SPARSE SIGNAL RECOVERY FROM NONLINEAR MEASUREMENTS

Amir Beck

Faculty of Industrial Engineering and Management
Technion, Haifa, Israel
becka@ie.technion.ac.il

Yonina C. Eldar

Faculty of Electrical Engineering
Technion, Haifa, Israel
yonina@ee.technion.ac.il

ABSTRACT

We treat the problem of minimizing a general continuously differentiable function subject to sparsity constraints. We present and analyze several different optimality criteria which are based on the notions of stationarity and coordinate-wise optimality. These conditions are then used to derive three numerical algorithms aimed at finding points satisfying the resulting optimality criteria: the iterative hard thresholding method and the greedy and partial sparse-simplex methods. The theoretical convergence of these methods and their relations to the derived optimality conditions are studied.

1. INTRODUCTION

Sparsity has long been exploited in signal processing, statistics and computer science. Recent years have witnessed a growing interest in algorithms for sparse recovery [3, 2, 16]. Despite the great interest in exploiting sparsity in various applications, most of the work to date has focused on recovering a sparse vector $\mathbf{x} \in \mathbb{R}^n$ from linear measurements of the form $\mathbf{b} = \mathbf{A}\mathbf{x}$. For example, the rapidly growing field of compressed sensing [7, 6, 10] considers recovery of a sparse $\mathbf{x}$ from a small set of linear measurements $\mathbf{b} \in \mathbb{R}^m$ where $m \ll n$.

In this paper we study the more general problem of minimizing a continuously differentiable objective function subject to a sparsity constraint. More specifically, we consider the problem

$$\text{(P):} \quad \min f(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{x}\|_0 \leq s$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function which we assume throughout is lower bounded, $s > 0$ is an integer smaller than n and $\|\mathbf{x}\|_0$ is the ℓ_0 norm of $\mathbf{x}$, which counts the number of nonzero components in $\mathbf{x}$. We do not assume that f is convex. This, together with the fact that the constraint function is nonconvex, and is not even continuous, renders the problem quite difficult. Our goal is to study necessary optimality conditions for problem (P) and develop algorithms that find points satisfying these conditions for general f.

Two examples of (P) that have been considered previously are compressed sensing and phase retrieval. As noted above, compressed sensing is concerned with recovery of a sparse vector $\mathbf{x}$ from linear measurements $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{b} \in \mathbb{R}^m$ and $m \ll n$. When noise is present in the measurements, it is natural to consider problem (P) with $f_{\text{LI}}(\mathbf{x}) \equiv \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$. A variety of algorithms have been proposed to approximate the solution to this problem [15, 16]. One approach is to replace the ℓ_0 norm with the convex ℓ_1 norm. Greedy methods are also popular, such as the matching pursuit (MP) and orthogonal MP (OMP) algorithms [12]. Another technique that is related to our approach below is the iterative hard thresholding (IHT) algorithm [5]. In [5] the authors consider a majorization-minimization approach to solve (P) with $f = f_{\text{LI}}$. In Section 3.1 we show how this approach can be applied to the general formulation (P), and discuss the quality of the limit points of the sequence generated by the algorithm.

Although linear measurements are pervasive, recently, attention has been given to quadratic measurements. Sparse recovery problems from quadratic measurements arise in a variety of different problems in optics, including sub-wavelength optical imaging [8, 14] and phase retrieval. Quadratic compressed sensing was recently proposed in [14], where the goal is to recover a sparse vector $\mathbf{x}$ from noisy quadratic measurements: $\mathbf{c}_i \approx \mathbf{x}^T\mathbf{A}_i\mathbf{x}$. The resulting problem can be written as in (P) with $f_{\text{QU}}(\mathbf{x}) \equiv \sum_{i=1}^m \left(\mathbf{x}^T\mathbf{A}_i\mathbf{x} - c_i\right)^2$. In this case, the objective function is nonconvex and quartic. In phase retrieval problems, a vector $\mathbf{x}$ is to be recovered from the magnitude of its measurements $y_i = |\mathbf{d}_i^*\mathbf{x}|$. Denoting by b_i the corresponding noisy measurements, the goal is to minimize $\sum_{i=1}^m (b_i^2 - |\mathbf{d}_i^*\mathbf{x}|^2)^2$ subject to $\|\mathbf{x}\|_0 \leq s$ for some s. In [14], an algorithm was developed to treat such problems based on a semidefinite relaxation, and low-rank matrix recovery. Similar approaches were recently proposed in [9, 13]. However, for large scale problems, these methods are inefficient and difficult to implement.

