

Article

# Near-Optimal Graph Signal Sampling by Pareto Optimization

Dongqi Luo <sup>1</sup>, Binqiang Si <sup>2</sup>, Saite Zhang <sup>1</sup>, Fan Yu <sup>1</sup> and Jihong Zhu <sup>3,\*</sup>

<sup>1</sup> Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China; ldq17@mails.tsinghua.edu.cn (D.L.); zhang-st17@mails.tsinghua.edu.cn (S.Z.); yuf18@mails.tsinghua.edu.cn (F.Y.)

<sup>2</sup> School of Instrumentation Science and Opto-Electronics Engineering, Beijing Information Science and Technology University, Beijing 100192, China; sbinqiang09@tsinghua.org.cn

<sup>3</sup> Department of Precision Instrument, Tsinghua University, Beijing 100084, China

\* Correspondence: jhzhu@tsinghua.edu.cn

**Abstract:** In this paper, we focus on the bandlimited graph signal sampling problem. To sample graph signals, we need to find small-sized subset of nodes with the minimal optimal reconstruction error. We formulate this problem as a subset selection problem, and propose an efficient Pareto Optimization for Graph Signal Sampling (POGSS) algorithm. Since the evaluation of the objective function is very time-consuming, a novel acceleration algorithm is proposed in this paper as well, which accelerates the evaluation of any solution. Theoretical analysis shows that POGSS finds the desired solution in quadratic time while guaranteeing nearly the best known approximation bound. Empirical studies on both Erdos-Renyi graphs and Gaussian graphs demonstrate that our method outperforms the state-of-the-art greedy algorithms.

**Keywords:** graph signal processing; graph signal sampling; supermodularity ratio; evolutionary algorithms; pareto optimization



**Citation:** Luo, D.; Si, B.; Zhang, S.; Yu, F.; Zhu, J. Near-Optimal Graph Signal Sampling by Pareto Optimization. *Sensors* **2021**, *21*, 1415. <https://doi.org/10.3390/s21041415>

Academic Editor: Vladimir Stankovic  
Received: 25 January 2021  
Accepted: 16 February 2021  
Published: 18 February 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Graph Signal Processing (GSP) is a powerful tool for irregular data modeling and processing, and has attracted extensive attention from the signal processing [1,2], machine learning [3], and computer vision communities [4,5]. Traditional signal processing usually treat the data as a sequence of vectors, then analyze the features in the time domain or frequency domain, and try to extract information from the features. This implies that the data rely on an Euclidean space  $\mathbb{R}^n$ . However, in many real-world applications such as computer graphics [6] and machine learning [7], this assumption is not accurate enough. For example, the data captured by sensors in a sensor network are always interconnected to each other, and they may rely on a manifold embedded in an Euclidean space. Moreover, a RGB image lies within the pixel space  $\mathbb{R}^{H \times W \times 3}$ , but the dimension of the natural image space is usually much lower. GSP brings the signal from the time (spatial) domain to the node domain, and allows us to use a similar tool in traditional signal processing, such as Fourier transform and wavelet analysis, to analyze the graph signals, and then extract the latent information. As an illustration, Figure 1 shows the graph signals supported on the Stanford bunny graph, the Swiss roll graph and the Minnesota road graph respectively.

However, in some applications, the underlying graph is very large, which makes it very expensive to collect and process the data from all nodes. Graph signal sampling is a basic problem in GSP that aims to select a subset with the minimal reconstruction error from the graph node set. Hence, we just need to collect the signal from a small-sized sampling set, and the complete graph signal can be obtained via optimal graph signal reconstruction. Graph signal sampling has been widely used in many fields, e.g., sensor placement [8] and 3D point-cloud sampling [9]. Previous works [10,11] first considered the graph signal reconstruction problem, which aims to recover the whole graph signal from partial observations, then formulated the graph signal sampling problem as a subset

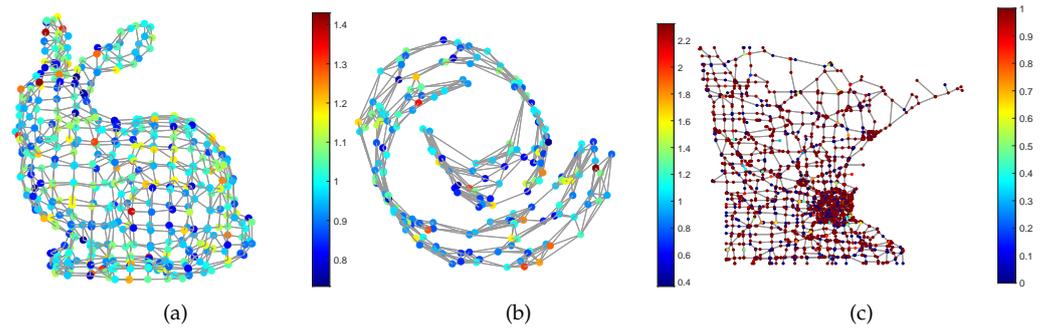
of nodes with the minimal optimal reconstruction error. This sampling problem can be regarded as a subset selection problem, which plays a fundamental role in many fields, such as sparse reciprocal graphical model estimation [12] and sparse plus low rank graphical model estimation [13]. This problem is proved to be generally NP-hard [14], so we may not find a global optimal solution of this problem in polynomial time. A sub-optimal method is the greedy algorithm, which selects an element with the greatest improvement to the objective function at each iteration. The greedy algorithm is very simple, but also very effective. In [15], it is proved that the greedy algorithm achieves  $1 - e^{-\gamma}$  approximation ratio for normalized (that means  $f(\emptyset) = 0, f(\mathcal{V}) = 1$ ) monotonic weak submodular objective functions in general subset selection problems, where  $\gamma$  is the submodularity ratio of the objective function. For submodular objective functions we have  $\gamma \geq 1$ , then the greedy algorithm guarantees  $1 - 1/e$  performance, which is the best-so-far theoretical guarantee.

Recently, many Multi-Objective Evolutionary Algorithms (MOEAs) such as the Pareto Optimization for Subset Selection (POSS) algorithm have exhibited better performance than greedy algorithms do [16–18]. POSS treats the cardinality constraint as an additional objective function of the problem, then the original problem is transformed to a Bi-objective Optimization Problem (BOP). POSS adopts the evolutionary mutation operation to search for different Pareto optimal solutions of the BOP, and finally select a desired Pareto optimal solution.

Though POSS has been proved to be a powerful tool to solve the subset selection problem, it is infeasible to directly apply it to the graph signal sampling problem, since the evaluation of the optimal reconstruction error is very expensive. In this work, we propose the Pareto Optimization for Graph Signal Sampling (POGSS) based on POSS, and an acceleration algorithm is proposed. By introducing the supermodularity ratio of the objective function, we prove that this method theoretically achieves comparable performance as greedy algorithms do in polynomial time, and show that the supermodularity ratio is bounded as well. Our major contributions can be summarized as follows:

1. Inspired by POSS, we formulate the graph signal sampling problem as a subset selection problem, and solve this problem by employing Pareto optimization, which has been proved to be superior to greedy methods.
2. To accelerate the objective estimation procedure, we propose a novel acceleration algorithm based on matrix inversion lemma. Our acceleration algorithm is scalable since it can be used to accelerate the evaluation of *any* solution, thus that can be directly applied to other evolutionary-based methods (e.g., POSS with recombination [17]).
3. We provide comprehensive theoretical analyses of the proposed algorithm, including complexity analysis, Pareto optimality analysis and error bound analysis.

