Aalborg Universitet

AALBORG UNIVERSITY

Cyclic Pure Greedy Algorithms for Recovering Compressively Sampled Sparse Signals

Sturm, Bob L.; Christensen, Mads Græsbøll; Gribonval, Remi

Published in:
Asilomar Conference on Signals, Systems and Computers. Conference Record

DOI (link to publication from Publisher):
10.1109/ACSSC.2011.6190193

Publication date:
2011

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Sturm, B. L., Christensen, M. G., & Gribonval, R. (2011). Cyclic Pure Greedy Algorithms for Recovering Compressively Sampled Sparse Signals. *Asilomar Conference on Signals, Systems and Computers. Conference Record*, 1143-1147. https://doi.org/10.1109/ACSSC.2011.6190193

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: December 05, 2025

Cyclic Pure Greedy Algorithms for Recovering Compressively Sampled Sparse Signals

Bob L. Sturm
Department of Architecture,
Design and Media Technology
Aalborg University Copenhagen, Denmark
Email: bst@create.aau.dk

Mads G. Christensen
Department of Architecture,
Design and Media Technology
Aalborg University, Aalborg, Denmark
Email: mgc@create.aau.dk

Rémi Gribonval
INRIA Rennes - Bretagne Atlantique
Campus de Beaulieu
35042 Rennes Cedex France
Email: remi.gribonval@inria.fr

***Abstract*—The pure greedy algorithms matching pursuit (MP) and complementary MP (CompMP) are extremely computationally simple, but can perform poorly in solving the linear inverse problems posed by the recovery of compressively sampled sparse signals. We show that by applying a cyclic minimization principle, the performance of both are significantly improved while remaining computationally simple. Our simulations show that while MP and CompMP may not be competitive with state-of-the-art recovery algorithms, their cyclic variations are. We discuss ways in which their complexity can be further reduced, but our simulations show these can hurt recovery performance. Finally, we derive the exact recovery condition of CompMP and both cyclic algorithms.**

I. Introduction

Under certain conditions, we can recover a vector $\mathbf{x} \in \mathbb{C}^N$ from measurements $\mathbf{u} = \mathbf{\Phi x}$ where $\mathbf{\Phi} \in \mathbb{C}^{m \times N}$. Several methods have been developed and studied for recovering $\mathbf{x}$ from $\mathbf{u}$, which are obviously sensitive to both the size and content of $\mathbf{\Phi}$ relative to $\mathbf{x}$. The overdetermined problem ($N \leq m$) has been formally studied in frame theory [1]. The underdetermined problem ($N > m$), with $\mathbf{x}$ a *sparse* vector, has been studied in statistics, inverse problems, sparse approximation [1], [2], and compressed sensing [1], [3], [4].

We propose solving the noiseless underdetermined problem by a cyclic application of the pure greedy algorithms matching pursuit (MP) [2], [5], [6], and complementary MP (CompMP) [7]. A pure greedy algorithm has an extremely low computational complexity, but at the same time does not generally perform well in solving the above problem. This can be seen by a comparison of their empirical phase transitions [8], [9] (Fig. 1). A cyclic approach to recovery involves alternating between augmentation and refinement of the model, and here we employ a pure greedy algorithm for each to great effect.

Of course one should expect that such an approach will perform better than MP or CompMP, but we find the surprising result that the performance equals or exceeds those of more complex approaches while being more computationally simple. We show empirically that a pure greedy algorithm within a cyclic minimization framework, its performance can equal or exceed those of orthogonal MP (OMP) [2], [10], [11], and orthogonal least squares (OLS) [12], [13], and yet have a complexity in their kth iteration of $\mathcal{O}(Nk)$. In comparison, the complexity of OMP it is at least $\mathcal{O}(Nk + mk)$, and for OLS it is at least $\mathcal{O}(mN + mk)$, both using the QR decomposition approach. We compare their performance to six state-of-the-art recovery algorithms, and find CompCMP is especially competitive. We propose a variety of ways to further reduce their algorithmic complexity, and test their performance in relation to their parameters. Finally, we derive an exact recovery condition (ERC) of CompMP [7], from which naturally follow the ERC of a cyclic pure greedy algorithm.

II. Pure Greedy Algorithms

MP attempts to iteratively solve

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{u} = \mathbf{\Phi x} \tag{1}$$

