A Preference-Based Multiobjective Evolutionary Approach for Sparse Optimization

Hui Li, Senior Member, IEEE, Qingfu Zhang, Fellow, IEEE, Jingda Deng, and Zong-Ben Xu

Abstract—Iterative thresholding is a dominating strategy for sparse optimization problems. The main goal of iterative thresholding methods is to find a so-called k-sparse solution. However, the setting of regularization parameters or the estimation of the true sparsity are nontrivial in iterative thresholding methods. To overcome this shortcoming, we propose a preference-based multiobjective evolutionary approach to solve sparse optimization problems in compressive sensing. Our basic strategy is to search the knee part of weakly Pareto front with preference on the true k-sparse solution. In the noiseless case, it is easy to locate the exact position of the k-sparse solution from the distribution of the solutions found by our proposed method. Therefore, our method has the ability to detect the true sparsity. Moreover, any iterative thresholding methods can be used as a local optimizer in our proposed method, and no prior estimation of sparsity is required. The proposed method can also be extended to solve sparse optimization problems with noise. Extensive experiments have been conducted to study its performance on artificial signals and magnetic resonance imaging signals. Our experimental results have shown that our proposed method is very effective for detecting sparsity and can improve the reconstruction ability of existing iterative thresholding methods.

Index Terms—Sparse optimization, regularization, iterative thresholding, multiobjective evolutionary approach.

I. INTRODUCTION

TN VARIOUS computational and engineering areas, such as data mining [1], variable selection [2], visual coding [3], signal and image processing [4], [5], and compressive sensing [6], one needs to consider the following sparse optimization problem:

$$\min \|x\|_0 \text{ s.t. } y = Ax$$
 (1)

where x is a signal vector in \mathbb{R}^N , y is an observation vector in \mathbb{R}^M , and A is a sensing matrix in $\mathbb{R}^{M \times N}$. If Problem (1) involves noise, then the constraint is $y = Ax + \epsilon$, where $\epsilon \in \mathbb{R}^M$ is the noise level. Usually, A is a very "flat" matrix,

Manuscript received May 18, 2016; revised October 26, 2016 and February 20, 2017; accepted February 20, 2017. This work was supported in part by the National Science Foundation of China under Grant 61573279, Grant 61175063, Grant 61473241, Grant 11626252, and Grant 11690011, and in part by the National Basic Research Program of China under Grant 2017CB329404.

H. Li and Z.-B. Xu are with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China (e-mail: lihui10@xjtu.edu.cn; zbxu@xjtu.edu.cn).

Qingfu Zhang and Jingda Deng are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: qingfu.zhang@cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2677973

i.e., $M \ll N$. This means that the linear system y = Ax is underdetermined. The objective function $||x||_0$, the L0-norm of x, is defined as the number of nonzero components in x. It takes discrete integer values from $\{0, 1, ..., N\}$. The goal of Problem (1) is to find the sparsest solution x^* of y = Ax, which is minimal to $||x||_0$. x^* is called k-sparse if $||x^*||_0 = k$. An equivalent formulation of Problem (1) can be stated as

$$\min \|y - Ax\|_2^2 \text{ s.t. } \|x\|_0 \le k \tag{2}$$

where x, y, and A are the same as in (1).

Using the penalty method, Problem (1) can be converted into an unconstrained optimization problem, called L0 regularization, which can be stated as follows:

$$\min_{x \in \mathcal{R}^{N}} ||y - Ax||_{2}^{2} + \lambda ||x||_{0}$$
(3)

where $||y - Ax||_2^2$, a quadratic function of x, is the loss term, and $||x||_0$ is the regularization term. λ is a positive regularization parameter, which balances the feasibility and the sparsity of x. It was proved that Problem (3) is \mathcal{NP} -hard [7]. The greedy strategies (e.g., match pursuit [8], orthogonal match pursuit (OMP) [9]), and hard iterative thresholding methods (ITH/L0) [10], [11] are two classes of commonly used methods for L0 regularization. OMP builds an approximate solution in an incremental and greedy manner. It is suitable for low-dimensional sparse optimization problems. In contrast, ITH/L0 is more effective, and applicable for high-dimensional problems [12], [13].

