Decentralized and Collaborative Subspace Pursuit: A Communication-Efficient Algorithm for Joint Sparsity Pattern Recovery

Gang Li, Senior Member, IEEE, Thakshila Wimalajeewa, Member, IEEE, and Pramod K. Varshney, Fellow, IEEE

Abstract:

In this paper, we consider the problem of joint sparsity pattern recovery in a decentralized network and propose an algorithm named decentralized and collaborative subspace pursuit (DCSP). The basic idea of DCSP is to embed collaboration among nodes into each iteration of the standard subspace pursuit (SP) algorithm. At each iteration of the DCSP algorithm, local estimation of the support set is carried out at every node by finding the subspace that the local measurement vector most probably lies in, and then the global estimate of the support set is obtained by fusion of all the local estimates. An attractive characteristic of DCSP is the small number of messages to be transmitted among nodes, which is helpful when the communication capacity of the network is limited. Compared to existing decentralized greedy algorithms, DCSP offers satisfactory accuracy of sparsity pattern recovery with much less communication cost. We further extend DCSP to the generalized DCSP (GDCSP) algorithm, by allowing each node to share more local information with its surrounding neighbors. GDCSP outperforms DCSP in terms of the accuracy of sparsity pattern recovery at the cost of slightly increased communication overhead.

Key words: Joint sparsity pattern recovery, Compressive sensing, Information fusion, Subspace pursuit
I. INTRODUCTION

Compressive sensing (CS) refers to the idea that a sparse signal can be accurately recovered from a small number of measurements [1]-[3]. It has been shown that CS is potentially useful in a wide range of applications including medical imaging [4][5], radar imaging [6]-[8], and source localization [9]-[11]. In particular, CS provides a new approach for data reduction in sensor network applications without compromising performance [12]-[15].

Consider a network composed of $Q$ distributed nodes. The measurements collected at the $q$-th node are given by

$$y_q = A_q x_q$$

(1)

where $y_q$ is an $M \times 1$ measurement vector, $A_q$ is an $M \times N$ dictionary matrix, $x_q$ is an $N \times 1$ vector which has $K$ nonzero entries, $q=1, 2, \cdots, Q$. Assume that all \{ $x_q$, $q=1, 2, \cdots, Q$ \} have the same sparsity pattern, i.e., the support set is defined as $T=\{ i : x_q(i) \neq 0, i=1, 2, \cdots, N \}$ with cardinality $|T|=K$. Such a joint sparsity model with a common support set is applicable in many practical scenarios including multi-sensor surveillance systems in which multiple sensors (such as video cameras, radars, and sonars) observe the same group of targets [16][17], and cognitive radio networks in which multiple cognitive radios identify the occupied spectrum [12][13]. Our goal is to recover the support set $T$ using \{ $y_q, A_q$, $q=1, 2, \cdots, Q$ \} in the case when $N \geq M \geq 2K$. Once $T$ is determined, the nonzero entries in $x_q$ can be easily obtained by least squares estimation.

In a centralized processing framework, it is required for each node to transmit its measurement vector and information regarding dictionary matrix to a central fusion center. Due to practical constraints on communication bandwidth and computational capacity, recovering sparsity pattern in a decentralized manner is more efficient. In this paper, we do not consider any specific communication protocol for inter-node
communication and assume the worst-case scenario in terms of communication complexity that all the nodes communicate with each other one-by-one. It is expected that the use of efficient communication protocols will reduce communication complexity. The joint orthogonal matching pursuit (JOMP) and the joint subspace pursuit (JSP) algorithms were presented in [18]. In these algorithms, after the iterative solution processes at all the nodes are completed, the global estimate of the support set is obtained by majority voting over all local estimates. In [19], the distributed version of JSP, the distributed subspace pursuit (DiSP) algorithm, was developed by restricting the number of connectable neighbors of each node. Compared to JOMP and JSP, DiSP reduces the requirement of network connectivity, however, at the cost of decreased accuracy of sparsity pattern recovery. The algorithms in [20]-[24] achieve better accuracy of sparsity pattern recovery than that in [18][19], by embedding collaboration among nodes into the iterative solution process. In the distributed basis pursuit algorithms [20][21], each node solves for a local sparse solution by convex optimization and refines the solution by communicating with other nodes in an iterative manner. The convex optimization based algorithms in [20][21] require large computational resources. In the simultaneous orthogonal matching pursuit (SOMP) [22][23] and the simultaneous subspace pursuit (SSP) algorithms [24], collaboration among nodes at each iteration is also employed, but local estimation of the support set is carried out by greedy pursuit procedures. Generally speaking, the greedy algorithms such as SOMP and SSP are computationally much simpler than the convex optimization based algorithms in [20][21].

To implement the algorithms in [20]-[24] in a decentralized network, at each iteration, each node has to send $O(N)$-length vectors consisting of locally estimated coefficients to all the other nodes in the network. Before the iterative process converges, these $O(N)$-length messages are probably not sparse, and therefore, the total number of messages to be transmitted is considerably large. Thus, the algorithms in [20]-[24] lead to large communication overhead of the network. A communication-efficient algorithm, the distributed and
collaborative orthogonal matching pursuit (DCOMP) algorithm, was proposed in [25], where the reduction of communication overhead came from the fact that transmission of $O(N)$-length vectors is restricted to a small neighborhood surrounding each node. However, due to lack of backtracking operations in DCOMP, once an index is deemed as reliable and added to the support set estimate, it is not removed from the support set in subsequent iterations. This means that the strategy of index selection in DCOMP is too strict, and therefore, a larger number of measurements is required to guarantee the success of sparsity pattern recovery.

In this paper, we develop a new communication-efficient algorithm named decentralized and collaborative subspace pursuit (DCSP) for joint sparsity pattern recovery. The basic idea of DCSP is to embed collaboration among nodes into the iterative solution process of the standard subspace pursuit (SP) algorithm. A part of this work was presented in [26]. Here we significantly extend the work in [26] by providing theoretical analysis on the reliability and the communication cost of DCSP. In addition, we perform more detailed experiments and present more extensive simulation results. Theoretical analysis and experimental evaluations show that, DCSP can exactly recover the joint sparsity pattern with relatively low communication cost. The approach for collaboration among nodes here is similar to that in [20]-[25] and different from that in [18], which leads to the fact that DCSP outperforms JOMP and JSP in terms of the accuracy of sparsity pattern recovery. Different from the unrestricted transmission of $O(N)$-length vectors among nodes in SOMP [22][23] and SSP [24], at each iteration of DCSP, each node communicates with others in the network by only exchanging $K$-length local estimates of the support set. Therefore, compared to SOMP and SSP, DCSP significantly reduces the number of messages to be transmitted and is helpful when the communication capacity of the network is limited. Different from DCOMP in which all of the index estimates from past iterations are considered reliable, at each iteration DCSP has the ability to remove wrong elements from and putting new candidates into the estimated support set, by first updating the local list of reliable indices and then fusing all the locally selected indices. Therefore, compared to DCOMP, DCSP
provides much better accuracy of sparsity pattern recovery at a comparable communication cost. We further propose a generalized DCSP (GDSCP) algorithm, in which, besides broadcasting the $K$-length local estimate of the support set to the whole network, sharing of $O(N)$-length messages in a small neighborhood surrounding each node is also allowed. GDSCP is superior to DCSP in terms of the accuracy of sparsity pattern recovery, at the cost of slightly increased communication overhead.

The rest of this paper is organized as follows. In Section II, the implementations of SSP [24] and DCOMP [25] in a decentralized network are reviewed. In Section III, the DCSP algorithm is proposed for joint sparsity pattern recovery, and the convergence and the communication overhead of DCSP are theoretically analyzed. The GDSCP algorithm is presented in Section IV. Simulation results are provided in Section V and concluding remarks are given in Section VI.

Notation: To simplify the presentation, we define the following notation used in this paper.

- $\text{proj}(y, A) = [A^HA]^{-1}A^Hy$ calculates projection coefficients of a vector $y$ onto the column space of a matrix $A$. Conjugate transpose is denoted by $(\cdot)^H$.

- $\text{resid}(y, A) = y - A[A^HA]^{-1}A^Hy$ outputs the projection residual vector.

- $\text{max\_ind}(y, K) = \{K \text{ indices corresponding to the largest magnitude entries in the vector } y\}$.

- $\text{max\_occ}(T, K) = \{K \text{ elements that have the highest frequency of occurrence in the set } T\}$.

- $A(T)$ denotes a sub-matrix composed of the columns of $A$ indexed by the set $T$.

- $y(T)$ denotes a sub-vector composed of the entries of $y$ indexed by the set $T$.

- $|y|^n$ computes the $n$-th power of the absolute value of the vector $y$ element-by-element.

- The set-subtraction $T - S$ outputs a set composed of those elements that belong to $T$ but not $S$.

