BACKTRACKING MATCHING PURSUIT WITH SUPPLEMENT SET OF ARBITRARY SIZE

*Laming Chen and Yuantao Gu**

State Key Laboratory on Microwave and Digital Communications
Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, CHINA

ABSTRACT

The idea of backtracking has been incorporated into the matching pursuit algorithms in sparse recovery, for example, subspace pursuit (SP) and compressive sampling matching pursuit (CoSaMP), to improve the recovery performance. In each iteration, a supplement set of size K or $2K$ is added to the candidate set to re-evaluate their reliability and then discard the unreliable indices, where K is the sparsity level of the original sparse signal. Yet the optimal choice of the size of the supplement set is still unclear. This paper aims to provide comprehensive analysis on the optimal choice of the size. The optimality is twofold: performance guarantees and computational complexity. By two theorems,we provide theoretical guarantees for the supplement set of arbitrary size, and computational complexity needed for perfect recovery. Numerical simulations demonstrate that a moderate size, such as $0.25K$, results in computational efficiency without loss of recovery quality.

***Index Terms*—** Sparse recovery, backtracking matching pursuit, size of supplement set, restricted isometry property, computational complexity.

1. INTRODUCTION

Finding the sparsest solution to the under-determined equation,

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{1}$$

is an essential issue in many fields of signal processing, especially in compressive sampling (CS) over the past decades [1–3]. In (1), $\mathbf{A} \in \mathbb{R}^{M\times N}$ is a sensing matrix with more columns than rows, $\mathbf{y} \in \mathbb{R}^M$ is a measurement vector, and $\mathbf{x} \in \mathbb{R}^N$ is a K-sparse vector to be recovered, which means only K out of its N entries are nonzero. Directly finding the sparest solution to (1) is NP-hard, which is not practical when facing large scale problems. This leads to one of the major aspects of CS theory—designing effective recovery algorithms with fine recovery performance and low computational complexity.

Plentiful algorithms have been proposed to derive the sparse solution to (1). A family of convex relaxation algorithms [4] had been introduced before the theory of CS was established. Based on linear programming (LP) techniques, it is demonstrated that l_1 norm optimization problem yields an identical solution to the l_0 norm optimization problem, provided that $\mathbf{A}$ satisfies the restricted isometry property (RIP) with a constant parameter [5–7]. The computational complexity of LP algorithms based on interior point methods is $O(M^2N^{3/2})$ [8], which is still high for large signal dimension N.

Another family of iterative greedy algorithms has received much attention due to their simple implementation and low computational complexity. The basic idea underlying these algorithms is to iteratively estimate the support set of the original sparse signal. In each iteration, one or more indices are added to the support estimation by correlating the columns of $\mathbf{A}$ with the residual vector. Typical examples include orthogonal matching pursuit (OMP) [9, 10], regularized OMP (ROMP) [11, 12], and stage-wise OMP (StOMP) [13]. Compared with convex relaxation algorithms, matching pursuits need more number of measurements, but they tend to be more computationally efficient. The total computational complexity of matching pursuits is approximately $O(KMN)$.

1.1. Related Works

Recently, several greedy pursuits including subspace pursuit (SP) [14] and compressive sampling matching pursuit (CoSaMP) [15] have been proposed by incorporating the idea of backtracking. With the prior information of sparsity level K, they iteratively refine the candidate support set by adding a supplement set of fixed size and discarding the unreliable candidates. The size of the supplement set of SP is K, while that of CoSaMP is $2K$. By re-evaluating the reliability of all candidates in each iteration, these algorithms can provide comparable performance to convex relaxation algorithms, and exhibit low computational complexity as matching pursuit algorithms.

Based on SP and CoSaMP, several algorithms have been proposed to further improve the recovery performance. A fast version of SP is developed in [16] at the expense of accuracy loss. Sparsity adaptive matching pursuit (SAMP) [17] and its fast version [18] iteratively estimate the sparsity level and adopt the SP strategy for sparse recovery. In [19], a sparsity-constrained minimization problem is introduced and an algorithm inspired by CoSaMP is proposed. It is clarified that optimally tuned SP dominates optimally tuned CoSaMP [20], yet the optimal size of the supplement set is still unavailable, which is the main motivation of this paper. In [21], the size of the supplement set varies dynamically and the criterion of selecting indices is improved, but it lacks theoretical performance guarantees.

