(± 2e+01) | 1(± 4e-09) |
| | | HTP | 994/994 | 1(± 0) | 386.2(± 1e+01) | 29.53(± 1) | 1(± 0) |
| | | CSMPSP | 991/995 | 0.9964(± 0.06) | 1507(± 6e+02) | 107(± 4e+01) | 1(± 9e-06) |
| | 4428 | NIHT | 1/998 | 0.674(± 0.1) | 258.9(± 2e+02) | 227.6(± 1e+02) | 1(± 0) |
| | | HTP | 0/998 | 0.6419(± 0.06) | 948.8(± 6e+02) | 71.91(± 4e+01) | 0.991(± 0.09) |
| | | CSMPSP | 996/999 | 0.9984(± 0.03) | 1761(± 4e+02) | 118.7(± 3e+01) | 1(± 2e-05) |
| 40322 | 7656 | NIHT | 997/997 | 1(± 0) | 81.28(± 1e+01) | 72.41(± 1e+01) | 1(± 0) |
| | | HTP | 995/995 | 1(± 0) | 333.5(± 1e+02) | 23.84(± 8) | 1(± 0) |
| | | CSMPSP | 998/998 | 1(± 0) | 1774(± 7e+02) | 104.9(± 4e+01) | 1(± 4e-06) |
| | 11484 | NIHT | 993/993 | 1(± 9e-06) | 250.7(± 3e+01) | 194.3(± 2e+01) | 1(± 7e-09) |
| | | HTP | 498/996 | 0.9194(± 0.08) | 1449(± 5e+02) | 91.27(± 3e+01) | 1(± 5e-05) |
| | | CSMPSP | 993/993 | 1(± 0.0001) | 2282(± 7e+02) | 114.8(± 3e+01) | 1(± 2e-05) |
| | 13780 | NIHT | 0/995 | 0.7397(± 0.02) | 500.3(± 2e+02) | 354.2(± 2e+02) | 1(± 0) |
| | | HTP | 0/997 | 0.6944(± 0.009) | 1541(± 7e+02) | 90.54(± 4e+01) | 1(± 0) |
| | | CSMPSP | 993/995 | 1(± 0.0002) | 2710(± 3e+02) | 125.5(± 1e+01) | 1(± 4e-05) |

Table VI. Performance of NIHT, HTP, and CSMPSP for $(\mathcal{S}_7, B_\epsilon)$ with n=65536

(iv) For NIHT and HTP, $\rho^{alg}_{(Mat,N_\epsilon)}(\delta) > \rho^{alg}_{(Mat,B_\epsilon)}(\delta)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ and for CSMPSP with $\delta < \frac{1}{2}$.

There have been several cases [32, 33, 34] where it has been shown that of all problem classes $(\mathcal{N}, vec)$, $vec = B$ is the most challenging vector ensemble for greedy algorithms. Observation 4.2.7 supports these claims while extending them to the matrix ensembles $\mathcal{S}_p$ and $DCT$ and to the setting of noisy measurements $vec \in \{B_\epsilon, U_\epsilon, N_\epsilon\}$. Figure 15 depicts this behavior with matrix ensemble $\mathcal{S}_7$ where the recovery phase transition curves for NIHT, HTP, and CSMPSP are plotted for $vec$ taking all values $\{B, U, N\}$ or $\{B_\epsilon, U_\epsilon, N_\epsilon\}$ for each algorithm. Figures 36-38 depict this behavior for the remaining matrix ensembles.

## 5. CONCLUSION

In compressed sensing one seeks to identify both a measurement process of a $k$ sparse vector that requires very few measurements and a computationally efficient recovery algorithm which will identify the measured vector. In this paper, previous empirical studies identifying the recovery regions for a particular problem class have been extended to include alternate vector ensembles and the effects of noise. As the problem instances approach the recovery phase transition curve, the convergence of the iterative algorithms can be arbitrarily slow. Therefore, a practitioner needs to identify the number of measurements required to push the ratio $k/m$ into the algorithms' recovery regions. Often, this results in a problem size where multiple algorithms will reliably return the correct solution. This paper focuses on distinguishing the performance of the algorithms,

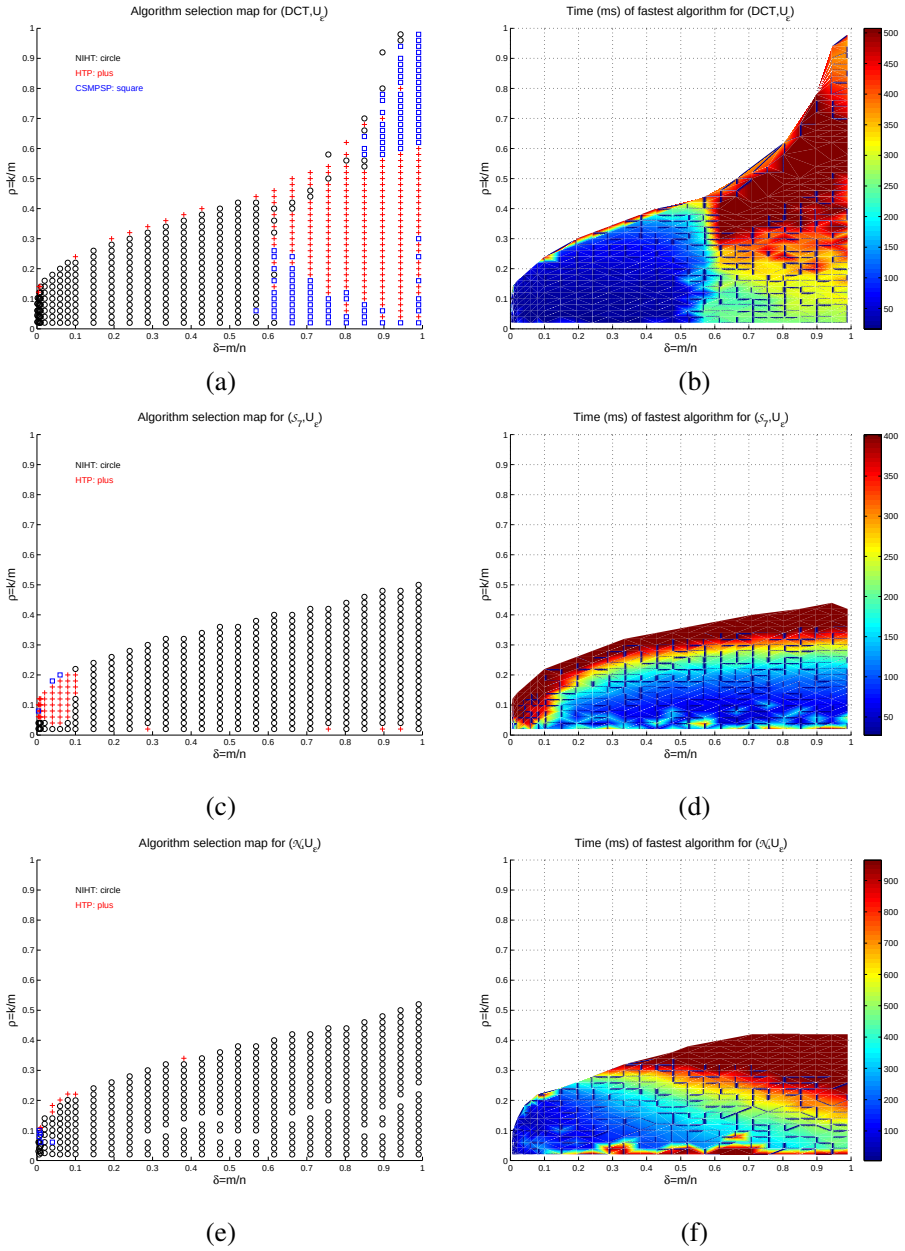



Figure 13. Left panels: Algorithm selection maps with moderate noise $\epsilon = 1/10$. Right panels: Average time for the fastest algorithm. Panels (a-b) $(DCT, U_\epsilon)$ with $n = 2^{17}$, (c-d) $(\mathcal{S}_7, U_\epsilon)$ with $n = 2^{17}$, and (e-f) $(\mathcal{N}, U_\epsilon)$ with $n = 2^{13}$.

particularly in the regions where multiple algorithms typically succeed. The algorithm selection maps included in this paper inform practitioners and theorists of the algorithm that is able to reliably recover the measured vector in the least computation time. Furthermore, the concentration of performance characteristics for each algorithm tells practitioners that an algorithm will have consistent, predictable behavior and provides a foundation for the observations and claims made by this article.

This manuscript is quantitative where precise statements provide valuable information and is qualitative for statements where further precision is of limited value. Highly quantitative statements can be readily obtained by performing a large number of tests for a given problem instance, and

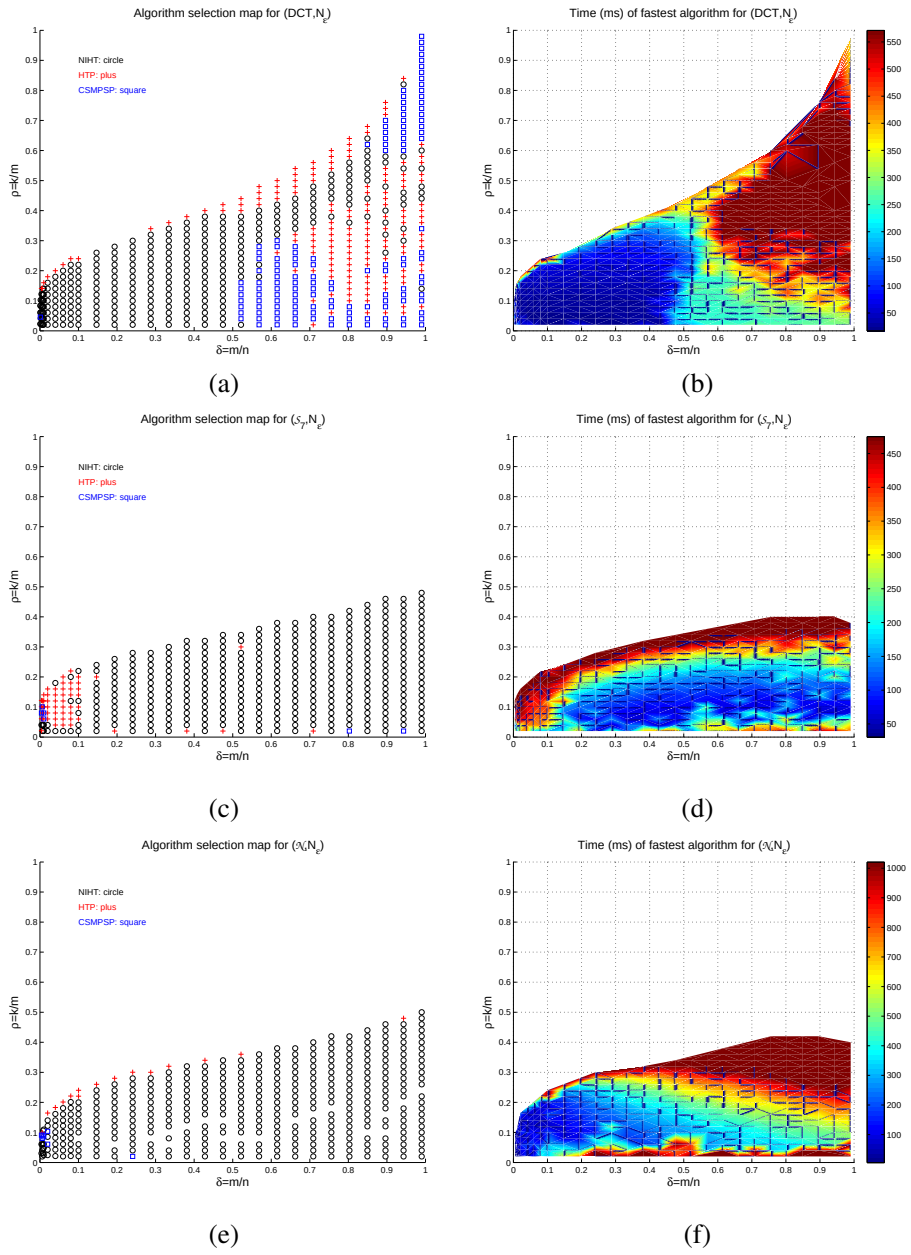



Figure 14. Left panels: Algorithm selection maps with moderate noise $\epsilon = 1/10$. Right panels: Average time for the fastest algorithm. Panels (a-b) $(DCT, N_\epsilon)$ with $n = 2^{17}$, (c-d) $(\mathcal{S}_7, N_\epsilon)$ with $n = 2^{17}$, and (e-f) $(\mathcal{N}, N_\epsilon)$ with $n = 2^{13}$.

the software, GAGA [36], is designed to handle such large-scale testing. However, many of the most interesting observations in this paper would not be substantially more informative when made more quantitative. The algorithm selection maps identify the fastest algorithm and the associated recovery time ratios provide a general idea of how each algorithm compares to the fastest; a precise identification of this ratio is of secondary importance. Claims 1–6 provide a general summary of the substantial amount of data contained in this manuscript. These claims are informed by the more specific observations from Secs. 3–4 and Apps. A–F. Conducting additional performance comparisons for other algorithms and formally testing hypotheses regarding algorithm behavior can be readily accomplished using GAGA [35, 36] and is slated for future work.