To make the L0 regularization problem (3) solvable by continuous optimization methods, one can consider its relaxed problem

$$\min_{x \in \mathcal{R}^N} \|y - Ax\|_2^2 + \lambda \|x\|_q^q \tag{4}$$

where $||x||_q^q = (\sum_{i=1}^n |x_i|^q)^{1/q}$, $q \in (0, 1]$, is called the Lq-norm of x. The smaller the value of q is, the sparser the solution of Problem (4) becomes. In practice, only very few special values of q, such as 0.5 and 1.0, are taken into consideration due to the existence of closed-form optimal solution. In the case of q = 1, Problem (4) is a well-known convex optimization problem, called L1 regularization. It can be transformed to a quadratic optimization problem and solved by soft iterative thresholding methods (ITH/L1) [14]–[16]. In the case of q = 0.5, Problem (4) is a non-Lipschitz-continuous and nonsmooth optimization problem called L0.5 regularization. Recently, a fast and efficient iterative thresholding solver based on the L0.5-norm (ITH/L0.5) was proposed in [17].

2162-237X © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



Fig. 1. Distribution of weakly Pareto solutions in the biobjective sparse optimization problem (5) including all solutions of y = Ax.

Regularization method is a popular technique for dealing with overfitting problem in statistics and machine learning [18]–[21]. In fact, the choice of suitable regularization parameter is a very critical task [22]-[24]. Over the recent few years, various strategies for choosing regularization parameters have been studied in the area of machine learning [25]-[29]. In [25], the improved variants of Akaike information criteria [30] and Bayesian information criteria [31] based on Bayesian arguments and the Kullback-Leibler divergence are proposed to choose regularization parameters for the image restoration problem. In [26], the regularization parameters in regularized negative correlation learning are optimized by an automatic algorithm based on Bayesian inference. In [27], the regularization parameters in faulty radial basis function networks are optimized by means of mean prediction error results. In [28], the cross validation method based on the rule of "trial and error" is employed in choosing regularization parameters for solving the support vector machine regularization path. In [29], the homotopy methods are proposed to overcome the difficulty of the iterative hard thresholding method on the choice of the regularization parameter. Moreover, the sparsity regularization also has the difficulties on the choice of regularization parameters, where the true sparsity value k or its estimation value should be provided in advance [17]. However, the value of k is often unknown in practice.

So far, many challenging optimization problems in machine learning have been solved via multiobjective framework [32]–[35], where the constraints are often converted into an extra objective. Following this idea, Problem (1) with equality constraints can be naturally transformed into the following biobjective optimization problem:

$$\min_{x \in \mathcal{R}^N} \left\{ f_1(x) = ||x||_0, \, f_2(x) = ||y - Ax||_2^2 \right\}$$
(5)

where x, y, and A are the same as in Problem (1). Compared with the regularization problems in (3) and (4), the advantage of Problem (5) is that no regularization parameter is needed. Note that the k-sparse solution of Problem (1) is the Pareto solution of Problem (5) with $f_2 = 0$ (i.e., y = Ax). This is visualized in an illustrative example in Fig. 1, where the k-sparse solution Q is located in the knee part of weakly Pareto front.

In the area of evolutionary computation, multiobjective evolutionary algorithms (MOEAs) have attracted much attention for approximating the Pareto front. Recently, two pioneering MOEAs—MOEA/D(half) [36] and StEMO(soft) [37]—have been suggested for sparse signal reconstruction. The former is interested to search the knee region of weakly Pareto front of Problem (5) with preference, while the latter aims at approximating the whole Pareto front. In both algorithms, the existing ITH methods, such as ITH/L0.5 or ITH/L1, are used for local improvement. It should be pointed out that their performances in solution precision and computational efficiency are not competitive to ITH/L0 or ITH/L0.5 although the setting of regularization parameters is not needed. The main reason is that the majority of their computational costs are wasted on approximating the region far from the k-sparse solution.