This paper aims to provide comprehensive analysis on the optimal choice of the size of the supplement set. The SP algorithm is modified to the situation where the size of the supplement set is an arbitrary number L. We term this class of algorithms as *backtracking matching pursuit* (BMP). It is proved that the recovery performance is guaranteed as long as the sensing matrix $\mathbf{A}$ satisfies the RIP of order $2K + L$ with a constant parameter. Furthermore, the analysis of computational complexity suggests a moderate L would be more suitable for sparse recovery in practice. Notice that the theoretical analysis is also applicable to other sparse recovery algorithms based on backtracking, such as SAMP [17] and GraSP [19], and helps to improve their performance. Numerical simulations reveal

The corresponding author of this paper is Yuantao Gu (gyt@tsinghua.edu.cn).

Table 1. The Procedure of Backtracking Matching Pursuit

Input: $\mathbf{A}, \mathbf{y}, K, L$;

Initialization: $l = 0, T^0 = \mathrm{maxind}\{\mathbf{A}^{\mathrm{T}}\mathbf{y}, K\}$,

$\mathbf{y}_r^0 = \mathrm{resid}(\mathbf{y}, \mathbf{A}_{T^0})$;

Output: T^l, solution $\hat{\mathbf{x}}_{T^l} = \mathbf{A}_{T^l}^{\dagger}\mathbf{y},\ \hat{\mathbf{x}}_{\{1,\ldots,N\}\setminus T^l} = \mathbf{0}$.

Repeat:

$l = l + 1$;

Adding supplement set:

$\tilde{T}^l = T^{l-1} \cup \mathrm{maxind}\{\mathbf{A}^{\mathrm{T}}\mathbf{y}_r^{l-1}, L\}$;

Updating candidate set:

$T^l = \mathrm{maxind}\{\mathbf{A}_{\tilde{T}^l}^{\dagger}\mathbf{y}, K\}$;

Updating residual vector:

$\mathbf{y}_r^l = \mathrm{resid}(\mathbf{y}, \mathbf{A}_{T^l})$;

Until: Stop criterion satisfied;

that a moderate size of supplement set, such as $0.25K$, results in computational efficiency without loss of recovery quality.

2. BACKTRACKING MATCHING PURSUIT

For better understanding of the description of backtracking matching pursuit (BMP), we adopt the notations in [14] about projection and its residual vector. Let $\mathbf{A}_I$ and $\mathbf{x}_I$ denote the submatrix formed by columns of $\mathbf{A}$ and subvector composed of elements of $\mathbf{x}$ indexed by the set I, respectively. $\mathrm{span}(\mathbf{A}_I)$ denotes the subspace spanned by the columns of $\mathbf{A}_I$. Suppose that $\mathbf{A}_I^{\mathrm{T}}\mathbf{A}_I$ is invertible, then the projection of $\mathbf{y}$ onto $\mathrm{span}\,(\mathbf{A}_I)$ is defined as

$$\mathbf{y}_p = \mathrm{proj}\,(\mathbf{y}, \mathbf{A}_I) = \mathbf{A}_I\mathbf{A}_I^{\dagger}\mathbf{y}, \tag{2}$$

where $\mathbf{A}_I^{\dagger} = \left(\mathbf{A}_I^{\mathrm{T}}\mathbf{A}_I\right)^{-1}\mathbf{A}_I^{\mathrm{T}}$ denotes the pseudo-inverse of matrix $\mathbf{A}_I$. The residual vector of the projection equals

$$\mathbf{y}_r = \mathrm{resid}\,(\mathbf{y}, \mathbf{A}_I) = \mathbf{y} - \mathbf{y}_p. \tag{3}$$

Define $\mathrm{maxind}\{\mathbf{x}, K\}$ as the set composed of the indices of K largest magnitude entries of $\mathbf{x}$. Let $I \setminus J$ denote the set consisting of elements in the set I but not in the set J.

The procedure of BMP is demonstrated in Table 1. In each iteration, BMP generates a supplement set of size L, and adds to the candidate set. Then the most reliable K indices is selected out of these $K + L$ ones. This algorithm is a direct generalization of the SP algorithm [14] by setting the size of the supplement set to arbitrary size. If $L = K$, BMP is identical to SP. The stop criterion of BMP is when the candidate set remains the same as the last one, or the number of iterations reaches the bound Q. It is easy to check that if the candidate set remains the same, it won't change in the sequential iterations. In addition, once the support set is successfully recovered, it will remain the same as well.