II. BACKGROUND

In this paper, we make the following assumptions. 1) The nonzero coefficients \( \{x_q(T), q=1, 2, \ldots, Q\} \)
are $K$-dimensional random variables and they are statistically independent of each other. This scenario is applicable when multiple sensor nodes collect independent snapshots [30], and when optical, radar and sonar sensors observe the same group of targets [31][32].

2) The dictionary matrix $\mathbf{A}_q$ satisfies the restricted isometry property (RIP) with constant parameters, for $q=1, 2, \ldots, Q$. The definition of the RIP is given as follows.

**Definition 1** RIP [1][2]: A $M \times N$ matrix $\mathbf{A}$ satisfies the RIP with parameters $(K, \delta)$ for $K \leq M$ and $0 \leq \delta < 1$, if

$$
(1 - \delta) \|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta) \|\mathbf{x}\|_2^2
$$

holds for any sparse signal $\mathbf{x}$ whose $l_0$ norm is not larger than $K$, where $\|\cdot\|_2$ denotes the $l_2$ norm.

In what follows, we briefly review two relevant algorithms for common sparsity pattern recovery based on the observation model (1), SSP [24] and DCOMP [25], which will be compared to the proposed DCSP algorithm later.

A. **SSP in a decentralized network**

The standard SP algorithm was proposed in [27] for sparse signal reconstruction from a single measurement vector. The most attractive feature of SP is that it offers comparable quality of sparsity pattern recovery to the linear programming methods along with much lower computational complexity. In [24], the centralized SSP algorithm extends the standard SP algorithm from the case of single measurement vector to the case of multiple measurement vectors. Here we consider the implementation of SSP in a decentralized network with no fusion center. To achieve the same accuracy of sparsity pattern recovery as with the centralized SSP algorithm, all the nodes in the decentralized network need to share local processing results, e.g. correlation coefficients and projection coefficients, with each other at each iteration. The operations of SSP at the $q$-th node are summarized in Algorithm 1, where the set $G$ contains the indices of all the nodes, i.e., $G=\{1,2, \ldots, Q\}$. 6
Algorithm 1 The decentralized SSP algorithm at the \( q \)-th node

Input: \( K, \ y_q, A_q \)

Initialization:

1) Send the vector \( c_q^0 = |A_q^0 y_q| \) to and receive \( c_j^0 \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

2) Let \( T^0 = \max \text{ind} \left( \sum_{q \in G} c_q^0, K \right) \); calculate the residual \( r_q^0 = \text{resid}(y_q, A_q(T^0)) \).

Iteration: at the \( l \)-th iteration (\( l \geq 1 \))

3) Send the vector \( c_q^l = |A_q^l r_q^{l-1}| \) to and receive \( c_j^l \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

4) Let \( \hat{T}^l = T^{l-1} \cup \max \text{ind} \left( \sum_{q \in G} c_q^l, K \right) \).

5) Send the vector \( d_q^l = |\text{proj}(y_q, A_q(\hat{T}^l))| \) to and receive \( d_j^l \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

6) Let \( T^l = \max \text{ind}(\sum_{q \in G} d_q^l, K) \); update the residual \( r_q^l = \text{resid}(y_q, A_q(T^l)) \).

7) Send the value of \( \|r_q^l\|_2^2 \) to and receive \( \|r_j^l\|_2^2 \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

8) If \( \sum_{q \in G} \|r_q^l\|_2^2 \geq \sum_{q \in G} \|r_q^{l-1}\|_2^2 \), let \( T^l = T^{l-1} \) and stop the iteration; otherwise, let \( l = l+1 \), and return to Step 3.

Output: The estimated support set \( \hat{T}_q = T^l \).

The communication cost is an important concern in most decentralized networks, which can be considered proportional to the number of messages to be transmitted. In the decentralized implementation of SSP, the communications among nodes appear in Steps 1, 3, 5 and 7 of Algorithm 1, and the lengths of the messages transmitted from each node are \( N, N, 2K \) and 1, respectively. Thus, the total number of messages transmitted from all the nodes is

\[
C_{\text{SSP}} = \left[ N + L_{\text{SSP}}(N + 2K + 1) \right] (Q-1) Q ,
\]

where \( L_{\text{SSP}} \) is the required number of iterations to successfully recover the support set for SSP. The total communication cost in (3) will dramatically increase as the network scale increases.
B. DCOMP in a decentralized network

The DCOMP algorithm [25] provides a promising way to reduce the communication overhead in the network by restricting the transmission of O(N)-length vectors to a small neighborhood surrounding each node. In the DCOMP algorithm, the communication among nodes is divided into two kinds, i.e., local communication and global communication. In the step of local communication, each node shares the correlation coefficients between its own residual and the columns of its own dictionary matrix with its several neighbors. Through such a local communication, each node finds a local reliable index that likely belongs to the support set. In the step of global communication, each node broadcasts the locally selected index to the whole network so that every node in the network knows all of the Q index candidates. Then fusion is performed by selecting some of these indices that occur more than once. The operations of DCOMP at the q-th node are summarized in Algorithm 2, where the set $G_q$ records the indices of the neighbors of the q-th node and itself.

---

**Algorithm 2 The DCOMP algorithm at the q-th node**

**Input:** $K$, $y_q$, $A_q$

**Initialization:**

1) Let $T^0 = \emptyset$ and the residual $r^0_q = y_q$.

**Iteration:** at the $l$-th iteration ($l \geq 1$)

2) Send the vector $c'_q = |A_q^T r_q^{l-1}|$ to and receive $c'_j$ from the $j$-th node, for all $j \in G_q \setminus \{q\}$.

3) Let $\lambda_q^l = \max_{q=1}^{s_q} \text{ind}(\sum_{q=1}^{s_q} c'_q, 1)$.

4) Send the locally selected index $\lambda_q^l$ to and receive $\lambda_j^l$ from the $j$-th node, for all $j \in G \setminus \{q\}$.
5) Let $\mathbf{u}'$ be a $Q$-length vector whose $q$-th element records the index selected by the $q$-th node, i.e., $u'(q) = \lambda^i_q$ for $q=1,2,\cdots, Q$. Let $\tilde{T}^i = \{\text{elements in } \mathbf{u}' \text{ that occur more than once}\}$; if $\tilde{T}^i = \emptyset$, i.e., if all elements in $\mathbf{u}'$ are distinct, let $\tilde{T}^i = \{u'(q)\}$ where $q$ is randomly selected from $G$. Let $T^i = T^{i-1} \cup \tilde{T}^i$.

6) If the cardinality $|T^i| < K$, increment $l = l+1$, calculate $r_q' = \text{resid}(\mathbf{y}_q, A_q(T^i))$, and return to Step 2; otherwise, stop the iteration.

Output: The estimated support set $\hat{T}_q = T^i$.

In DCOMP, the communications among nodes appear in Steps 2 and 4 of Algorithm 2, and the lengths of the messages transmitted from each node are $N$ and 1, respectively. Thus, the total number of messages transmitted from all the nodes is

$$C_{DCOMP} = L_{DCOMP} \left( \sum_{q \in G} N(g_q - 1) + (Q-1)Q \right),$$

(4)

where $L_{DCOMP}$ is the number of iterations of DCOMP, and $g_q$ is the cardinality of $G_q$. As reported in [25], the value of $L_{DCOMP}$ is probably smaller than $K$, since multiple indices may be selected per iteration by the fusion in Step 5 of DCOMP. Moreover, as shown in [26], the value of $L_{SSP}$ is comparable with that of $L_{DCOMP}$. Therefore, when $g_q \ll Q$, i.e., when each node in a large scale network is allowed to share $O(N)$-length vectors only with a few of its several neighbors, the communication cost of DCOMP in (4) is much less than that of SSP in (3). However, due to lack of backtracking operations, once an index is deemed as reliable and selected by DCOMP as an element of the estimated support set, it is not removed from there in subsequent iterations. This means that the strategy of index selection in DCOMP is too strict, and therefore, a large number of measurements is required to guarantee the success of each iteration of DCOMP, which is consistent with the simulation results in Section V.
III. DCSP FOR JOINT SPARSITY PATTERN RECOVERY

In this section, we develop the DCSP algorithm for joint sparsity pattern recovery and theoretically analyze its performance.

A. Algorithm description

Note that the majority of communication overhead in SSP comes from the frequent transmission of \( O(N) \)-length messages among nodes. From this observation, the DCSP algorithm significantly reduces the communication cost by avoiding transmission of \( O(N) \)-length messages in the network.

Algorithm 3 The DCSP algorithm at the \( q \)-th node

Input: \( K, y_q, A_q \).

Initialization:

1) Let \( \Gamma_q^0 = \text{max\_ind}(|A_q^T y_q|, K) \).

2) Send \( \Gamma_q^0 \) to and receive \( \Gamma_j^0 \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

3) Let \( \Gamma^0 = \{\Gamma_1^0, \Gamma_2^0, \cdots, \Gamma_q^0\} \) and \( T^0 = \text{max\_occ}(\Gamma^0, K) \), set the residual \( r_q^0 = \text{resid}(y_q, A_q(T^0)) \).