Over the past 20 years, the preference-based MOEAs have been widely studied in the area of evolutionary multiobjective optimization [38]-[41]. The major goal of preference-based MOEAs is to approximate a local part of Pareto front. In [42], a general decomposition-based MOEA called MOEA/D was proposed. It optimizes multiple subproblems in a collaborative manner. Unlike other MOEAs, the preference information can be easily combined with the subproblems of MOEA/D. To improve the performance of multiobjective methods for Problem (5), the knee region of weakly Pareto front near the k-sparse solution must be preferred for the exploitation of the search. On the basis of MOEA/D framework, this paper proposed a new preference-based multiobjective evolutionary approach for sparse optimization, called sparse preferencebased local search (SPLS). Its new features include the following.

- Incremental Steady-State Search Mode: In each iteration, only one solution close to the knee region is preferred in the selection of starting solution for local search based on thresholding search. As the search processes, the number of nonzero components in the solutions of current population is gradually increased until the knee region is reached.
- Multilevel Truncation Strategy: In thresholding search, each solution obtained by gradient descent method is truncated at multiple sparsity levels, which belong to a *T*-neighborhood of the current sparsity level. This is very crucial for diversifying the search along weakly Pareto front.
- 3) *Two-Archived Removal Rule:* When the size of population exceeds its maximal size, part of solutions with large values of loss function f_2 in one of its subsets or those with large values of sparsity f_1 in the other subset are removed with priority.

The major contribution of SPLS for the community of machine learning is that it provides a new effective and efficient preference-based multiobjective framework for sparse reconstruction without the difficulty on the choice of regularization parameter.

To study the performance of our proposed method-SPLS, we have done the following experimental simulations in this paper.

1) We have conducted extensive experiments to compare the performance of three versions of SPLS with three existing ITH methods, OMP, as well as two MOEAs for sparse reconstruction on both short-length artificial signals and long-length artificial signals.

- 2) We have made the sensitivity analysis on the major parameters in SPLS to study the stability of SPLS. The performances of SPLS on different values of $(\mathcal{N}, \mathcal{M}, k)$ have also been studied.
- 3) We have analyzed the sparsity of five magnetic resonance imaging (MRI) signals from the benchmark library and compared SPLS/L0.5 and ITH/L0.5 on reconstructing the signals in the sparse versions of these MRI images.

The rest of this paper is organized as follows. Section II briefly reviews several ITH methods. In Section III, the motivations and the detailed description of SPLS are presented. The experimental results on the performance of SPLS on artificial signals are reported and analyzed in Section IV. One application of SPLS on the sparsity detection of MRI signals is presented in Section V. Section VI concludes this paper.

II. BRIEF REVIEW ON ITERATIVE THRESHOLDING METHODS

The main goal of ITH methods is to find the solution of the regularization optimization problem formulated in (3) or (4). A typical ITH method involves three major steps—gradient descent, setting of regularization parameter, and thresholding truncation. In ITH methods, the estimation of the true sparsity k, let us say \bar{k} , or an empirical value of regularization parameter should be provided in advance. Three well-known ITH methods, i.e., ITH/L0, ITH/L0.5, and ITH/L1, mainly differ in the last two steps. In the following, we introduce these steps in brief.

1) *Gradient Descent:* Assume that $x^{(n)}$ is the current solution, a trial solution $\tilde{x} \in \mathcal{R}^N$ is generated by

$$\tilde{x} = x^{(n)} + \mu \times A^T (Ax^{(n)} - y)$$

where $A^T(Ax^{(n)} - y)$ is the negative gradient vector of $f_2 = \|y - Ax\|_2^2$ at $x^{(n)}$, and $\mu > 0$ is the step size. As suggested in [17], μ is set to 1 in this paper.