The remainder of this paper is organized as follows. We present the preliminaries and formulate the problem in Section 2.1. We give a brief introduction to the POSS method and introduce the Pareto optimization for graph signal sampling (POGSS) algorithm in Section 3. In Section 4, we provide comprehensive theoretical analyses of POGSS. The Numerical results and the conclusions are presented in Sections 5 and 6, respectively.



**Figure 1.** Illustration of smooth graph signals. The signals are supported on (a) The stanford bunny graph; (b) The swiss roll graph; (c) The Minnesota road graph.

## 2. Preliminaries

### 2.1. Notations

In this paper, we use the bold uppercase letters (e.g.,  $\mathbf{A}$ ) to denote matrices and use bold lowercase letters (e.g.,  $\mathbf{a}$ ) to denote vectors.  $(i, j)$ -th entry of  $\mathbf{A}$  is denoted as  $A_{ij}$  or  $[\mathbf{A}]_{ij}$ .  $(\cdot)^T$  denote the transpose,  $(\cdot)^{-1}$  denote the inverse,  $(\cdot)^\dagger$  denote the Moore-Penrose pseudo-inverse.  $\text{Tr}(\cdot)$  denote the trace.  $\mathbb{E}[\cdot]$  denote the expectation.  $\text{diag}(\mathbf{a})$  stands for a diagonal matrix with the diagonal entries given by a vector  $\mathbf{a}$ . Sets are denoted by calligraphic letters, e.g.,  $\mathcal{A}$  (except the set of real number, which is denoted as  $\mathbb{R}$ ).  $[n]$  denote the set of all integers from 1 to  $n$ , i.e.,  $[n] = \{1, 2, \dots, n\}$ .  $\lceil \cdot \rceil$  denote the ceil operator, e.g.,  $\lceil 0.1 \rceil = 1$ .

### 2.2. Graph Signal Processing

We consider a weighted, undirected graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ , where  $\mathcal{V}$  is the set of nodes with  $|\mathcal{V}| = n$ ,  $\mathcal{E}$  is the set of edges. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be the adjacency matrix of  $\mathcal{G}$ , where  $A_{ij}$  is the weight of the edge between nodes  $i$  and  $j$ . The graph signal  $\mathbf{x}$  is a function that maps each node in  $\mathcal{V}$  to a real number, and can be written as a vector with its  $i$ -th entry denotes the signal value at nodes  $i$ .

Defined the unnormalized graph Laplacian matrix of  $\mathcal{G}$  as

$$\mathbf{L} = \mathbf{D} - \mathbf{A},$$

where  $\mathbf{D}$  is a diagonal degree matrix with  $D_{ii} = \sum_{j=1}^n A_{ij}$ . Then we write the eigenvalue decomposition of the graph Laplacian matrix as

$$\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T,$$

where  $\mathbf{\Lambda} = \{0 = \lambda_1, \lambda_2, \dots, \lambda_n\}$  is a diagonal matrix of graph frequencies, and  $\mathbf{V}$  is the matrix of graph Fourier bases. Then the graph Fourier transform of  $\mathbf{x}$  is defined as  $\bar{\mathbf{x}} = \mathbf{V}^T \mathbf{x}$ , and the inverse Fourier transform is given by  $\mathbf{x} = \mathbf{V} \bar{\mathbf{x}}$ . We say a graph signal  $\mathbf{x}$  is  $\mathcal{K}$ -bandlimited, if its graph Fourier coefficients indexed by  $\mathcal{V} \setminus \mathcal{K}$  are zero, i.e.,  $\bar{\mathbf{x}}_{\mathcal{V} \setminus \mathcal{K}} = \mathbf{0}$ . In [19], it was proved that if the sampling set size is not smaller than  $|\mathcal{K}|$ , then the  $\mathcal{K}$ -bandlimited graph signal can be perfectly recovered. However, in many applications we can only observe noise corrupted signals, we cannot find a perfect estimator under noise. Consequently, Bayesian estimation is adopted in the graph signal reconstruction problem.

### 2.3. Problem Formulation

Let  $\mathbf{x}$  be the original noise-free graph signal, and  $\mathbf{n}$  be the noise vector. In the *graph signal reconstruction* problem, only the noise corrupted samples  $\mathbf{y}_S$  in a sampling set  $\mathcal{S} \subseteq \mathcal{V}$  are observable, thus we take the observation model as

$$\begin{cases} \mathbf{y} = \mathbf{x} + \mathbf{n}, \\ \mathbf{y}_S = \mathbf{C}_S \mathbf{y}, \end{cases} \quad (1)$$

where  $\mathbf{C}_S \in \{0, 1\}^{|\mathcal{S}| \times n}$  is the sampling matrix. We assume that  $\mathbf{n}$  is a zero-mean vector with covariance matrix  $\sigma^2 \mathbf{I}$ ,  $\mathbf{x}$  is  $\mathcal{K}$ -bandlimited and  $\bar{\mathbf{x}}_{\mathcal{K}}$  is a zero-mean vector with covariance matrix  $\bar{\Sigma}$ , hence the covariance matrix of  $\mathbf{x}$  is  $\Sigma = \mathbf{V}_{\mathcal{K}} \bar{\Sigma} \mathbf{V}_{\mathcal{K}}^T$ . Graph signal reconstruction is to approximately recover the original graph signal  $\mathbf{x}$  from partial noisy observation  $\mathbf{y}_S$  via a linear transform  $\Phi_S \in \mathbb{R}^{n \times |\mathcal{S}|}$ , i.e.,

$$\hat{\mathbf{x}}(\Phi_S) = \Phi_S \mathbf{y}_S. \quad (2)$$

The error covariance matrix with respect to  $\Phi_S$  is defined as

$$\mathbf{K}_S(\Phi_S) = \mathbb{E}[(\mathbf{x} - \hat{\mathbf{x}}(\Phi_S))(\mathbf{x} - \hat{\mathbf{x}}(\Phi_S))^T].$$

To find the optimal reconstruction matrix  $\Phi_S^*$ , one needs to minimize the scalar loss function  $J(\Phi_S) = \mathbf{z}^T \Phi_S \mathbf{z}$  for all  $\mathbf{z} \in \mathbb{R}^n$ . The derived optimal recover operator is [10]

$$\Phi_S^* = \Sigma \mathbf{C}^T (\mathbf{C}(\Sigma + \sigma^2 \mathbf{I}) \mathbf{C}^T)^{\dagger}, \quad (3)$$

and the corresponding error covariance matrix is given by

$$\mathbf{K}_S^* = \mathbf{K}_S(\Phi_S^*) = \mathbf{V}_{\mathcal{K}} \left( \bar{\Sigma}^{-1} + \sigma^{-2} \sum_{i \in \mathcal{S}} \mathbf{v}_i \mathbf{v}_i^T \right)^{-1} \mathbf{V}_{\mathcal{K}}^T, \quad (4)$$

where  $\mathbf{V}_{\mathcal{K}}^T = [\mathbf{v}_1, \dots, \mathbf{v}_n]$ , and the optimal mean squared error (MSE) is given by

$$\text{MSE}_S^* = \mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}(\Phi_S^*)\|_2^2] = \text{Tr}(\mathbf{K}_S^*). \quad (5)$$