The candidate set, the supplement set and several other relevant notations are exhibited in Fig. 1 for a visualization. Let T and $|T|$ denote the exact support set of $\mathbf{x}$ and the cardinality of T, respectively. Define T_Δ as the supplement set, then

$$T_\Delta \cap T^{l-1} = \emptyset,\ \ T_\Delta \cup T^{l-1} = \tilde{T}^l. \tag{4}$$

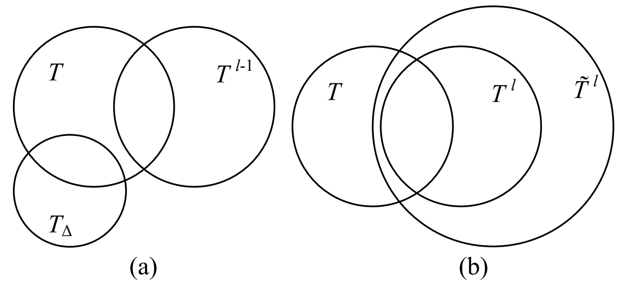

Fig. 1. The relationship of (a) T, T^{l-1} and T_Δ, as well as (b) T, $\tilde{T}^l$ and T^l.

The new candidate set satisfies $T^l \subset \tilde{T}^l$. By the means of re-evaluating, more indices of T are contained in the candidate set T^l than those in the previous iteration.

It needs to be emphasized that the main contribution of this paper is not the derivation of the BMP algorithm, but the analysis on the optimal choice of the size of supplement set. The optimality is twofold: performance guarantees and computational complexity, and they are demonstrated in Section 3 and Section 4, respectively.

3. PERFORMANCE GUARANTEES

In this section, the theoretical recovery performance guarantees of BMP are provided. Define δ_K to be the K-restricted isometry constant (RIC) as in [6], and let $\delta^* = \delta_{2K+L}$ for short. The following theorem demonstrates the main result.

Theorem 1. *Define $\rho = K/L$. The set of undetected indices of the support set in the l-th iteration, i.e. $T \setminus T^l$, satisfies*

$$\|\mathbf{x}_{T\setminus T^l}\|_2 \le C\|\mathbf{x}_{T\setminus T^{l-1}}\|_2, \tag{5}$$

where

$$C = \begin{cases} \dfrac{\left(1+\sqrt{\rho}\right)\delta^*(1+\delta^*)}{\left(1-\delta^*\right)^3} & 0 < \rho \le 1; \\ \dfrac{1+\delta^*}{1-\delta^*}\sqrt{\dfrac{\rho-1}{\rho} + \dfrac{8\delta^*}{\left(1-\delta^*\right)^4}} & \rho > 1. \end{cases} \tag{6}$$

Furthermore, if $(2K + L)$-RIC of matrix A satisfies

$$0 < \delta^* \le \begin{cases} \dfrac{1}{\rho+4} & 0 < \rho \le 1; \\ \dfrac{1}{12\rho+4} & \rho > 1, \end{cases} \tag{7}$$

then the constant C is less than 1, which implies that the original sparse signal is guaranteed to be recovered in finite iterations.

Proof. For each scenario, the proof of (5) mainly consists of two parts. First, by adding the supplement set T_Δ to the candidate set T^{l-1} to construct $\tilde{T}^l$, the undetected energy $\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2$ is much reduced compared to $\|\mathbf{x}_{T\setminus T^{l-1}}\|_2$. Second, after discarding the unreliable indices, the undetected energy $\|\mathbf{x}_{T\setminus T^l}\|_2$ will not increase too much compared to $\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2$. Refer to Fig. 1 to comprehend the big picture of the proof.

As for the first part of the proof of (5), we first prove that if $\hat{T} \subset T \setminus \tilde{T}^l$ and $|\hat{T}| = \alpha|T_\Delta \setminus T|$, it holds that

$$\|\mathbf{x}_{\hat{T}}\|_2 \le \frac{\left(1+\sqrt{\alpha}\right)\delta^*}{\left(1-\delta^*\right)^2}\|\mathbf{x}_{T\setminus T^{l-1}}\|_2. \tag{8}$$

And if $|\hat{T}| = \alpha|T_\Delta \cap T|$, it holds that

$$\|\mathbf{x}_{\hat{T}}\|_2 \le \sqrt{\alpha}\frac{1+\delta^*}{1-\delta^*}\|\mathbf{x}_{T_\Delta\cap T}\|_2 + \frac{\left(1+\sqrt{\alpha}\right)\delta^*}{\left(1-\delta^*\right)^2}\|\mathbf{x}_{T\setminus T^{l-1}}\|_2. \tag{9}$$

We establish the inequality (9) since for small L, $|T_\Delta \setminus T|$ may be rather small or even zero, and (8) is not sufficient for the proof of this scenario.