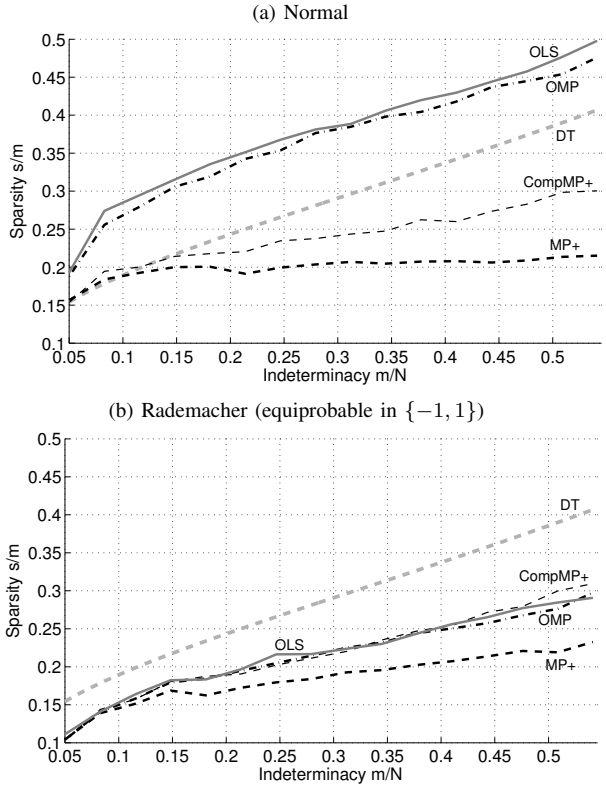

Fig. 1. Empirical phase transitions of four greedy algorithms (labeled) for two different sparse vector distributions (we describe their derivation in the experimental section). The region below a line shows where the majority of signals are successfully recovered by that algorithm. Successful recovery here implies that the true support is recovered. "DT" marks theoretical phase transition of ℓ_1-minimization [3], [8], [14].

where $\mathbf{\Phi} := [\boldsymbol{\varphi}_1 | \boldsymbol{\varphi}_2 | \cdots | \boldsymbol{\varphi}_N]$ consists of unit norm columns. Given the ordered index set Ω_k, which is a subset of $\Omega := \{1, 2, \ldots, N\}$ indexing the columns of $\mathbf{\Phi}$, MP augments Ω_k by $\Omega_{k+1} := \Omega_k \cup \{n_k\}$ using the selection criterion

$$n_k := \arg\min_{n \in \Omega} \|\mathbf{r}_k - \langle \mathbf{r}_k, \boldsymbol{\varphi}_n \rangle \boldsymbol{\varphi}_n\|_2^2 = \arg\max_{n \in \Omega} |\langle \mathbf{r}_k, \boldsymbol{\varphi}_n \rangle|^2 \tag{2}$$

where the residual $\mathbf{r}_k := \mathbf{u} - \mathbf{\Phi x}_k$. MP then updates the solution

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \langle \mathbf{r}_k, \boldsymbol{\varphi}_{n_k} \rangle \mathbf{e}_{n_k} \tag{3}$$

where $\mathbf{e}_k$ is the kth vector from the standard basis of $\mathbb{C}^N$, i.e., all zeros with a 1 in its kth row. For initialization, $\Omega_0 := \emptyset$ and $\mathbf{x}_0 := \mathbf{0}$. Usually, we can make MP stop once $\|\mathbf{r}_k\|$ drops below some threshold, or k exceeds a maximum number of iterations, e.g., twice the assumed sparsity of $\mathbf{x}$. MP has complexity $\mathcal{O}(N)$ if one

precomputes and stores the Gramian of $\mathbf{\Phi}$ [5].

Instead of (1), CompMP [7] attempts to iteratively solve

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = \mathbf{\Xi}\mathbf{x} \tag{4}$$

where $\mathbf{y} := \mathbf{\Phi}^H(\mathbf{\Phi}\mathbf{\Phi}^H)^{-1}\mathbf{u}$, and $\mathbf{\Xi} = [\xi_1|\xi_2|\cdots|\xi_N] := \mathbf{\Phi}^H(\mathbf{\Phi}\mathbf{\Phi}^H)^{-1}\mathbf{\Phi}$. As MP, CompMP augments Ω_k with the index

$$n_k := \arg\min_{n\in\Omega} \frac{\|\mathbf{y}_k - \langle \mathbf{y}_k, \xi_n\rangle \xi_n\|_2^2}{\|\xi_n\|_2^2} = \arg\max_{n\in\Omega} \frac{|\langle \mathbf{y}_k, \xi_n\rangle|^2}{\|\xi_n\|_2^2} \tag{5}$$

where $\mathbf{y}_k := \mathbf{\Phi}^H(\mathbf{\Phi}\mathbf{\Phi}^H)^{-1}[\mathbf{u} - \mathbf{\Phi}\mathbf{x}_k]$, and then updates the solution

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \frac{\langle \mathbf{y}_k, \xi_{n_k}\rangle}{\|\xi_{n_k}\|_2^2}\mathbf{e}_{n_k}. \tag{6}$$

As for MP, we initialize CompMP by $\Omega_0 := \emptyset$ and $\mathbf{x}_0 := \mathbf{0}$, and can use the same stopping conditions, and procedure for duplicate atoms.

Note that we cannot guarantee in general that $\mathbf{\Xi}$ consists of unit norm columns, and so CompMP must take their norms into account for the projections. If we define $\mathbf{\Psi} := \mathbf{\Xi}\mathbf{D}_{\mathbf{\Xi}}$, where $\mathbf{D}_{\mathbf{\Xi}}$ is a $N \times N$ diagonal full rank matrix, with its ith diagonal element defined $\|\xi_i\|_2^{-1}$, then each column of $\mathbf{\Psi}$ is unit norm. With $\mathbf{\Psi}$ and $\mathbf{y}$, then MP attempts to solve the problem $\min_{\mathbf{z}} \|\mathbf{z}\|_0$ subject to $\mathbf{y} = \mathbf{\Psi}\mathbf{z}$ from which we obtain the solution to (4) if we then define $\mathbf{x} := \mathbf{D}_{\mathbf{\Xi}}\mathbf{z}$. Thus, CompMP with $\mathbf{\Xi}$ on $\mathbf{y}$ is exactly the same as MP with $\mathbf{\Psi}$ on $\mathbf{y}$, ans a final weight adjustment step. In any case, $\text{Supp}(\mathbf{z}) = \text{Supp}(\mathbf{x})$ always since $\mathbf{D}_{\mathbf{\Xi}}$ is diagonal and full rank. From this, we see that each iteration of CompMP and MP have the same complexity.