The goal of *graph signal sampling* is to select a subset  $\mathcal{S} \subseteq \mathcal{V}$  with at most  $k$  nodes, which minimizes the optimal reconstruction error  $\text{MSE}_S^*$ , i.e., finding

$$\begin{aligned} \mathcal{S}^* &= \arg \min_{\mathcal{S} \subseteq \mathcal{V}, |\mathcal{S}| \leq k} \text{MSE}_S^* \\ &= \arg \min_{\mathcal{S} \subseteq \mathcal{V}, |\mathcal{S}| \leq k} \text{Tr}(\mathbf{K}_S^*). \end{aligned} \quad (6)$$

Define  $\bar{\mathbf{K}}_S = (\bar{\Sigma}^{-1} + \sigma^{-2} \sum_{i \in \mathcal{S}} \mathbf{v}_i \mathbf{v}_i^T)^{-1}$ ,  $\mathbf{M} = \mathbf{V}_{\mathcal{K}}^T \mathbf{V}_{\mathcal{K}}$ . By using the circular commutation property of trace, (6) is simplified to

$$\mathcal{S}^* = \arg \min_{\mathcal{S} \subseteq \mathcal{V}, |\mathcal{S}| \leq k} \text{Tr}(\mathbf{M} \bar{\mathbf{K}}_S) \quad (7)$$

### 3. Method

In this study, we propose to solve (7) by Pareto optimization. For representable convenience, we use the binary vector  $\mathbf{s} \in \{0, 1\}^n$  to represent the subset  $\mathcal{S} \subseteq \mathcal{V}$ , the entry  $s_i = 1$  means that the node  $v_i$  is included in  $\mathcal{S}$  and  $s_i = 0$  otherwise. We do not distinguish between the two notations. Please note that the original problem is a constrained problem with single objective function, we now reformulate this problem as an unconstrained BOP:

$$\arg \min_{\mathbf{s} \in \{0, 1\}^n} \mathbf{f}(\mathbf{s}) = (f(\mathbf{s}), c(\mathbf{s}))^T, \quad (8)$$

where

$$f(\mathbf{s}) = \text{MSE}_S^* = \text{Tr}(\mathbf{M}\bar{\mathbf{K}}_S),$$

$$c(\mathbf{s}) = \begin{cases} \|\mathbf{s}\|_0, & \|\mathbf{s}\|_0 < 2k \\ +\infty, & \|\mathbf{s}\|_0 \geq 2k \text{ or } \|\mathbf{s}\|_0 = 0 \end{cases}$$

To compare two solutions, the Pareto dominance relationship and the Pareto optimality are defined as follows:

**Definition 1** (Pareto dominance). Let  $\mathbf{s}_1$  and  $\mathbf{s}_2$  be two solutions, then

- (Weak dominance)  $\mathbf{s}_1$  weakly dominates  $\mathbf{s}_2$  ( $\mathbf{s}_1 \preceq \mathbf{s}_2$ ), if and only if  $f(\mathbf{s}_1) \leq f(\mathbf{s}_2)$  and  $c(\mathbf{s}_1) \leq c(\mathbf{s}_2)$ ;
- (Dominance)  $\mathbf{s}_1$  dominates  $\mathbf{s}_2$  ( $\mathbf{s}_1 \prec \mathbf{s}_2$ ), if and only if  $\mathbf{s}_1 \preceq \mathbf{s}_2$  and  $(f(\mathbf{s}_1) < f(\mathbf{s}_2) \text{ or } c(\mathbf{s}_1) < c(\mathbf{s}_2))$ .

**Definition 2** (Pareto optimality). We say  $\mathbf{s}^*$  is a Pareto optimal solution with respect to  $\mathcal{M}$ , if there is no solution  $\mathbf{s} \in \mathcal{M}$  such that  $\mathbf{s} \prec \mathbf{s}^*$ .

To summarize, a solution  $\mathbf{s}_1$  is said to weakly dominates  $\mathbf{s}_2$  if  $\mathbf{s}_1$  performs not worse than  $\mathbf{s}_2$  for all metrics, and for two solutions such that  $\mathbf{s}_1 \preceq \mathbf{s}_2$ , we say  $\mathbf{s}_1$  dominates  $\mathbf{s}_2$  if there is at least one metric for which  $\mathbf{s}_1$  strictly outperforms  $\mathbf{s}_2$ .  $\mathbf{s}^*$  is said to be a Pareto optimal solution with respect to  $\mathcal{M}$  if there is no solution in  $\mathcal{M}$  that dominates  $\mathbf{s}^*$ .

### 3.1. Pareto Optimization for Subset Selection (Poss)

POSS is an evolutionary algorithm for subset selection, and its overall procedure is shown in Algorithm 1. Denote  $\mathcal{T} = 2^{\mathcal{V}}$  as the set of all solutions,  $\mathcal{M} \subseteq \mathcal{T}$  as the set of all solutions found so far. POSS maintains a set  $\mathcal{P} \subseteq \mathcal{M}$  of all Pareto optimal solutions with respect to  $\mathcal{M}$ . At each step, POSS first randomly selects a solution from the set of Pareto optimal solutions  $\mathcal{P}$ , and then adopts the mutation operator to search for a potential Pareto optimal solution. If there is no solution that dominates the generated solution  $\mathbf{s}'$ , then POSS includes the generated solution to  $\mathcal{P}$ , and remove the solutions which are weakly dominated by  $\mathbf{s}'$  from  $\mathcal{P}$ ; Otherwise POSS discards the solution  $\mathbf{s}$ . Such update strategy in POSS guarantees that if  $\mathcal{P}$  is the Pareto optimal solution set at current iteration, then the updating operation does not change the optimality of  $\mathcal{P}$ . After  $T$  iterations, POSS outputs a solution  $\mathbf{s}^* \in \mathcal{P}$  with the minimal reconstructed error and a cardinality smaller than  $k$ , i.e.,  $\mathbf{s}^* = \arg \min_{\mathbf{s} \in \mathcal{P}, \|\mathbf{s}\|_0 \leq k} f(\mathbf{s})$ .

---

#### Algorithm 1 POSS

---

**Input:** An universal set  $\mathcal{V}$ ,  $f: 2^{\mathcal{V}} \rightarrow \mathbb{R}$ ,  $k \in [n]$ ,  $T \in \mathbb{N}_+$ .