According to the proof of Theorem 3 in [14], the residual vector can be written as

$$\mathbf{y}_r^{l-1} = \mathbf{A}_{T\cup T^{l-1}}\mathbf{x}_r^{l-1}, \tag{10}$$

where $\mathbf{x}_r^{l-1}$ satisfies

$$\|\mathbf{x}_r^{l-1}\|_2 \le \frac{1}{1-\delta_{2K}}\|\mathbf{x}_{T\setminus T^{l-1}}\|_2. \tag{11}$$

Since T_Δ is the indices of L largest magnitude entries in $\mathbf{A}^{\mathrm{T}}\mathbf{y}_r^{l-1}$ and $|\hat{T}| = \alpha|T_\Delta \setminus T|$, it holds that

$$\|\mathbf{A}_{\hat{T}}^{\mathrm{T}}\mathbf{y}_r^{l-1}\|_2 \le \sqrt{\alpha}\|\mathbf{A}_{T_\Delta\setminus T}^{\mathrm{T}}\mathbf{y}_r^{l-1}\|_2. \tag{12}$$

On one hand, applying Lemma 1 in [14] and according to (10),

$$\|\mathbf{A}_{T_\Delta\setminus T}^{\mathrm{T}}\mathbf{y}_r^{l-1}\|_2 \le \delta_{2K+L}\|\mathbf{x}_r^{l-1}\|_2. \tag{13}$$

On the other hand, (10) and triangle inequality implies

$$\begin{aligned}&\|\mathbf{A}_{\hat{T}}^{\mathrm{T}}\mathbf{y}_r^{l-1}\|_2\\ \ge&\|\mathbf{A}_{\hat{T}}^{\mathrm{T}}\mathbf{A}_{\hat{T}}\mathbf{x}_{\hat{T}}\|_2 - \|\mathbf{A}_{\hat{T}}^{\mathrm{T}}\mathbf{A}_{T\cup T^{l-1}\setminus\hat{T}}\left(\mathbf{x}_r^{l-1}\right)_{T\cup T^{l-1}\setminus\hat{T}}\|_2\\ \ge& \left(1-\delta_K\right)\|\mathbf{x}_{\hat{T}}\|_2 - \delta_{2K}\|\mathbf{x}_r^{l-1}\|_2.\end{aligned} \tag{14}$$

Substituting (13) and (14) into (12), and together with (11), the inequality (8) can be derived.

According to the definition of T_Δ and $|\hat{T}| = \alpha|T_\Delta \cap T|$,

$$\|\mathbf{A}_{\hat{T}}^{\mathrm{T}}\mathbf{y}_r^{l-1}\|_2 \le \sqrt{\alpha}\|\mathbf{A}_{T_\Delta\cap T}^{\mathrm{T}}\mathbf{y}_r^{l-1}\|_2. \tag{15}$$

On one hand, similar to the proof of (14),

$$\|\mathbf{A}_{T_\Delta\cap T}^{\mathrm{T}}\mathbf{y}_r^{l-1}\|_2 \le \left(1+\delta_K\right)\|\mathbf{x}_{T_\Delta\cap T}\|_2 + \delta_{2K}\|\mathbf{x}_r^{l-1}\|_2. \tag{16}$$

Substituting (16) and (14) into (15), and according to (11), the inequality (9) can be derived.

Now turn to the the first part of the proof of (5). Consider the scenario of $L \ge K$, which means $0 < \rho \le 1$. It is easy to check that $|T \setminus \tilde{T}^l| \le \rho|T_\Delta \setminus T|$, thus (8) implies

$$\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2 \le \frac{(1+\sqrt{\rho})\,\delta^*}{(1-\delta^*)^2}\|\mathbf{x}_{T\setminus T^{l-1}}\|_2, \tag{17}$$

which means the undetected energy $\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2$ is upper bounded by $\|\mathbf{x}_{T\setminus T^{l-1}}\|_2$.