- 2) Setting of λ : The current regularization parameter λ_n can be set to a constant empirical value (e.g., ITH/L1) or determined by the prior estimation \bar{k} . In ITH/L0 and ITH/L0.5, λ_n is set to $|\tilde{x}_{i_{\bar{k}+1}}|$ and $(\sqrt{96}/9)|\tilde{x}_{i_{\bar{k}+1}}|^{(3/2)}$, respectively, where $|\tilde{x}_{i_{\bar{k}+1}}|$ is the $(\bar{k} + 1)$ th largest value in $\{|\tilde{x}_j|| j = 1, ..., N\}$.
- Thresholding Truncation: To make x̃ sparse, some of its smallest components in absolute values are set to zeros. The following are three examples.
 - a) ITH/L0(hard)

$$x_i^{(n+1)} = \begin{cases} \tilde{x}_i & \text{if } |\tilde{x}_i| > \lambda_n \\ 0 & \text{otherwise.} \end{cases}$$
(6)

b) ITH/L1(soft)

$$x_i^{(n+1)} = \begin{cases} \psi_1(\tilde{x}_i, \lambda_n) & \text{if } |\tilde{x}_i| > 0.5\lambda_n \\ 0 & \text{otherwise.} \end{cases}$$
(7)

where $\psi_1(\tilde{x}_i, \lambda_n) = \operatorname{sgn}(\tilde{x}_i)(\tilde{x}_i - 0.5\lambda_n)$.

c) ITH/L0.5(half)

$$x_{i}^{(n+1)} = \begin{cases} \psi_{2}(\tilde{x}_{i}) & \text{if } |\tilde{x}_{i}| > \frac{\sqrt[3]{54}}{4}\lambda_{n}^{\frac{2}{3}} \\ 0 & \text{otherwise.} \end{cases}$$
(8)

where
$$\psi_2(\tilde{x}_i) = (2/3)\tilde{x}_i (1 + \cos((2\pi/3) - (2\phi(\tilde{x}_i)/3))))$$
 with $\phi(\tilde{x}_i) = \arccos\left((\lambda_n/8) ((|\tilde{x}_i|/3))^{-(3/2)}\right).$

According to the experimental results reported in [17], ITH/L0.5 is clearly superior to ITH/L0 and ITH/L1 in terms of sparsity recovery ability and solution precision. Over the past few years, ITH/L0.5 has been successfully used in many applications [43]–[45].

III. SPARSE PREFERENCE-BASED MULTIOBJECTIVE EVOLUTIONARY APPROACH

In this section, the motivations on the use of multiobjective algorithms based on preference for sparse optimization are first discussed. Then, the description of the proposed SPLS algorithm is provided.

A. Motivations

Solving constrained optimization problems via MOEAs has attracted much attention in the area of evolutionary constrained optimization [46]–[49]. The basic idea is to optimize an alternative biobjective optimization problem, which involves an extra objective defined by constraints. To ensure the search efficiency, only a local part of the weakly Pareto front of the biobjective optimization problem, which includes the optimal solution of the original constrained optimization problem, should be examined. Following this line, few attempts have been made on the use of MOEAs for sparse optimization.

- In [36], a decomposition-based multiobjective method with half iterative thresholding method, i.e., MOEA/D, was proposed to approximate the knee part of the weakly Pareto front of Problem (5) in an online mode. In each generation, multiple subprobelms with different sparsity levels are optimized by existing ITH methods in a parallel way. During the search, the subproblems with boundary sparsity levels are adaptively changed with the preference to move the population toward *k*-sparse solution. Note that the conventional crossover operators are not used in MOEA/D. Instead, the solution of each subproblem is disturbed by mutation before local search.
- 2) In [37], a variant of NSGA-II based on soft iterative thresholding method, called StEMO, was developed to find the sparse solution of Problem (1) in an offline mode. First, StEMO needs to approximate the whole Pareto front of Problem (5). The BLX-α crossover and the nonuniform mutation are used for producing offspring solutions. Then, the B-spline curve fitting method and an angle-based method are adopted to locate the knee solution of the approximation of Pareto front in a posterior way.

