# A MIXED INTEGER LINEAR PROGRAMMING FORMULATION FOR THE SPARSE RECOVERY PROBLEM IN COMPRESSED SENSING

*N. Burak Karahanoğlu*[a,b]*, Hakan Erdoğan*[b]*, and Ş. İlker Birbil*[b]

[a]Advanced Technologies Research Institute, TÜBİTAK-BİLGEM, Kocaeli, Turkey
[b]Faculty of Engineering and Natural Sciences, Sabancı University, Istanbul, Turkey

**ABSTRACT**

We propose a new mixed integer linear programming (MILP) formulation of the sparse signal recovery problem in compressed sensing (CS). This formulation is obtained by introduction of an auxiliary binary vector, where ones locate the recovered nonzero indices. Joint optimization for finding this auxiliary vector together with the underlying sparse vector leads to the proposed MILP formulation. By addition of a few appropriate constraints, this problem can be solved by existing MILP solvers. In contrast to other methods, this formulation is not an approximation of the sparse optimization problem, but is its equivalent. Hence, its solution is exactly equal to the optimal solution of the original sparse recovery problem, once it is feasible. We demonstrate this by recovery simulations involving different sparse signal types. The proposed scheme improves recovery over the mainstream CS recovery methods especially when the underlying sparse signals have constant amplitude nonzero elements.

***Index Terms—*** compressed sensing, $\ell_0$ norm minimization, sparse signal recovery, mixed integer linear programming, branch-and-cut algorithm

## 1. INTRODUCTION

In contrast to the conventional acquisition process, where a signal is captured as a whole before dimensionality reduction can be applied via transform coding, the rapidly emerging compressed sensing (CS) field targets acquisition of *sparse* or *compressible* signals directly in reduced dimensions. Concentrating on the sparse case, let us define $\mathbf{x} \in \mathrm{R}^{N\times 1}$ as a $K$-sparse signal, i.e. $\mathbf{x}$ has at most $K$ nonzero elements. Employing an observation matrix $\mathbf{\Phi} \in \mathrm{R}^{M\times N}$, where $K < M < N$, the "compressed" observations $\mathbf{y} \in \mathrm{R}^{M\times 1}$ can be obtained as

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x}. \tag{1}$$

This compressed observations introduce the fundamental problem of CS: Because of the dimensionality reduction via $M < N$, there exists infinitely many solutions for $\mathbf{x}$. To overcome this difficulty, (1) is casted as the sparsity-promoting optimization problem

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{x}\|_0 \\ \text{subject to} \quad & \mathbf{\Phi}\mathbf{x} = \mathbf{y}, \end{aligned} \tag{2}$$

where $\|\mathbf{x}\|_0$, called the $\ell_0$ norm by abuse of terminology, denotes the number of nonzero elements in $\mathbf{x}$. Since direct solution of (2) is computationally intractable, it has been shown that $\mathbf{x}$ can be recovered by alternative formulations [1, 2, 3, 4, 5] once the observation matrix $\mathbf{\Phi}$ satisfies the restricted isometry property (RIP) [1]. Random observation matrices, such as those with independent and identically distributed Gaussian or Bernoulli entries and random selections from the discrete Fourier transform, are common in CS, since these satisfy the RIP with high probabilities if $K$, $M$ and $N$ satisfy some specific conditions [6].

### 1.1. Related Work in Compressed Sensing

The CS literature contains a vast number of sparse signal recovery algorithms which exploit different properties of the underlying recovery problem, such as [4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15]. An insightful overview and classification of CS sparse signal recovery methods can be found in [16].