For the scenario of $L < K$ which means $\rho > 1$, since $|T\setminus\tilde{T}^l| \le K - |T_\Delta \cap T|$, there exist T_1 and T_2 such that $T_1 \cup T_2 = T \setminus \tilde{T}^l$, $T_1 \cap T_2 = \emptyset$, and $|T_1| \le K - \rho|T_\Delta \cap T| = \rho|T_\Delta \setminus T|$, $|T_2| \le (\rho-1)|T_\Delta \cap T|$. According to (8), it can be derived that

$$\|\mathbf{x}_{T_1}\|_2 \le \frac{(1+\sqrt{\rho})\,\delta^*}{(1-\delta^*)^2}\|\mathbf{x}_{T\setminus T^{l-1}}\|_2, \tag{18}$$

while (9) implies that

$$\|\mathbf{x}_{T_2}\|_2 \le \sqrt{\rho-1}\frac{1+\delta^*}{1-\delta^*}\|\mathbf{x}_{T_\Delta\cap T}\|_2 + \frac{(1+\sqrt{\rho-1})\delta^*}{\left(1-\delta^*\right)^2}\|\mathbf{x}_{T\setminus T^{l-1}}\|_2. \tag{19}$$

It is easy to check that

$$\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2^2 = \|\mathbf{x}_{T_1}\|_2^2 + \|\mathbf{x}_{T_2}\|_2^2 \tag{20}$$

$$\|\mathbf{x}_{T_\Delta\cap T}\|_2^2 = \|\mathbf{x}_{T\setminus T^{l-1}}\|_2^2 - \|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2^2. \tag{21}$$

Substituting (18) and (19) into (20), and together with (21), it can be derived that

$$\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2 < \sqrt{\frac{\rho-1}{\rho} + \frac{8\delta^*}{(1-\delta^*)^4}}\|\mathbf{x}_{T\setminus T^{l-1}}\|_2, \tag{22}$$

which also reveals that $\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2$ is upper bounded.

Now turn to the the second part of the proof of (5). For both scenarios, it holds that

$$\|\mathbf{x}_{T\setminus T^l}\|_2 \le \frac{1+\delta^*}{1-\delta^*}\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2. \tag{23}$$

The proof of (23) is much the same as the proof of Theorem 4 in [14], except for that $|\tilde{T}^l \setminus T^l| = L$. Thus we omit the detailed proof. (23) reveals that $\|\mathbf{x}_{T\setminus T^l}\|_2$ may be slightly larger than $\|\mathbf{x}_{T\setminus\tilde{T}^l}\|_2$.

Combining (17) and (22) with (23), the inequality (5) can be derived. In addition, it is easy to check that if (7) is satisfied, then $C < 1$, which means that the undetected energy $\|\mathbf{x}_{T\setminus T^l}\|_2$ can be arbitrary small. If at least one element of the support set is undetected, $\|\mathbf{x}_{T\setminus T^l}\|_2$ will always be larger than the smallest magnitude nonzero entry of $\mathbf{x}$, which leads contradiction. Thus the support set will be exactly recovered, and the sparse signal will be as well. □

Theorem 1 reveals that for arbitrary L, the recovered signal is identical to the original sparse one as long as $\mathbf{A}$ satisfies the RIP of order $2K + L$ with a constant parameter. For constant ρ, set the size of the supplement set $L = \rho^{-1}K$, and the requirement of the RIC, $\delta_{(2+\rho^{-1})K}$, is a constant independent of the sparsity K. For the case of $\rho = 1$, which is exactly the scenario of the SP algorithm, the constant C in Theorem 1 is identical to that of Theorem 2 in [14]. Notice that when $L = 2K$, the demand of δ_{4K} in BMP is more relaxed than that in CoSaMP. This is mainly due to the fact that the residual vector of BMP is calculated by projecting, while that of CoSaMP is derived by pruning.

4. COMPUTATIONAL COMPLEXITY

Let $|x_{\min}|$ denote the smallest magnitude nonzero entry of sparse signal $\mathbf{x}$, and define $\eta_{\min} = |x_{\min}|/\|\mathbf{x}\|_2$, which scales the worst element to be recovered. The upper bound of the iteration number n_{it}, which is needed for perfect recovery, is given as follows.