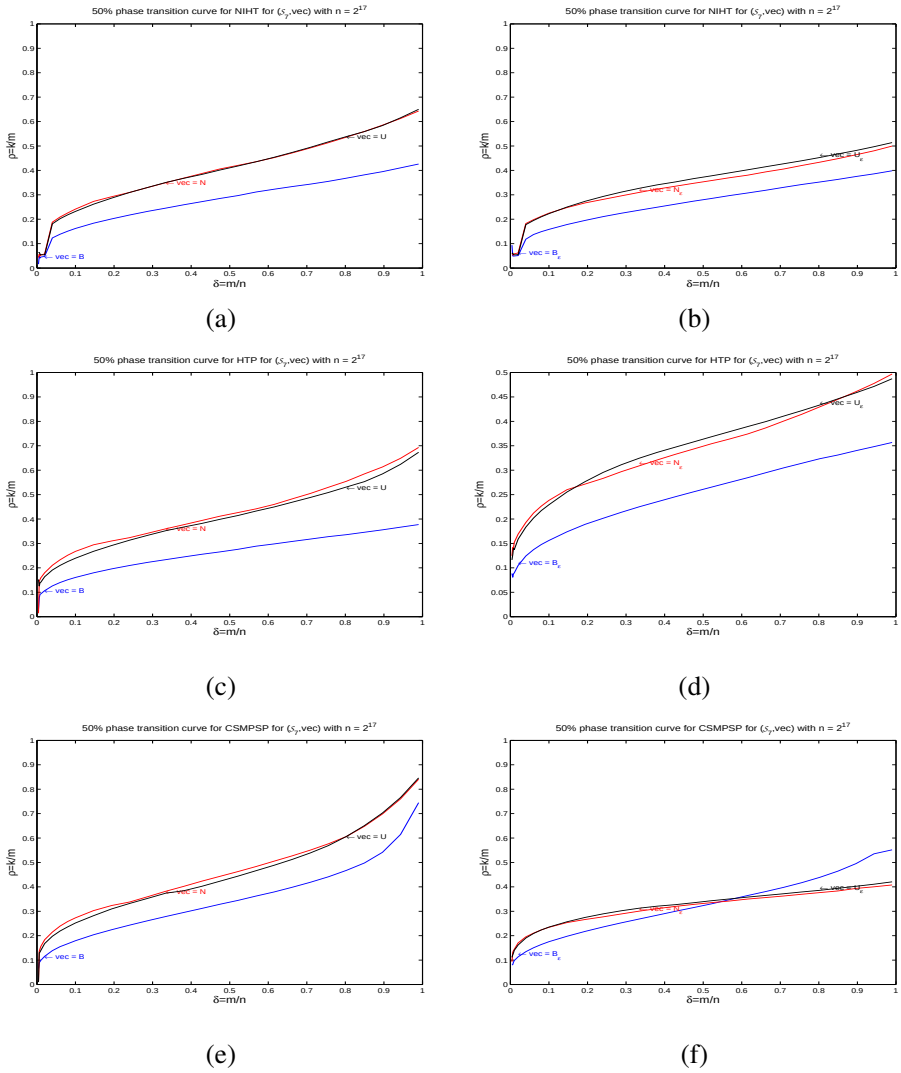



Figure 15. 50% recovery probability logistic regression curves for $Mat = \mathcal{S}_7$ with $n = 2^{17}$ and for each $vec \in \{B, U, N\}$ for (a) NIHT, (c) HTP, and (e) CSMPSP, and for each $vec \in \{B_\epsilon, U_\epsilon, N_\epsilon\}$ with $\epsilon = 1/10$ for (b) NIHT, (e) HTP, and (f) CSMPSP.

ACKNOWLEDGEMENTS

JDB was supported by NSF DMS 1112612 and NSF OISE 0854991. JT was supported by the Leverhulm Trust and an Nvidia Professor Partnership.

## A. CONCENTRATION OF RECOVERY PHASE TRANSITIONS FOR $(Mat, B)$

Sec. 3.1 discusses the recovery phase transitions of the problem classes $(Mat, B)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ with $p = 4$ and 7, but only presents data for a single value of $n$ per problem class. This appendix includes further data on the recovery phase transition curves for this problem class at smaller values of $n$ and including the algorithm IHT. Along with presenting the 50% recovery phase transitions in the left panels of each figure, the right panels include the gap between the 10% and 90% recovery curves of the logistic regressions. The plots show quantitative convergence of the recovery phase transitions toward those shown in Fig. 1.

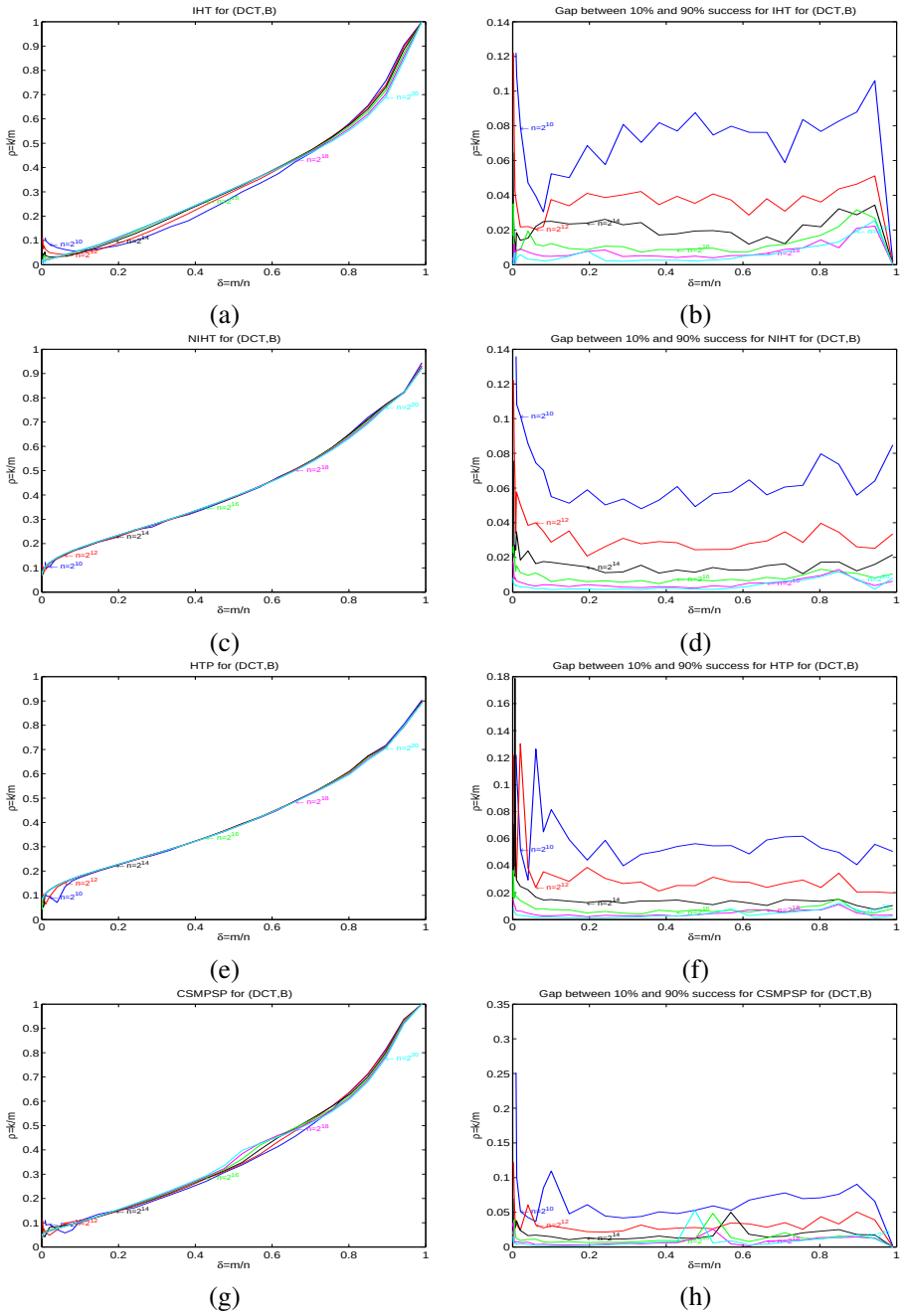



Figure 16. Left panels: 50% recovery probability logistic regression curves for $(DCT, B)$ and $n = 2^j$ with $j = 10, 12, 14, 16, 18, 20$. Right panels: gap between 10% and 90% recovery probability curves. Results shown for the algorithms: IHT (a-b), NIHT (c-d), HTP (e-f), and CSMPSP (g-h).

## B. ALGORITHM SELECTION MAPS FOR $(Mat, B)$ AT SMALLER VALUES OF $n$

Sec. 3.2 presents algorithm selection maps and average recovery times for the problem classes $(Mat, B)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ with $p = 7$, but only presents data for a single value of $n$ per problem class. This appendix includes similar algorithm selection maps for smaller values of $n$, showing that the selection maps are consistent for large $n$. Algorithm selection maps and average recovery times are also presented here for the problem class $(\mathcal{S}_4, B)$, showing that the selection maps for $(\mathcal{S}_p, B)$ are consistent for smaller values of $p$. Larger values of $p$ are considered in App. F.

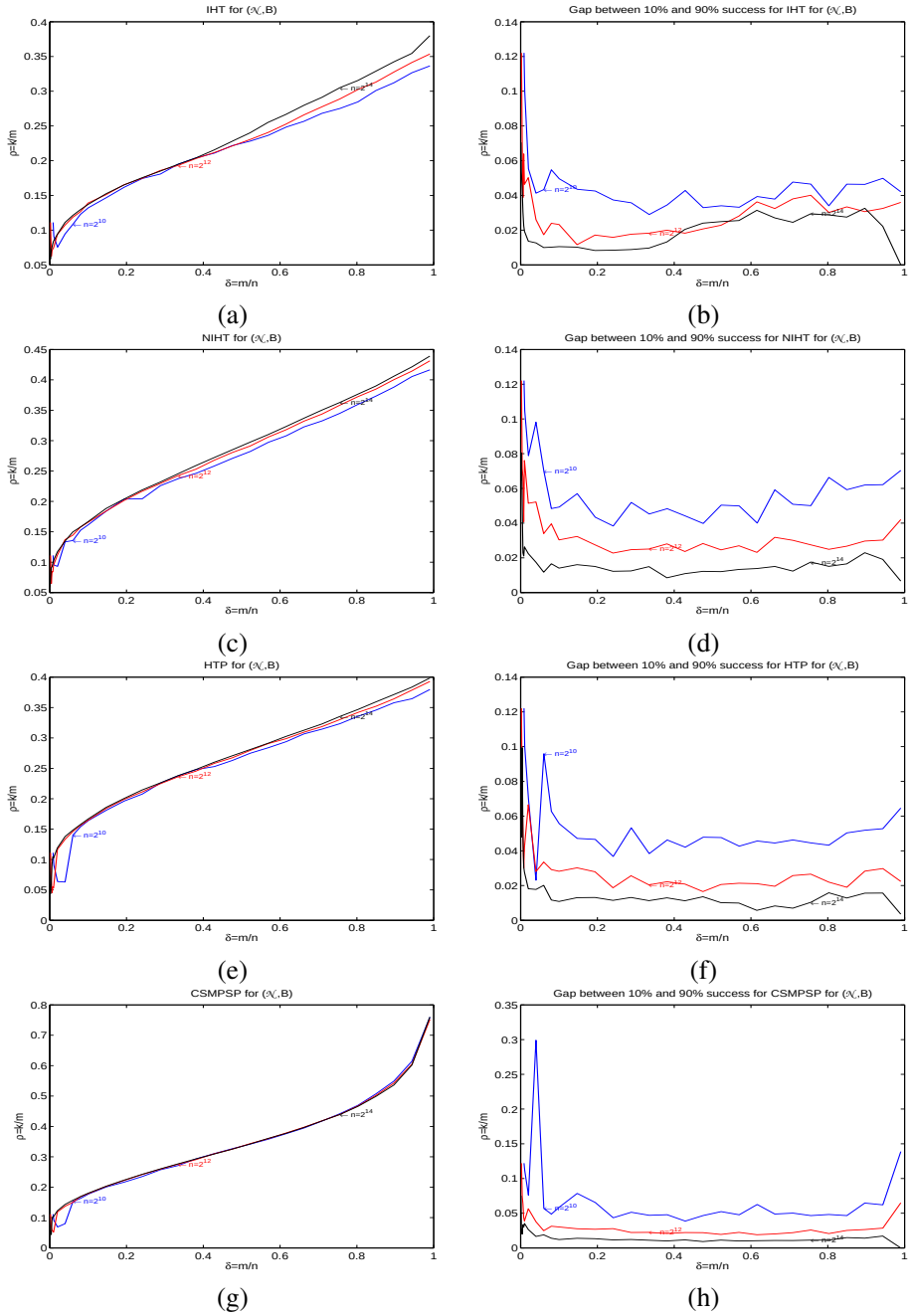



Figure 17. Left panels: 50% recovery probability logistic regression curves for $(\mathcal{N}, B)$ and $n = 2^j$ with $j = 10, 12, 14$. Right panels: gap between 10% and 90% recovery probability curves. Results shown for the algorithms: IHT (a-b), NIHT (c-d), HTP (e-f), and CSMPSP (g-h).