Iteration: at the \( l \)-th iteration (\( l \geq 1 \))

4) Let \( \hat{T}_q^l = T^{l-1} \cup \text{max\_ind}(|A_q^T y_q|, K) \) and \( \Gamma_q^l = \text{max\_ind}(|\text{proj}(y_q, A_q(\hat{T}_q^l))|, K) \).

5) Send \( \Gamma_q^l \) to and receive \( \Gamma_j^l \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

6) Let \( \Gamma^l = \{\Gamma_1^l, \Gamma_2^l, \cdots, \Gamma_q^l\} \) and \( T^l = \text{max\_occ}(\Gamma^l, K) \); update the residual \( r_q^l = \text{resid}(y_q, A_q(T^l)) \).

7) Send \( \|r_q^l\|_2 \) to and receive the \( \|r_q^l\|_2 \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

8) If \( \sum_{q \in G} \|r_q^l\|_2 \geq \sum_{q \in G} \|r_q^{l-1}\|_2 \), let \( T^l = T^{l-1} \) and stop the iteration; otherwise, let \( l = l+1 \), and return to Step 4.

Output: The estimated support set \( \hat{T}_q = T^l \).

The DCSP algorithm is summarized in Algorithm 3. The basic idea is that, the local estimation of the
support set is carried out at every node by finding the subspace that the local measurement vector most probably lies in, and then the global estimate of the support set is obtained by fusion based on a majority vote of all the local estimates. In the initialization steps, the local estimate of the support set is obtained by finding $K$ maximum correlation coefficients between the local measurement vector and the columns of the local dictionary matrix, and then the global support set estimate is initialized by finding $K$ elements in $\Gamma^*$ that have the highest frequency of occurrence. After initialization, in Step 4 of DCSP, each node first enlarges the set of index candidates and then selects $K$ indices corresponding to the largest projection coefficients. This finds a $K$-dimensional subspace that the local measurement vector most likely lies in. Then the $q$-th node broadcasts the $K$-length vector $\Gamma^*_q$, the local estimate of the support set, to the whole network, and as a result, the index set $\Gamma^i$ in Step 6 of DCSP is the same for all the nodes. There exist $KQ$ elements in $\Gamma^i$ and some of them may occur more than once. In Step 6 of DCSP, fusion of all the local estimates of the support set is carried out by majority voting, i.e., $K$ elements in $\Gamma^i$ that have the highest frequency of occurrence are selected as the global estimate of the support set. In Step 7 of DCSP, each node broadcasts once again to report the local recovery error. When the global recovery error reaches the minimum, iterations at all the nodes are terminated.

**B. Performance analysis**

In this subsection, we theoretically analyze the performance of DCSP. At the $l$-th iteration, define a $N \times 1$ binary vector $\mathbf{\beta}_q^l$ by

$$
\mathbf{\beta}_q^l(i) = \begin{cases} 
1, & \text{if } i \in \Gamma^i_q \\
0, & \text{otherwise}
\end{cases},
$$

for $i=1,2, \cdots, N$. That is to say, the entries “1” in $\mathbf{\beta}_q^l$ are indexed by the set $\Gamma^i_q$. Let

$$
\mathbf{\beta}' = \sum_{q=1}^{Q} \mathbf{\beta}_q^l.
$$

In the Step 6 of DCSP, the fusion based on a majority vote among multiple nodes is performed by finding $K$ elements from $\{\Gamma_1^i, \Gamma_2^i, \cdots, \Gamma_Q^i\}$ that have the highest frequency of occurrence. This is equivalent to saying
that $T'$ is composed of $K$ indices corresponding to the biggest coefficients in $\mathbf{p}'$.

1) Convergence of DCSP

Now we investigate the convergence of DCSP in the noise-free case.

**Proposition 1** When each dictionary matrix $A_q$ satisfies the RIP with the constant $\delta_{jk} \leq 0.165$ and the number of nodes is large enough, the support set estimated by DCSP converges to the true value as the iterations proceed (i.e. $T' \rightarrow T$) with a notably high probability.

**Proof:** The proof is composed of two parts. We will discuss the reliability of $T^0$ (in the initialization phase) and $T^l$ (at the $l$-th iteration with $l \geq 1$), respectively.

- **In the initialization phase**

We first look at the initialization steps in DSCP. At Step 1 of DCSP, $K$ indices corresponding to the largest correlation coefficients between $y_q$ and the columns of $A_q$ are recorded in the set $0\Gamma_q^0$. As reported in [27], when $A_q$ satisfies the RIP with constant $\delta_{jk} \leq 0.165$, we have

$$\|x_q(T^0 - \Gamma_q^0)\|_1 \leq \frac{8\delta_{jk} - 8\delta_{jk}^2}{1 + \delta_{jk}}\|x_q\|_1 \leq \sqrt{8\delta_{jk} - 8\delta_{jk}^2}\|x_q\|_1,$$

where the second inequality holds because $\delta_{jk} \leq \delta_{jk}$ [27] and the function $f(x) = \frac{\sqrt{8x-8x^2}}{1+x}$ is monotone increasing on $[0, 1/3]$. This implies that $T \cap \Gamma_q^0 \neq \emptyset$, for $q = 1, 2, \cdots, Q$. Assume that $K_q^0 = |T \cap \Gamma_q^0|$, i.e., $K_q^0$ indices in $\Gamma_q^0$ belong to $T$. It is obvious that $1 \leq K_q^0 \leq K$. Accordingly, other $K - K_q^0$ indices in $\Gamma_q^0$ belong to $T_c$, where $T_c = \{1, 2, \cdots, N\} \setminus T$. Note that $\Gamma_q^0 = \text{max}_\text{ind}(\|A_q^\mu y_q\|, K) = \text{max}_\text{ind}(\|A_q^\mu A_q(T)x_q(T)\|, K)$. Due to the randomness of the coefficients of $x_q(T)$, there is no a priori knowledge that some indices in $T$ have higher priority to be selected into $\Gamma_q^0$ than others. Thus, it is reasonable to assume that the indices in $T$ have an equal opportunity to be selected into $\Gamma_q^0$, i.e., the elements of $|T \cap \Gamma_q^0|$ are uniformly distributed within $T$. Similarly, the elements of $|T_c \cap \Gamma_q^0|$ are also uniformly distributed within $T_c$. Therefore, $\mathbf{p}_q^0(n)$ with $n \in T$ and $\mathbf{p}_q^0(m)$ with $m \in T_c$ follow Bernoulli distributions.
\[
\text{Pr}(\mathbf{p}_q(n) = 1) = \frac{C_{k-1}^{N,k} \cdot C_{N-k}^{k}}{C_{k}^{k} \cdot C_{N-k}^{k}} = \frac{K_0}{K}, \quad n \in T.
\]
\[
\text{Pr}(\mathbf{p}_q(n) = 0) = 1 - \text{Pr}(\mathbf{p}_q(n) = 1) = 1 - \frac{K_0}{K} 
\]
respectively, where \(K_0 \triangleq \min\{K_1, K_2, \ldots, K_Q\}\), and \(C' = \frac{p!}{j!(r-j)!}\). The definition of \(K_0\) implies that every node has correctly selected at least \(K_0\) indices after the Step 1 of DCSP. This is also equivalent to saying that at most \(K-K_0\) indices in \(\Gamma_q^0\) are incorrect for \(q=1,2,\ldots, Q\), since all \(\{\Gamma_1^0, \Gamma_2^0, \ldots, \Gamma_Q^0\}\) have the same cardinality \(K\). Furthermore, for \(n \in T\) and \(m \in T\), the covariance between \(\mathbf{p}_q(n)\) and \(\mathbf{p}_q(m)\) is given by
\[
\text{cov}(\mathbf{p}_q(n), \mathbf{p}_q(m)) = E(\mathbf{p}_q(n) \cdot \mathbf{p}_q(m)) - E(\mathbf{p}_q(n)) \cdot E(\mathbf{p}_q(m))
\]
\[
= C_{k-1}^{N,k} \cdot C_{N-k}^{k} \cdot \frac{K_0}{K} \cdot \frac{K - K_0}{N - K}
\]
\[
= 0.
\]
As proved in [29], two Bernoulli variables are independent if and only if they are uncorrelated. Thus, (10) implies that \(\mathbf{p}_q^0(n)\) and \(\mathbf{p}_q^0(m)\) are statistically independent of each other for \(n \in T\) and \(m \in T\).