In this paper we present a uniform approach to treating problems of the form (P). In Section 2 we derive 3 classes of necessary optimality conditions: basic feasibility, L-stationarity, and coordinate-wise (CW) optimality. We then show that CW-optimality implies L-stationarity for suitable values of L, and they both imply basic feasibility. In Section 3 we present two classes of algorithms for solving (P). The first is a generalization of IHT, and is based on the notion of L-stationarity. The second class of methods are based on the concept of CW-optimality. These are coordinate descent type algorithms which update the support at each iteration by one or two variables. Due to their resemblance with the celebrated simplex method for linear programming, we refer to these methods as "sparse-simplex" algorithms. As we show, these algorithms are as simple as IHT, while obtaining stronger optimality guarantees.

Throughout the paper we state our results without proofs; detailed proofs of the theorems can be found in [1].

2. NECESSARY OPTIMALITY CONDITIONS

Throughout, we denote by $\mathbf{x}_R$ the subvector of $\mathbf{x}$ corresponding to the indices in R. The support set of $\mathbf{x}$ is defined by $I_1(\mathbf{x}) \equiv \{i : x_i \neq 0\}$, and its complement is $I_0(\mathbf{x}) \equiv \{i : x_i = 0\}$. We denote the set of s-sparse vectors by $C_s = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq s\}$. The ith largest absolute value component in $\mathbf{x}$ is denoted by $M_i(\mathbf{x})$, so that in particular $M_1(\mathbf{x}) = \max_{i=1,\ldots,n} |x_i|$ and $M_n(\mathbf{x}) =$

This work is supported by the Israel Science Foundation under Grants no. 170/10 and 253/12, by the Ollendorf Foundation, and by a Magnet grant Metro450 from the Israel Ministry of Industry and Trade.

$\min_{i=1,\ldots,n} |x_i|$.

2.1. Basic Feasibility

As a first step in studying (P), we consider its optimality conditions, and then use them to generate algorithms. Since (P) is nonconvex, it does not seem to posses necessary and sufficient conditions for optimality. Therefore, we derive several necessary conditions, and analyze the relationship between them. We will then show in Section 3 how these conditions lead to algorithms that are guaranteed to generate a point satisfying the respective conditions.

For unconstrained differentiable problems, a necessary optimality condition is that the gradient is zero. It is therefore natural to expect that a similar necessary condition will be true over the support $I_1(\mathbf{x}^*)$ of an optimal point $\mathbf{x}^*$. Inspired by linear programming terminology, we will call a vector satisfying this property a *basic feasible* vector.

Definition 2.1. *A vector $\mathbf{x}^* \in C_s$ is called a basic feasible (BF) vector of (P) if:*

1. *when $\|\mathbf{x}^*\|_0 < s$, $\nabla f(\mathbf{x}^*) = 0$;*
2. *when $\|\mathbf{x}^*\|_0 = s$, $\nabla_i f(\mathbf{x}^*) = 0$ for all $i \in I_1(\mathbf{x}^*)$.*

Theorem 2.1. *Let $\mathbf{x}^*$ be an optimal solution of (P). Then $\mathbf{x}^*$ is a BF vector.*

It turns out that BF is a weak necessary condition, namely, there are many BF points that are not optimal points. We next consider stricter necessary optimality conditions.

2.2. L-Stationarity

In this subsection we consider L-stationarity, which is an extension of the concept of stationarity for convex constrained problems. We begin by recalling some well known concepts on optimality for convex constrained differentiable problems [4].

Consider a problem of the form $\min\{g(\mathbf{x}) : \mathbf{x} \in C\}$ where C is a closed convex set and g is a continuously differentiable function, which is possibly nonconvex. A vector $\mathbf{x}^* \in C$ is called stationary if

$$\langle \nabla g(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0 \text{ for all } \mathbf{x} \in C. \tag{2.1}$$

If $\mathbf{x}^*$ is an optimal solution of (P), then it is also stationary. Therefore, stationarity is a necessary condition for optimality. It is often useful to use the property that *for any $L > 0$*, a vector $\mathbf{x}^*$ is a stationary point if and only if

$$\mathbf{x}^* = P_C\left(\mathbf{x}^* - \frac{1}{L}\nabla g(\mathbf{x}^*)\right), \tag{2.2}$$

where for a closed subset $D \subseteq \mathbb{R}^n$, $P_D(\mathbf{y}) \equiv \operatorname{argmin}_{\mathbf{x}\in D} \|\mathbf{x}-\mathbf{y}\|^2$.