## C. RECOVERY PHASE TRANSITIONS FOR ALTERNATE VECTOR ENSEMBLES

Section 4.2 includes a discussion of the recovery phase transitions for the sparse vector ensembles $U$ and $N$. This appendix contains the representations of the data used for Obs. 4.2.1–4.2.3. Figures 26–29 present the recovery phase transition curves for the problem classes $(Mat, vec)$ for $Mat \in \{\mathcal{N}, \mathcal{S}_7, DCT\}$ and $vec \in \{U, U_\epsilon, N, N_\epsilon\}$ with $\epsilon = 1/10$. For some problem classes, the recovery phase transition curves lack smoothness and for $(\mathcal{N}, N)$ the recovery phase transition curve is not monotonically increasing. The

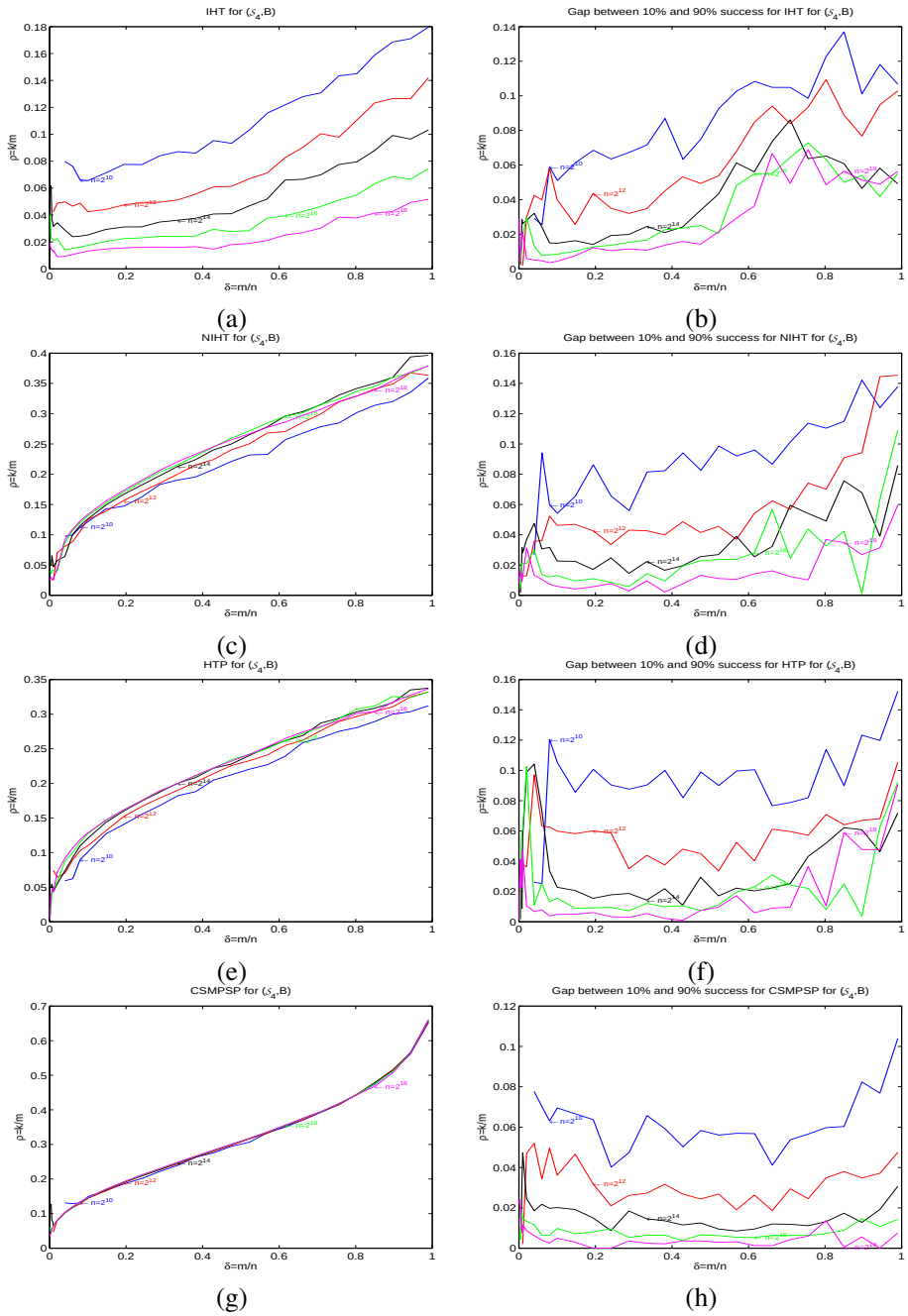



Figure 18. Left panels: 50% recovery probability logistic regression curves for $(\mathcal{S}_4, B)$ and $n = 2^j$ with $j = 10, 12, 14, 16, 18$. Right panels: gap between 10% and 90% recovery probability curves. Results shown for the algorithms: IHT (a-b), NIHT (c-d), HTP (e-f), and CSMPSP (g-h).

following observation is useful when interpreting the data presented in Figures 26–29 and is explicitly invoked in Obs. 4.2.3 and 4.2.7.

*Observation C.1*
For the problem class $(Mat, vec)$ with $Mat \in \{\mathcal{N}, \mathcal{S}_p, DCT\}$ and $vec \in \{U, N\}$, the phase transition region for each algorithm can be much wider than observed for $vec = B$. In some cases, the phase transition region is large enough to cause $\rho^{alg}_{(Mat,vec)}(\delta)$ to lack the expected smoothness and monotonicity. (See Fig. 25.)

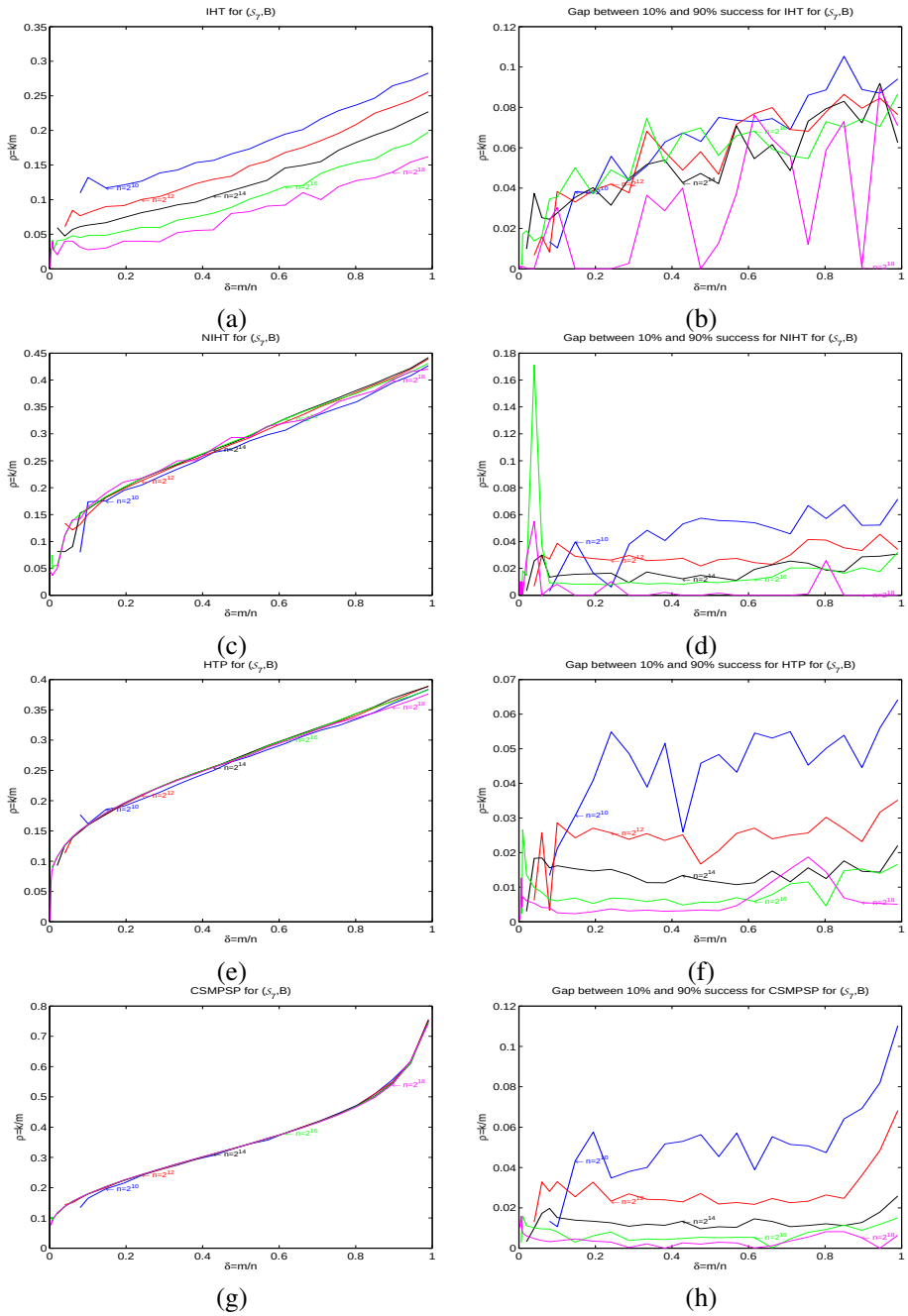



Figure 19. Left panels: 50% recovery probability logistic regression curves for $(\mathcal{S}_7, B)$ and $n = 2^j$ with $j = 10, 12, 14, 16, 18$. Right panels: gap between 10% and 90% recovery probability curves. Results shown for the algorithms: IHT (a-b), NIHT (c-d), HTP (e-f), and CSMPSP (g-h).

As stated in Obs. C.1, the greedy algorithms' transition regions can be much wider when recovering vectors from the alternate distributions, even in the noiseless setting. From App. A the transition region widths depicted in Figs. 16–19 are essentially bounded above by $0.02$ for problem classes $(Mat, B)$ with $n \geq 2^{16}$ for $Mat \in \{\mathcal{S}_p, DCT\}$ and $n \geq 2^{12}$ for $Mat = \mathcal{N}$. For comparable problem sizes on the alternate vector ensembles $vec \in \{U, N\}$, the transition regions can be 5 to 20 times as wide. In Fig. 29, we show three representative recovery phase transitions $\rho^{alg}_{(Mat,vec)}(\delta)$ (a,c,e) together with the associated transition region

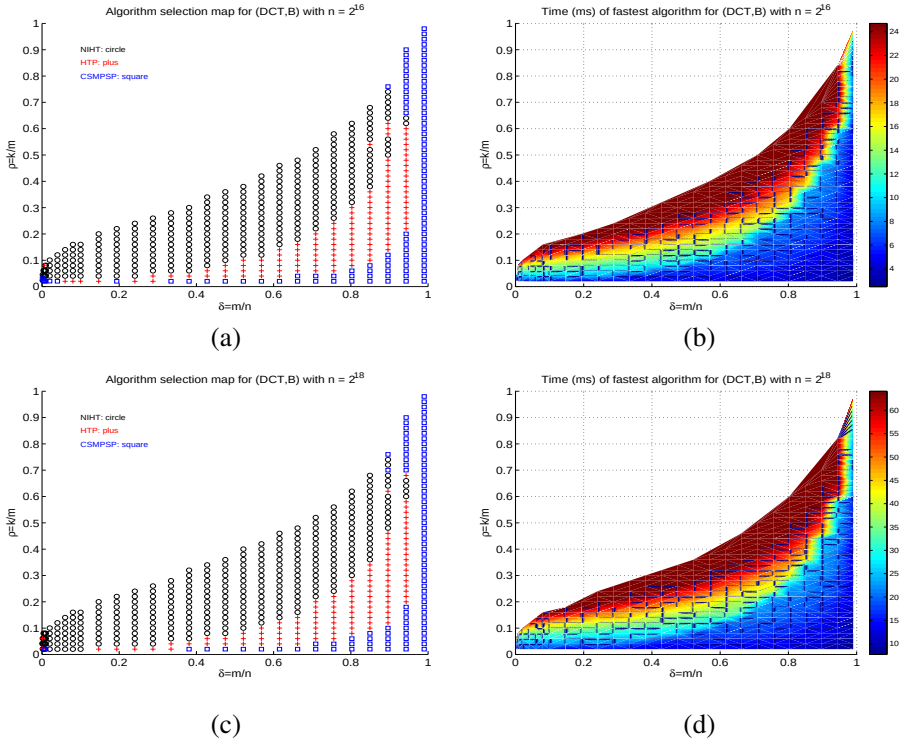



Figure 20. Left panels: Algorithm selection maps for problem class $(DCT, B)$. Right panels: Average time for the fastest algorithm. Panels (a-b) with $n = 2^{16}$ and (c-d) with $n = 2^{18}$.