Among different classes of algorithms, convex relaxation [1, 5, 6, 17] replaces the $\ell_0$ minimization in (2) with its closest convex approximation, the $\ell_1$ minimization, which can be solved by convex optimization algorithms. Being the first convex relaxation algorithm, basis pursuit (BP) [17] solves the corresponding $\ell_1$ norm minimization problem by a primal-dual interior-point method. On the other hand, greedy pursuit algorithms [3, 7, 18, 19, 20, 21, 22, 23] provide iterative approximations of (2). Among these, orthogonal matching pursuit (OMP) [7] builds up the support of $\mathbf{x}$ by adding one element per iteration. subspace pursuit (SP) [3] and compressive sampling matching pursuit (CoSaMP) [2] apply two-stage iterative schemes, where each iteration first expands and then shrinks the support estimate by the same number of elements. Another two-stage scheme is proposed by forward-backward pursuit (FBP) [24], where the expansion is larger than the shrinkage, enlarging the support estimate iteratively. Iterative hard thresholding (IHT) [4] combines gradient descent with a thresholding step. Among other types of algorithms, $\mathrm{A}^{\star}$OMP [25, 26] is a semi-greedy approach, which employs best-first search in combination with OMP to recover $\mathbf{x}$. Smoothed $\ell_0$ (SL0) [27] is a non-convex procedure which minimizes a smoothed version of $\|\mathbf{x}\|_0$ by a series of optimizations where the quality of the approximation is gradually improved. Among iterative re-weighting algorithms, iterative support detection (ISD) [28] solves a series of re-weighted $\ell_1$ minimizations where only the indices of $\mathbf{x}$ out of the detected support are penalized.

### 1.2. Our Contributions

This paper proposes a mixed integer linear programming (MILP) model to solve an equivalent formulation of problem (2). Previous work in the field has examined the use of LP, however, to the best of our knowledge, MILP appears for the first time in this context. Moreover, the formulation we introduce here is not a relaxation of (2) as for the mainstream CS sparse signal recovery methods, but it is equivalent to (2). That is, the solution of the proposed formulation is exactly equal to the solution of the original sparse optimization problem, once it is feasible. To obtain this MILP formulation, we introduce an auxiliary binary vector $\mathbf{z}$ of length $N$, on which the nonzero indices of $\mathbf{x}$ are located by ones. Then, (2) can be casted into an equivalent MILP problem which is based on the joint optimization of $\mathbf{z}$ and $\mathbf{x}$. The feasibility of the solution is demonstrated by a number of simulations for recovery of sparse signals from noise-free measurements. These simulations not only reveal the performance of the proposed approach for recovery of sparse signals with different characteristics, but also compare it to a number of well-known algorithms in the field such as SP, BP, OMP, IHT, ISD, SL0 and A*OMP.

## 2. MILP FORMULATION OF THE CS SPARSE RECOVERY PROBLEM

The CS recovery problem in (2) may be considered as an optimization problem involving two subproblems which should be solved simultaneously: The first one of these problems is identifying the locations of the nonzero elements of $\mathbf{x}$, i.e. the support of $\mathbf{x}$, and the other is finding the values of these nonzero elements. An equivalent MILP formulation of (2) might be obtained by exploiting this basic observation.

### 2.1. Problem Formulation

Let $T$ be the support of $\mathbf{x}$, and $\mathbf{x}_T$ be the vector consisting of the elements of $\mathbf{x}$ indexed by $T$. Next, we define the auxiliary vector $\mathbf{z} = [z_1\ z_2\ \ldots\ z_N]^T$ to mark the nonzero locations of $\mathbf{x}$:

$$z_i = \begin{cases} 1, & \text{if } i \in T; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Now, our original problem (2) can be equivalently written as

$$\begin{aligned} \text{minimize} \quad & \mathbf{e}^T\mathbf{z} \\ \text{subject to} \quad & \mathbf{\Phi}\mathbf{x} = \mathbf{y}, \\ & c_l z_i \le x_i \le c_u z_i, \qquad i = 1, \cdots, N, \\ & z_i \in \{0, 1\}, \qquad i = 1, \cdots, N, \end{aligned} \tag{4}$$

where $\mathbf{e}$ is a vector of ones, and $c_l, c_u \in \mathrm{R}$ are chosen large enough so that the range $[c_l, c_u]$ covers all nonzero values in $\mathbf{x}$. The bound constraints given in the third line of (4) force the nonzero elements of $\mathbf{x}$ to appear only at the locations marked by $\mathbf{z}$.