The general stability and convergence properties of MP are well-known within the area of non-linear approximation [2], [5], [6], [15]. Let $\Omega^* := \{1, 2, \ldots, s\} \subset \Omega, s \leq m$, where, without loss of generality, the atoms a pure greedy algorithm must detect are the first s columns of $\mathbf{\Phi}$. We assume that $\mathbf{\Phi}_{\Omega^*}$, is made of the columns of $\mathbf{\Phi}$ indexed by Ω^*, and has full column rank s. Consider the vector space $\mathcal{U}_{\Omega^*} := \text{span}\{\boldsymbol{\varphi}_i\}_{i\in\Omega^*}$. Theorem 1 of [15] proves for every $\mathbf{u} \in \mathcal{U}_{\Omega^*}$, any solution assembled by a general MP algorithm involves only atoms indexed by Ω^* if

$$\max_{n\in\Omega\setminus\Omega^*} \|\mathbf{\Phi}_{\Omega^*}^{\dagger}\boldsymbol{\varphi}_n\|_1 < 1. \qquad (\text{ERC}_{\text{MP}}(\mathbf{\Phi}, \Omega^*))$$

The general stability and convergence properties of CompMP have yet to be formally studied, but from its near equivalence to MP shown above, and the fact that it is a general MP algorithm, it must have similar properties. For any $\mathbf{u} \in \mathcal{U}_{\Omega^*}$, we see that $\mathbf{y} := \mathbf{\Phi}^H(\mathbf{\Phi}\mathbf{\Phi}^H)^{-1}\mathbf{u} = \mathbf{\Psi}_{\Omega^*}\mathbf{z}'$, where $\mathbf{\Psi}_{\Omega^*}$ are the first s columns of $\mathbf{\Psi}$, and $\mathbf{z}'$ is made of the first s elements of $\mathbf{z}$. So, $\mathbf{y} \in \mathcal{Y}_{\Omega^*} := \text{Range}\{\mathbf{\Psi}_{\Omega^*}\}$. Now, given a $\mathbf{u} \in \mathcal{U}_{\Omega^*}$, any solution assembled by CompMP involves only atoms indexed by Ω^* if

$$\max_{n\in\Omega\setminus\Omega^*} \|\mathbf{\Psi}_{\Omega^*}^{\dagger}\boldsymbol{\psi}_n\|_1 < 1. \qquad (\text{ERC}_{\text{CompMP}}(\mathbf{\Psi}, \Omega^*))$$

The follows directly from the proof of Theorem 1 in [15], which comes from that of Theorem 3.1 in [11].

III. Greedy Algorithms with Cyclic Minimization

Given a set of estimated parameters $\Theta^{(0)} := \{\theta_i^{(0)}\}$, and an objective function $J(\Theta^{(0)})$ that maps the estimate quality to a scalar, the cyclic minimization principle [16] modifies the estimate of the jth parameter by holding constant all others and evaluating

$$\theta_j^{(1)} := \arg\min_{\theta} J(\ldots, \theta_{j-1}^{(1)}, \theta, \theta_{j+1}^{(0)}, \ldots). \tag{7}$$

Cycling through each parameter in this way, using all new and old estimates, one generates a new set of refined estimates $\Theta^{(1)}$. This principle guarantees that $J(\Theta^{(k+1)}) \leq J(\Theta^{(k)})$.

Cyclic Matching Pursuit Algorithm

For $k = 1$ to maximum model size
Augmentation
Add atom to model by MP (2) and (3)
Refinement
$\mathbf{r}_{k\setminus 0} \leftarrow \mathbf{r}_k$
For $i = 1$ to k
Remove atom i from residual and solution
$\mathbf{r}_{k\setminus i} := \mathbf{r}_{k\setminus i-1} + [\mathbf{x}_k]_{\Omega_k(i)}\boldsymbol{\varphi}_{\Omega_k(i)}$
$[\mathbf{x}_k]_{\Omega_k(i)} \leftarrow 0$
Replace atom i in model
$n_i := \arg\max_{n\in\Omega} |\langle \mathbf{r}_{k\setminus i}, \boldsymbol{\varphi}_n\rangle|^2$
$\Omega_k(i) \leftarrow n_i$
$\mathbf{x}_k \leftarrow \mathbf{x}_k + \langle \mathbf{r}_{k\setminus i}, \boldsymbol{\varphi}_{n_i}\rangle \mathbf{e}_{\Omega_k(i)}$
$\mathbf{r}_{k\setminus i} \leftarrow \mathbf{r}_{k\setminus i} - \langle \mathbf{r}_{k\setminus i}, \boldsymbol{\varphi}_{n_i}\rangle \boldsymbol{\varphi}_{n_i}$
$\mathbf{r}_{k\setminus 0} \leftarrow \mathbf{r}_{k\setminus k}$
Repeat until refinement stopping condition
$\mathbf{r}_k \leftarrow \mathbf{r}_{k\setminus k}$
Repeat until augmentation stopping condition

The cyclic minimization principle has been applied to MP for audio coding and modeling [17], [18], as well as to OLS [18]. Recent work [19] applies it to the compressibility of synthetic sparse signals using OLS and a complementary form of OLS. This work [19] is related to what we propose here, however, Dymarski et al. refine the atoms using cyclic minimization at the very end of assembling the signal model, while we propose alternating between atom selection and model refinement. Our experiments below show that the latter approach significantly outperforms the former.