Theorem 2. *If the constant C defined in Theorem 1 satisfies $C < 1$, then the number of iterations for perfect recovery is upper bounded by*

$$n_{\mathrm{it}} \le \min\left(\frac{-\log\eta_{\min}}{-\log C} + 1, \frac{1.5K}{-\log C}\right). \tag{24}$$

The detailed proof of Theorem 2 is referred to the proof of Theorem 6 in [14], since their principles are similar. According to Theorem 2, the number of iterations needed satisfies $n_{\mathrm{it}} \le O(K)$. The

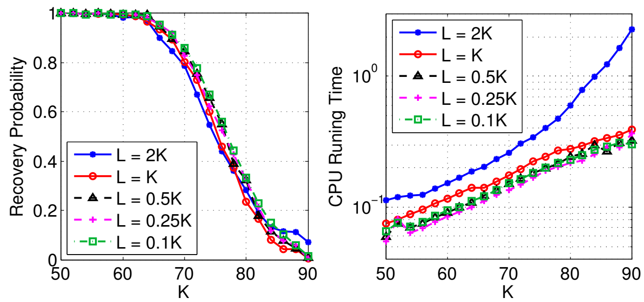

Fig. 2. For BMP algorithm, the probability of perfect recovery and the average CPU running time versus sparsity level K with respect to $L = 2K$, K, $0.5K$, $0.25K$, and $0.1K$ when $N = 3000$ and $M = 300$ are demonstrated.

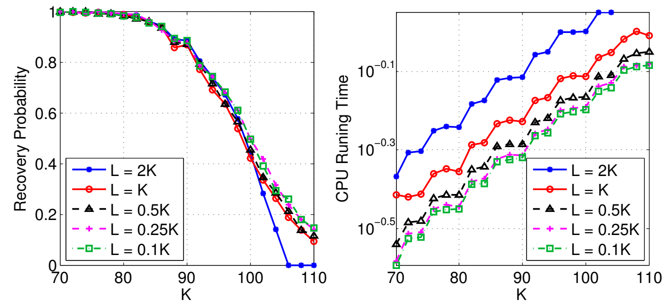

Fig. 3. For SAMP algorithm, the probability of perfect recovery and the average CPU running time versus sparsity level K with respect to $L = 2K$, K, $0.5K$, $0.25K$, and $0.1K$ when $N = 3000$ and $M = 300$ are demonstrated.

computational complexity for arbitrary L can be easily estimated by multiplying the number of iterations with the complexity in each iteration.

Generally speaking, the computational complexity of computing $\mathbf{A}^{\mathrm{T}}\mathbf{y}_r^{l-1}$ is $O(MN)$, computing $\mathbf{A}_{\tilde{T}^l}^{\dagger}\mathbf{y}$ is $O(M(K+L)^2)$, and updating $\mathbf{y}_r^l$ is $O(MK^2)$. Thus the complexity in each iteration is $O(M(N+(K+L)^2))$, which is a monotone strictly increasing function of L. Together with Theorem 2, the total computational complexity is upper bounded by $O(KM(N+(K+L)^2))$, which is comparable to that of matching pursuit algorithms.

To reduce the computational complexity, we should reduce the complexity in each iteration as well as decrease the number of iterations needed. L is a tradeoff parameter between them: bigger L may introduce more reliability indices in each iteration to reduce the number of iterations needed, but results in higher complexity in each iteration. However, extremely small L, such as $L = 1$, will cause huge number of iterations, which greatly increases the total complexity. Thus moderate L is preferred for sparse recovery with high recovery probability and low computational complexity.

5. NUMERICAL SIMULATIONS

Two Monte Carlo simulations are performed in this section to compare the recovery performance, including the probability of perfect recovery and the average CPU running time, with respect to different size of supplement set. In the experiments, the entries of sensing matrix $\mathbf{A}$ are independently and identically distributed Gaussian with zero mean and variance $1/M$. The locations of nonzero entries of the sparse signal $\mathbf{x}$ are randomly chosen among all possible choices. These nonzero entries are independently Gaussian distributed with zero mean and the same variance. The sparse signal is finally normalized to have unit energy. If the support set is successfully detected, this recovery is considered perfect, and the CPU running time is recorded.