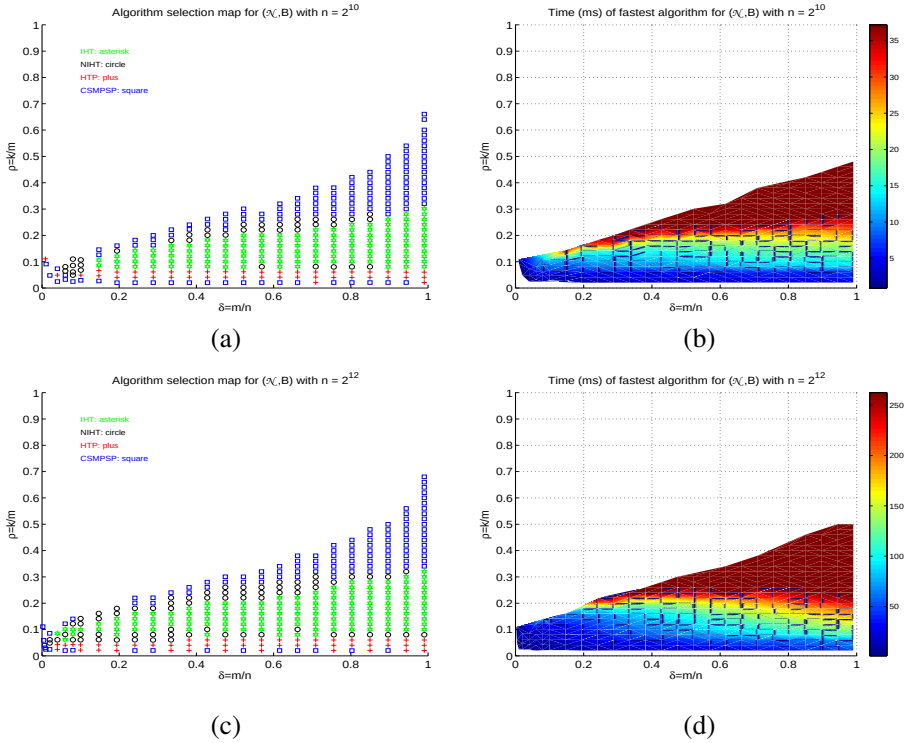



Figure 21. Left panels: Algorithm selection maps for problem class $(\mathcal{N}, B)$. Right panels: Average time for the fastest algorithm. Panels (a-b) with $n = 2^{10}$ and (c-d) with $n = 2^{12}$.

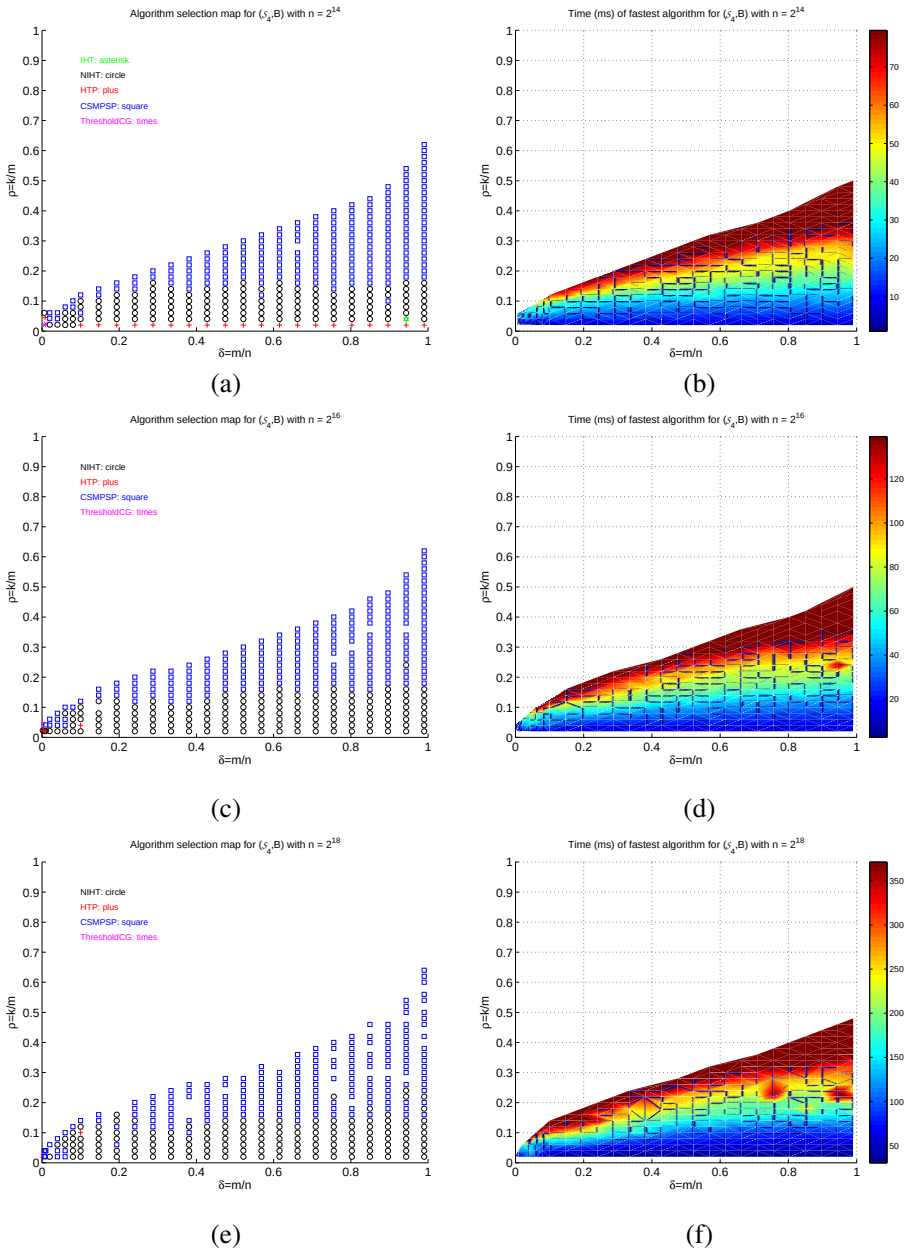



Figure 22. Left panels: Algorithm selection maps for problem class $(\mathcal{S}_4, B)$. Right panels: Average time for the fastest algorithm. Panels (a-b) with $n = 2^{14}$, (c-d) with $n = 2^{16}$, and (e-f) with $n = 2^{18}$.

width (b,d,f). In each of these cases, $\rho^{alg}_{(Mat,vec)}(\delta)$ lacks the now familiar smoothness and monotonicity of recovery phase transitions in sparse approximation.

## D. ALGORITHM SELECTION MAPS FOR ALTERNATE VECTOR ENSEMBLES

In Sec. 4.2, algorithm selection maps are presented for $(Mat, vec)$ with $vec \in \{U_\epsilon, N_\epsilon\}$. This appendix contains the representations of the data used for Obs. 4.2.4 –4.2.6. Figures 30–35 present the ratio of the recovery time for each algorithm to the fastest recovery time in the union of the recovery regions for problem classes $(Mat, U_\epsilon)$ and $(Mat, N_\epsilon)$. Note the ratio of recovery time for NIHT to the fastest algorithm is consistently small and almost always one for the matrix ensembles $\mathcal{S}_7$ and $\mathcal{N}$.

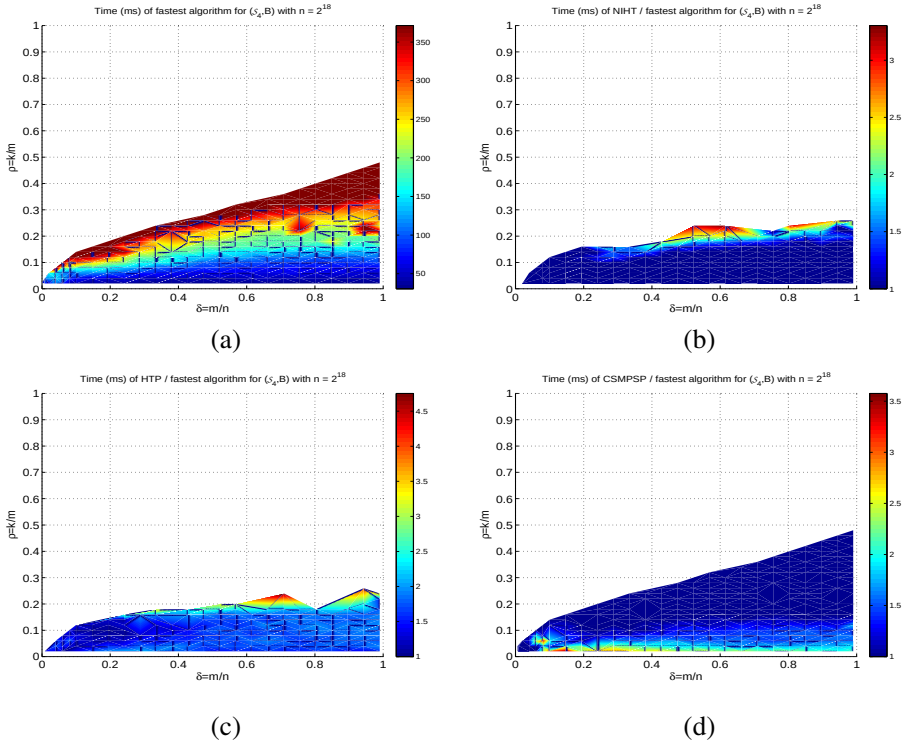



Figure 23. Panel (a), time for fastest algorithm for $(\mathcal{S}_4, B)$ with $n = 2^{14}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

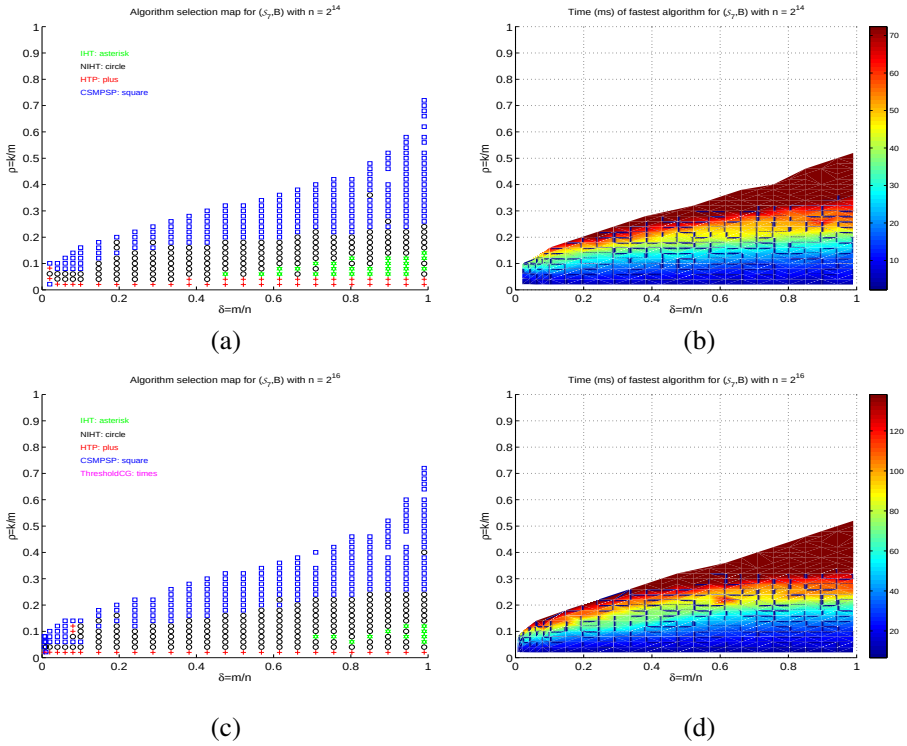



Figure 24. Left panels: Algorithm selection maps for problem class $(\mathcal{S}_7, B)$. Right panels: Average time for the fastest algorithm. Panels (a-b) with $n = 2^{14}$ and (c-d) with $n = 2^{16}$.

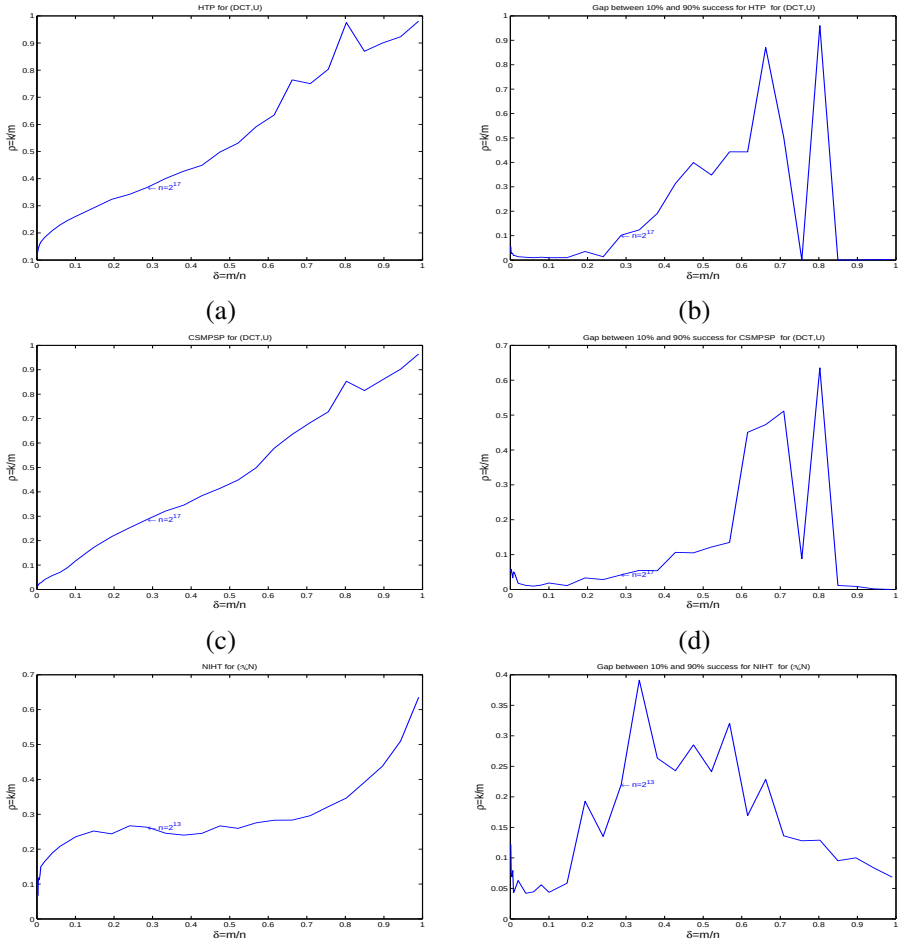



Figure 25. Left panels: 50% recovery probability logistic regression curves. Right panels: gap between 10% and 90% recovery probability curves. (a-b) HTP for $(DCT, U)$; (c-d) CSMPSP for $(DCT, U)$; (e-f) NIHT with $(\mathcal{N}, U)$.