Unlike existing ITH methods, both MOEA/D and StEMO are free to set appropriate regularization parameter. However, their performances in solution precision and computational efficiency are clearly worse than some existing efficient ITH methods, such as ITH/L0 and ITH/L0.5. The main reason lies in the unbalance between the exploration of the whole Pareto front and the exploitation of the knee part near *k*-sparse solution. To overcome this weakness, the knee part must be exploited with preference. Compared with the NSGA-II framework used in StEMO, MOEA/D is more advantageous to exploit a local part of weakly Pareto front by integrating preference due to its decomposition strategy for fitness assignment.

Assume that the population size is L in MOEA/D, it needs to optimize L subproblems defined by

$$\min_{x \in \mathcal{R}^N} \|y - Ax\|_2^2, \text{ s.t. } \|x\|_0 = k_i$$
(9)

where $K = \{k_1, \ldots, k_L\}$ is a set of L initial sparsity levels in $[k_{\min}, k_{\max}]^1$ including the true sparsity k. In MOEA/D, each sparsity level k_i is associated with one solution $x^{(i)}$. In each generation, a fixed number of thresholding iterations are used in the minimization of every subproblem defined in (9) for local search. According to the proportion of nondominated solutions in the population, the boundary subproblem with the sparsity level $k_l = \min_{s \in K} s$ or $k_r = \max_{s \in K} s$ is adaptively adjusted. Note that the adjustment rules used in MOEA/D make use of the preferences on the knee part of weakly Pareto front in selection, which contains some solutions with similar structures. In the early stage of search, the sparsity levels of all subproblems are diversely distributed in $[k_{\min}, k_{\max}]$. Since each subproblem consumes the same amount of computational cost, the parallel optimization (i.e., generational search mode) of multiple subproblems with sparsity levels far from the knee part will waste lots of computational cost. Moreover, the performance of MOEA/D highly depends on the setting of population size L.

B. Description of SPLS

To overcome the weakness in MOEA/D, the knee part should be approximated with a steady-state mode, where only one subproblem with the sparsity level closer to the true sparsity k is selected for optimization in each iteration. In this paper, we further suggest an improved version of MOEA/D with preference for sparse optimization with the following major changes.

- 1) The number of subproblems is dynamically changed. Some subproblems with large or small sparsity levels will be removed with the preference on the knee part.
- 2) Only one subproblem (solution) is selected with the preference for thresholding search at a time. The solutions with small values of loss function f_2 are preferred in the thresholding search.
- 3) During the thresholding search, multiple neighboring solutions with similar sparsity levels to the starting solutions are obtained with a multilevel thresholding truncation.
- 4) The population is improved by thresholding search in an incremental manner like OMP. That is, the sparsity

¹In cross validation of the exisiting ITH methods, k_{\min} and k_{\max} are the prior lower bound and upper bound of the range of sparsity levels.

levels of the solutions in the population are very small in the beginning of the search and increased gradually until the knee region is reached.

Since our proposed method aims at finding some weakly Pareto solutions in the knee part with preference using existing thresholding method as local search operator, we also called our proposed algorithm SPLS.

In the SPLS algorithm, we need to maintain the following data structures.

- 1) The sensing matrix A and the observation vector y: They are used to calculate the loss function $f_2(x) = ||y Ax||_2^2$.
- 2) The archived population E: It consists of two subsets E_1 and E_2 with the following forms:

$$E_1 = \{ x \in E | f_2(x) < f_2^{\min} + \beta \}$$
(10)

and

$$E_2 = \{x \in E | f_2(x) \ge f_2^{\min} + \beta\}$$
(11)

where f_2^{\min} is the minimal value of the loss function found so far, and β is a positive parameter.