Since \(T^0\) is composed of \(K\) indices corresponding to the biggest coefficients in \(\mathbf{p}^0\), the probability \(\text{Pr}(\mathbf{p}^0(n) > \mathbf{p}^0(m))\) for \(n \in T\) and \(m \in T\) represents the reliability of \(T^0\). From (8) and (9), it is clear that \(\text{Pr}(\mathbf{p}_q^0(n) = 1)\) increases and \(\text{Pr}(\mathbf{p}_q^0(m) = 1)\) decreases with an increasing \(K_q^0\), for \(n \in T\) and \(m \in T\). This implies, from (6), that \(\text{Pr}(\mathbf{p}^0(n) > \mathbf{p}^0(m))\) reaches the minimum when \(K_q^0 = K_q^0\) for \(q=1,2,\ldots, Q\). In what follows, we investigate the lower bound of \(\text{Pr}(\mathbf{p}^0(n) > \mathbf{p}^0(m))\) for \(n \in T\) and \(m \in T\), by assuming \(K_q^0 = K_q^0\) for \(q=1,2,\ldots, Q\).

Note that \(\Gamma_q^0 = \max\{\|A_{q}^0 x_q\|, K\}\) and \(\text{Pr}(\mathbf{p}_q^0(T) = \mathbf{x}_q(T))\), \(K\). We can conclude that \(\{\Gamma_q^0, q=1,2,\ldots, Q\}\) are independent due to the independence among \(\{\mathbf{x}_q(T), \mathbf{p}_q^0, q=1,2,\ldots, Q\}\).
\( q = 1, 2, \cdots, Q \}. \) Then, it can be deduced that \( \mathbf{b}^q(n) \) with \( n \in T \) and \( \mathbf{b}^q(m) \) with \( m \in T_p \) follow Binomial distributions \( B(Q, p^n_0) \) and \( B(Q, p^m_0) \), respectively, where

\[
\begin{align*}
  p^n_0 & \triangleq \frac{K^n_0}{K} \\
  p^m_0 & \triangleq \frac{K - K^n_0}{N - K} = \frac{1 - K^n_0 / K}{N / K - 1}
\end{align*}
\]  

(11)

From (10) it follows that \( \mathbf{b}^q(n) \) and \( \mathbf{b}^q(m) \) are statistically independent for \( n \in T \) and \( m \in T_p \). Thus, we have

\[
\Pr(\mathbf{b}^q(n) > \mathbf{b}^q(m)) = \sum_{i=0}^{Q} \sum_{j=0}^{Q} \Pr(\mathbf{b}^q(n) = i) \cdot \Pr(\mathbf{b}^q(m) = j)
\]

\[
= \sum_{i=0}^{Q} \sum_{j=0}^{Q} \Pr(\mathbf{b}^q(n) = i) \cdot \Pr(\mathbf{b}^q(m) = j)
\]

\[
= \sum_{i=0}^{Q} \sum_{j=0}^{Q} C^Q_0 p^n_0^i (1 - p^n_0)^{Q - i} \cdot \sum_{i=0}^{Q} C^Q_0 p^m_0^j (1 - p^m_0)^{Q - j}
\]

\[
\triangleq F_Q \left( \frac{K^n_0}{K}, \frac{K}{N} \right)
\]

(12)

According to (11), a necessary and sufficient condition for \( p^n_0 > p^m_0 \) is

\[
K^n_0 > \frac{K^2}{N},
\]

(13)

which is quite easy to satisfy under the assumption that \( N \gg K \). Hence, it is reasonable to assume that \( p^n_0 > p^m_0 \), since we are interested in the scenario of sparse pattern recovery. Then from (12) we have

\[
\lim_{Q \to \infty} \Pr(\mathbf{b}^q(n) > \mathbf{b}^q(m)) > 0.99
\]

(14)

for \( n \in T \) and \( m \in T_p \), by using the well-known central limit theorem. The proof of (14) is given in Appendix A.

From (14) it is clear that, with a sufficient number of nodes, every element in \( T^o \) is correct with high probability. Thus, it can be concluded that

\[
\| \mathbf{x}_q (T - T^o) \|_2^2 \leq \| \mathbf{x}_q (T - \Gamma_q^o) \|_2^2
\]

holds with high probability when the value of \( Q \) is large enough.

- At the l-th iteration \((l \geq 1)\)

In Step 4 of DCSP, the set of index candidates is enlarged and a new group of \( K \) indices corresponding to the largest projection coefficients is selected at each node. By doing so, it is possible to remove some
wrong indices that were considered reliable during past iterations and to add some new indices into the support set estimate, i.e., the reliability of the support set estimate is reevaluated based on the local measurement data. This step is the same with that in the standard SP algorithm, so according to Theorem 3 and Theorem 4 in [27] we have

\[ \|x_q(T - \Gamma_q^l)\|_2 \leq \frac{2\delta_k(1 + \delta_k)}{(1 - \delta_k)^2} \|x_q(T - \Gamma_q^{l-1})\|_2 < \|x_q(T - \Gamma_q^{l-1})\|_2. \]  

(16)

where the second inequality holds since the RIP constant \( \delta_{kk} \leq 0.165 \). From (15) and (16) it follows that

\[ \|x_q(T - \Gamma_q^l)\|_2 < \|x_q(T - \Gamma_q^{l-1})\|_2. \]  

(17)

holds with high probability. Assume that, at the \( l \)-th iteration, the \( q \)-th node has correctly selected \( K_q^l \) indices that belong to the true support set after the Step 4 of DCSP, i.e., \( K_q^l = |T \cap \Gamma_q^l| \). Let \( K^l = \min\{K_q^l, K_{q1}^l, \cdots, K_{qQ}^l\} \). Similar to the analysis on the initialization steps of DCSP, here we also investigate the lower bound of \( \Pr\{p^l(n) > p^l(m)\} \) for \( n \in T \) and \( m \in T_c \), by assuming \( K_q^l = K^l \) for \( q = 1, 2, \cdots, Q \). Note from (17) that

\[ K^l > K_q^{l-1}. \]  

(18)

As stated in Step 4 of DCSP, \( \Gamma_q^l \) is obtained from the projection coefficients of \( y_q = A_q(T)x_q(T) \) onto a determined subspace. Thus, when \( K^l < K \) (i.e., before the termination of DCSP), \( \{\Gamma_q^l, q = 1, 2, \cdots, Q\} \) and, therefore, \( \{p^l_q, q = 1, 2, \cdots, Q\} \) are independent due to the independence among \( \{x_q(T), q = 1, 2, \cdots, Q\} \). Moreover, due to the randomness of the coefficients of \( x_q(T) \), there is no a priori knowledge that some indices in \( T \) have higher priority to be selected into \( \Gamma_q^l \) than others. Thus, it is reasonable to assume that the elements of \( |T \cap \Gamma_q^l| \) are uniformly distributed within \( T \). Similarly, elements of \( |T_c \cap \Gamma_q^l| \) are also uniformly distributed within \( T_c \). Similar to the analysis in the first part of this proof, it can be deduced that \( \mathcal{B}(n) \) with \( n \in T \) and \( \mathcal{B}(m) \) with \( m \in T_c \) follow Binomial distributions \( \mathcal{B}(Q, p^l_q) \) and \( \mathcal{B}(Q, p^l_{q1}) \), respectively, where
Moreover, $\mathbf{b}^i(n)$ and $\mathbf{b}^i(m)$ are statistically independent, for $n \in T$ and $m \in T^c$, since $\text{cov}(\mathbf{b}^i_q(n), \mathbf{b}^i_q(m)) = 0$ similar to (10). From (18) and (19) it follows

$$p^i_1 > p^i_{1-1} > p^i_{2-1} > p^i_2.$$  

Similar to the proof in Appendix A, by using the central limit theorem, we can deduce that

$$\lim_{Q \to \infty} \Pr(\mathbf{b}^i(n) > \mathbf{b}^i(m)) > 0.99,$$  

for $n \in T$ and $m \in T^c$. When $K^i > 0.5 K$, which happens as the iterations proceed according to (18), we further have

$$\lim_{Q \to \infty} \Pr(\mathbf{b}^{i+1}(n) > \mathbf{b}^{i+1}(m)) = \lim_{Q \to \infty} \Pr(\mathbf{b}^i(n) > \mathbf{b}^i(m))$$

for $n \in T$ and $m \in T^c$. The proof of (22) is given in Appendix B. Both (21) and (22) implies that, as the iterations proceed, every element in $T^i$ is correct with high probability if the number of nodes is sufficiently large. Therefore,

$$\|x_q(T - T^i)\| \leq \|x_q(T - T^i_q)\|$$

holds with high probability if the value of $Q$ is large enough. From (16) and (23), it follows that

$$\|x_q(T - T^i)\| < \|x_q(T - T^{i+1})\|,$$

holds with high probability, when the value of $Q$ is large enough. This implies that DCSP with sufficient number of nodes ensures the continuous reduction of the error of support set estimation. This completes the proof of Proposition 1. ■