A. Cyclic MP (CMP) and Complementary CMP (CompCMP)

The cyclic pure greedy algorithm (CMP) [17], runs as MP with $\mathbf{\Phi}$ at each iteration, but includes a solution refinement step that also runs as MP. Similarly, CompCMP runs as CompMP with $\mathbf{\Xi}$ (or equivalently MP with $\mathbf{\Psi}$) at every iteration, but includes a refinement step that also runs as CompMP (or MP). Since it requires only a trivial change of notation — $\boldsymbol{\varphi}$ becomes $\boldsymbol{\psi}$, and $\mathbf{r}_k := \mathbf{u} - \mathbf{\Phi}\mathbf{x}_k$ becomes $\mathbf{y}_k := \mathbf{\Phi}^H(\mathbf{\Phi}\mathbf{\Phi}^H)^{-1}\mathbf{u} - \mathbf{\Psi}\mathbf{z}_k$ — we present only CMP.

Define $\Omega_k(i)$ as the ith index in Ω_k, $1 \leq i \leq |\Omega_k|$, and $\mathbf{r}_{k\setminus 0} := \mathbf{u} - \mathbf{\Phi}\mathbf{x}_k$ as the residual before refinement. Starting with the first element of Ω_k, CMP removes $\boldsymbol{\varphi}_{\Omega_k(1)}$ from the residual by adding it back with the same weight with which it was taken

$$\mathbf{r}_{k\setminus 1} := \mathbf{r}_{k\setminus 0} + [\mathbf{x}_k]_{\Omega_k(1)}\boldsymbol{\varphi}_{\Omega_k(1)} \tag{8}$$

then removes it from the solution by $[\mathbf{x}_k]_{\Omega_k(1)} \leftarrow 0$, and finally finds the index of a replacement atom by the MP selection criterion (2):

$$n_1 := \arg\max_{n\in\Omega} |\langle \mathbf{r}_{k\setminus 1}, \boldsymbol{\varphi}_n\rangle|^2. \tag{9}$$

CMP then updates Ω_k such that $\Omega_k(1) \leftarrow n_1$, and adjusts the solution

$$\mathbf{x}_k \leftarrow \mathbf{x}_k + \langle \mathbf{r}_{k\setminus 1}, \boldsymbol{\varphi}_{n_1}\rangle \mathbf{e}_{\Omega_k(1)}. \tag{10}$$

Note that n_1 could be the same as the original index $\Omega_k(1)$, in which case only its weight is adjusted. Finally, the residual is updated by $\mathbf{r}_{k\setminus 1} \leftarrow \mathbf{r}_{k\setminus 1} - \langle \mathbf{r}_{k\setminus 1}, \boldsymbol{\varphi}_{n_1}\rangle \boldsymbol{\varphi}_{n_1}$, and the second atom is refined. CMP in general is shown above.

Refinement can be done any number of times for all elements of Ω_k. Once finished, CMP then augments Ω_k, and performs refinement cycles on the enlarged set. One can implement CMP to stop this refinement process after a certain number of cycles, or until the improvement between cycles decreases below a threshold. For

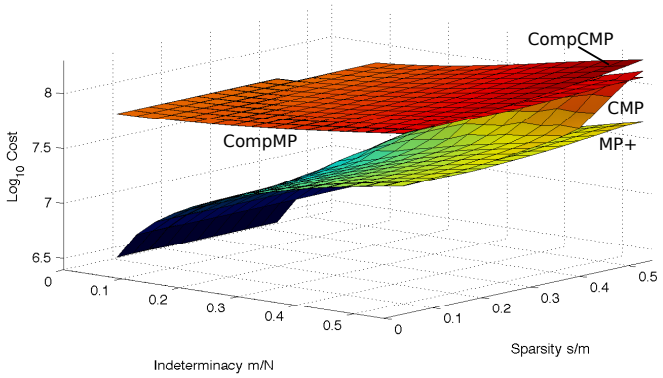

Fig. 2. Computational cost (number of multiplies) of MP+, CMP-1, CompMP, and CompCMP-1 over a portion of the phase space (lower).