It is natural to try and extend (2.1) or (2.2) to the nonconvex (feasible set) setting. Condition (2.1) with $g = f$ and $C = C_s$ is actually not a necessary optimality condition so we do not pursue it further. To extend (2.2) to the sparsity constrained problem (P), we introduce the notion of "L-stationarity".

Definition 2.2. *A vector $\mathbf{x}^* \in C_s$ is called an L-stationary point of (P) if it satisfies the relation*

$$[\mathrm{NC}_L] \quad \mathbf{x}^* \in P_{C_s}\left(\mathbf{x}^* - \frac{1}{L}\nabla f(\mathbf{x}^*)\right). \tag{2.3}$$

Below we will show that under an appropriate Lipschitz condition, L-stationarity is a necessary condition for optimality. We first describe a more explicit representation of $[\mathrm{NC}_L]$.

Lemma 2.1. *For any $L > 0$, $\mathbf{x}^*$ satisfies $[\mathrm{NC}_L]$ if and only if $\|\mathbf{x}^*\|_0 \leq s$ and*

$$|\nabla_i f(\mathbf{x}^*)| \begin{cases} \leq LM_s(\mathbf{x}^*) & \text{if } i \in I_0(\mathbf{x}^*), \\ = 0 & \text{if } i \in I_1(\mathbf{x}^*). \end{cases} \tag{2.4}$$

A direct result of Lemma 2.1 is the following:

Corollary 2.1. *Suppose that $\mathbf{x}^*$ is an L-stationary point for some $L > 0$. Then $\mathbf{x}^*$ is a BF point.*

In general, L-stationarity is not a necessary optimality condition for problem (P). To establish such a result, we need to assume a Lipschitz continuity property of ∇f:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L(f)\|\mathbf{x} - \mathbf{y}\| \quad \text{for every } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \tag{2.5}$$

This assumption holds for $f = f_{\mathrm{LI}}$ with $L(f) = 2\lambda_{\max}(\mathbf{A}^T\mathbf{A})$, but not for $f = f_{\mathrm{QU}}$. We will *not* make this assumption throughout the paper; it will be stated explicitly when needed.

Under (2.5) we now show that an optimal solution of (P) is an L-stationary point for any $L > L(f)$.

Theorem 2.2. *Suppose that* (2.5) *holds, $L > L(f)$ and let $\mathbf{x}^*$ be an optimal solution of (P). Then*

(i) $\mathbf{x}^*$ is an L-stationary point.

(ii) The set $P_{C_s}\left(\mathbf{x}^* - \frac{1}{L}\nabla f(\mathbf{x}^*)\right)$ is a singleton.

To conclude this section, we have shown that under a Lipschitz condition on ∇f, L-stationarity for any $L > L(f)$ is a necessary optimality condition, which also implies the basic feasibility property. In Section 3.1 we will show how IHT for solving the general problem (P), can be used in order to find L-stationary points (for $L > L(f)$).

2.3. Coordinate-Wise Minima

The L-stationarity condition has two major drawbacks: first, it requires the gradient to be Lipschitz continuous and second, in order to validate it, we need to know a bound on the Lipschitz constant. We now consider a different and stronger necessary optimality condition that does not even require (2.5) to hold.

For a general unconstrained optimization problem, a vector $\mathbf{x}^*$ is called a "coordinate-wise (CW)" minimum if for every $i = 1, 2, \ldots, n$ the scalar x_i^* is a minimum of f with respect to the ith component x_i while keeping all other variables fixed:

$$x_i^* \in \operatorname{argmin} f(x_1^*, \ldots, x_{i-1}^*, x_i, x_{i+1}^*, \ldots, x_n^*).$$

Clearly, any optimal $\mathbf{x}^*$ is also a coordinate-wise minimum. It is therefore natural to extend this definition to problem (P).