## E. COMPARISONS OF ALTERNATE VECTOR ENSEMBLES

Observation 4.2.7 in Sec. 4.2 states that the recovery phase transition for NIHT, HTP, and CSMPSP is lowest for the sparse vector ensemble $B$. This appendix contains the representations of the data used for Obs. 4.2.7. Figures 36–38 present the recovery phase transition curves for NIHT, HTP, and CSMPSP for the problem classes $(Mat, vec)$ where $Mat$ is a fixed member of the set $\{\mathcal{N}, \mathcal{S}_4, DCT\}$ while $vec$ takes on each value in one of the two sets $\{B, U, N\}$ or $\{B_\epsilon, U_\epsilon, N_\epsilon\}$. The problem classes with $Mat = \mathcal{S}_7$ appear in Sec. 4.2 as Fig. 15. For NIHT and HTP and all matrix ensembles, the recovery phase transition curves $\rho^{alg}_{(Mat,vec)}(\delta) > \rho^{alg}_{(Mat,B)}(\delta)$ for $vec \in \{U, N\}$ and $\rho^{alg}_{(Mat,vec)}(\delta) > \rho^{alg}_{(Mat,B_\epsilon)}(\delta)$ for $vec \in \{U_\epsilon, N_\epsilon\}$. While $\rho^{csmpsp}_{(Mat,U)}(\delta) > \rho^{csmpsp}_{(Mat,B)}(\delta)$ and $\rho^{csmpsp}_{(Mat,N)}(\delta) > \rho^{csmpsp}_{(Mat,B)}(\delta)$, the addition of moderate levels of noise ($\epsilon = 1/10$)) forces the recovery phase transition curves for problem classes $(Mat, vec)$, $Mat \in \{\mathcal{S}_p, \mathcal{N}\}$ and $vec \in \{U_\epsilon, N_\epsilon\}$ to fall below the related curves for vector ensemble $B_\epsilon$ as $m/n$ approaches one.

## F. ROLE OF NONZEROS PER COLUMN ON PROBLEM CLASSES $(\mathcal{S}_p, vec)$

This appendix briefly investigates the role of the number of nonzeros per column in the sparse matrix ensemble $\mathcal{S}_p$. For computational reasons, using very few nonzeros per column provides a relatively fast

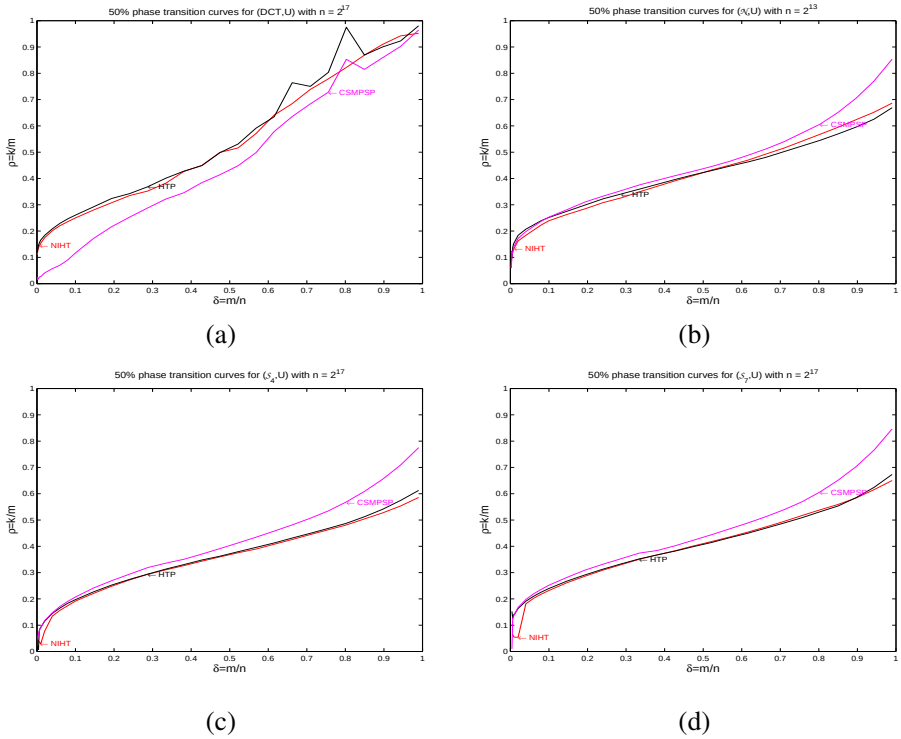



Figure 26. 50% recovery probability logistic regression curves for $vec = U$ and algorithms NIHT, HTP, and CSMPSP: (a) $Mat = DCT$ with $n = 2^{17}$, (b) $Mat = \mathcal{N}$ with $n = 2^{13}$, (c) $Mat = \mathcal{S}_4$ with $n = 2^{17}$, and (d) $Mat = \mathcal{S}_7$ with $n = 2^{17}$.

matrix multiplication. Sections 3 and 4 focus on $p = 4$ or $p = 7$ with an emphasis placed on $p = 7$. Increasing the number of nonzeros from $p = 7$ to $p = 13$ requires roughly 1.85 times as many computations, yet there is only a minor increase in the recovery phase transition. Figures 39 and 40 present the recovery phase transitions of a fixed algorithm for the problem classes $(\mathcal{S}_p, vec)$, $p = 4, 7, 13$, displaying the three recovery phase transition curves together on a single plot for a fixed vector ensemble $vec \in \{B, U, N, B_\epsilon, U_\epsilon, N_\epsilon\}$. This data supports our focus on $\mathcal{S}_7$ while also demonstrating that very few nonzeros per column $p = 4$ can still lead to a reasonably sized recovery region.

*Observation F.1*
For problem classes $(\mathcal{S}_p, vec)$ with $vec \in \{B, U, N, B_\epsilon, U_\epsilon, N_\epsilon\}$ increasing the number of nonzeros per column from $p = 4$ to $p = 7$ increases the area of the recovery phase transition curve for NIHT, HTP, and CSMPSP from between 8% and 17%; in contrast, further increasing the number of nonzeros per column from $p = 7$ to $p = 13$ never increases the area of the recovery phase transition by more than 3%. (See Figs. 39 and 40.)

## REFERENCES

1. Donoho DL. Compressed sensing. *IEEE Trans. Inform. Theory* 2006; **52**(4):1289–1306.
2. Candès EJ. Compressive sampling. *International Congress of Mathematicians. Vol. III*. Eur. Math. Soc., Zürich, 2006; 1433–1452.
3. APS M. Mosek optimization software. `http://www.mosek.com`.
4. CVX Research, Inc. CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx` Sep 2012.
5. ILOG CPLEX. Mathematcial programming system. `http://www.cplex.com`.
6. Figueiredo MAT, Nowak RD, Wright SJ. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Selected Topics in Signal Processing* 2007; **1**(4):586–597.

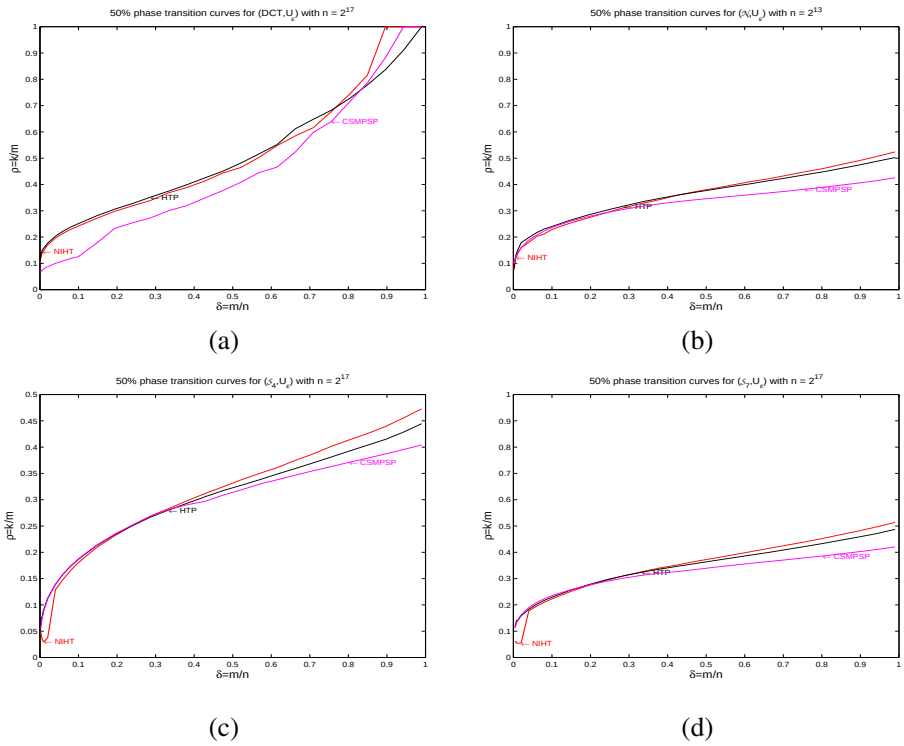



Figure 27. 50% recovery probability logistic regression curves for $vec = U_\epsilon$ with $\epsilon = 1/10$ and algorithms NIHT, HTP, and CSMPSP: (a) $Mat = DCT$ with $n = 2^{17}$, (b) $Mat = \mathcal{N}$ with $n = 2^{13}$, (c) $Mat = \mathcal{S}_4$ with $n = 2^{17}$, and (d) $Mat = \mathcal{S}_7$ with $n = 2^{17}$.


7. Berg Evd, Friedlander MP. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing* 2008; **31**(2):890–912.
8. Wright SJ, Nowak RD, Figueiredo MAT. Sparse reconstruction by separable approximation. *Proc. International Conference on Acoustics, Speech, and Signal Processing* 2008; .
9. Yin W, Osher S, Goldfarb D, Darbon J. Bregman iterative algorithms for $\ell^1$-minimization with applications to compressed sensing. *SIAM Journal on Imaging Science* 2008; **1**(1):143–168.
10. Donoho DL. Neighborly polytopes and sparse solution of underdetermined linear equations 2004. Technical Report, Department of Statistics, Stanford University.
11. Donoho DL. High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension. *Discrete Comput. Geom.* 2006; **35**(4):617–652.
12. Xu W, Hassibi B. Compressed sensing over the grassmann manifold: A unified analytical framework. Forty-Sixth Annual Allerton Conference, 2008.
13. Xu W, Hassibi B. Precise stability phase transitions for minimization: A unified geometric framework. *Information Theory, IEEE Transactions on* oct 2011; **57**(10):6894 –6919.
14. Donoho DL, Tanner J. Neighborliness of randomly projected simplices in high dimensions. *Proc. Natl. Acad. Sci. USA* 2005; **102**(27):9452–9457 (electronic).
15. Donoho DL, Tanner J. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proc. Natl. Acad. Sci. USA* 2005; **102**(27):9446–9451 (electronic).
16. Donoho DL, Tanner J. Counting faces of randomly projected polytopes when the projection radically lowers dimension. *J. AMS* 2009; **22**(1):1–53.
17. Donoho DL, Tanner J. Counting the faces of randomly-projected hypercubes and orthants, with applications 2008. Submitted.
18. Benjamin Recht WX, Hassibi B. Null space conditions and thresholds for rank minimization. *Mathematical Programming. Ser B.* 2011; **127**:175–211.
19. Stojnic M. Optimization in block-sparse compressed sensing and its strong thresholds. *Selected Topics in Signal Processing, IEEE Journal of* 2010; **4**(2):350–357.
20. Donoho DL, Maleki A, Montanari A. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences* 2009; **106**(45):18 914–18 919.
21. Donoho DL, Maleki A, Montanari A. The noise-sensitivity phase transition in compressed sensing. *IEEE Trans. Inform. Theory* 2011; **57**(10):6920–6941.
22. Donoho D, Tanner J. 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.* 2009; **367**(1906):4273–4293, doi:10.1098/rsta.2009.0152. URL