Proposition 1 states that, when the number of nodes is large enough, $T^i$ estimated by DCSP approaches the true support set $T$ with a notably high probability as the iterations proceed. This also allows us to analyze the number of nodes required for DCSP to guarantee the reliability of the support set estimate. According to (12), given different values of $K^0 / K$ and $N / K$, in Fig. 1 we plot the probability
\[ \Pr(\beta'(n) > \beta'(m)) \] for \( n \in T \) and \( m \in T_c \) versus the number of nodes. We can see that, to ensure that
\[ \Pr(\beta'(n) > \beta'(m)) \geq 0.99 \] for \( n \in T \) and \( m \in T_c \), 13, 17, 47, and more than 50 nodes are required when the parameter pair \( (K^0/K, N/K) \) is equal to \((0.4, 20), (0.4, 10), (0.2, 20)\) and \((0.2, 10)\), respectively. It is clear that the required value of \( Q \) decreases with the increasing values of \( K^0/K \) and \( N/K \). This conclusion is straightforward from the following considerations: 1) the more correct indices in the local estimate of the support set, the fewer nodes are required to achieve a desired accuracy of the fusion; and 2) the sparser the signal, the smaller the probability that different nodes select the same wrong indices from \( T_c \). As stated in the proof of Proposition 1, \( K' \), the lower bound of the number of correct indices selected at each node, increases with the increasing iteration index \( l \), i.e., \( K \geq K' > K'^{-1} > \cdots > K^0 \geq 1 \). Simultaneous consideration of this fact and the trend in Fig. 1 provide that, if \( Q \) is large enough so that \( \Pr(\beta'(n) > \beta'(m)) \geq 0.99 \) holds for \( l=0 \), it is also sufficient to guarantee \( \Pr(\beta'(n) > \beta'(m)) \geq 0.99 \) for \( l \geq 1 \). Therefore, to ensure that \( \Pr(\beta'(n) > \beta'(m)) \geq 0.99 \) with \( n \in T \) and \( m \in T_c \) holds through the whole iterative process, the sufficient number of nodes can be given by
\[ Q_{\text{uf}} = Q \left| F_{\Theta} \left( \frac{1}{K \cdot N} \right) \geq 0.99 \right., \] (25)
where \( F_{\Theta}(\cdot) \) is defined in (12). It is worth emphasizing that the actual number of nodes required for DCSP to successfully recover the joint sparsity pattern may be much smaller than \( Q_{\text{uf}} \), as shown by simulations in Section V. The reason is that the above analysis is carried out for the lower bound of \( \Pr(\beta'(n) > \beta'(m)) \) with \( n \in T \) and \( m \in T_c \) based on the assumption that \( K'_q = K' = \min \{ K'_1, K'_2, \ldots, K'_Q \} \) for \( q=1,2, \ldots, Q \). When \( K'_q > K' \) happens at some nodes, i.e., when at some nodes the accuracy of local support set estimate is better than expected, the number of nodes required to guarantee the desired fusion performance will be decreased.
2) The number of iterations and the communication cost

As indicated by Proposition 1, fusion of the local support set estimates in DCSP can increase the number of correctly selected indices at each iteration, compared to the standard SP algorithm with no collaboration. This is equivalent to saying that the number of iterations can be significantly reduced. In the following, we investigate the number of iterations needed for DCSP to successfully recover the support set.

From (7), (15), (16) and (23), it follows that

$$\| \mathbf{x}_i(T - T') \| \leq \left[ \frac{2\delta_{i,k} (1 + \delta_{i,k})}{(1 - \delta_{i,k})^2} \right] \frac{\sqrt{8\delta_{i,k}^2 - 8\delta_{i,k}^2}}{1 + \delta_{i,k}} \| \mathbf{x}_i \|,$$

holds with high probability, if the number of nodes is sufficiently large. The necessary and sufficient condition for $T = T'$ is that $\| \mathbf{x}_i(T - T') \| < \min \{ |\mathbf{x}_i(i), i \in T \}$. Hence, the number of iterations required for DCSP to accurately recover the joint sparsity pattern can be expressed by

$$L_{DCSP} = \left\lceil \frac{\log \left( \frac{\alpha \cdot 1 + \delta_{i,k}}{\sqrt{8\delta_{i,k}^2 - 8\delta_{i,k}^2}} \right)}{\log \left( \frac{2\delta_{i,k} (1 + \delta_{i,k})}{(1 - \delta_{i,k})^2} \right)} \right\rceil,$$

Fig. 1 The probability $\Pr(\beta^0(n) > \beta^0(m))$ for $n \in T$ and $m \in T_c$ versus the value of $Q$. 

---

18
where \( \lceil \cdot \rceil \) is the ceiling function and
\[
\alpha \triangleq \frac{\min\{|x_q(i)|, i \in T\}}{|x_q|_E}. \tag{28}
\]
It is clear that the value of \( L_{DCSP} \) depends on the decay of the nonzero coefficients of the sparse signal.

Consider a special case of random coefficients of \( \{x_q(T), q=1, 2, \ldots, Q\} \), where the coefficients of \( x_q(T) \) are generated from the same decay function but arranged in random order for \( q=1, 2, \ldots, Q \). In this case, the value of \( L_{DCSP} \) can be exactly determined by (27). For example, for the sparse signals with exponentially decaying entries, in which the \( n \)-th large amplitude is constrained by \( c_{\text{exp}} \cdot e^{-p_{\text{exp}}(n-1)} \) for some constants \( c_{\text{exp}} > 0 \) and \( p_{\text{exp}} > 0 \), (28) can be specified as
\[
\alpha_{\text{exp}} = e^{-p_{\text{exp}}(K-1)} \cdot \frac{1-e^{-2p_{\text{exp}}}}{1-e^{-2p_{\text{exp}}^b}}. \tag{29}
\]
For the sparse signals with power-law decaying entries, in which the \( n \)-th large amplitude is constrained by \( c_{\text{power}} \cdot n^{-p_{\text{power}}} \) for some constants \( c_{\text{power}} > 0 \) and \( p_{\text{power}} > 1 \), (28) becomes
\[
\alpha_{\text{power}} = \frac{K^{-p_{\text{power}}}}{\sum_{n=1}^{K} n^{-p_{\text{power}}}}. \tag{30}
\]

Based on (27)-(30), in Fig. 2 we plot the number of iterations required for DCSP to exactly recover the joint sparsity pattern versus the varying RIP constant \( \delta_{3K} \). The constants of exponentially decaying function are \( c_{\text{exp}} = 1 \) and \( p_{\text{exp}} = 0.3 \), while the constants of power-law decaying function are \( c_{\text{power}} = 1 \) and \( p_{\text{power}} = 1.5 \). For both kinds of sparse signals, the sparsity \( K \) is fixed to 10. It is clear from Fig. 2 that the required number of iterations of DCSP is in the order of \( O(K) \). Moreover, when the value of \( \delta_{3K} \) goes down (approximately corresponding to the increase of the number of measurements per node), the joint sparsity pattern can be well recovered after just one iteration of DCSP. This is consistent with the simulation results provided in section V. It should be pointed out that, even if the fusion in initialization phase of DCSP (Step 3 in Algorithm 3) successfully recovers the support set, one iteration is still needed to check whether the
algorithm should be terminated (Step 8 in Algorithm 3). That is to say, the minimum number of iterations of DCSP is 1.

![Graph showing required number of iterations in DCSP versus RIP constant $\delta_{1,k}$](graph.png)

Fig. 2 The required number of iterations in DCSP versus RIP constant $\delta_{1,k}$.

With the bound on the number of iterations, the communication cost of DCSP can be easily estimated. In DCSP, the communications among nodes appear in Steps 2, 5 and 7 of Algorithm 3, and the lengths of the messages transmitted from each node are $K$, $K$ and 1, respectively. Thus, the total number of messages to be transmitted from all the nodes is

$$C_{DCSP} = [K + L_{DCSP}(K + 1)](Q - 1)Q,$$

where the bound of $L_{DCSP}$ is given in (27). Compared to SSP and DCOMP, DCSP completely avoids the transmission of $O(N)$-length vectors. Also note that all of $L_{SSP}$, $L_{DCOMP}$ and $L_{DCSP}$ are in the order of $O(K)$. Therefore, from (3) and (31) it is clear that the communication overhead of DCSP is much less than that of SSP. The comparison between (4) and (31) indicates that the communication overheads of DCOMP and DCSP are comparable when $g_q \ll Q$. However, in contrast to DCOMP, DCSP is capable of removing wrong indices that were deemed reliable in past iterations from the estimated support set. This is achieved by two kinds of operations: 1) in Steps 4 of DCSP, each node reevaluates the reliability of the previously selected indices and the newly added indices; 2) in Step 6 of DCSP, poor index candidates corresponding to low
frequency of occurrence are rejected by fusion based on majority vote. Thus the accuracy of sparse pattern recovery of DCSP is expected to be much better than that of DCOMP, which will also be shown by simulations in Section V.