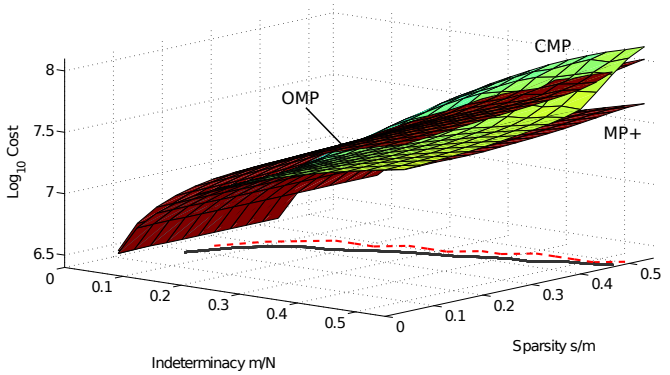

Fig. 3. Computational cost (number of multiplies) of MP+, CMP-1, OMP, over a portion of the phase space (lower). Black line marks empirical phase transition of OMP and CMP-1. Dotted red line marks region above which CMP becomes more expensive than OMP (implemented with QR).

instance, defining an error measure of the kth-order solution after l refinement cycles $J(\Omega_k) := ||\mathbf{u} - \mathbf{\Phi}\mathbf{x}_k||_2^2$, we can stop refinement when $J(\Omega_k') > \eta J(\Omega_k)$, $0 < \eta < 1$, where Ω_k' is the set after a refinement cycle. We can also stop CMP from augmenting the model when $J(\Omega_k) > \epsilon J(\Omega_{k-1})$ for $0 < \epsilon < 1$. Each model refinement and augmentation step of CMP (as well as CompCMP) never increases the residual ℓ_2-norm because of the atom selection criterion [18].

B. Computational Efficiency

When $\mathbf{\Phi}$ is small enough such that its Gramian can be precomputed and stored, we can efficiently perform the refinement cycles using the same tricks as for MP [5]. Define $\Gamma_l := \{\langle \boldsymbol{\varphi}_l, \boldsymbol{\varphi}_n \rangle\}_{n\in\Omega}$, which is the ordered set of the elements from the lth column of $\mathbf{\Phi}^H\mathbf{\Phi}$. In the lth refinement, $1 \le l \le |\Omega_k|$, CMP defines the intermediate set

$$\mathcal{I}'_{k\setminus l-1} := \{\mathcal{I}_{k\setminus l-1}(n) + x_{n_l}\Gamma_{n_l}(n)\}_{n\in\Omega} \tag{11}$$

where $\mathcal{I}_{k\setminus 0} := \mathcal{I}_k$. This operation requires $\mathcal{O}(N)$ multiplies and adds. CMP then sets $\alpha \leftarrow x_{n_l}$, and $x_{n_l} \leftarrow 0$. CMP then finds the index n_l' of the largest magnitude in $\mathcal{I}'_{k\setminus l-1}$, updates the set

$$\mathcal{I}_{k\setminus l} := \{\mathcal{I}'_{k\setminus l-1}(n) - \mathcal{I}'_{k\setminus l-1}(n_l')\Gamma_{n_l'}(n)\}_{n\in\Omega} \tag{12}$$

and updates the atom amplitude $x_{n_l'} \leftarrow x_{n_l'} + \mathcal{I}'_{k\setminus l-1}(n_l')$, and computes the new residual energy by $||\mathbf{r}_k||_2^2 \leftarrow ||\mathbf{r}_k||_2^2 + |\alpha|^2 + 2\mathrm{Re}\{\alpha\mathcal{I}_{k\setminus l-1}(n_l)\} - |\mathcal{I}_{k\setminus l}(n_l')|^2$. Once refinement is finished, CMP sets $\mathcal{I}_k \leftarrow \mathcal{I}_{k\setminus k}$, and augments the model with a new atom.

All of this takes $\mathcal{O}(N)$ multiplies and adds, and a search with cost at most $\mathcal{O}(N)$. Thus, each refinement cycle of a k order model takes $\mathcal{O}(Nk)$. If CMP refines the model a total of L times at each of s augmentations, the computational complexity is $\mathcal{O}\left(sN + \sum_{k=2}^{s} kLN\right) = \mathcal{O}(sN + NLs^2)$. For a naive implementation, the computational cost of each model augmentation is $\mathcal{O}(Nm + N)$, and for each refinement of a k-order model the total cost is $\mathcal{O}(Nmk+Nk)$. This brings its total computational complexity to $\mathcal{O}(Nms+NLms^2)$, for L refinements at each of s augmentations. The first approach thus provides savings, but still scales with s^2. We might overcome this problem by refining the model only after $K \ge 1$ additional augmentations (CMP-K), or only once the target sparsity is reached. Another possibility is refining the atom amplitudes while holding the atom indices constant. In this case, no search in the dictionary is needed, and CMP only changes the atom amplitudes by (11) and (12). Since CompCMP with $\mathbf{\Xi}$ on $\mathbf{u}$ is identical to CMP with $\mathbf{\Psi}$ on $\mathbf{y}$, it has the same computational complexity per iteration.

C. Stability Condition of CMP-K and CompCMP-K

Since CMP-K and CompCMP-K are general MP algorithms [15], they have the same stability conditions as MP and CompCMP, respectively. In short, as long as either ($\mathrm{ERC}_{\mathrm{MP}}(\mathbf{\Phi}, \Omega^*)$) or ($\mathrm{ERC}_{\mathrm{CompMP}}(\mathbf{\Psi}, \Omega^*)$) hold, all refinement steps of either MP or CompMP, respectively, will select atoms from those indexed by Ω^*.