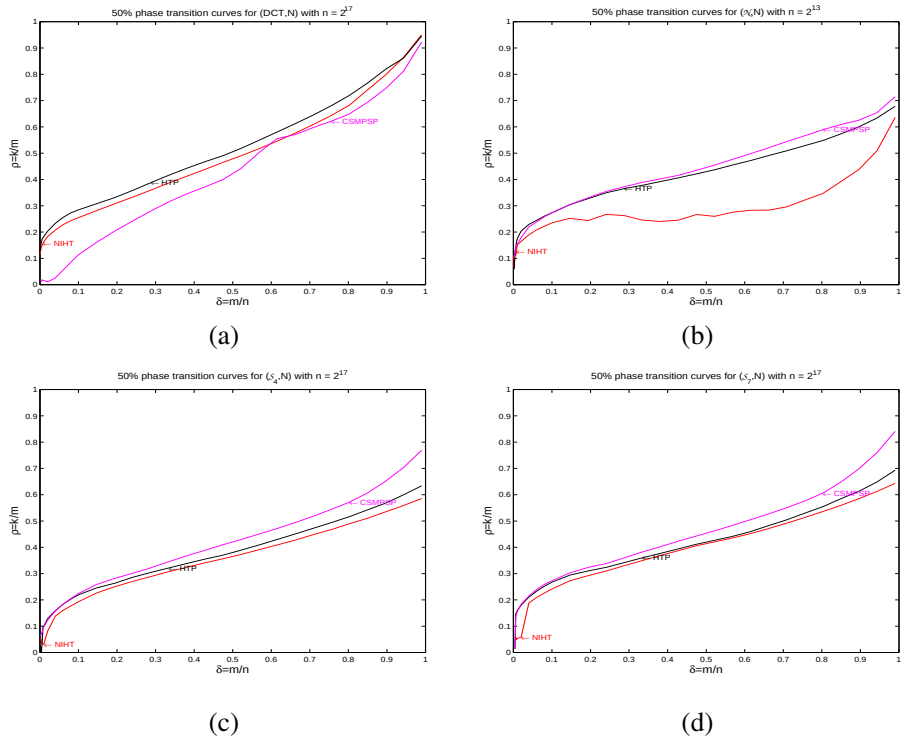



Figure 28. 50% recovery probability logistic regression curves for $vec = N$ and algorithms NIHT, HTP, and CSMPSP: (a) $Mat = DCT$ with $n = 2^{17}$, (b) $Mat = \mathcal{N}$ with $n = 2^{13}$, (c) $Mat = \mathcal{S}_4$ with $n = 2^{17}$, and (d) $Mat = \mathcal{S}_7$ with $n = 2^{17}$.


http://dx.doi.org/10.1098/rsta.2009.0152, with electronic supplementary materials available online.

23. Monajemi H, Jafarpour S, Gavish M, Stat 330/CME 362 Collaboration, Donoho DL. Deterministic matrices matching the compressed sensing phase transitions of gaussian random matrices. *Proceedings of the National Academy of Sciences* 2013; **110**(4):1181–1186.
24. Needell D, Tropp J. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comp. Harm. Anal.* 2009; **26**(3):301–321.
25. Dai W, Milenkovic O. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Inform. Theory* 2009; **55**(5):2230–2249.
26. Blumensath T, Davies ME. Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.* 2009; **27**(3):265–274.
27. Blumensath T, Davies ME. Normalised iterative hard thresholding; guaranteed stability and performance. *IEEE Selected Topics in Signal Processing* 2010; **4**(2):298–309.
28. Foucart S. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis* 2011; **49**(6):2543–2563.
29. Candes EJ, Tao T. Decoding by linear programming. *IEEE Trans. Inform. Theory* 2005; **51**(12):4203–4215.
30. Blanchard JD, Cartis C, Tanner J, Thompson A. Phase transitions for greedy sparse approximation algorithms. *Appl. Comput. Harmon. Anal.* 2011; **30**(2):188–203.
31. Cartis C, Thompson A. Phase transitions for greedy sparse approximation algorithms 2011. Preprint.
32. Donoho JI David, Montanari A. Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising 2013. URL http://front.math.ucdavis.edu/1111.1041.
33. Maleki A, Donoho D. Optimally tuned iterative reconstruction algorithms for compressed sensing. *Selected Topics in Signal Processing, IEEE Journal of* april 2010; **4**(2):330 –341.
34. Sturm B. Sparse vector distributions and recovery from compressed sensing 2011. ArXiv:1103.6246v2.
35. Blanchard JD, Tanner J. GPU accelerated greedy algorithms for compressed sensing. *Submitted, www.math.grinnell.edu/∼blanchaj/GAGACS.pdf* 2012; .
36. Blanchard JD, Tanner J. GAGA: GPU Accelerated Greedy Algorithms 2012. URL webpage coming soon, version 0.9.9.
37. Schnass K, Vandergheynst P. Average performance analysis for thresholding. *Signal Processing Letters, IEEE* nov 2007; **14**(11):828 –831.
38. Blanchard JD, Cartis C, Tanner J. Compressed Sensing: How sharp is the restricted isometry property? *SIAM Review* 2011; **53**(1):105–125.
39. Donoho DL, Tanner J. Precise undersampling theorems. *Proceedings of the IEEE* 2010; **98**(6):913–924.

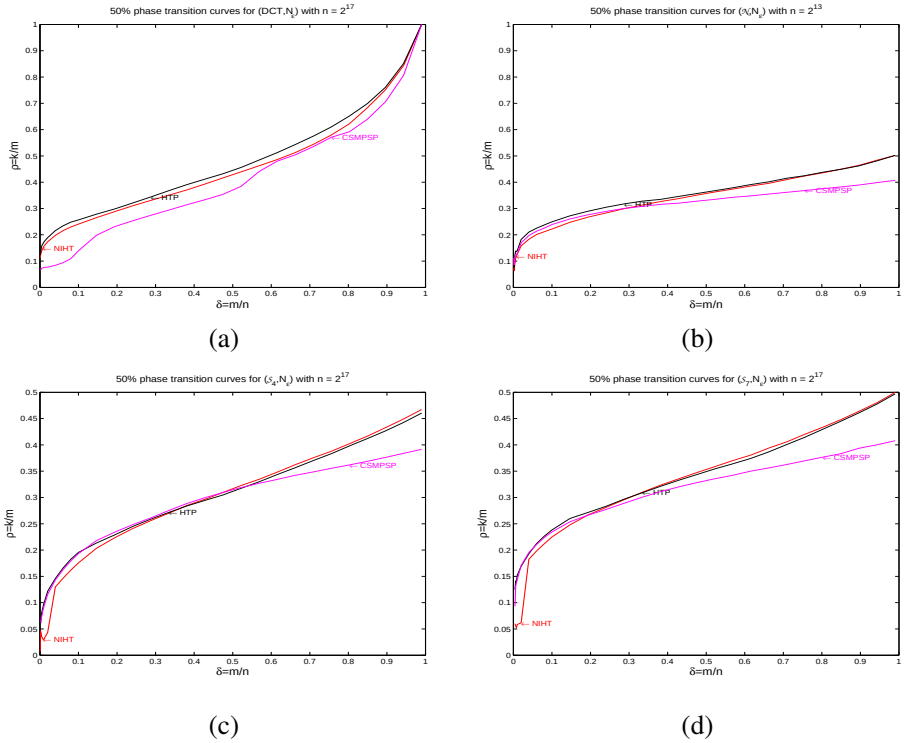



Figure 29. 50% recovery probability logistic regression curves for $vec = N_\epsilon$ with $\epsilon = 1/10$ and algorithms NIHT, HTP, and CSMPSP: (a) $Mat = DCT$ with $n = 2^{17}$, (b) $Mat = \mathcal{N}$ with $n = 2^{13}$, (c) $Mat = \mathcal{S}_4$ with $n = 2^{17}$, and (d) $Mat = \mathcal{S}_7$ with $n = 2^{17}$.

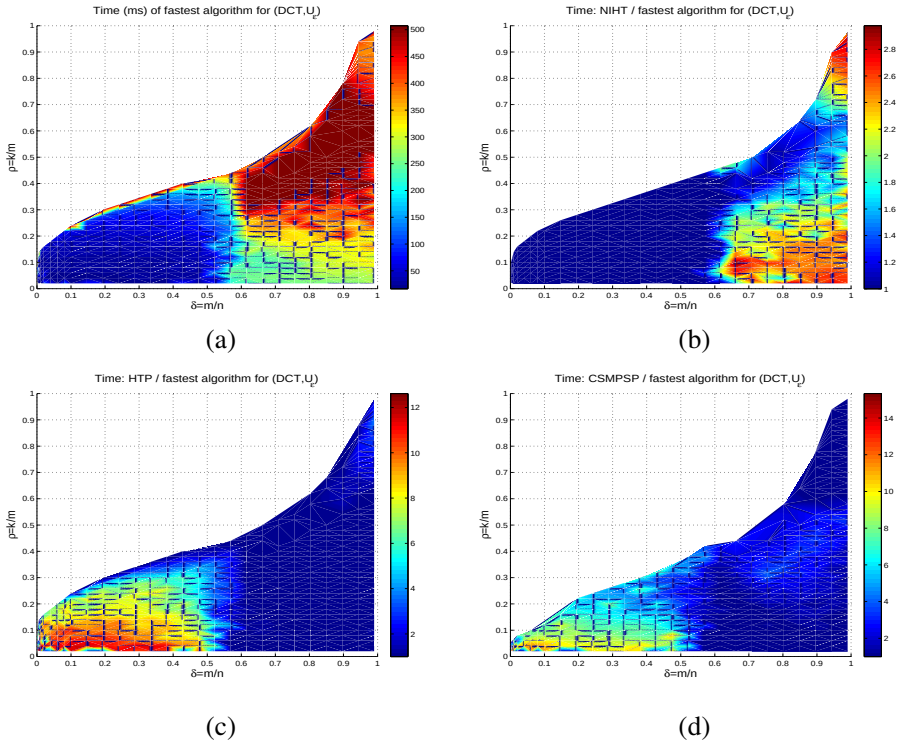



Figure 30. Panel (a), time for fastest algorithm for $(DCT, U_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{17}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

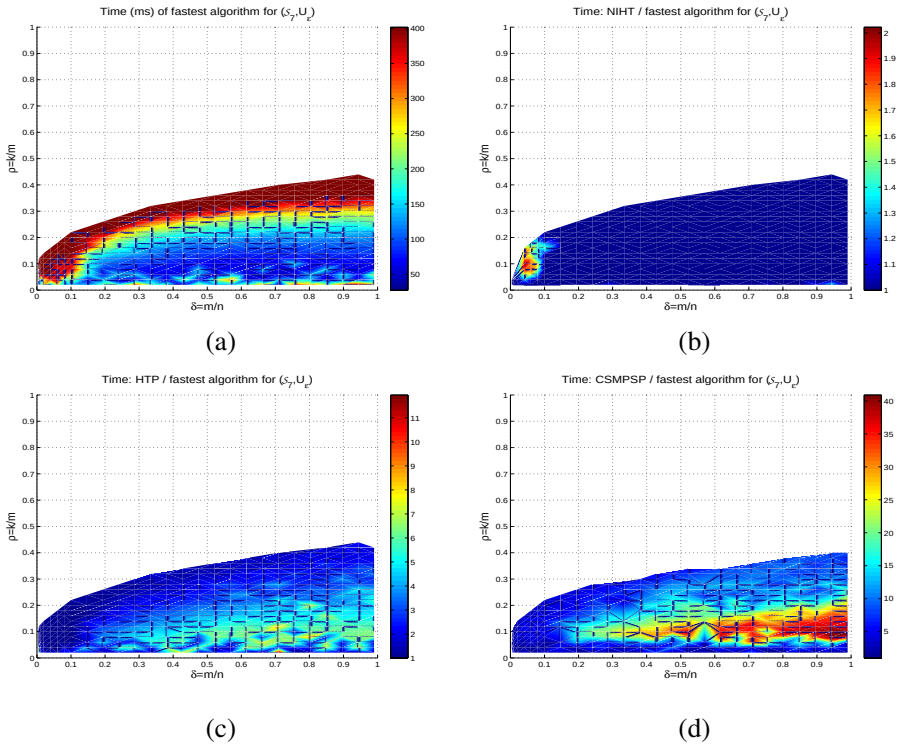



Figure 31. Panel (a), time for fastest algorithm for $(\mathcal{S}_7, U_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{17}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

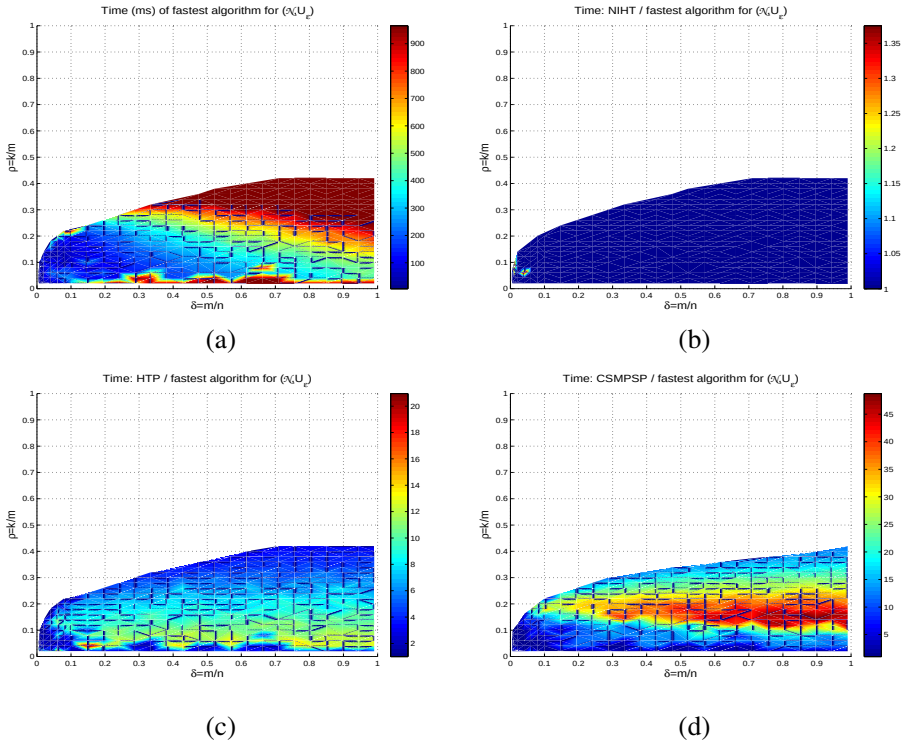



Figure 32. Panel (a), time for fastest algorithm for $(\mathcal{N}, U_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{13}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