**Output:** a subset  $\mathcal{S} \subseteq \mathcal{V}$  with at most  $k$  nodes.

- 1:  $\mathbf{s} = \{0\}^n$ ,  $\mathcal{P} = \{\mathbf{s}\}$ .
  - 2: **for**  $t = 1, 2, \dots, T$  **do**
  - 3:   Select  $\mathbf{s}$  from  $\mathcal{P}$  uniformly at random.
  - 4:   Generate  $\mathbf{s}'$  by bit-wise flipping with probability  $\frac{1}{n}$ .
  - 5:   **if**  $\nexists \mathbf{z} \in \mathcal{P}$  such that  $\mathbf{z} \prec \mathbf{s}'$  **then**
  - 6:      $\mathcal{P} = (\mathcal{P} \setminus \{\mathbf{z} \in \mathcal{P} | \mathbf{s}' \preceq \mathbf{z}\}) \cup \{\mathbf{s}'\}$ .
  - 7:   **end if**
  - 8: **end for**
  - 9:  $\mathbf{s}^* = \arg \min_{\mathbf{s} \in \mathcal{P}, \|\mathbf{s}\|_0 \leq k} f(\mathbf{s})$ .
- 

### 3.2. Proposed Pareto Optimiazation for Graph Signal Sampling (Pogss)

However, due to the general  $\mathcal{O}(n^3)$  complexity of the matrix inversion operation, the evaluation step of  $f(\mathbf{s})$  is very expensive in general, which means we cannot efficiently compare two solutions. Please note that at each iteration the new solution is generated from a solution evaluated before, we can accelerate the evaluation of the objective function

by leveraging the information of the previous solution. More explicitly, by using the matrix inversion lemma [20] we deduced the following lemma:

**Lemma 1.** Let  $S \subseteq \mathcal{V}$ , then for all  $v_1 \in \mathcal{V} \setminus S, v_2 \in S$ , we have

$$\bar{\mathbf{K}}_{S \cup \{v_1\}} = \bar{\mathbf{K}}_S - \frac{\bar{\mathbf{K}}_S \mathbf{v}_{v_1} \mathbf{v}_{v_1}^T \bar{\mathbf{K}}_S}{\sigma^2 + \mathbf{v}_{v_1}^T \bar{\mathbf{K}}_S \mathbf{v}_{v_1}} \quad (9)$$

$$\bar{\mathbf{K}}_{S \setminus \{v_2\}} = \bar{\mathbf{K}}_S + \frac{\bar{\mathbf{K}}_S \mathbf{v}_{v_2} \mathbf{v}_{v_2}^T \bar{\mathbf{K}}_S}{\sigma^2 - \mathbf{v}_{v_2}^T \bar{\mathbf{K}}_S \mathbf{v}_{v_2}} \quad (10)$$

Let  $\mathbf{t}_I = \mathbf{s}' \setminus \mathbf{s} = \{t_{I_1}, \dots, t_{I_l}\}$  be the elements to be included at the current iteration,  $\mathbf{t}_E = \mathbf{s} \setminus \mathbf{s}' = \{t_{E_1}, \dots, t_{E_m}\}$  denote the elements to be excluded, and  $\mathbf{t}_R = \mathbf{s} \cap \mathbf{s}'$  denote the elements that will be retained. Instead of directly computing the  $\bar{\mathbf{K}}_{\mathbf{s}'}$ , we propose to compute  $\bar{\mathbf{K}}_{\mathbf{s} \cup \mathbf{t}_I}$  firstly, and then compute the desired  $\bar{\mathbf{K}}_{\mathbf{s}'} = \bar{\mathbf{K}}_{(\mathbf{s} \cup \mathbf{t}_I) \setminus \mathbf{t}_E}$ . By using Lemma 1, we can easily compute  $\bar{\mathbf{K}}_{\mathbf{s} \cup \mathbf{t}_I}$  by including the elements in  $\mathbf{t}_I$  to  $\mathbf{s}$  one by one, and then compute  $\bar{\mathbf{K}}_{\mathbf{s}'}$  by excluding the elements in  $\mathbf{t}_E$  from  $\mathbf{s} \cup \mathbf{t}_I$  separately. The full procedure of POGSS is described in Algorithm 2, and the solution flipping algorithm for acceleration is presented in Algorithm 3.

Denote  $N_t = |\mathbf{t}_I| + |\mathbf{t}_E|$  as the number of bits that will be flipped at iteration  $t$ , then we have  $\mathbb{E}(N_t) = 1$ , which reveals the fact that we are expected to flip only 1 bit at each iteration. Please note that the evaluation of (9) and (10) can be done at a cost of  $\mathcal{O}(n^2)$ , so the cost of evaluating the objective function in Algorithm 2 is expected to be  $\mathcal{O}(n^2)$ , while the naive matrix inversion update requires a cost of  $\mathcal{O}(n^3)$ . Therefore, this proposed procedure significantly accelerates the algorithm.

---

#### Algorithm 2 POGSS

---

**Input:**  $\Sigma, \mathbf{V}$ , integer  $k \in [n], T \in \mathbb{N}_+$ .

**Output:** a subset  $S^* \subseteq \mathcal{V}$  with at most  $k$  nodes.

- 1:  $\mathbf{s} = \{0\}^n, \bar{\mathbf{K}}_{\mathbf{s}} = \Sigma, \mathcal{P} = \{\mathbf{s}\}$ .
  - 2: **for**  $t = 1, 2, \dots, T$  **do**
  - 3:   Select  $\mathbf{s}$  from  $\mathcal{P}$  uniformly at random.
  - 4:   Generate  $\mathbf{s}'$  by bit-wise flipping with probability  $\frac{1}{n}$ .
  - 5:   Compute  $\bar{\mathbf{K}}_{\mathbf{s}'}$  by invoking Algorithm 3, then evaluate the objective function  $f(\mathbf{s}') = \text{Tr}(\mathbf{M}\bar{\mathbf{K}}_{\mathbf{s}'})$ .
  - 6:   **if**  $\nexists \mathbf{z} \in \mathcal{P}$  such that  $\mathbf{z} \prec \mathbf{s}'$  **then**
  - 7:      $\mathcal{P} = (\mathcal{P} \setminus \{\mathbf{z} \in \mathcal{P} | \mathbf{s}' \preceq \mathbf{z}\}) \cup \{\mathbf{s}'\}$ .
  - 8:   **end if**
  - 9: **end for**
  - 10: Output  $\mathbf{s}^* = \arg \min_{\mathbf{s} \in \mathcal{P}, \|\mathbf{s}\|_0 \leq k} f(\mathbf{s})$ .
- 

---

#### Algorithm 3 Solution Flipping

---

**Input:**  $\mathbf{s}, \bar{\mathbf{K}}_{\mathbf{s}}, \mathbf{s}'$ .

**Output:**  $\bar{\mathbf{K}}_{\mathbf{s}'}$ .

- 1: Initialize  $\mathbf{s}_0 = \mathbf{s}, \bar{\mathbf{K}}_{\mathbf{s}_0} = \bar{\mathbf{K}}_{\mathbf{s}}$ .
  - 2: Compute the difference set  $\mathbf{t}_I = \mathbf{s}' \setminus \mathbf{s} = \{t_{I_1}, \dots, t_{I_l}\}$  and  $\mathbf{t}_E = \mathbf{s} \setminus \mathbf{s}' = \{t_{E_1}, \dots, t_{E_m}\}$ .
  - 3: **for**  $i = 1, \dots, l$  **do**
  - 4:    $\mathbf{s}_i = \mathbf{s}_{i-1} \cup \{t_{I_i}\}$ .
  - 5:   Compute  $\bar{\mathbf{K}}_{\mathbf{s}_i} = \bar{\mathbf{K}}_{\mathbf{s}_{i-1} \cup \{t_{I_i}\}}$  by using (9).
  - 6: **end for**
  - 7: **for**  $i = 1, \dots, m$  **do**
  - 8:    $\mathbf{s}_{l+i} = \mathbf{s}_{l+i-1} \setminus \{t_{E_i}\}$ .
  - 9:   Compute  $\bar{\mathbf{K}}_{\mathbf{s}_{l+i}} = \bar{\mathbf{K}}_{\mathbf{s}_{l+i-1} \setminus \{t_{E_i}\}}$  by using (10).
  - 10: **end for**
  - 11: Output  $\bar{\mathbf{K}}_{\mathbf{s}'} = \bar{\mathbf{K}}_{\mathbf{s}_{l+m}}$ .
- 

#### 4. Theoretical Analysis

In this section, we theoretically analyze the performance of POGSS, including the Pareto optimality of  $\mathcal{P}$  and the overall reconstruction error of POGSS.