IV. Experimental Results

In our tests, we investigate the performance of the cyclic algorithms in recovering the support of the true solution to $\mathbf{u} = \mathbf{\Phi}\mathbf{x}$, i.e., the locations of the non-zero values in $\mathbf{x}$. Our problem suite lives in an ambient dimension $N = 400$, and has 480 pairs of sparsities and indeterminaces: 30 linearly-spaced sparsities $\rho := s/m \in [0.05, 1]$, and 16 linearly-spaced indeterminacies $\delta := m/N \in [0.05, 0.5414]$. For each sparsity and indeterminacy pair, we sample $\mathbf{\Phi}$ from the uniform spherical ensemble [9], and find the proportion of 100 random vectors sensed by $\mathbf{\Phi}$ that are exactly recovered.

Our criteria for exact recovery is that the support of the recovered solution matches that of the true solution, i.e., no missed detections or false alarms. From these results, we find the empirical phase transition [8] by estimating the parameters of a linear model of the logistic of the frequencies [9]. To synthesize a s-sparse vector, we first determine indices of its non-zero elements by drawing at random and independently s members from $\Omega := \{1, 2, \ldots, N\}$. Then, we find the values at these indices by sampling from one of two distributions: Normal (and all magnitudes greater than 10^{-10}), and Rademacher, or equiprobable in $\{-1, 1\}$. The first distribution is favored by greedy algorithms, while the second appears to be the least favorable for all algorithms [9], [20].

We initialize every algorithm with the empty set, and set each one to perform a maximum of $2s$ augmentations, where s is the true signal sparsity. We debias the support of each solution from an algorithm by performing the following steps:

1) find indices of elements with magnitudes $> \min(m, 2s)$ largest
2) refine amplitudes of these elements by least squares projection
3) define solution support as the indices of elements with magnitudes above 10^{-10}.

Because of this addition, we denote MP as MP+, and CompMP as CompMP+. All algorithms are made to exit once either it has performed $2s$ iterations, or $\|\mathbf{u} - \mathbf{\Phi}\mathbf{x}_k\|^2/\|\mathbf{u}\|^2 < 10^{-10}$ for MP and CMP, or $\|\mathbf{y} - \mathbf{\Psi}\mathbf{z}_k\|^2/\|\mathbf{y}\|^2 < 10^{-10}$ for CompMP and CompCMP.

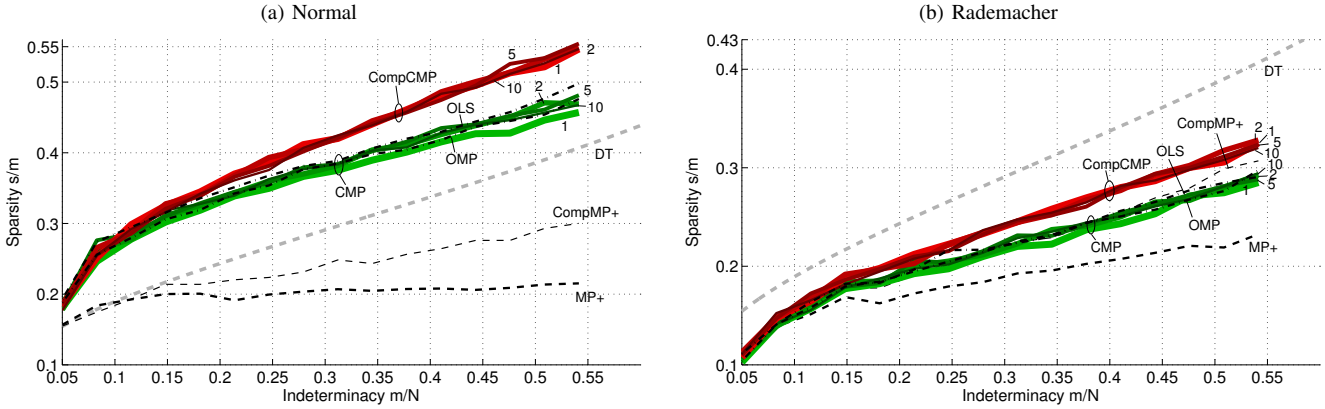

Fig. 4. Empirical phase transitions for CMP-1 (green) and CompCMP-1 (red) and differing number of refinement cycles L (labeled).

A. Computational Cost

In this experiment, we compare the computational complexities of all four algorithms, with respect to the number of multiplies, over the portion of the phase space which we are testing. We count the number of multiplies performed at each of 480 points, regardless of successful recovery, and average the results from 100 vectors. Figure 2 compares the complexities of all four algorithms. Though CompMP and CompCMP-1 have a high initial cost, they both scale remarkably well over the phase plane, and CompCMP shows nearly the same complexity as CompMP. Likewise, the complexity of CMP-1 stays close to that of MP+ until more iterations are required when the solution support is larger. In the same way, Fig. 3 compares the complexity of MP, CMP-1, and OMP. We see that though the empirical phase transitions of CMP-1 and OMP are practically identical, it is compleetly in the region where CMP is less expensive than OMP overall. Though each of its iterations have a lower computational complexity than that of OMP, CMP-1 can require many more iterations to reach the same model order.