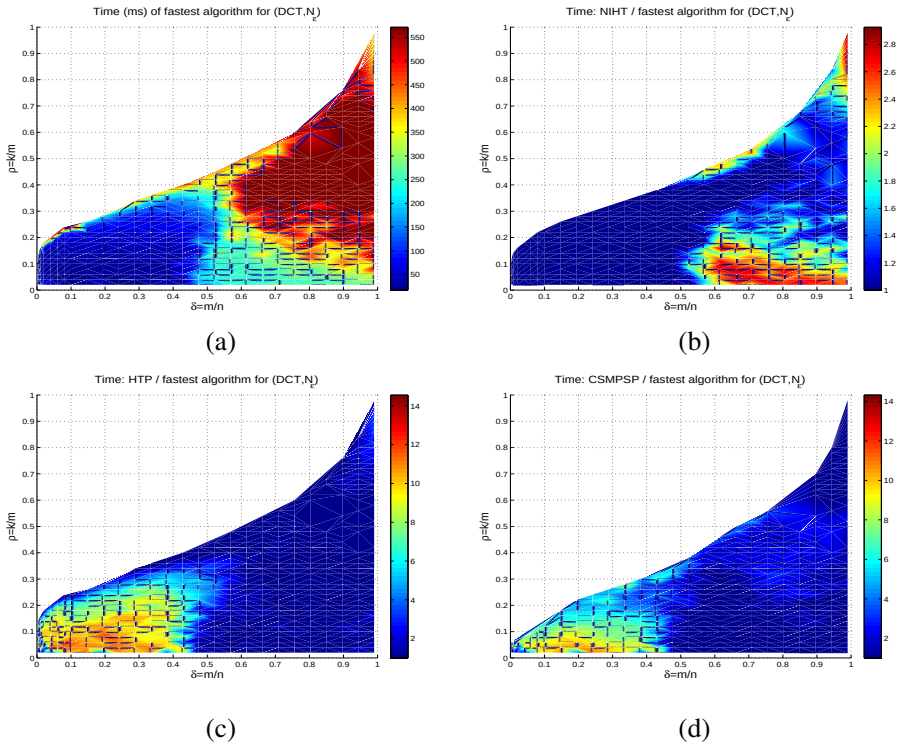



Figure 33. Panel (a), time for fastest algorithm for $(DCT, N_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{17}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

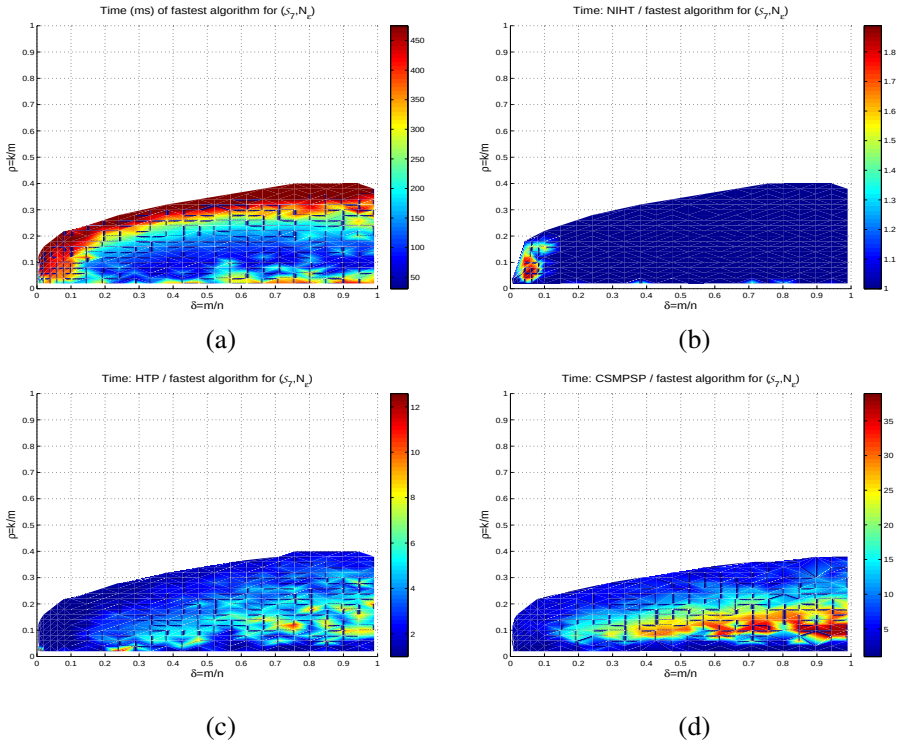



Figure 34. Panel (a), time for fastest algorithm for $(\mathcal{S}_7, N_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{17}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

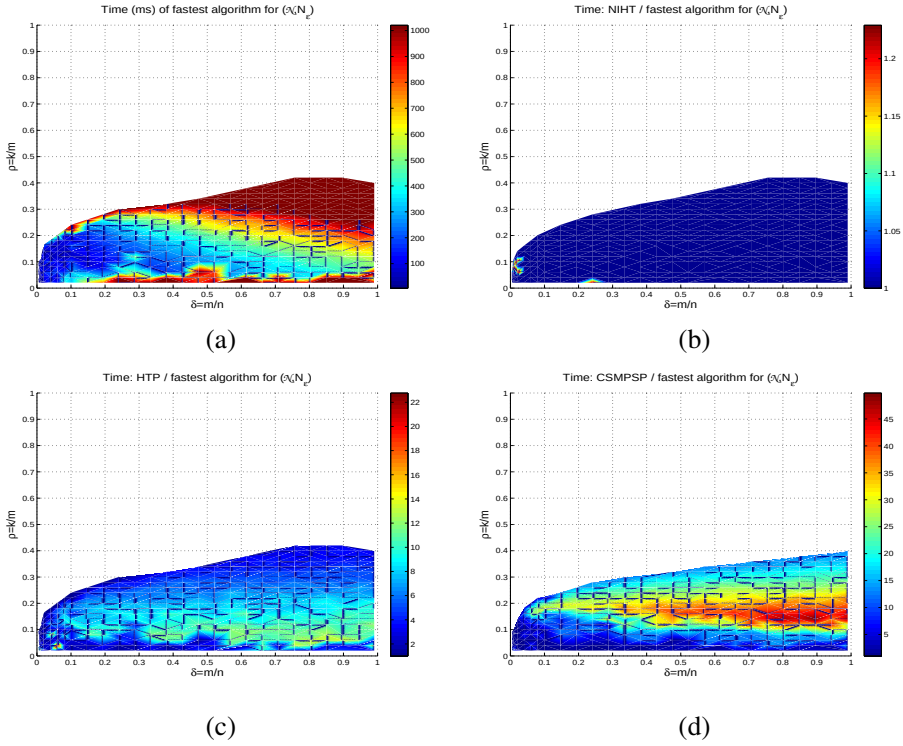



Figure 35. Panel (a), time for fastest algorithm for $(\mathcal{N}, N_\epsilon)$ with $\epsilon = 1/10$ and $n = 2^{13}$. Ratio of average time for NIHT, HTP, and CSMPSP over the fastest algorithm in (b-d) respectively.

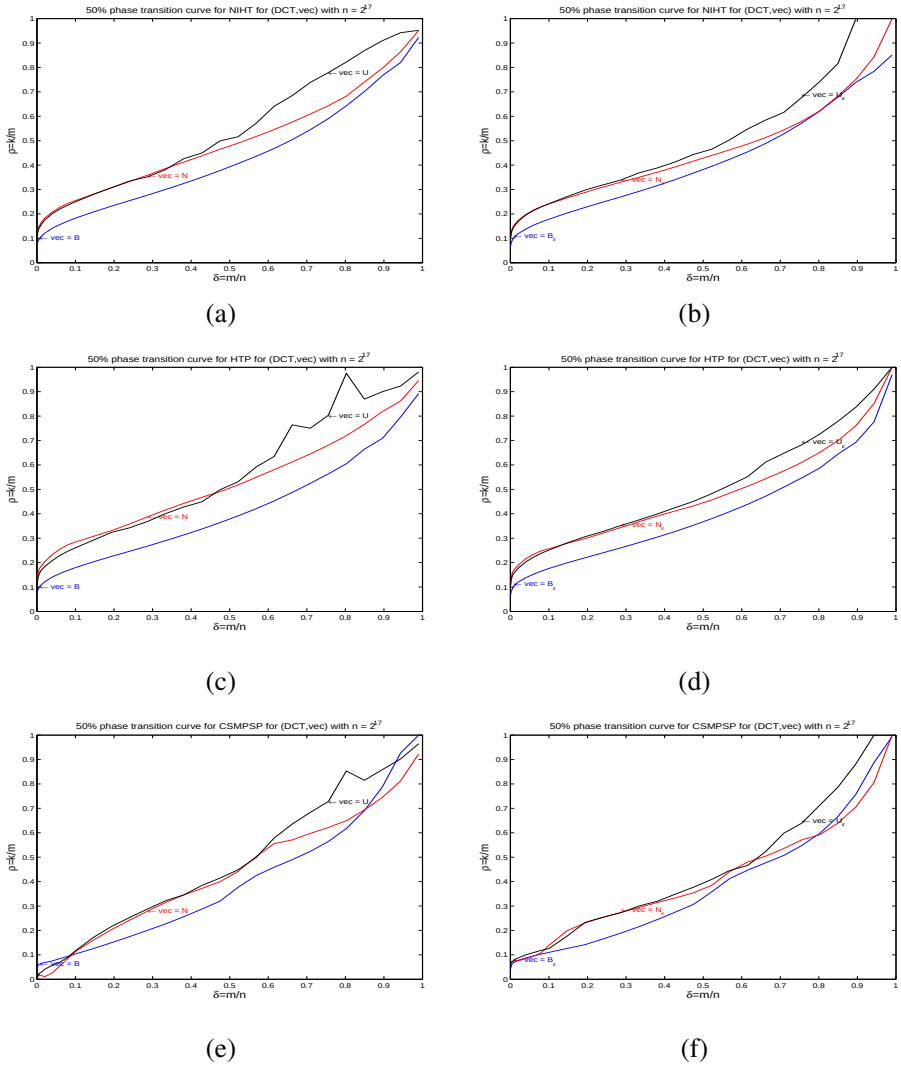



Figure 36. 50% recovery probability logistic regression curves for $Mat = DCT$ with $n = 2^{17}$ and for each $vec \in \{B, U, N\}$ for (a) NIHT, (c) HTP, and (e) CSMPSP, and for each $vec \in \{B_\epsilon, U_\epsilon, N_\epsilon\}$ with $\epsilon = 1/10$ for (b) NIHT, (e) HTP, and (f) CSMPSP.

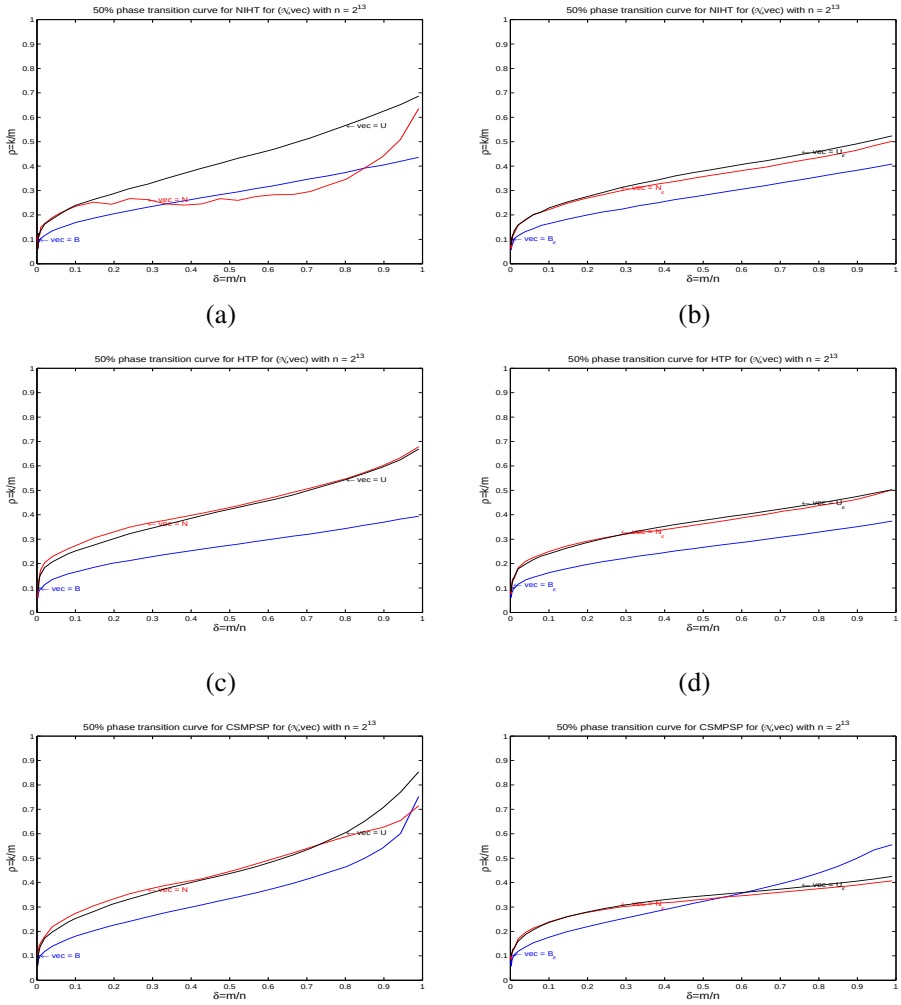



Figure 37. 50% recovery probability logistic regression curves for $Mat = \mathcal{N}$ with $n = 2^{13}$ and for each $vec \in \{B, U, N\}$ for (a) NIHT, (c) HTP, and (e) CSMPSP, and for each $vec \in \{B_\epsilon, U_\epsilon, N_\epsilon\}$ with $\epsilon = 1/10$ for (b) NIHT, (e) HTP, and (f) CSMPSP.