IV. GENERALIZED DCSP ALGORITHM FOR JOINT SPARSITY PATTERN RECOVERY

In this section, considering the tradeoff between the communication cost and the accuracy of support set estimation, we propose the GDCSP algorithm. In [24] and [25], it was proved that sharing of $N$-length correlation coefficients and projection coefficients among some nodes is helpful to improve the accuracy of the local support set estimation and, therefore, the accuracy of global estimation after fusion. In DCSP proposed in the previous section, the transmission of $N$-length vectors is completely prohibited. In GDCSP proposed below, the transmission of $O(N)$-length vectors is allowed but restricted to a small neighborhood surrounding each node. As a result, the accuracy of support set estimation is expected to be improved in comparison with DCSP, at the price of slightly increased communication overhead.

Algorithm 4 The GDCSP algorithm at the $q$-th node

Input: $K$, $y_q$, $A_q$.

Initialization:

1) Send the vector $c^{i}_{q} = |A_{q}^{H}y_{q}|^2$ to and receive $c^{i}_{j}$ from the $j$-th node, for all $j \in G_q \{q\}$.

2) Let $\Gamma^{0}_{q} = \text{max}\_\text{ind}(\sum_{q \in G_q} c^{0}_{q}, K)$.

3) Send $\Gamma^{0}_{q}$ to and receive $\Gamma^{0}_{j}$ from the $j$-th node, for all $j \in G \{q\}$.

4) Let $\Gamma^{0} = \{\Gamma^{0}_{1}, \Gamma^{0}_{2}, \ldots, \Gamma^{0}_{Q}\}$ and $T^{0} = \text{max}\_\text{occ}(\Gamma^{0}, K)$, set the residual $r^{0}_{q} = \text{resid}(y_{q}, A_{q}(T^{0}))$.

Iteration: at the $l$-th iteration ($l \geq 1$)

5) Send the vector $c^{i'}_{q} = |A_{q}^{H}r^{l-1}_{q}|^2$ to and receive $c^{i'}_{j}$ from the $j$-th node, for all $j \in G_q \{q\}$.
6) Let \( \tilde{T}_q^i = T_q^{i-1} \cup \max_{\text{ind}}(\sum_{q \in G_q} \bar{e}_q^i, K) \); calculate the projection coefficients \( \bar{d}_q^i = \text{proj}(\bar{y}_q, A_q(\tilde{T}_q^i)) \).

7) Send \( \bar{d}_q^i \) to and receive \( \bar{d}_j^i \) from the \( j \)-th node, for all \( j \in G_q \setminus \{q\} \).

8) Let \( \Gamma_q^i = \max_{\text{ind}}(\sum_{qG_q} \bar{d}_q^i, K) \).

9) Send \( \Gamma_q^i \) to and receive \( \Gamma_j^i \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

10) Let \( \Gamma_q^i = \{\Gamma_q^i, \Gamma_1^i, \ldots, \Gamma_q^i\} \) and \( T_q^i = \max_{\text{occ}}(\Gamma_q^i, K) \); update the residual \( r_q^i = \text{resid}(\bar{y}_q, A_q(T_q^i)) \).

11) Send \( \|r_q^i\| \) to and receive the \( \|r_j^i\| \) from the \( j \)-th node, for all \( j \in G \setminus \{q\} \).

12) If \( \sum_{q \in G} \|r_q^i\| \geq \sum_{q \in G} \|r_q^{i-1}\| \), let \( T_q^i = T_q^{i-1} \) and stop; otherwise, let \( l = l + 1 \), and return to Step 5.

Output: The estimated support set \( \hat{T}_q = T_q^i \).

The GDCSP algorithm is summarized in Algorithm 4. Here, each node first collaborates with its several neighbors to obtain the local estimate of the support set, and then fusion based on majority vote is performed based on all the local estimates. Different from DCSP, in GDCSP each node communicates with its neighbors twice per iteration besides broadcasting the \( K \)-length local estimate of the support set to the whole network. After initialization, the first collaboration among neighboring nodes appears in Step 5 of GDCSP, where the \( q \)-th node shares an \( N \)-length correlation coefficient vector \( \bar{e}_q^i \) with its neighbors. The indices of the neighbors of the \( q \)-th node and itself are recorded in the set \( G_q \) with the cardinality \( |G_q| = g_q \). The coefficients in \( \bar{e}_q^i \) are probably not sparse before the iterations converge, so the transmission of the entire \( \bar{e}_q^i \) is necessary. The neighboring nodes collaborate once again in Step 7 of the GDCSP, where the \( q \)-th node shares a \( 2K \)-length projection coefficient vector \( \bar{d}_q^i \) with its neighbors. By such two-stage collaboration, the \( q \)-th node selects \( K \) indices from \( \Gamma_q^i \) as the local estimate of the support set by finding a \( K \)-dimensional subspace that the measurement vectors collected within the neighborhood most probably lie in. Other steps of GDCSP such as the broadcast of locally estimated support set and the majority voting fusion among all the nodes are the same with that of DCSP. When \( G_q = 1 \) for all \( q \in G \), GDCSP will degenerate into DCSP.
described in Algorithm 3. It is predictable that the accuracy of sparsity pattern recovery of GDCSP will be further improved as the value of $g_q$ increases according to the analysis in [24] and [25]. When $G_q = G$ for all $q \in G$, the fusion of locally estimated support sets in GDCSP can be omitted since the support set estimates at all the nodes will be the same, and accordingly, GDCSP will be equivalent to the decentralized version of SSP described in Algorithm 1.

The required number of iterations for GDCSP to exactly recover the support set is slightly smaller than that in DCSP, since sharing of $O(N)$-length coefficients among neighbors improves the accuracy of local estimation of the support set. Let $L_{GDCSP}$ denote the required number of iterations for GDCSP. Then the communication cost of GDCSP can be estimated by

$$
C_{GDCSP} = N \sum_{q \in G} (g_q - 1) + K(Q - 1)Q + L_{GDCSP} \left( (N + 2K) \sum_{q \in G} (g_q - 1) + (K + 1)(Q - 1)Q \right).
$$

(32)

In what follows, we compare different algorithms in terms of the communication cost. Without loss of generality, a symmetric network is considered and two assumptions are made: 1) $g_q = g < Q$ for all $q \in G$; 2) the neighbors of the $l$-th node are indexed by \{mod($q + 1, Q$) + 1, mod($q + 2, Q$) + 1, $\ldots$, mod($q + g - 1, Q$) + 1\}, where mod(·) is the modulus operation. The numbers of messages to be transmitted in different algorithms are listed in Table 1. JOMP and JSP have the smallest communication overhead in Table 1, since in these two algorithms fusion is performed only after the iterative solution processes are completed at all the nodes. However, as shown in the next section, the accuracy of sparsity pattern recovery of JOMP and JSP is unsatisfactory. Since $K \ll N$, $g < Q$ and all values of $L_{DCOMP}$, $L_{SSP}$, $L_{DCSP}$, and $L_{GDCSP}$ are in the order of $O(K)$, we can see that the communication costs of DCSP and GDCSP are in the same order of magnitude as that of DCOMP and much less than that of SOMP and SSP.

| Algorithm     | Number of transmitted messages |
|--------------|--------------------------------|
| JSP, JOMP    | $K(Q - 1)Q$                    |
V. SIMULATION RESULTS

In this section, some simulation results are provided to demonstrate the performance of the proposed DCSP and GDCSP algorithms.

A. Accuracy of sparsity pattern recovery

We evaluate the accuracy of sparsity pattern recovery of different algorithms. The sparsity is fixed to $K=10$, and the length of sparse signal at each node is set to $N=200$. There are $Q=6$ nodes in the decentralized network. The simulations are carried out as follows.

a) Randomly generate a set of $M \times N$ dictionary matrices $\{A_q, q=1, 2, \cdots, Q\}$ from the standard independent and identically distributed (iid) Gaussian ensemble.

b) Randomly select $K$ indices from $\{1, 2, \cdots, N\}$ as the support set $T$, and draw the entries of $x_q$ supported on $T$ from the standard iid Gaussian ensemble.

c) Generate the measurement vectors $y_q=A_qx_q$, for $q=1, 2, \cdots, Q$. Apply different algorithms to recover the support set. If the recovery result $T^l = T$ at the $l$-th iteration, the recovery is considered as successful.

d) Count the success rate of sparsity pattern recovery over 500 Monte Carlo trials.