B. Number of Refinement Cycles in CMP-1 and CompCMP-1

In this experiment, we force the cyclic algorithms to perform L refinements after each model augmentation. Figure 4 shows the resulting empirical phase transitions from each signal distribution. For even just a single refinement cycle, we see for Normal signals significant improvements in both algorithms over MP+ and CompMP+. When $L = 2$, the performance of CMP is comparable to that of OMP, but changes little for larger numbers of refinements. For only one refinement cycle, the performance of CompCMP exceeds that of OLS, but gains little with more refinements. The improvements are not as large for signals distributed Rademacher, however, the performance of CMP is comparable to that of OMP, and CompCMP still performs better than OLS.

C. Effect of K in CMP-K and CompCMP-K

In this experiment, we look at the effect of refining the model only every K augmentations. Each algorithm stops refinement when either $\|\mathbf{u} - \mathbf{\Phi}\mathbf{x}'_k\|^2 > (1 - 10^{-5})\|\mathbf{u} - \mathbf{\Phi}\mathbf{x}_k\|^2$ (or for the complementary approaches $\|\mathbf{y} - \mathbf{\Psi}\mathbf{z}'_k\|^2 > (1 - 10^{-5})\|\mathbf{y} - \mathbf{\Psi}\mathbf{z}_k\|^2$), where $\mathbf{x}_k$ and $\mathbf{x}'_k$, or $\mathbf{z}_k$ and $\mathbf{z}'_k$, are the solutions before after a refinement cycle, respectively, or when ten refinement cycles have been completed. Even with this maximum number of refinements to 10, neither algorithm refined the model that number of times. Figure 5 shows how the empirical phase transitions change for each signal distribution. We can clearly see that waiting several augmentations to refine the model reduces the computational complexity, but results in poorer recovery performance. Surprisingly, however, the performance of CMP is hurt less by this than CompCMP for signals distributed Normal.

D. Comparison with other recovery algorithms

In this final experiment, we compare the performance of CMP-1 and CompCMP-1 to the following collection of algorithms: MP+ and CompMP+; OMP [2], [10], [11]; orthogonal least squares (OLS) [12], [13]; "recommended versions" of iterative hard thresholding (IHT) and two-stage thresholding (TST) [9]; probabilistic OMP (PrOMP) [21]; smoothed ℓ_0 (SL0) [22]; subspace pursuit (SP) [23]; approximate message passing (AMP) [24]. The cyclic algorithms stop refinement when either it performs two refinement cycles (based on the results of experiment 1 above), or using the same ℓ_2-norm criteria in part C above. Figure 6 shows the empirical phase transitions for the two signal distributions. In both cases, CMP-1 performs nearly the same as OMP, but with smaller iteration-wise complexity: $\mathcal{O}(Nk)$ versus $\mathcal{O}(Nk + Mk)$ for the kth iteration. CompCMP-1 performs surprisingly well for Normal sparse signals, exceeding all others except SL0, and PrOMP at low indeterminaces. For Rademacher signals we see that CompMP still performs better than CompMP+, but all algorithms are here exceeded by AMP and ℓ_1 minimizations (DT). Experiments at larger ambient dimensions ($N \geq 1000$) show only small changes.

V. Conclusion

We have shown emprically that in solving the inverse problem posed by CS, the performance of the pure greedy algorithms MP and CompMP can be greatly improved with a cyclic minimization principle for refining the model after a number of model augmentations. In this way, we can maintain their computational simplicity while boosting their performance. Under certain conditions too, we see they perform better than many more complex algorithms for the recovery of sparse vectors from compressive measurements. However, it appears from our experiments that waiting to refine the model can significantly hurt performance. Our future work will look at the performance of these algorithms when the measurements are noisy, and alternative strategies for deciding which support elements should be refined in order to further reduce the complexity of the algorithms.

Acknowledgment

This work is supported in part by a grant from the French Ambassador to Denmark, *Afdelingen for Forsknings- og Universitetssamarbejde*, No. 31/2011-CSU 8.2.1. R. Gribonval acknowledges support from EU Framework 7 FET-Open project FP7-ICT-225913-SMALL.

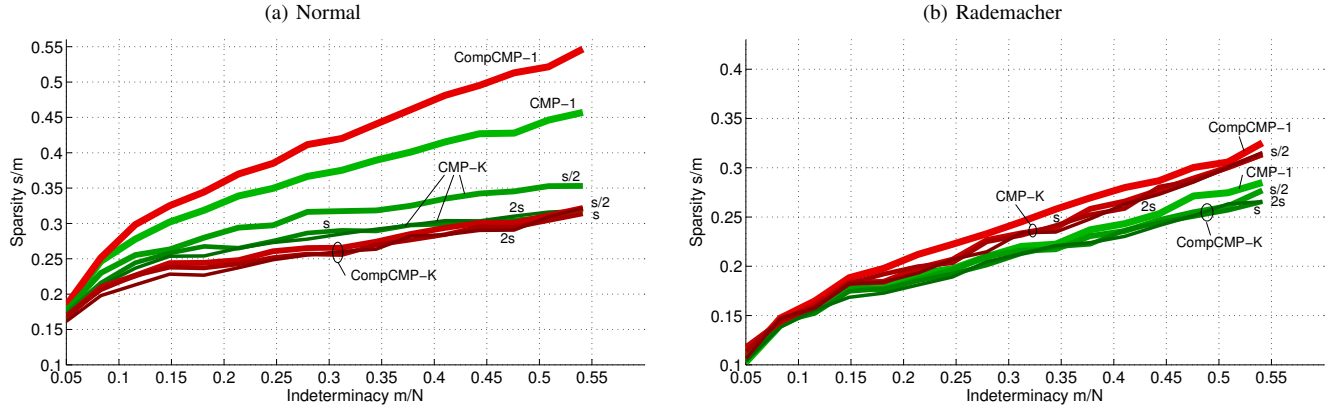

Fig. 5. Empirical phase transitions for CMP-K and CompCMP-K for several K as a function of signal sparsity s (labeled).