The first experiment compares the recovery performance of BMP versus sparsity level K with respect to different L, and the results are depicted in Fig. 2. The parameter L is set to $2K$, K, $0.5K$, $0.25K$, and $0.1K$, respectively, where $L = K$ corresponds to SP algorithm and $L = 2K$ to a variant of CoSaMP. The dimension parameters are $N = 3000$, $M = 300$, and the sparsity level K varies from 50 to 90. The maximal iteration number $Q = K$, which means the support is expected to be recovered in K iterations. The experiment is repeated for 500 trials to calculate the probability of perfect recovery and the average CPU running time. As can be seen from Fig. 2, for different choices of L, the probability of perfect recovery is not much influenced. Their crucial sparsity $K_{\max}$, which is the largest integer which guarantees 100% successful recovery, are all 52. The comparison of the average running time demonstrates that $L = 0.25K$ costs the least running time, which indicates BMP with supplement set of size $0.25K$ may be a better option than SP and CoSaMP for sparse recovery.

The second experiment revises the size of the supplement set in the SAMP algorithm [17] to an arbitrary number, and the size L is also set to $2K$, K, $0.5K$, $0.25K$, and $0.1K$, respectively. The step size of SAMP is $s = 5$. The dimension parameters are the same as those in the first experiment, and the sparsity level K varies from 70 to 110. The experiment is repeated for 500 trials to calculate the probability of perfect recovery and the average CPU running time. The results are presented in Fig. 3. As is revealed in this scenario, the recovery probability is again not much influenced for different choices of L. The average running time reveals that the optimal choice of L is $0.25K$ and $0.1K$, which occupies the least computational resources. Comparing the recovery performances of these two algorithms according to Fig. 2 and Fig. 3, the SAMP guarantees to recover signals with larger sparsity, but costs more running time for the estimation of sparsity level.

6. CONCLUSION

In this paper, we aim to provide comprehensive analysis on the optimal size of supplement set for matching pursuit algorithms with the idea of backtracking. Theoretical guarantees for sparse recovery are derived. Provided that the sensing matrix satisfies the RIP of order $2K + L$ with a constant parameter, the original sparse signal is guaranteed to be recovered in finite iterations. The theoretical analysis of computational complexity reveals that moderate L results in more efficiency for large scale problems. Monte Carlo simulations compare the recovery performances of BMP and SAMP, including the probability of perfect recovery and the average CPU running time, for $L = 2K$, K, $0.5K$, $0.25K$, and $0.1K$, respectively. It is demonstrated that the recovery probability is rather robust for a large range of L, while $L = 0.25K$ costs the least running time, which is more efficient for practical use.

7. REFERENCES

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

[2] D. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289-1306, Apr. 2006.

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

[4] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33-61, Aug. 1998.

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

[6] E. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589-592, May 2008.

[7] T. Cai, L. Wang, and G. Xu, "New bounds for restricted isometry constants," *IEEE Trans. Information Theory*, vol. 56, no. 9, pp. 4388-4394, Sept. 2010.

[8] I. Nesterrov, A. Nemirovskii, and Y. Nesterov, *Interior-Point Polynomial Algorithms in Convex Programming*, Philadelphia, PA: SIAM, 1994.

[9] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *Conf. Rec. 27th Asilomar Conf. Signals, Systems and Computers*, vol. 1, pp. 40-44, Nov. 1993.

[10] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Information Theory*, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[11] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317-334, June 2009.

[12] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310-316, Apr. 2010.

[13] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Information Theory*, vol. 58, no. 2, pp. 1094-1121, Feb. 2012.

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

[15] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301-321, May 2009.

[16] K. Tan, Q. Wan, A. Huang, and J. Wang, "A fast subspace pursuit for compressive sensing," *IET International Radar Conference*, pp. 1-4, Apr. 2009.

[17] T. Do, L. Gan, N. Nguyen, and T. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," *42nd Asilomar Conf. Signals, Systems and Computers*, pp. 581-587, Oct. 2008.

[18] C. Ma, C. Xu, L. Shen, and S. Zheng, "A fast sparsity adaptive matching pursuit algorithm for compressed sensing," *Advances in Automation and Robotics*, vol. 1, pp. 363-368, 2012.

[19] S. Bahmani, P. Boufounos, and B. Raj, "Greedy sparsity-constrained optimization," *Conf. Rec. 45th Asilomar Conf. Signals, Systems and Computers*, pp. 1148-1152, Nov. 2011.

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

[21] B. Varadarajan, S. Khudanpur, and T. Tran, "Stepwise optimal subspace pursuit for improving sparse recovery," *IEEE Signal Processing Letters*, vol. 18, no. 1, pp. 27-30, Jan. 2011.