In Fig. 3, we compare DCSP and GDCSP with five other algorithms (i.e. SSP [24], SOMP [22], DCOMP [25], JOMP and JSP [18]) in terms of accuracy of sparsity pattern recovery. Assuming that all of the 6 nodes fully collaborate without any restriction, SSP and SOMP requires $M \geq 24$ and $M \geq 28$.
to achieve success rate greater than 95%, respectively, while the former is the best performance in Fig. 3. With the size of neighborhood of each node $g=3$, GDCSP and DCOMP require $M \geq 26$ and $M \geq 32$ to achieve success rates greater than 95%, respectively, which means that GDCSP has much better recovery capability than DCOMP. It is worth emphasizing that, GDCSP yields slightly higher success rate than SOMP, although the former restricts the transmission of $O(N)$-length messages to a small neighborhood consisting of 3 nodes and the latter allows the transmission of $O(N)$-length messages among all the 6 nodes. The reason is that GDCSP is capable of removing wrong indices from the estimated support set while SOMP cannot do it. Because of the same reason, the DCSP algorithm offers better accuracy ($M \geq 30$ for the success rate greater than 95%) than DCOMP with $g=3$. If the transmission of $O(N)$-length messages is prohibited in the whole network, which applies when the communication capacity of the network is extremely limited, only DCSP and DCOMP with $g=1$ are eligible among these algorithms. As shown in Fig. 3, the success rate of DCSP is much better than that of DCOMP with $g=1$. In other words, DCSP is the only one among these algorithms that can achieve satisfactory accuracy of sparsity pattern recovery with no transmission of $O(N)$-length messages among nodes.

Next, we compare the accuracy of sparsity pattern recovery of different algorithms in a noisy environment. The measurement vector at the $q$-th node is $y_q = A_q x_q + w_q$, where $w_q$ is the additive Gaussian noise. We assume that the noise is statistically independent of the sparse signals. The signal-to-noise ratio (SNR) is defined as $\text{SNR} \triangleq \sum_{q=1}^{Q} \| A_q x_q \|_2^2 / (QM \sigma_w^2)$, where $\sigma_w^2$ is the variance of the noise. Let $K=10$, $M=50$, $N=200$ and $Q=6$. Through 500 Monte Carlo trials, Fig. 4 plots the success rates of sparse pattern recovery of five algorithms (i.e. SSP, SOMP, DCOMP, DCSP, and GDCSP) that have higher success rates in Fig. 3. JOMP and JSP are not included here, because their success rates are unsatisfactory even in noise-free case under this parameter setting. When the transmission of $O(N)$-length messages is prohibited in the network, i.e., $g=1$, DCSP provides higher success rate than DCOMP. When the size of neighborhood of each node is
$g=3$, GDSCP has better accuracy than DCOMP. As motioned in the previous experiment, the reason is that DCSP and GDCSP have the capability of removing poor index candidates that were deemed reliable in past iterations from the estimated support set while DCOMP cannot.

![Graph showing success rate vs. number of measurements per node](image1)

Fig. 3 The success rate of sparsity pattern recovery versus the number of measurements per node in noise-free case, for independent sparse signal model. $Q=6$, $K=10$, and $N=200$.

![Graph showing success rate vs. SNR](image2)

Fig. 4 The success rate of sparsity pattern recovery versus SNR, for independent sparse signal model. $Q=6$, $K=10$, and $N=200$. 

26
\[ K=10, M=50, \text{ and } N=200. \]

**B. Communication cost and convergence speed**

In this subsection, we compare communication costs and convergence speeds of five algorithms (i.e. SSP, SOMP, DCOMP, DCSP, and GDCSP) that have higher success rates in Fig. 3, i.e., JOMP and JSP are not considered. Let \( M=50, N=200 \) and \( K=10 \). Assume that the network scale is increasing, i.e., \( Q \) is varying from 5 to 40. The average numbers of iterations required for these algorithms to exactly recover the sparsity pattern are plotted in Fig. 5 over 500 Monte Carlo trials. Thanks to the efficient selection of indices at each node and the effective collaboration among nodes, SSP, DCSP and GDCSP have faster convergence speeds compared to DCOMP and SOMP. In particular, SSP, DCSP and GDCSP can even recover the joint sparsity pattern from only one iteration when the number of nodes is sufficiently large. This means that, fusion in the initialization phase of DCSP (Step 3 in Algorithm 3) has correctly recovered the support set, and only one iteration is needed to check whether the algorithm should be terminated (Step 8 in Algorithm 3). This is consistent with the theoretical analysis in Proposition 1.

The comparison of these algorithms in terms of communication cost is demonstrated in Fig. 6, which is consistent with Table 1. Although DCOMP with \( g=1 \) has the least communication cost in Fig. 6, its accuracy of sparsity pattern recovery is unsatisfactory as shown in Fig. 3. The superiority of DCSP in terms of communication efficiency lies in the following facts: 1) the exchange of \( O(N) \)-length messages in the network is completely avoided; 2) less iterations are needed for convergence. With the fixed size of the neighborhood surrounding each node \( g=3 \), the communication overhead of GDCSP is comparable with that of DCOMP: in a small scale network, the number of transmitted messages of GDCSP is slightly lower than that of DCOMP, while the situation is reversed in a large scale network.
Fig. 5 The number of required iterations versus the network scale, for independent sparse signal model.

\[ M=50, N=200 \text{ and } K=10. \]

Fig. 6 The number of messages to be transmitted versus the network scale, for independent sparse signal model. \[ M=50, N=200 \text{ and } K=10. \]

C. When all the nodes observe the same sparse signal

Although the performance analysis in Section III is provided for the case where the coefficients of the
sparse signals at different nodes are statistically independent, we would like to experimentally evaluate the proposed DCSP and GDCSP algorithms when the sparse signals at all the nodes are the same. This signal model can be expressed as $y_q = A_q x$, and $K$ nonzero coefficients of the sparse signal $x$ are supported on $T$. Let $N=200$, $K=10$, and $Q=6$. In Fig. 7, we compare multiple algorithms in terms of the accuracy of sparsity pattern recovery with varying number of measurements. The simulations are carried out in a similar way as the first experiment, but here in each trial we randomly generate a $K$-sparse signal $x$ and assign it to all the nodes, i.e., $x_q = x$ for $q=1, 2, \cdots, Q$. Compared to Fig. 3, the success rate of every algorithm in Fig. 7 is increased. That is to say, when all the nodes observe the same sparse signal, these algorithms offer better accuracy of sparsity pattern recovery than that in the case of independent sparse signals. From Fig. 7, we also see the superiority of DCSP and GDCSP to DCOMP, similar to what we observed in Fig. 3.

The comparisons of five algorithms in terms of the convergence speed and the communication cost are provided in Figs. 8 and 9, respectively. Compared to Figs. 5 and 6 where the sparse signals at different nodes are independent, the communication costs and the numbers of iterations of these algorithms are slightly increased, except for SOMP. With the same sparse signal at all the nodes, $\{p_q, q=1, 2, \cdots, Q\}$ are probably similar to each other, and the difference among them is only caused by the difference among $\{A_q, q=1, 2, \cdots, Q\}$. This definitely improves the performance of fusion based on majority vote, as shown in Fig. 7; however, this reduces the chance to select more than $K$ correct indices from the majority voting fusion at the $l$-th iteration. This is the reason why DCSP, GDCSP, DCOMP, and SSP require more iterations and have relatively higher communication costs when all the nodes observe the same sparse signal (Figs. 8 and 9), compared to that in the independent sparse signal case (Figs. 5 and 6). Similar to those we observed in Figs. 5 and 6, from Figs. 8 and 9 we can also see the communication-efficient characteristic and the fast convergence speed of DCSP and GDCSP compared to other algorithms.
Fig. 7 The success rate of sparsity pattern recovery versus the number of measurements per node, when all the nodes observe the same sparse signal. $Q=6$, $K=10$, and $N=200$.

Fig. 8 The number of required iterations versus the network scale, when all the nodes observe the same sparse signal. $M=50$, $N=200$, and $K=10$. 
VI. CONCLUSION

In this paper, we addressed the problem of collaborative recovery of common sparsity pattern with a decentralized network. By embedding collaboration among all the nodes into each iteration of the standard SP algorithm, we developed an algorithm named DCSP for decentralized recovery of the sparsity pattern. An attractive characteristic of DCSP is small communication overhead, since the transmission of $O(N)$-length messages in the network is completely avoided. This is helpful when the communication cost is the most concern of a practical system. We further extend DCSP to GDCSP by restricting the transmission of $O(N)$-length messages to a small neighborhood surrounding each node. Compared to DCSP, GDCSP has better accuracy of sparsity pattern recovery at the price of the slightly increased communication overhead. Simulation results show that, compared to other decentralized greedy algorithms, DCSP and GDCSP offer satisfactory recovery accuracy with much less communication costs. Our approach presented here can be easily combined with the compressive sampling matching pursuit (CoSaMP) algorithm [28], since CoSaMP
and SP are very similar to each other. Our future work will include the development of decentralized greedy algorithms for recovery and approximation of structured sparse signals.