Though (4) is already enough for finding the correct support of $\mathbf{x}$, we also define the sparsity constraint as

$$\mathbf{e}^T\mathbf{z} \le rM, \tag{5}$$

where $0 < r \le 1$. This constraint sets an upper limit on the sparsity of the recovered vector, hence reduces the feasible solution space. We discuss the choice of $r$ below.

Next, we define a combined representation to complete the MILP formulation of the CS sparse recovery problem. Let us first introduce the auxiliary vector

$$\mathbf{f} = [\mathbf{z}^T\mathbf{x}^T]^T \tag{6}$$

and the weight vector

$$\mathbf{w} = [\mathbf{e}^T \quad \mathbf{0}_{1\times N}]^T, \tag{7}$$

where $\mathbf{0}_{b\times c} \in \mathrm{R}^{b\times c}$ denotes a matrix consisting of zeros only. Using these, we write the MILP equivalent of the CS sparse signal recovery problem as

$$\begin{aligned} \text{minimize} \quad & \mathbf{w}^T\mathbf{f} \\ \text{subject to} \quad & \mathbf{A}_{\text{eq}}\mathbf{f} = \mathbf{b}_{\text{eq}}, \\ & \mathbf{A}_{\text{ineq}}\mathbf{f} \le \mathbf{b}_{\text{ineq}}, \\ & z_i \in \{0, 1\}, \qquad i = 1, \cdots, N, \end{aligned} \tag{8}$$

where

$$\mathbf{A}_{\text{eq}} = [\mathbf{0}_{M\times N} \quad \mathbf{\Phi}], \tag{9}$$

$$\mathbf{b}_{\text{eq}} = \mathbf{y}, \tag{10}$$

$$\mathbf{A}_{\text{ineq}} = \underbrace{\begin{bmatrix} -c_u & & \mathbf{0} & 1 & & \mathbf{0} \\ & \ddots & & & \ddots & \\ \mathbf{0} & & -c_u & \mathbf{0} & & 1 \\ c_l & & \mathbf{0} & -1 & & \mathbf{0} \\ & \ddots & & & \ddots & \\ \mathbf{0} & & c_l & \mathbf{0} & & -1 \\ 1 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix}}_{2N}, \tag{11}$$

$$\mathbf{b}_{\text{ineq}} = [\mathbf{0}_{1\times 2N} \quad rM]^T. \tag{12}$$

Note that (9) and (10) represent the observation constraint $\mathbf{\Phi}\mathbf{x} = \mathbf{y}$. The first $2N$ rows of (11) and (12) represent the constraints on the nonzero elements of $\mathbf{x}$, i.e. $c_l z_i \le x_i \le c_u z_i$, while the last rows of these correspond to the sparsity constraint in (5).

### 2.2. Practical Issues

In this work, we employ the IBM ILOG CPLEX optimization studio [29] to solve the MILP problem (8). In practice, (8) might take too long to solve, even when powerful solvers like CPLEX are employed. The parameters $c_l$, $c_u$ and $r$, which should be chosen properly, are very important for this purpose. We discuss these parameters below.

The parameters $c_l$ and $c_u$ define the range which the nonzero values of $\mathbf{x}$ are allowed to span. If the chosen range is narrower than the actual range for $\mathbf{x}$, recovery failure is obvious. On the other hand, if the range $[c_l, c_u]$ is chosen too wide, then the constraints are clearly not tight enough and they are not useful in reducing the size of the search tree employed in solving the MILP by the solver. Consequently, the computational effort increases along with the solution time. Hence, $c_l$ and $c_u$ should be chosen properly. Having said that, our main concern in this work is not finding the optimal $[c_l, c_u]$ range, but demonstrating application of MILP in the CS problem. Hence, we do not attempt at finding the optimal $c_l$ and $c_u$ range, but employ appropriate assumptions during the simulations.