Definition 2.3. *Let $\mathbf{x}^*$ be a feasible solution of (P). Then $\mathbf{x}^*$ is called a coordinate-wise (CW) minimum of (P) if one of the following cases hold true:*
Case I: $\|\mathbf{x}^*\|_0 < s$ and for every $i = 1, 2, \ldots, n$ one has:

$$f(\mathbf{x}^*) = \min_{t\in\mathbb{R}} f(\mathbf{x}^* + t\mathbf{e}_i). \tag{2.6}$$

Case II: $\|\mathbf{x}^*\|_0 = s$ and for every $i \in I_1(\mathbf{x}^*)$ and $j = 1, 2, \ldots, n$ one has:

$$f(\mathbf{x}^*) \leq \min_{t\in\mathbb{R}} f(\mathbf{x}^* - x_i^*\mathbf{e}_i + t\mathbf{e}_j). \tag{2.7}$$

Based on this definition, we have the following result.

Theorem 2.3. *Let $\mathbf{x}^*$ be an optimal solution of (P). Then $\mathbf{x}^*$ is a CW-minimum of (P). Furthermore, if $\mathbf{x}^* \in C_s$ is a CW-minimum of (P), then $\mathbf{x}^*$ is also a BF vector.*

We next show that being a CW-minimum is a stronger, i.e. more restrictive, condition than being L-stationary for any $L \geq L(f)$. To state this result, we note that under (2.5), for any $i \neq j$ there exists a constant $L_{i,j}(f)$ for which

$$\|\nabla_{i,j} f(\mathbf{x}) - \nabla_{i,j} f(\mathbf{x}+\mathbf{d})\| \leq L_{i,j}(f)\|\mathbf{d}\|, \qquad (2.8)$$

for any $\mathbf{x} \in \mathbb{R}^n$ and any $\mathbf{d} \in \mathbb{R}^n$ which has at most two nonzero components. Here $\nabla_{i,j} f(\mathbf{x})$ denotes a vector of length-2 whose elements are the ith and jth elements of $\nabla f(\mathbf{x})$. Define *the local Lipschitz constant*:

$$L_2(f) \equiv \max_{i \neq j} L_{i,j}(f).$$

Clearly (2.8) is satisfied when replacing $L_{i,j}(f)$ by $L(f)$. Therefore, in general, $L_2(f) \leq L(f)$.

Theorem 2.4. *Suppose that* (2.5) *holds and let* $\mathbf{x}^*$ *be a CW-minimum of (P). Then* $\mathbf{x}^*$ *is an* $L_2(f)$*-stationary point. Furthermore, any optimal solution of (P) is an* $L_2(f)$*-stationary point.*

3. NUMERICAL ALGORITHMS

We now develop two classes of algorithms that achieve the necessary conditions defined in the previous section: Iterative hard thresholding (IHT) and sparse-simplex methods.

3.1. The IHT Method

One approach for solving problem (P) is to employ the following fixed point method in order to "enforce" the L-stationary condition (2.3):

$$\mathbf{x}^{k+1} \in P_{C_s}\left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right), \quad k = 0, 1, 2, \ldots \qquad (3.1)$$

Convergence of this method can be shown when (2.5) holds; we therefore make this assumption when using this approach.

The IHT method
Input: a constant $L \geq L(f)$.

- **Initialization:** Choose $\mathbf{x}_0 \in C_s$.
- **General step :** $\mathbf{x}^{k+1} \in P_{C_s}\left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right)$

For the case $f \equiv f_{\text{LI}}$, and under the assumption that $\|\mathbf{A}\|_2 < 1$, our algorithm coincides with IHT [5]. Our approach extends this algorithm to the general case under (2.5).

The following theorem states that all accumulation points of the sequence generated by the IHT method with constant stepsize $\frac{1}{L}$ are indeed L-stationary points.

Theorem 3.1. *Let* $\{\mathbf{x}^k\}_{k\geq 0}$ *be the sequence generated by the IHT method with stepsize* $\frac{1}{L}$ *where* $L > L(f)$*. Then any accumulation point of* $\{\mathbf{x}^k\}_{k\geq 0}$ *is an* L*-stationary point.*

3.2. The Greedy Sparse-Simplex Method

The IHT method is able to find L-stationary points for any $L > L(f)$ under (2.5). However, by Theorem 2.4, any optimal solution is also an $L_2(f)$-stationary point, and $L_2(f)$ can be significantly smaller than $L(f)$. It is therefore natural to seek a method that is able to generate such points. An even better approach would be to derive an algorithm that converges to a CW-minima, which by Theorem 2.4, is a stronger notion than L-stationary. An additional drawback of IHT is that it requires the validity of (2.5) and the knowledge of $L(f)$.