We first show in Proposition 1 that the subset  $\mathcal{P}$  is the set of all Pareto optimal solutions with respect to  $\mathcal{M}$ .

**Proposition 1** (Pareto optimality of  $\mathcal{P}$ ). *At any iteration of the algorithm 2, let  $\mathcal{M}$  be the set of all solutions found so far, then*

- for all  $\mathbf{s} \in \mathcal{P}$ , there is no  $\mathbf{s}' \in \mathcal{M}$  such that  $\mathbf{s}' \prec \mathbf{s}$ ,
- for all  $\mathbf{s} \in (\mathcal{M} \setminus \mathcal{P})$ , there is a solution  $\mathbf{s}' \in \mathcal{M}$  such that  $\mathbf{s}' \prec \mathbf{s}$ .

We then analyze the performance of POGSS by introducing the supermodularity ratio. A set function  $f : 2^{\mathcal{V}} \rightarrow \mathbb{R}_+$  is said to be a *supermodular* function, if  $f(\mathcal{A} \cup \{u\}) - f(\mathcal{A}) \leq f(\mathcal{B} \cup \{u\}) - f(\mathcal{B})$  holds for any  $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$  and  $v \in \mathcal{V} \setminus \mathcal{B}$ . For a general criterion  $f$ , Definition 3 defines how close it is to a supermodular function, i.e., the supermodularity ratio  $\gamma_{S,k}(f)$  [21]. It is easy to prove that  $f$  is supermodular if and only if  $\gamma_{S,k}(f) \geq 1$  holds for any  $S$  and  $k$ .

**Definition 3** (Supermodularity ratio). *Let  $f$  be a non-negative set function. The submodularity ratio of  $f$  with respect to a set  $S$  and a parameter  $k \geq 1$  is*

$$\gamma_{S,k}(f) = \min_{\mathcal{L} \subseteq S, \mathcal{U}: |\mathcal{U}| \leq k, \mathcal{L} \cap \mathcal{U} = \emptyset} \frac{f(\mathcal{L} \cup \mathcal{U}) - f(\mathcal{L})}{\sum_{u \in \mathcal{U}} (f(\mathcal{L} \cup \{u\}) - f(\mathcal{L}))} \quad (11)$$

The most time-consuming operation in POGSS is the evaluation of the objective function after bit flipping. So here comes a question: how many bit flipping operations we need to obtain a desired solution? We provide an approximation upper bound of the optimal reconstruction MSE with  $\mathcal{O}(k^2n)$  bits flipped in Theorem 1. Please note that the greedy algorithm guarantees  $(1 - e^{-\gamma_{S^G,k}}) \cdot OPT + e^{-\gamma_{S^G,k}} f(\mathbf{0})$  performance, where  $\mathbf{s}^G$  is the solution given by the greedy algorithm, POGSS achieves nearly the best known approximation guarantee as greedy algorithms do in quadratic time.

**Theorem 1.** *Define  $\gamma_{\min} = \min_{\mathbf{s}: \|\mathbf{s}\|_0 = k-1} \gamma_{\mathbf{s},k}$ , let  $N$  be the total number of flipped bits,  $OPT$  be the optimal value of (7). Then POGSS finds a solution  $\mathbf{s}$  such that*

$$f(\mathbf{s}) \leq (1 - e^{-\gamma_{\min}}) \cdot OPT + e^{-\gamma_{\min}} f(\mathbf{0}),$$

with

$$\mathbb{E}(N) \leq 2k^2 \exp\left(-\frac{n}{n-0.5}\right) n \leq 2ek^2n.$$

The proof of Theorem 1 relies on Lemma 2 and 3 stated bellow. The complete proof is similar to the proof of theorem 14.1 in [22], which is omitted due to the limited space.

**Lemma 2.** *For any  $n \geq 2$ ,*

$$\left(1 - \frac{1}{n}\right)^{n-1} \geq e^{-(n-0.5)/n} \geq e^{-1}$$

holds.

**Lemma 3** ([16]).  $\forall \mathbf{s} \in \{0, 1\}^n$ , there exists one nodes  $v \in \mathbf{s}$  such that

$$f(\mathbf{s}) - f(\mathbf{s} \cup \{v\}) \geq \frac{\gamma_{\mathbf{s},b}}{b} (f(\mathbf{s}) - OPT)$$

Please note that the approximation upper bound in Theorem 1 involves  $\gamma_{\min}$ , we show in Theorem 2 that  $\gamma_{\min}$  is also bounded.

**Proposition 2** (Lower bound for  $\gamma_{\min}$ ). *The supermodularity ratio satisfies*

$$\gamma_{\min} \geq \frac{1}{k}.$$

This conclusion is natural and obvious, since the equality holds when  $f(\mathcal{L}) - f(\mathcal{L} \cup \mathcal{U}) = \max_{u \in \mathcal{U}} f(\mathcal{L}) - f(\mathcal{L} \cup \{u\})$ , and is also relatively loose. However, it is still meaningful because it is independent of the graph structure and the Signal-to-Noise Ratio (SNR), so it is tighter than the bound in [10] in high SNR scenarios.

## 5. Experiments

In this section, we conduct several experiments to evaluate the performance of POGSS. We first state the experimental settings, and then report the experimental results.

### 5.1. Experimental Settings

Two types of undirected graphs are adopted for evaluation:

- The Erdos-Renyi (ER) graph: For ER graphs, any two nodes are connected with probability  $p$ , i.e.,  $\mathbb{P}(\mathbf{W}_{i,j} = \mathbf{W}_{j,i} = 1) = p$ ,  $\mathbb{P}(\mathbf{W}_{i,j} = \mathbf{W}_{j,i} = 0) = 1 - p$ .
- The Gaussian graph: For Gaussian graphs, we uniformly sample  $n$  points  $\{\mathbf{p}_i\}_{i=1}^n$  in a unit square  $[0, 1]^2$ . The edges are first connected with weights

$$\mathbf{W}_{i,j} = \mathbf{W}_{j,i} = \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}{2s^2}\right),$$

and we then set the weights which are smaller to some threshold  $W_T$  to 0.

In our experiments, the number of nodes  $|\mathcal{V}| = n$  is set to 100 for all graphs, and we set  $p = 0.2$  for ER graphs, set  $s = 0.3$ ,  $W_T = 0.6$  for Gaussian graphs.

The graph signal  $\mathbf{x}$  is sampled from a Gaussian distribution:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_{\mathcal{K}} \bar{\Sigma} \mathbf{V}_{\mathcal{K}}^T) \quad (12)$$

where  $\mathbf{V}_{\mathcal{K}}$  is the matrix of graph Fourier bases,  $\bar{\Sigma}$  is the covariance matrix of  $\bar{\mathbf{x}}_{\mathcal{K}}$ . In our experiments,  $\bar{\Sigma}$  is generated by  $\bar{\Sigma} = \mathbf{P}\mathbf{P}^T$ , where  $\mathbf{P} \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{K}|}$  is a random matrix with  $\mathbf{P}_{i,j} \sim \mathcal{N}(0, 1)$ . The additive Gaussian noise  $\mathbf{n}$  is adopted in our experiments, and the noise power  $\sigma^2$  is set to 0.01.