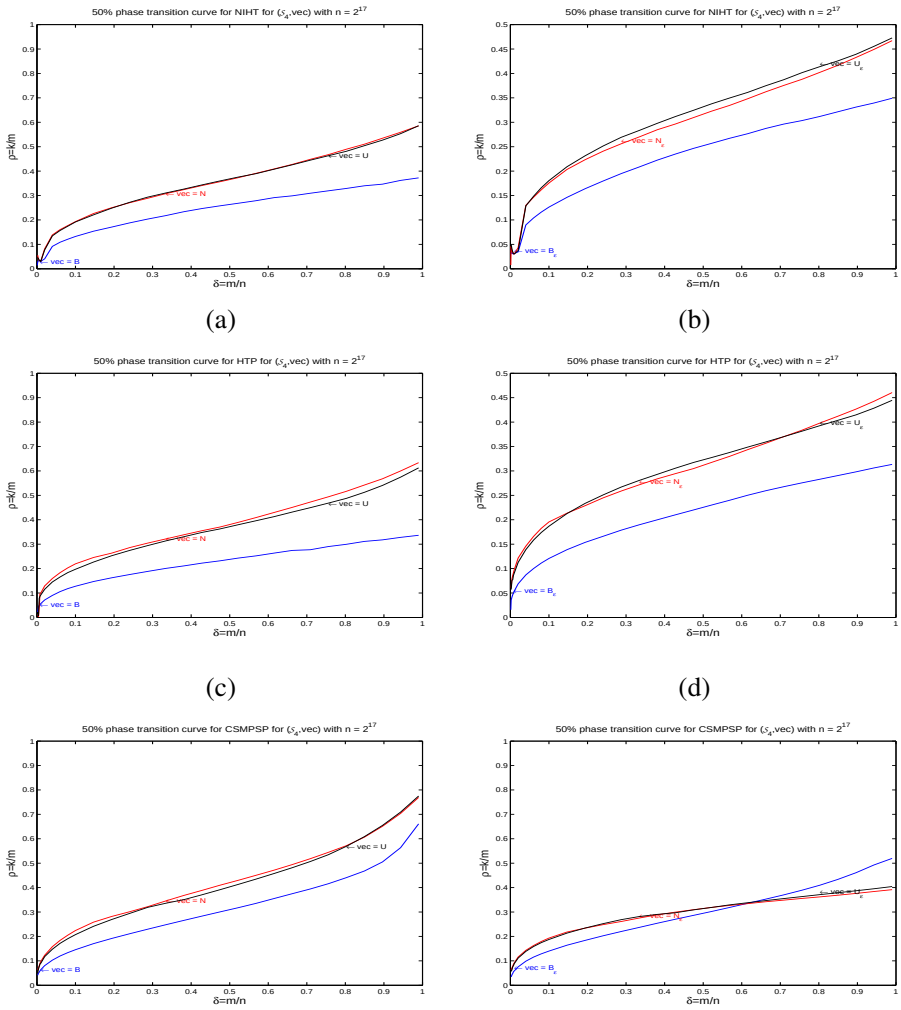
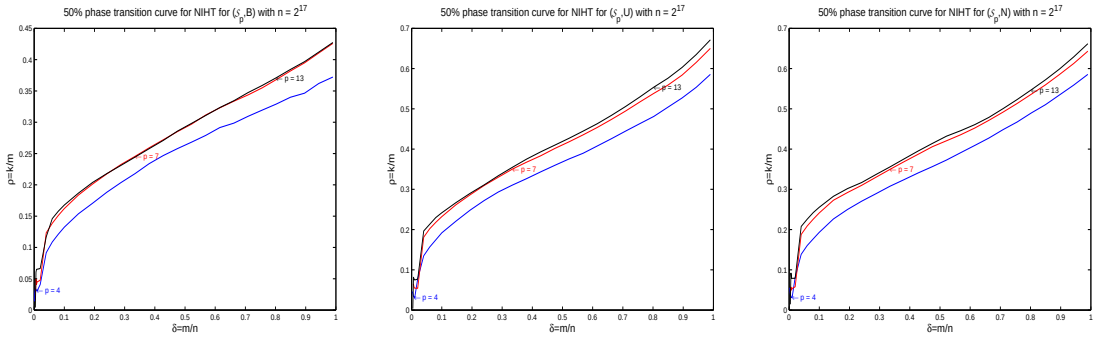
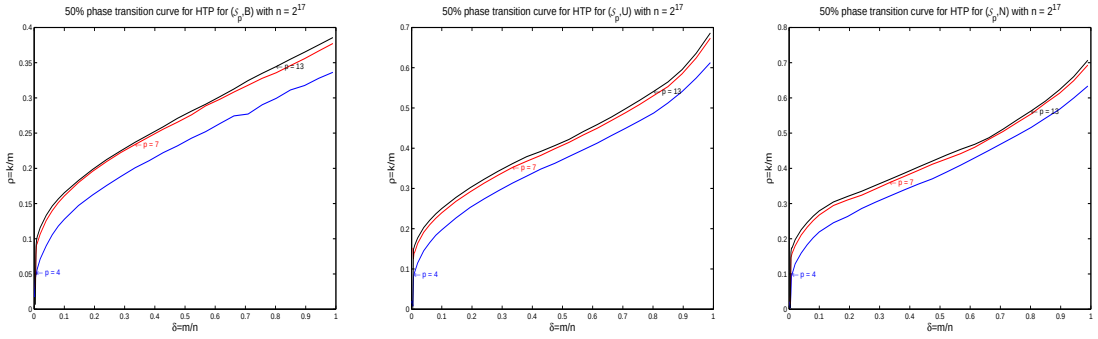



Figure 38. 50% recovery probability logistic regression curves for $Mat = \mathcal{S}_4$ with $n = 2^{17}$ and for each $vec \in \{B, U, N\}$ for (a) NIHT, (c) HTP, and (e) CSMPSP, and for each $vec \in \{B_\epsilon, U_\epsilon, N_\epsilon\}$ with $\epsilon = 1/10$ for (b) NIHT, (e) HTP, and (f) CSMPSP.

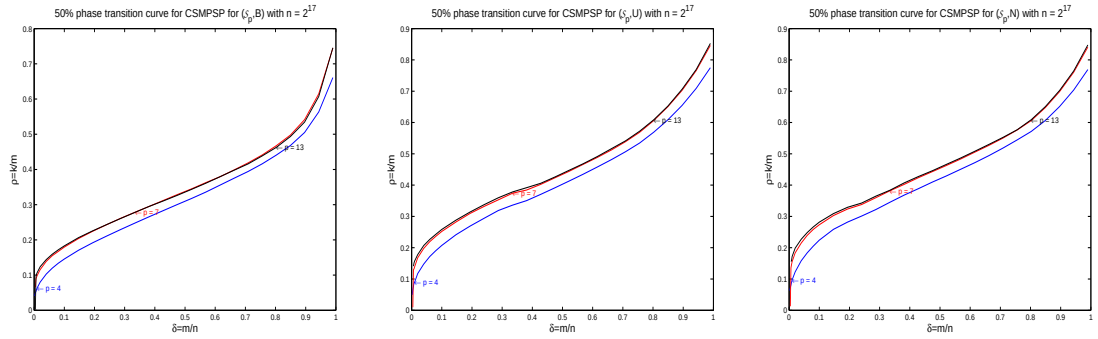
 




(a) (b) (c)



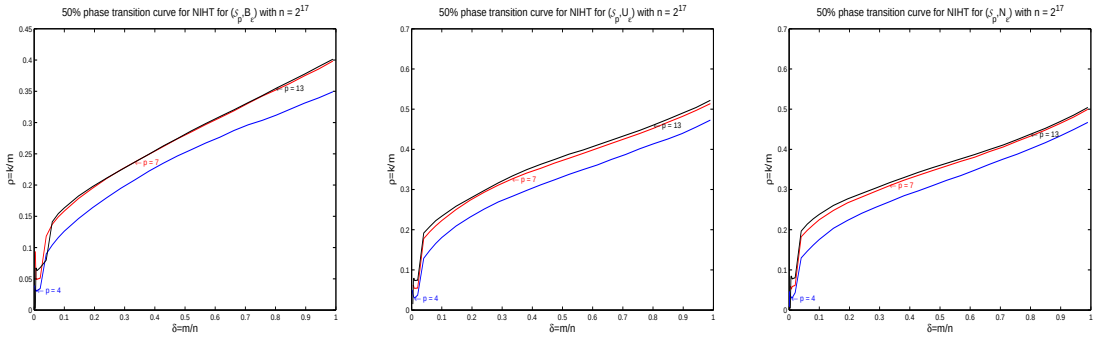

(d) (e) (f)



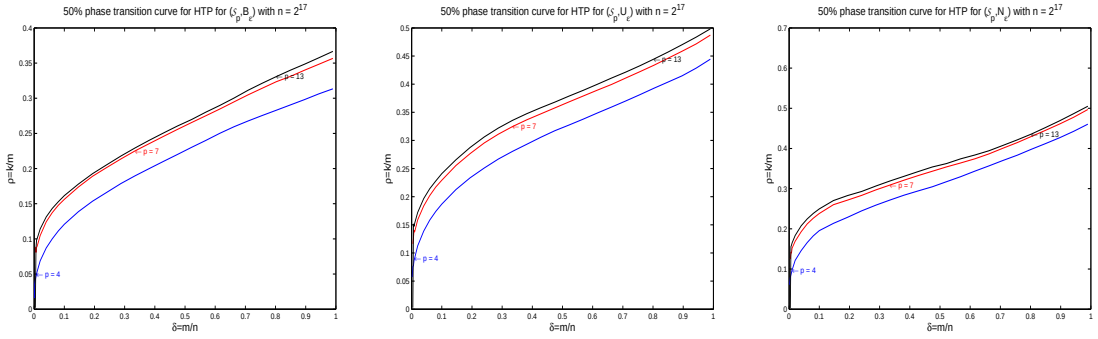

(g) (h) (i)

Figure 39. 50% recovery probability logistic regression curves for $Mat = \mathcal{S}_p$ with $n = 2^{17}$ and for $p = 4$, 7, and 13 in each plot. Left, center, and right panels are $vec = B$, $vec = U$, and $vec = N$ respectively. Algorithms: NIHT (a-c), HTP (d-f), and CSMPSP (g-i).

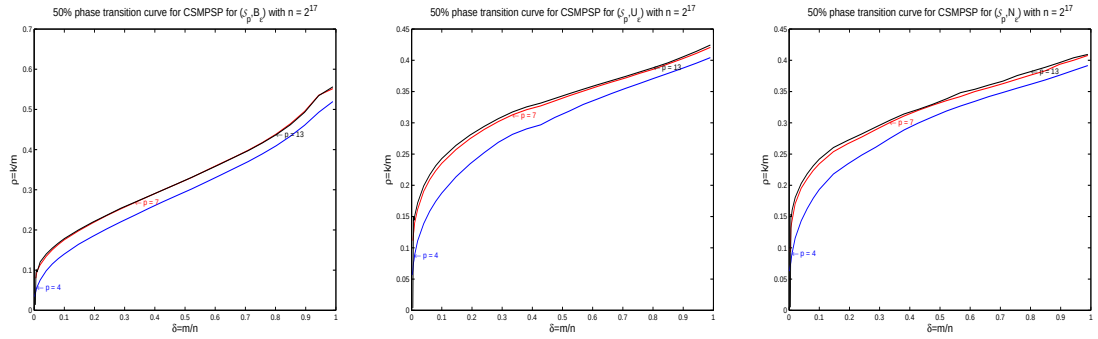
 




(a) (b) (c)




(d) (e) (f)




(g) (h) (i)

Figure 40. 50% recovery probability logistic regression curves for $Mat = \mathcal{S}_p$ with $n = 2^{17}$ and for $p = 4$, 7, and 13 in each plot. Left, center, and right panels are $vec = B_\epsilon$, $vec = U_\epsilon$, and $vec = N_\epsilon$ respectively with $\epsilon = 1/10$. Algorithms: NIHT (a-c), HTP (d-f), and CSMPSP (g-i).