APPENDIX

A. Proof of (14)

When the value of $Q$ is large enough, by the well-known central limit theorem, the probability distributions of $\beta^o(n)$ and $\beta^o(m)$ can be approximated by normal distributions $\mathcal{N}(Qp_1^o, Qp_1^o(1-p_1^o))$ with mean $Qp_1^o$ and variance $Qp_1^o(1-p_1^o)$ and $\mathcal{N}(Qp_2^o, Qp_2^o(1-p_2^o))$ with mean $Qp_2^o$ and variance $Qp_2^o(1-p_2^o)$, respectively. Furthermore, $\beta^o(n)$ and $\beta^o(m)$ are statistically independent of each other according to (10). Therefore, it can be concluded that the probability that the variable pair $\{\beta^o(n), \beta^o(m)\}$ is located within the region $[Qp_1^o - 3\sqrt{Qp_1^o(1-p_1^o)}, Qp_1^o + 3\sqrt{Qp_1^o(1-p_1^o)}] \times [Qp_2^o - 3\sqrt{Qp_2^o(1-p_2^o)}, Qp_2^o + 3\sqrt{Qp_2^o(1-p_2^o)}]$ is larger than 0.99. Also note $p_1^o > p_2^o$. Therefore, when $Q > \frac{9}{4(p_1^o - p_2^o)^2}$, we have

\[
\begin{align*}
Qp_1^o - 3\sqrt{Qp_1^o(1-p_1^o)} &\geq Qp_1^o - \frac{3}{2}\sqrt{Q} > Qp_2^o \\
Qp_2^o + 3\sqrt{Qp_2^o(1-p_2^o)} &\leq Qp_2^o + \frac{3}{2}\sqrt{Q} < Qp_1^o
\end{align*}
\]

(33)

This means that $[Qp_1^o - 3\sqrt{Qp_1^o(1-p_1^o)}, Qp_1^o + 3\sqrt{Qp_1^o(1-p_1^o)}] \times [Qp_2^o - 3\sqrt{Qp_2^o(1-p_2^o)}, Qp_2^o + 3\sqrt{Qp_2^o(1-p_2^o)}]$ is a subset of the region of integration in (12). Therefore, the probability in (12) is larger than 0.99 when $Q$ is large enough. This completes the proof of (14). ■

B. Proof of (22)

With a sufficient number of nodes, the probability distributions of $\beta^i(n)$ and $\beta^i(m)$ can be approximated as the normal distributions $\mathcal{N}(Qp_1^i, Qp_1^i(1-p_1^i))$ and $\mathcal{N}(Qp_2^i, Qp_2^i(1-p_2^i))$, respectively, according to the well-known central limit theorem. When $K^i > 0.5K$ and $N \gg K$, we have $p_1^{i+1} > p_1^i > 0.5 > p_2^i > p_2^{i+1}$ according to (18), (19) and (20). This leads to the increase of $Qp_1^i$ and the decrease of $Qp_2^i$, $Qp_1^i(1-p_1^i)$ and $Qp_2^i(1-p_2^i)$ as the iterations proceed. Therefore, the probability that
\[ \lim_{q \to \infty} \Pr(\beta'(n) > \beta'(m)) \] is monotonically increasing with respect to the iteration index \( l \). This completes the proof of (22). ■

REFERENCES

[1] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] E. Candes and M. B. Wakin, “An introduction to compressive sampling,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[3] R. G. Baraniuk, “Compressive sensing,” IEEE Signal Processing Magazine, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[4] D. Wipf and S. Nagarajan, “A unified bayesian framework for MEG/EEG source imaging,” NeuroImage, vol. 44, no. 3, pp. 947–966, Feb. 2009.

[5] M. Lustig, D. L. Donoho; J. M. Santos, and J. M. Pauly, “Compressed Sensing MRI,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 72 – 82. March. 2008.

[6] J. H. G. Ender, “On compressive sensing applied to radar”, Signal Processing, vol. 90, no. 5, pp. 1402 – 1414, May 2010.

[7] G. Li, H. Zhang, X. Wang, and X.-G. Xia, "ISAR 2-D imaging of uniformly rotating targets via matching pursuit," IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 2, pp. 1838-1846, Apr. 2012.

[8] W. Rao, G. Li, X. Wang, and X.-G. Xia, “Adaptive Sparse Recovery by Parametric Weighted L1 Minimization for ISAR Imaging of Uniformly Rotating Targets,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 6, no. 2, pp. 942 – 952, Apr. 2013.
[9] D. Malioutov, M. Cetin, A. Willsky, “A sparse signal reconstruction perspective for source localization with sensor arrays,” IEEE Transactions on Signal Processing, vol. 53, no. 8, pp. 3010–3022, Aug. 2005.

[10] C. Zheng, G. Li, and X. Wang, “Combination of weighted L2,1 minimization with unitary transformation for DOA estimation,” Signal Processing, vol. 93, no. 12, pp. 3430-3434, Dec. 2013.

[11] C. Zheng, G. Li, Y. Liu and X. Wang, "Subspace Weighted L2,1 Minimization for Sparse Signal Recovery," EURASIP Journal on Advances in Signal Processing, Special Issue on Sparse Signal Processing, doi:10.1186/1687-6180-2012-98, 2012.

[12] F. Zeng, C. Li, and Z. Tian, “Distributed compressive spectrum sensing in cooperative multihop cognitive networks,” IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 1, pp. 37–48, Feb. 2011.

[13] Q. Ling and Z. Tian, “Decentralized support detection of multiple measurement vectors with joint sparsity,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 2996–2999.

[14] Z. Li, Y. Zhu, H. Zhu, and M. Li, “Compressive sensing approach to urban traffic sensing,” in Proceedings of the 31st International Conference on Distributed Computing Systems, 2011, pp. 889–898.

[15] X. Yu, H. Zhao, L. Zhang, S. Wu, B. Krishnamachari, and V. O. K. Li, “Cooperative sensing and compression in vehicular sensor networks for urban monitoring,” in Proceedings of the IEEE International Conference on Communications (ICC), 2010, pp. 1–5.

[16] D. W. Rieken and D. R. Fuhrmann, “Generalizing MUSIC and MVDR for multiple noncoherent arrays,” IEEE Transactions on Signal Processing, vol. 52, no. 9, pp. 2396 – 2406, Sept. 2004.
[17] T. E. Derham, S. Doughty, K. Woodbridge, and C. J. Baker, "Design and evaluation of a low-cost multistatic netted radar system," IET - Radar, Sonar & Navigation, vol. 1, no. 5, pp. 362–368, Oct. 2007.

[18] D. Sundman, S. Chatterjee, and M. Skoglund, “Greedy pursuits for compressed sensing of jointly sparse signals,” in Proceedings of the 19th European Signal Processing Conference (EUSIPCO), 2011, pp. 368-372.

[19] D. Sundman, S. Chatterjee, and M. Skoglund, “A greedy pursuit algorithm for distributed compressed sensing,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 2729-2732.

[20] J. A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” IEEE Transactions on Signal Processing, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.

[21] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, “Distributed basis pursuit,” IEEE Transactions on Signal Processing, vol. 60, no. 4, pp. 1942-1956, Apr. 2012.

[22] J. A. Tropp, A. C. Gilbert, Martin J. Strauss, “Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit,” Signal Processing, vol. 86, no. 3, pp. 572-588, Mar. 2006.

[23] D. Baron, M. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk, “Distributed compressed sensing,” preprint, 2009. Online available: http://arxiv.org/abs/0901.3403

[24] J.-M. Feng and C.-H. Lee, “Generalized subspace pursuit for signal recovery from multiple-measurement vectors,” in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2013, pp. 2874-2878.

[25] T. Wimalajeewa and P. K. Varshney, “Cooperative sparsity pattern recovery in distributed networks via distributed-OMP,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 5288-5292. The full version is online available: http://arxiv.org/abs/1307.8320
[26] G. Li, T. Wimalajeewa and P. K. Varshney, “Decentralized Subspace Pursuit for Joint Sparsity Pattern Recovery,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.

[27] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” IEEE Transactions on Information Theory, vol. 55, no. 5, pp. 2230-2249, May 2009.

[28] D. Needell and J.A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” Applied and Computational Harmonic Analysis, no. 26, no. 3, pp. 301-321, May 2009.

[29] B. Dai, S. Ding, and G. Wahba, “Multivariate Bernoulli distribution,” Bernoulli, vol. 19, no. 4, pp.1465–1483, 2013.

[30] N. H. Lehmann, E. Fishler, A. M. Haimovich, R. S. Blum, D. Chizhik, L. J. Cimini Jr., and R. Valenzuela, “Evaluation of Transmit Diversity in MIMO-Radar Direction Finding,” IEEE Transactions on Signal Processing, vol. 55, no. 5, pp. 2215 – 2225, May 2007.

[31] B. Ma, S. Lakshmanan, and A. O. Hero, “Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion,” IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 3, pp. 135 – 147, Sept. 2000.

[32] K. Goebel and A. Agogino, “An architecture for fuzzy sensor validation and fusion for vehicle following in automated highways,” In Proceedings of the 29th International Symposium on Automotive Technology and Automation (ISATA), Florence, Italy, June 4, 1996.