The sparsity constraint (5) also plays an important role in practice. Note that $r = 1$ is a natural upper bound due to the problem definition. Choosing $r$ smaller, on the other hand, reduces the feasible solution space and therefore allows for faster termination of the algorithm. However, as for $c_l$ and $c_u$, $r$ should also not be chosen smaller than the actual sparsity level, since this makes the actual solution infeasible. For many practical applications, $K$ is not known *a priori*, however $K \ll M$ holds in general. In accordance, we choose $r = 0.5$, i.e. $\|\mathbf{x}\|_0 \le 0.5M$ in the simulations below, while this choice might be modified according to the *a priori* information about a particular recovery problem. Choosing $r = 0.5$ also provides another important advantage. Following the assumption that

$\mathbf{\Phi}$ is full row rank, this choice guarantees that the optimization problem has only one possible solution when $K \leq M/2$. [1] This allows us to configure the optimization parameters such that CPLEX returns the first solution it encounters, without running until the actual termination point where all MILP subproblems are covered. This results in faster termination of the algorithm.

## 3. SIMULATIONS

Below, we demonstrate the performance of MILP for the CS signal recovery problem in comparison to A$^\star$OMP, BP, SP, OMP and IHT. As discussed above, MILP problem is solved by running the "cplexmilp" optimizer of the CPLEX optimization studio [29] from the MATLAB environment. We set a time limit of 100 seconds on CPLEX for each recovery, and terminate optimization just after the first solution is found. That is, if no solution is found in 100 seconds, the algorithm fails. As discussed above, we set $r = 0.5$. The other algorithms are run using freely available software such as $\ell_1$−magic [30], Sparsify [31] and AStarOMP [32], Threshold-ISD [33] and the Matlab implementation of SL0 [34]. A$^\star$OMP and OMP are run using a residue-based termination criterion with $\varepsilon = 10^{-6}$. That is, they run until $\|\mathbf{r}\|_2 \leq \varepsilon \|\mathbf{y}\|_2$, where $\mathbf{r}$ denotes the residue of the observation $\mathbf{y}$. A$^\star$OMP parameters are set as $I = 3$, $B = 2$ and $P = 200$, and the Adaptive-Multiplicative cost model [26] is employed with $\alpha = 0.97$. For SL0, we decrement the smoothing parameter $\sigma$ slowly by 0.95 in order to decrease the risk of falling into local minima. The algorithms are run to recover sparse signals with different characteristics from noise-free measurements. Each test is repeated over 200 randomly generated sparse samples. The signal length is chosen as $N = 256$, while $M = 100$. The sparsity level $K$ varies in $[10, 50]$. For each test sample, the elements of $\mathbf{\Phi}$ are modelled as independent and identically distributed Gaussian random variables with mean 0 and standard deviation $1/N$. The recovery results are expressed in terms of Average Normalized Mean-Squared-Error (ANMSE) and the exact recovery rates. ANMSE is defined as

$$\text{ANMSE} = \frac{1}{100} \sum_{i=1}^{L} \frac{\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2}{\|\mathbf{x}_i\|_2^2} \tag{13}$$

where $\hat{\mathbf{x}}_i$ is the reconstruction of the $i$th test vector $\mathbf{x}_i$.

The tests involve sparse samples with different characteristics. The nonzero entries of these samples are selected from four different random ensembles. The nonzero entries of the so-called Gaussian sparse signals are drawn from the standard Gaussian distribution. Nonzero elements of uniform sparse signals are distributed uniformly in $[-1, 1]$. In addition to these, we consider two types of sparse signals with constant amplitude nonzero elements: The nonzero elements of the binary sparse signals are set to 1. Finally, the constant amplitude random sign (CARS) sparse signals, where the naming follows [35], involve nonzero elements with unit amplitude and random sign.