Below we present *the greedy sparse-simplex (GSS) algorithm* which overcomes the faults of IHT alluded to above: its limit points are CW-minima, it does not require the validity of (2.5), but if the assumption does hold, then its limit points are $L_2(f)-$stationary points (without the need to know any information on Lipschitz constants).

The Greedy Sparse-Simplex Method
- **Initialization:** Choose $\mathbf{x}_0 \in C_s$.
- **General step :** ($k = 0, 1, \ldots$)
- If $\|\mathbf{x}^k\|_0 < s$, then compute for every $i = 1, 2, \ldots, n$

$$t_i \in \operatorname*{argmin}_{t\in\mathbb{R}} f(\mathbf{x}^k + t\mathbf{e}_i), \qquad (3.2)$$
$$f_i = \min_{t\in\mathbb{R}} f(\mathbf{x}^k + t\mathbf{e}_i).$$

Let $i_k \in \operatorname*{argmin}_{i=1,\ldots,n} f_i$. If $f_{i_k} < f(\mathbf{x}^k)$, then set

$$\mathbf{x}^{k+1} = \mathbf{x}^k + t_{i_k}\mathbf{e}_{i_k}.$$

Otherwise, STOP.
- If $\|\mathbf{x}^k\|_0 = s$, then for every $i \in I_1(\mathbf{x}^k)$ and $j = 1, \ldots, n$ compute

$$t_{i,j} \in \operatorname*{argmin}_{t\in\mathbb{R}} f(\mathbf{x}^k - x_i^k\mathbf{e}_i + t\mathbf{e}_j), \qquad (3.3)$$
$$f_{i,j} = \min_{t\in\mathbb{R}} f(\mathbf{x}^k - x_i^k\mathbf{e}_i + t\mathbf{e}_j).$$

Let $(i_k, j_k) \in \operatorname{argmin}\{f_{i,j} : i \in I_1(\mathbf{x}^k), j = 1, \ldots, n\}$. If $f_{i_k,j_k} < f(\mathbf{x}^k)$, then set

$$\mathbf{x}^{k+1} = \mathbf{x}^k - x_{i_k}^k\mathbf{e}_{i_k} + t_{i_k,j_k}\mathbf{e}_{j_k}.$$

Otherwise, STOP.

Theorem 3.2 establishes a convergence result for GSS:

Theorem 3.2. *Let* $\{\mathbf{x}^k\}$ *be the sequence generated by the GSS method. Then any accumulation point of* $\{\mathbf{x}^k\}$ *is a CW-minimum of (P).*

Combining Theorem 3.2 with Theorem 2.4 leads to the following corollary.

Corollary 3.1. *Suppose that* (2.5) *holds and let* $\{\mathbf{x}^k\}$ *be the sequence generated by the GSS method. Then any accumulation point of* $\{\mathbf{x}^k\}$ *is an* $L_2(f)$*-stationary point.*

When $f \equiv f_{\text{LI}}$, GSS is similar to MP. Specifically, it can be readily shown that MP coincides with our algorithm as long as the support is smaller than s. Our approach however has several advantages:

- We do not need to initialize it with a zero vector. It is possible to improve its performance by using several starting points and then choosing the solution with minimal objective function value;
- In MP once an index is added to the support it will generally not be removed. Our approach allows to remove elements from the support under broad conditions, allowing for an inherent "correction" scheme;
- In MP the algorithm stops once the maximal support is achieved. In contrast, in our approach, further iterations are made by utilizing the correction mechanism.

We note that once our method converges to a fixed support set, it continues to update the values on the support. Ultimately,

it converges to the least-squares solution on the support since in this situation the method is a simple coordinate descent method employed on a convex function. This is similar in spirit to the OMP approach [11].