We compare our methods with several algorithms, including the state-of-the-art greedy algorithm, the randomized greedy algorithm, and the convex relaxation algorithm. In all experiments, the number of iterations of POGSS is set to  $\lceil 2ek^2n \rceil$  according to Theorem 1. The naive greedy graph signal sampling algorithm [10] simply selects a node with the largest marginal gain at each iteration. Assume subset of nodes obtained by the greedy algorithm at the current iteration is  $\mathcal{S}$ , then the node  $v = \arg \max_{v \in \mathcal{V} \setminus \mathcal{S}} f(\mathcal{S} \cup \{v\})$  is included to  $\mathcal{S}$ . The randomized greedy algorithm [11] uses the similar strategy to the greedy algorithm, but accelerates the algorithm by selecting the best node from a random subset. More explicitly, it first samples a subset  $\mathcal{R} \subset \mathcal{V} \setminus \mathcal{S}$  with size  $-n/k \log(\epsilon)$  uniformly, where  $0 < \epsilon < 1$  is a hyper-parameter, then includes the node  $v = \arg \max_{v \in \mathcal{R}} f(\mathcal{S} \cup \{v\})$  to  $\mathcal{S}$ . Recall the original problem of graph signal sampling (7)

$$\mathbf{s}^* = \arg \min_{\mathbf{s} \in \{0,1\}^n} f(\mathbf{s}) = \text{Tr} \left( \mathbf{M} \left( \bar{\Sigma}^{-1} + \sigma^{-2} \sum_{i=0}^n s_i \mathbf{v}_i \mathbf{v}_i^T \right) \right), \text{ s.t. } \|\mathbf{s}\|_1 \leq k, \quad (13)$$

since the original problem is neither differentiable nor convex, we relax this problem to a convex optimization problem [23]

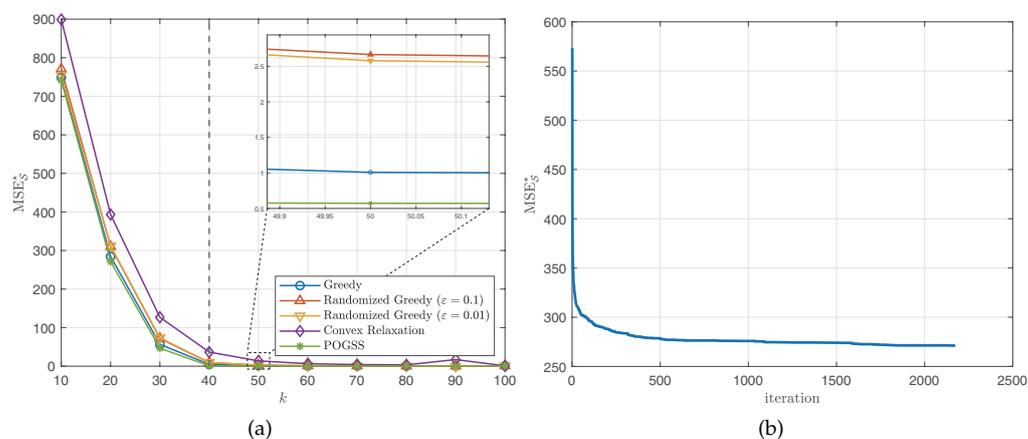
$$\mathbf{d}^* = \arg \min_{\mathbf{d} \in [0,1]^n} f(\mathbf{d}), \text{ s.t. } \|\mathbf{d}\|_1 \leq k, \quad (14)$$

then let the largest  $k$  entries in  $\mathbf{d}^*$  to be included to the sampling set.

### 5.2. Experimental Results

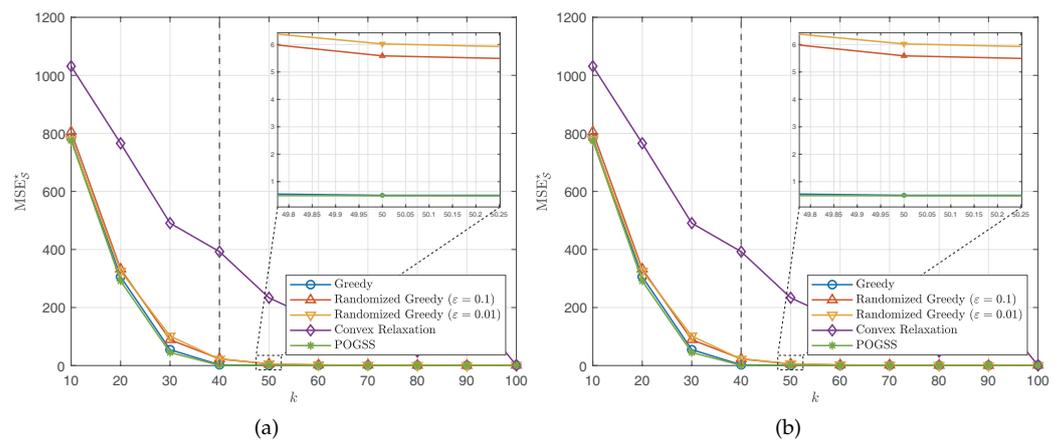
We first evaluate the algorithms under the  $\mathcal{K}$ -bandlimited signal assumption. Under this assumption, the entries in  $\bar{\mathbf{x}}$  indexed by  $\mathcal{K}$  are zero, thus  $\mathbf{x}$  can be perfectly represented by only  $|\mathcal{K}|$  graph Fourier bases. In the following experiments we set  $|\mathcal{K}| = 40$  and use the first 40 graph Fourier bases to form  $\mathbf{V}_{\mathcal{K}}$ .

We set the subset size  $k$  from 10 to 100 with stepsize 10, then independently run the simulation 10 times and report the average optimal reconstruction MSE of the obtained subset of each algorithm. Figures 2 and 3 illustrate the results obtained on ER graphs and Gaussian graphs respectively. From Figures 2a and 3a we observe that POGSS and greedy-based algorithms achieve near-optimal performance, the performance convex relaxation method is worse than other method, and POGSS outperforms others for any subset size. We note that when the subset size  $k$  is approaching  $|\mathcal{K}| = 40$  the MSE decreases dramatically, and when  $k > |\mathcal{K}| = 40$  (shown in the right-hand side of the vertical dotted line) all of the methods provide the solutions with very small reconstruction MSE. Figures 2b and 3b plot the convergence curve of POGSS when  $k = 20$ . We can see that the MSE decreases very fast at first, and then becomes flat when  $k > 500$ . However, it keeps decreasing throughout the whole process. To be precise, POGSS reaches its' final solution at the 2144-th iteration, while  $T = \lceil 2ek^2n \rceil = 2174$ .