Figure 1(a) depicts the recovery results for binary sparse signals. For this case, we assume that the nonzero coefficients of $\mathbf{x}$ lie in $[0, 1]$, i.e. $c_l = 0$, and $c_u = 1$. The other algorithms are also provided with similar *a priori* information. Interestingly, we observe that once such *a priori* information is available, MILP formulation leads to exact recovery of all binary sparse signals with sparsity level $K \in [10, 50]$. In practice, this provides a clear advantage for problems, where the sparse signal is known to have nonzero elements with equal or close values. As for the CARS case, which is similar to the binary problem except the random sign, we set $c_l = 0$, and $c_u = 1$. Figure 1(b) depicts the superior recovery accuracy of MILP formulation for this case. We observe that the highest exact recovery rate is obtained by employing MILP. In addition, the ANMSE for the MILP formulation is exactly related to the exact recovery rate. That is, if MILP is able to find a solution in at most 100 seconds, this solution is correct. Otherwise, an empty solution is returned, and the normalized $\ell_2$ norm of the residue is equal to unity. Hence, the ANMSE becomes equal to one minus the exact recovery rate of the MILP formulation. This indicates that the solution found by MILP is exactly equal to the exact solution of the original $\ell_0$ minimization problem, as discussed above.

The recovery results for the Gaussian and uniform sparse signals are illustrated in Figures 1(c) and 1(d). For uniform sparse signals, we assume that the signal is known to lie in $[-1, 1]$, that is $c_l = -1$ and $c_u = 1$. For the Gaussian ensemble, we set $-c_l = c_u = \|x\|_\infty$. We observe that MILP formulation still yields the highest accuracy for uniform sparse signals, while A*OMP performs very close to it. When the nonzero entries are normally distributed, A*OMP has the highest recovery accuracy, while SL0 and ISD also perform better than MILP. Clearly, MILP performance degrades when the range which is spanned by the nonzero elements of the underlying sparse signals gets wider. Among the examples we considered, Gaussian sparse signals are ones with the widest span of nonzero elements, hence they constitute the case where MILP shows the worst performance.

In addition to the recovery accuracy, run times of the MILP optimization are also extremely important for the evaluation of the proposed approach. Most integer programming problems are naturally NP-hard. However, the average run times depicted in Table 1 state that the proposed formulation can be solved in reasonable time for the recovery of sparse signals having constant amplitude nonzero elements with appropriate assumptions which effectively reduce the feasible solution space. We observe that the run time increases when $K$ exceeds 40 for CARS case. This is due to the failed recoveries, for each of which the algorithm runs for 100 seconds. For cases where MILP formulation provides exact recovery of all signals, the run times are reasonable for many applications.

**Table 1**. Average run-time in seconds per sparse vector

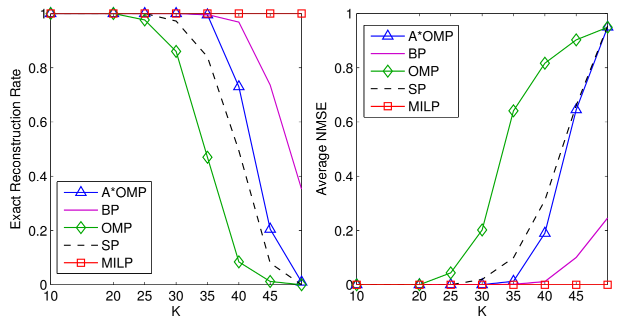
| | $K$ | | | | |
|---|---|---|---|---|---|
| | 10 | 20 | 30 | 40 | 50 |
| Binary | 0.18 | 0.19 | 0.19 | 0.2 | 0.34 |
| CARS | 0.23 | 0.27 | 0.37 | 22.1 | 81.9 |

## 4. CONCLUSIONS AND FUTURE WORK

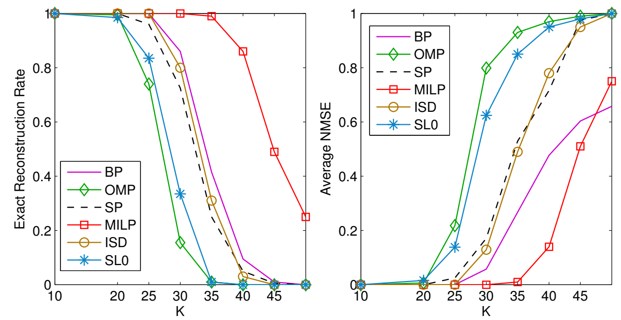
In this paper, we have concentrated on a new formulation for the sparse signal recovery problem. This formulation casts the problem into a MILP problem. Though MILP problems are mostly NP-hard, introduction of appropriate constraints help making it tractable for our case.