3.2.1. Examples

Example 3.1. Consider the sparse least squares problem

$$(P_2) \quad \min\{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 : \mathbf{x}\in C_2\},$$

where $\mathbf{A}\in\mathbb{R}^{4\times 5}$ and $\mathbf{b}\in\mathbb{R}^4$ are given by:

$$\mathbf{A}=\begin{pmatrix} 0.889 & -0.435 & 0.530 & -0.232 & 0.374\\ 0.079 & -0.347 & 0.094 & 0.968 & -0.491\\ 0.442 & 0.324 & 0.692 & 0.092 & 0.757\\ 0.077 & 0.764 & -0.480 & 0.014 & 0.209\end{pmatrix},\ \mathbf{b}=\begin{pmatrix}1.325\\0.427\\0.117\\-0.687\end{pmatrix}.$$

The matrix $\mathbf{A}$ was constructed as follows: first, the components were randomly and independently generated from a standard normal distribution, and then all the columns were normalized. The vector $\mathbf{b}$ was chosen as $\mathbf{b}\equiv\mathbf{A}\mathbf{x}_{\rm true}$, where $\mathbf{x}_{\rm true}=(1,-1,0,0,0)^T$. The problem has 10 BF vectors (corresponding to the 5-choose-2 options for the support) and they are denoted by $1,2,\ldots,10$, where the first solution is the optimal solution $\mathbf{x}_{\rm true}$. We compared three methods: 1) the IHT method with $L_1=1.1L(f)$ 2) the IHT method with $L_2=2L(f)$ 3) the GSS method. Each algorithm was run 1000 times with different randomly generated starting points. All the runs converged to one of the 10 BF vectors. The number of times each method converged to each of the BF vectors is given in Table 1.

BF	1	2	3	4	5	6	7	8	9	10
N_1	329	50	63	92	229	0	130	0	61	46
N_2	340	59	0	89	256	0	187	0	69	0
N_3	813	0	0	112	0	0	75	0	0	0
N_4	772	0	0	92	0	0	93	0	43	0

Table 1. Distribution of limit points among the 10 BF vectors. N_1, N_2, N_3, N_4 are the amount of runs for which the IHT method with L_1, L_2, the GSS and PSS algorithms (introduced in Section 3.3) converged to the ith BF vector.

Note that as L gets larger, there are more non-optimal candidates to which the IHT method can converge. The GSS method exhibits the best results with more than 80% chance to converge to the true optimal solution. This method will never converge to the BF vectors $3, 6, 8$ and 10 since they are not $L_2(f)$-stationary points. Moreover, there are only three possible BF vectors to which the GSS method converged: $1, 4$ and 7. The reason is that among the 10 BF vectors, there are only three CW-minima. This illustrates the fact that even though any CW-minimum is an $L_2(f)$-stationary point, the reverse claim is not true – there are $L_2(f)$-stationary points which are not CW-minima.

Example 3.2. We next compare the performance of MP and OMP to that of GSS. To this end we generated 1000 realizations of $\mathbf{A}$ and $\mathbf{b}$ as described in Example 3.1. We ran both MP and OMP with $s=2$. Each of these methods were considered "successful" if it found the correct support (MP usually does not find the correct values). GSS was run with an initial vector of all zero, so that the first two iterations were identical to MP. The results were the following: out of the 1000 realizations MP and OMP found the correct support in 452 cases. The GSS method, which adds "correcting" steps to MP, recovered the correct support in 652 instances.

An additional advantage of GSS is that it is capable of running from various starting points. We therefore added the following experiment: for each realization of $\mathbf{A}$ and $\mathbf{b}$, we ran GSS from 5 different initial vectors generated as in Example 3.1. If at least one of these 5 runs detected the correct support, then the experiment is considered successful. In this case the correct support was found 952 times out of the 1000 realizations.

3.3. The Partial Sparse-Simplex Method

As we noted, the GSS method has several advantages over IHT. On the other hand, the computational effort per iteration of the GSS algorithm is larger than the one required by IHT. This computational burden is caused by the fact that the method has no index selection strategy. To overcome this drawback, we introduce the *partial sparse-simplex method (PSS)*. The only difference from the GSS algorithm is in the case when $\|\mathbf{x}^k\|_0=s$, where there are two options. Either perform a minimization with respect to the variable in the support of $\mathbf{x}^k$ which causes the maximum decrease in function value; or replace the variable in the support with the smallest absolute value (that is, substituting zero instead of the current value), with the non-support variable corresponding to the largest absolute value of the partial derivative – the value of the new non-zero variable is set by performing a minimization procedure with respect to it. Finally, the best of the two choices (in terms of objective function value) is selected.