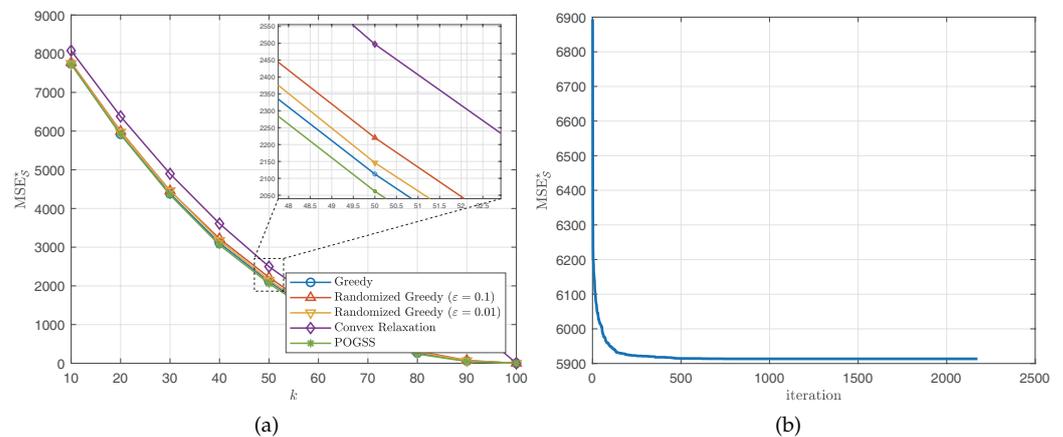


**Figure 2.** Results for bandlimited graph signals obtained on ER graphs. (a) MSE<sub>S</sub><sup>\*</sup> v.s.  $k$ . (b) Convergence curve when  $k = 20$ .

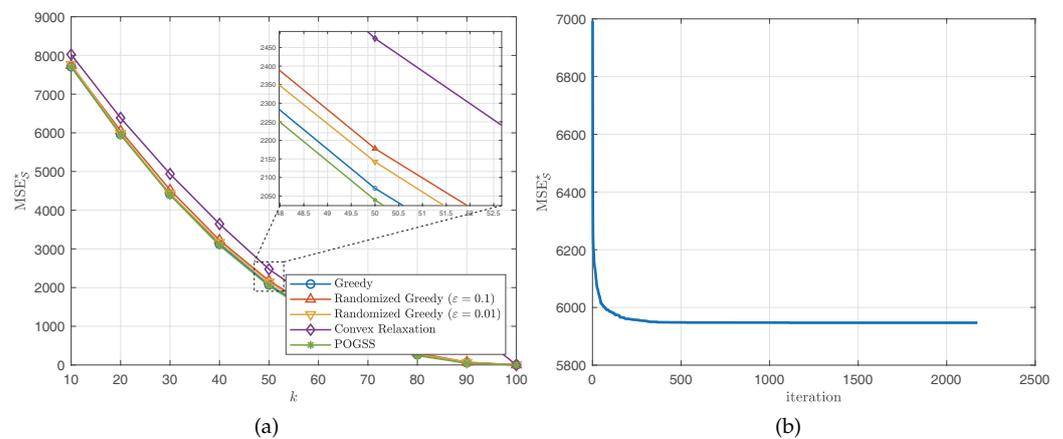
We next evaluate the algorithms for non-bandlimited graph signals, i.e.,  $\mathbf{V}_{\mathcal{K}} = \mathbf{V}$ . Similar to the bandlimited scenario, we set the subset size from 10 to 100 with step size 10 and present the average optimal reconstruction MSE of the obtained subset of each algorithm. Figures 4 and 5 show the results obtained on ER graphs and Gaussian graphs respectively. From Figures 4a and 5a we observe the similar results to the ones in Figures 2a and 3a, where POGSS still achieves the best performance. It is worth noting that there is no “turning point” in Figures 4 and 5, because the signals are non-bandlimited, and cannot be accurately reconstructed from partial observations.



**Figure 3.** Results for bandlimited graph signals obtained on Gaussian graphs. (a)  $MSE_S^*$  v.s.  $k$ . (b) Convergence curve when  $k = 20$ .



**Figure 4.** Results for non-bandlimited graph signals obtained on ER graphs. (a)  $MSE_S^*$  v.s.  $k$ . (b) Convergence curve when  $k = 20$ .



**Figure 5.** Results for non-bandlimited graph signals obtained on Gaussian graphs. (a)  $MSE_S^*$  v.s.  $k$ . (b) Convergence curve when  $k = 20$ .

In Figure 6 we compare the running time of different sampling set sizes. We observe that POGSS takes the most time to converge, which is as expected since Theorem 1 reveals that POGSS take  $2ek^2n$  iterations to converge, while greedy algorithms require only  $k$

iterations to do that (the time consumption of each method to finish one iteration is almost the same). Although POGSS takes the most time to find a desired solution, it is still practicable for two reasons:

1. if we fit the running time of POGSS with a quadratic function, then we have  $T \approx 0.26 \cdot k^2$  (sec), thus POGSS only takes only  $\Delta T \approx 5.15 \cdot k + 25.77$  (sec) when we add 10 nodes to the current sampling set, we think it is worthwhile to find a better solution with such a time consumption.
2. In real world applications of graph signal sampling, e.g., [8], the sampling procedure is offline and the reconstruction procedure is online. We only need to sample the graph once then the entire graph signal can be always reconstructed with a small error, while the reconstruction time depends only on the size of the sampling set, rather than the sampling method.

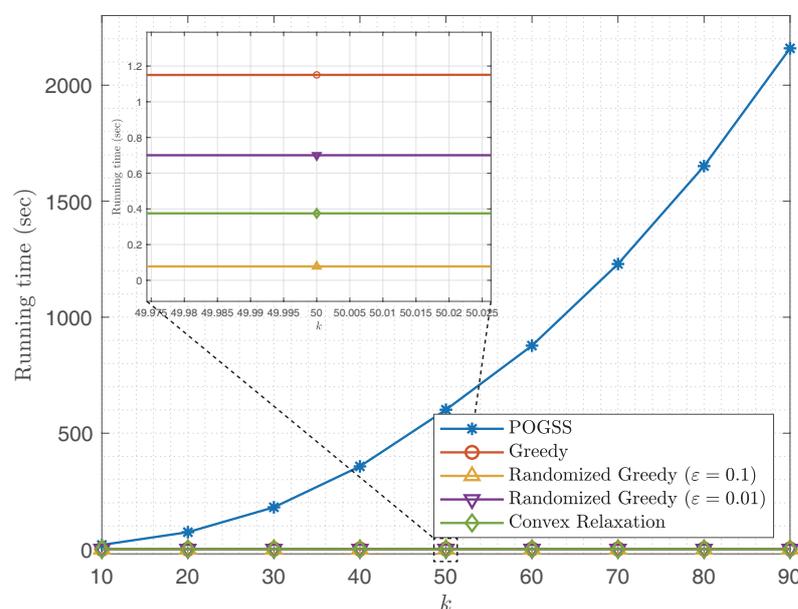
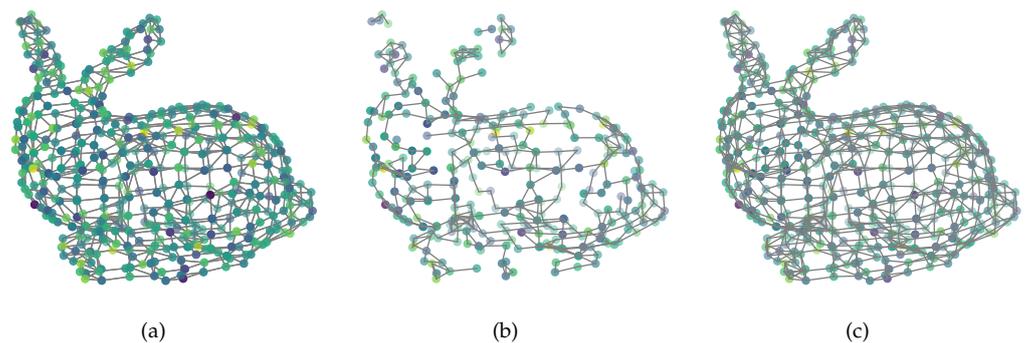


Figure 6. Running time v.s.  $k$  for non-bandlimited graph signals obtained on ER graphs.