We demonstrated the sparse signal recovery performance of the proposed approach via a number of simulation experiments involving sparse signals with different characteristics. These simulations

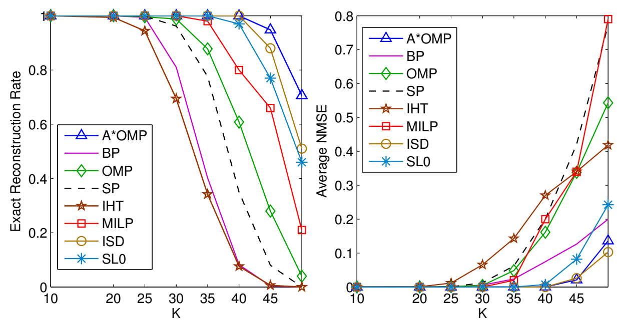
[1] This follows from the uniqueness of any $K$-sparse solution when $2K$-RIP is satisfied. See, for example, the discussion in [16].



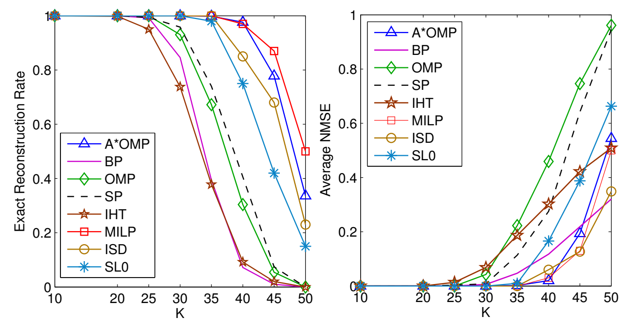

(a) Binary sparse signals




(b) CARS sparse signals




(c) Gaussian sparse signals




(d) Uniform sparse signals

**Fig. 1**. Average recovery results for the binary, CARS, uniform and Gaussian sparse signals. Each test is repeated over 100 random test samples. The signal length is 256, and the observation length is 100. The observation matrices are drawn from the Gaussian distribution.

indicate that the proposed approach yields high recovery rates when the underlying sparse signals have equal amplitude nonzero elements. Especially for binary sparse signals we have observed that the MILP formulation yields exact recovery until $K = M/2$ under some appropriate assumptions. Moreover, the algorithm is reasonably fast for such signals. The recovery accuracy of the proposed approach, however, begins to degrade when the nonzero elements vary in amplitude, in which case some other candidates yield similar or better recovery accuracy. Taking the complexity of the proposed algorithm also into account, we may conclude that the proposed approach is favorable for the recovery of sparse signals with constant or similar amplitude nonzero elements, especially the binary ones, where it provides both high recovery accuracy and reasonable termination speed.

Before concluding, we would like to note that future work on the constraints is necessary to take the full advantage of MILP in compressed sensing. Methods for finding tight bounds on the nonzero elements of the underlying sparse signals might especially be of interest. In addition, it is also worth investigating other possible constraints to further reduce the feasible solution space. One example of the latter might be structured sparsity, where the feasible solution space size may be further reduced by exploiting problem specific signal structures. In addition, our MILP reformulation is quite suitable to be solved with the well-known Benders decomposition technique [36] of integer programming. We believe implementing Benders decomposition would alleviate the computational burden, and hence, help us to solve much larger problems in shorter run times. We reserve such an implementation and the respective computational study for our future research. Finally, we believe that rapid advancements in computer hardware will be a vital key for the practical use of such methods in the near future.

## 5. REFERENCES


[1] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[2] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comp. Harmonic Anal.*, vol. 26, pp. 301–321, 2008.

[3] 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.

[4] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comp. Harmonic Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

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

[6] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[7] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, Los Alamitos, CA, 1993, vol. 1, pp. 40–44.