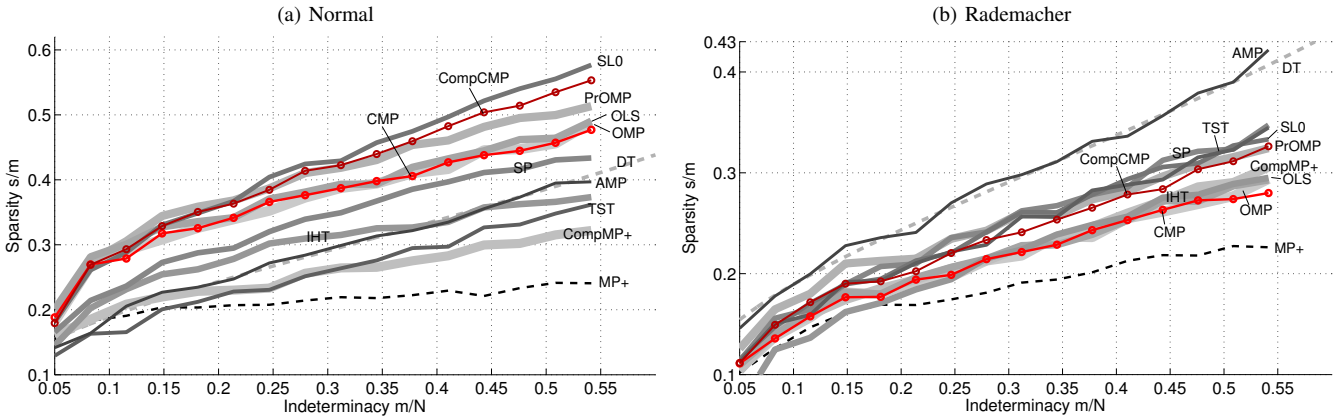

Fig. 6. Empirical phase transitions for several algorithms (labeled). Dashed line labeled 'DT' is theoretical phase transition for ℓ_1-minimization [3], [8], [14]

REFERENCES

[1] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, Elsevier, Amsterdam, 3rd edition, 2009.

[2] R. A. DeVore, "Nonlinear approximation," *Acta Numerica*, vol. 7, pp. 51–150, 1998.

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

[4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Info. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[6] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *J. Constr. Approx.*, vol. 13, no. 1, pp. 57–98, Jan. 1997.

[7] G. Rath and C. Guillemot, "A complementary matching pursuit algorithm for sparse approximation," in *Proc. European Signal Process. Conf.*, Lausanne, Switzerland, Aug. 2008.

[8] D. L. Donoho and J. Tanner, "Precise undersampling theorems," *Proc. IEEE*, vol. 98, no. 6, pp. 913–924, June 2010.

[9] A. Maleki and D. L. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," *IEEE J. Selected Topics in Signal Process.*, vol. 4, no. 2, pp. 330–341, Apr. 2010.

[10] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 1993, vol. 1, pp. 40–44.

[11] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Info. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[12] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Process. Lett.*, vol. 9, no. 4, pp. 137–140, Apr. 2002.

[13] T. Blumensath and M. E. Davies, "On the difference between orthogonal matching pursuit and orthogonal least squares," Tech. Rep., University of Edinburgh, Edinburgh, Scotland, U.K., Mar. 2007.

[14] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, Aug. 1998.

[15] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *IEEE Trans. Information Theory*, vol. 52, no. 1, pp. 255–261, 2006.

[16] P. Stoica and Y. Selén, "Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: A refresher," *IEEE Sig. Process. Mag.*, vol. 21, no. 1, pp. 112–114, Jan. 2004.

[17] M. G. Christensen and S. H. Jensen, "The cyclic matching pursuit and its application to audio modeling and coding," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2007.

[18] B. L. Sturm and M. Christensen, "Cyclic matching pursuit with multiscale time-frequency dictionaries," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2010.

[19] P. Dymarski, N. Moreau, and G. Richard, "Greedy sparse decompositions: a comparative study," *EURASIP Journal on Advances in Signal Processing*, Aug. 2011.

[20] B. L. Sturm, "Sparse vector distributions and recovery from compressed sensing," *arXiv:1103.6246v2*, July 2011.

[21] A. Divekar and O. Ersoy, "Probabilistic matching pursuit for compressive sensing," Tech. Rep., Purdue University, West Lafayette, IN, USA, May 2010.

[22] G. H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ_0 norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Jan. 2009.

[23] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[24] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. National Academy of the Sciences*, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.