We also provide a visual demonstration for our method. We use the stanford bunny data [6] to obtain the graph. The nodes are scanned from a 3D model, and the edges are obtained by connecting each node with its' four nearest neighbors. The smooth graph signal is first sampled from a Gaussian distribution, and then filtered by a heat diffusion low-pass filter [24]. The resulting graph Fourier coefficients after filtering are

$$\hat{\lambda}_i = h(\lambda_i) = \exp(-t\lambda_i),$$

where  $t$  is the scaling parameter and is set to 10 in our experiment. As displayed in Figure 7a is the original smooth graph signal with Gaussian noise, the noise power is  $\sigma^2 = 0.01$ . There are  $n = 453$  nodes in the original graph, and we sample a subset with  $k = 300$  nodes by using POGSS to obtain Figure 7b, note that all of the edges are from the original graph. Figure 7c illustrates the optimally recovered graph signal by using (3), and the corresponding reconstruction error is  $\text{MSE}_S^* = 137.2$ .



**Figure 7.** (a) Original noisy signal; (b) sampled noisy signal; (c) reconstructed signal.

## 6. Conclusions

In this paper, we consider the graph signal sampling problem, which can be formulated as a subset selection problem. The evolutionary Pareto optimization for graph signal sampling (POGSS) is proposed for solving this problem. An efficient procedure is also proposed to accelerate the evaluation of the objective function. Theoretical analysis shows that POGSS, in polynomial time, achieves the best approximation upper bound of  $(1 - e^{-\gamma_{\min}}) \cdot OPT + e^{-\gamma_{\min}} f(\mathbf{0})$  as the state-of-the-art greedy algorithm does, and a lower bound of  $\gamma_{\min}$  is built as well. Simulation results demonstrate that POGSS outperforms the previous greedy algorithms.

**Author Contributions:** Conceptualization, D.L. and B.S.; methodology, D.L.; software, D.L.; validation, S.Z., F.Y. and J.Z.; formal analysis, D.L.; investigation, D.L.; resources, D.L.; writing—original draft preparation, D.L.; writing—review and editing, D.L. and S.Z.; visualization, D.L.; supervision, J.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Ortega, A.; Frossard, P.; Kovačević, J.; Moura, J.M.F.; Vandergheynst, P. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE* **2018**, *106*, 808–828. [\[CrossRef\]](#)
- Huang, W.; Bolton, T.A.W.; Medaglia, J.D.; Bassett, D.S.; Ribeiro, A.; Van De Ville, D. A graph signal processing perspective on functional brain imaging. *Proc. IEEE* **2018**, *106*, 868–885. [\[CrossRef\]](#)
- Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [\[CrossRef\]](#)
- Sharma, V.K. Efficient image steganography using graph signal processing. *IET Image Process.* **2018**, *12*, 1065–1071. [\[CrossRef\]](#)
- Pang, J.; Cheung, G.; Hu, W.; Au, O.C. Redefining self-similarity in natural images for denoising using graph signal gradient. In Proceedings of the Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, Chiang Mai, Thailand, 9–12 December 2014; pp. 1–8. [\[CrossRef\]](#)
- Turk, G.; Levoy, M. Zippered polygon meshes from range images. In Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, Orlando, FL, USA, 24–29 July 1994; Association for Computing Machinery, New York, NY, USA, 1994; pp. 311–318. [\[CrossRef\]](#)
- Lin, T.; Zha, H. Riemannian manifold learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 796–809.
- Sakiyama, A.; Tanaka, Y.; Tanaka, T.; Ortega, A. Efficient sensor position selection using graph signal sampling theory. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 6225–6229. [\[CrossRef\]](#)
- Chen, S.; Tian, D.; Feng, C.; Vetro, A.; Kovačević, J. Fast resampling of three-dimensional point clouds via graphs. *IEEE Trans. Signal Process.* **2018**, *66*, 666–681. [\[CrossRef\]](#)
- Chamon, L.F.O.; Ribeiro, A. Greedy sampling of graph signals. *IEEE Trans. Signal Process.* **2018**, *66*, 34–47. [\[CrossRef\]](#)

11. Hashemi, A.; Shafipour, R.; Vikalo, H.; Mateos, G. Sampling and reconstruction of graph signals via weak submodularity and semidefinite relaxation. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4179–4183. [[CrossRef](#)]
12. Alpago, D.; Zorzi, M.; Ferrante, A. Identification of sparse reciprocal graphical models. *IEEE Control. Syst. Lett.* **2018**, *2*, 659–664. [[CrossRef](#)]
13. Ciccone, V.; Ferrante, A.; Zorzi, M. Robust identification of “Sparse Plus Low-rank” graphical models: An optimization approach. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 17–19 December 2018; pp. 2241–2246. [[CrossRef](#)]
14. Natarajan, B.K. Sparse approximate solutions to linear systems. *SIAM J. Comput.* **1995**, *24*, 227–234. [[CrossRef](#)]
15. Das, A.; Kempe, D. Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection. In Proceedings of the 2011 IEEE International Conference on Machine Learning (ICML), Washington, DC, USA, 28 June–2 July 2011.
16. Qian, C.; Yu, Y.; Zhou, Z.H. Subset selection by Pareto Optimization. In *Advances in Neural Information Processing Systems*; Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; MIT Press: Cambridge, MA, USA, 2015; Volume 28, pp. 1774–1782.
17. Qian, C.; Bian, C.; Feng, C. Subset selection by Pareto Optimization with recombination. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 2408–2415. [[CrossRef](#)]
18. Qian, C.; Shi, J.C.; Yu, Y.; Tang, K.; Zhou, Z.H. Subset selection under noise. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; MIT Press: Cambridge, MA, USA, 2017; Volume 30, pp. 3560–3570.
19. Chen, S.; Varma, R.; Sandryhaila, A.; Kovačević, J. Discrete signal processing on graphs: Sampling theory. *IEEE Trans. Signal Process.* **2015**, *63*, 6510–6523. [[CrossRef](#)]
20. Tylavsky, D.J.; Sohie, G.R.L. Generalization of the matrix inversion lemma. *Proc. IEEE* **1986**, *74*, 1050–1052. [[CrossRef](#)]
21. Bogunovic, I.; Zhao, J.; Cevher, V. Robust maximization of non-submodular objectives. In Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Lanzarote, Canary Islands, 9–11 April 2018; Storkey, A., Perez-Cruz, F., Eds.; PMLR: Brookline, MA, USA 2018; Volume 84, pp. 890–899.
22. Zhou, Z.H.; Yu, Y.; Qian, C. *Evolutionary learning: Advances in Theories and Algorithms*, 1st ed.; Springer: Singapore, 2019.
23. Joshi, S.; Boyd, S. Sensor selection via Convex Optimization. *IEEE Trans. Signal Process.* **2009**, *57*, 451–462. [[CrossRef](#)]
24. Zhang, F.; Hancock, E.R. Graph spectral image smoothing using the heat kernel. *Pattern Recognit.* **2008**, *41*, 3328–3342. [[CrossRef](#)]