[8] D. P. Wipf and B. D. Rao, "Bayesian learning for sparse signal reconstruction," in *Acoustics, Speech and Signal Processing (ICASSP), 2003 IEEE International Conference on*, 2003, vol. 6.

[9] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Wavelet-domain compressive signal reconstruction using a hidden markov tree model," in *Acoustics, Speech and Signal Processing (ICASSP), 2008 IEEE International Conference on*, 2008, pp. 5137–5140.

[10] R. Chartrand, "Nonconvex compressed sensing and error correction," in *Acoustics, Speech and Signal Processing (ICASSP), 2007 IEEE International Conference on*, 2007, vol. 3.

[11] N. B. Karahanoğlu and H. Erdoğan, "Compressed sensing signal recovery via A$^\star$ orthogonal matching pursuit," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 3732 –3735.

[12] A. Kyrillidis, G. Puy, and V. Cevher, "Hard thresholding with norm constraints," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, Mar. 2012, pp. 3645 –3648.

[13] D. Malioutov and M. Malyutov, "Boolean compressed sensing: Lp relaxation for group testing," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, Mar. 2012, pp. 3305 –3308.

[14] A.Y. Carmi, L. Mihaylova, and D. Kanevsky, "Unscented compressed sensing," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, Mar. 2012, pp. 5249 –5252.

[15] H. Mansour and Ö. Yılmaz, "Support driven reweighted $\ell_1$ minimization," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, Mar. 2012, pp. 3309 –3312.

[16] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, June 2010.

[17] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. on Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

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

[19] S. Krstulovic and R. Gribonval, "Mptk: Matching pursuit made tractable," in *Acoustics, Speech and Signal Processing (ICASSP), 2006 IEEE International Conference on*, 2006, vol. 3.

[20] S. F. Cotter and B. D. Rao, "Application of tree-based searches to matching pursuit," in *Acoustics, Speech and Signal Processing (ICASSP), 2001 IEEE International Conference on*, 2001, pp. 3933–3936.

[21] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, Apr. 2008, pp. 3869–3872.

[22] T. Blumensath and M. E. Davies, "Gradient pursuits," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2370–2382, June 2008.

[23] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Simultaneous sparse approximation via greedy pursuit," in *Acoustics, Speech and Signal Processing (ICASSP), 2005 IEEE International Conference on*, Mar. 2005, vol. 5.

[24] N. B. Karahanoğlu and H. Erdoğan, "Forward-backward search for compressed sensing signal recovery," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, aug. 2012.

[25] N. B. Karahanoğlu and H. Erdoğan, "A$^\star$ orthogonal matching pursuit: Best-first search for compressed sensing signal recovery," *Digital Signal Processing*, vol. 22, no. 4, pp. 555–568, July 2012.

[26] N. B. Karahanoğlu and H. Erdoğan, "A comparison of termination criteria for A$^\star$OMP," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, Aug. 2012.

[27] C. Jutten, H. Mohimani, and M. Babaie-Zadeh, "A fast approach for overcomplete sparse decomposition based on smoothed $\ell_0$ norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Jan. 2009.

[28] Y. Wang and W. Yin, "Sparse signal reconstruction via iterative support detection," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 462–491, 2010.

[29] "IBM ILOG CPLEX Optimization Studio V12.4," http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[30] E. Candès and J. Romberg, "$\ell_1$-magic: Recovery of sparse signals via convex programming," Tech. Rep., California Institute of Technology, Oct. 2005.

[31] T. Blumensath, "Sparsify software package," available at http://users.fmrib.ox.ac.uk/ tblumens/sparsify/sparsify.html.

[32] N. B. Karahanoğlu, "AStarOMP software package," available at http://myweb.sabanciuniv.edu/karahanoglu/research/.

[33] "Threshold-ISD software package," available at http://www.caam.rice.edu/ $\sim$optimization/L1/ISD/.

[34] "SL0 matlab code," available at http://ee.sharif.edu/ SLzero/.

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

[36] R. K. Martin, *Large Scale Linear and Integer Optimization: A Unified Approach*, Kluwer Academic Publishers, Boston, 1999.