The limit points of PSS are not necessarily CW-minima. However, when (2.5) holds, they are $L_2(f)$-stationary points, which is a better result than the one known for the IHT method.

Theorem 3.3. *Suppose that* (2.5) *holds and let* $\{\mathbf{x}^k\}$ *be the sequence generated by the PSS method. Then any accumulation point of* $\{\mathbf{x}^k\}$ *is an* $L_2(f)$*-stationary point.*

We now return to Example 3.1, and add a comparison to PSS. As can be seen, the method performs very well, much better than IHT. It is only slightly inferior to GSS (since it added BF vector 9), despite the fact that each iteration is much cheaper.

Example 3.3. We now consider an example of quadratic equations. Given m vectors $\mathbf{a}_1,\ldots,\mathbf{a}_m$, our problem is to find a sparse vector $\mathbf{x}\in\mathbb{R}^n$ satisfying $(\mathbf{a}_i^T\mathbf{x})^2=c_i$. The problem can be formulated as problem (P) with $f\equiv f_{\rm QI}$ where $\mathbf{A}_i=\mathbf{a}_i\mathbf{a}_i^T$. We compare GSS and PSS on an example with $m=80, n=120$ and $s=3,4,\ldots,10$. Each component of the 80 vectors $\mathbf{a}_1,\ldots,\mathbf{a}_{80}$ was randomly and independently generated from a standard normal distribution. The true vector $\mathbf{x}_{\rm true}$ was generated by choosing randomly the s nonzero components whose values were also randomly generated from a standard normal distribution. The vector $\mathbf{c}$ was determined by $c_i=(\mathbf{a}_i^T\mathbf{x}_{\rm true})^2$. For each value of s, we ran GSS and PSS from 100 different and randomly generated initial vectors. The numbers of runs out of 100 in which the methods found the correct solution is given in Table 2.

s	3	4	5	6	7	8	9	10
$N_{\rm PSS}$	27	22	8	5	9	5	3	2
$N_{\rm GSS}$	73	69	20	19	13	8	6	3

Table 2. The second (third) column contains the number of runs for which the partial (greedy) sparse-simplex method converged.

Note, that we can easily increase the success probability of the algorithms by starting them from several initial vectors and taking the best result.

4. REFERENCES

[1] A. Beck and Y. C. Eldar. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. 2012. arXiv:1203.4580v1.

[2] A. Beck and M Teboulle. Gradient-based algorithms with applications to signal recovery problems. In Yonina Eldar and Daniel Palomar, editors, *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2010.

[3] E. V. D. Berg and M. P. Friedlander. Sparse optimization with least-squares constraints. *SIAM J. Optim.*, 21:1201–1229.

[4] D. P. Bertsekas. *Nonlinear Programming*. Belmont MA: Athena Scientific, second edition, 1999.

[5] T. Blumensath and M. E. Davies. Iterative thresholding for sparse approximations. *The Journal of Fourier Analysis and Applications*, 14(5):629–654, 2008.

[6] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006.

[7] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006.

[8] A. Szameit et. al. Sparsity-based single-shot subwavelength coherent diffractive imaging. *Nature Materials*.

[9] K. Jaganathan, S. Oymak, and B. Hassibi. Recovery of sparse 1-D signals from the magnitudes of their Fourier transform. arXiv:1206.1405v1.

[10] Y. C. Eldar M. Davenport, M. Duarte and G. Kutyniok. *Compressed Sensing: Theory and Applications*, chapter Introduction to Compressed Sensing. Cambridge Univ. Press, 2012.

[11] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 2008.

[12] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Processing*, 41(12):3397–3415, 1993.

[13] H. Ohlsson, A. Y. Yang, R. Dong, and S. S. Sastry. Compressive phase retrieval from squared output measurements via semidefinite programming. 2012. arXiv:1111.6323v3.

[14] Y. Shechtman, Y. C. Eldar, A. Szameit, and M. Segev. Sparsity-based sub-wavelength imaging with partially spatially incoherent light via quadratic compressed sensing. *Optics Express*, 19:14807–14822, 2011.

[15] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50(10):2231–2242, October 2004.

[16] J. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *Proc. IEEE*, 98(6):948–958, 2010.