For each solution x in E, a *T*-neighborhood of sparsity levels, i.e., [||x||₀ − T × δ, ||x||₀ + T × δ], might be considered in thresholding search. Here, δ is the smallest gap between any two neighboring sparsity levels, *T* is the neighborhood size.

Ideally, *E* should contain some weakly Pareto solutions of Problem (5) in the knee part including the sparse solution *Q*. When $\beta = 0$ and $f_2^{min} = 0$, E_1 consists of some solutions of y = Ax while E_2 includes some Pareto solutions.

The algorithmic framework of our proposed SPLS algorithm is illustrated in *Algorithm* 1. In the initialization of the SPLS algorithm (*Step 1*), two major parameters T and β are initialized. β_0 is the starting value of β , which is set to 1.0 in this work. An initial solution $x^{(0)}$ is generated in a deterministic or random way. The commonly used deterministic method is to set $x^{(0)}$ as a zero vector. In the random way, a solution with a sparsity level in $[k_{\min}, k_{\max}]$ is randomly generated. In this paper, we generate $x^{(0)}$ with the sparsity level k_{\min} randomly.

In the following, the major steps in the main loop of the SPLS algorithm are explained in detail.

- 1) SelectFromArchive: The main goal of this step is to select one solution in E with the smaller value of f_2 shown in Fig. 2. In other words, f_2 is optimized with priority. The implementation of this idea is described in *Algorithm* 2, including the following two major steps.
 - a) In Step 1, a subset E_1 of E is first determined. The positive parameter β is used to control the quality of selected solutions for thresholding search. In fact, when β takes a relatively small value, the solutions in E_1 are superior to those in E_2 regarding the loss function f_2 .
 - b) In *Step 2*, a starting solution x' is selected from E_1 randomly. A *T*-neighborhood of the central sparsity level k', i.e., $||x'||_0$, is determined. k_l and k_r are the lower bound and the upper bound of the *T*-neighborhood, respectively. k'' is a random

Algorithm 1 SPLS

Step 1: Initialization Initialize T and $\beta = \beta_0$. Generate $x^{(0)}$ and let $E = \{x^{(0)}\}$.

Step 2: Main Loop

2.1 Select a starting solution x' and a random sparsity level k'' for thresholding search:

 $(x', k_l, k'') \leftarrow$ SelectFromArchive (E, β, T)

2.2 Perform thresholding search on x' with the sparsity level k'':

$x'' \leftarrow \text{ThresholdingSearch}(x', k'')$

2.3 Truncate x'' at multiple sparsity levels in a *T*-neighborhood:

$\tilde{E} \leftarrow \text{TruncateByMultiLevel}(x'', k_l, k'')$

2.4 Update the archived population E with \tilde{E} and limit the size of E:

$$E \leftarrow \text{UpdateArchive}(E, E, \beta, T)$$

Step 3: Stopping Criteria

If a certain stopping condition is met, then stop and **output** *E*; otherwise go to **Step 2**.

sparsity level between k' and k_r . It will be used as the new sparsity level in thresholding search.

Algorithm 2 SelectFromArchive (E, β, T)

Step 1: Determine a subset E_1 of E including a number of solutions with high quality regarding f_2 by (10); **Step 2:** Select x' from E_1 randomly and determine sparsity levels in a *T*-neighborhood:

$$k_{l} = \|x'\|_{0} - T \cdot \delta, \ k_{r} = \|x'\|_{0} + T \cdot \delta,$$

$$k' = \|x'\|_{0}, \qquad k'' = \|x'\|_{0} + |T \times rand| \cdot \delta$$

where *rand* is a uniform random number in [0, 1], and k'' is a random sparsity level between k' and k_r . Step 3: Output x', k_l and k''.

- 2) ThresholdingSearch : The implementation of thresholding search in SPLS is shown in Algorithm 3. x' and k" obtained in Algorithm 2 are the input parameters. The randomness of k" can be viewed as the mutation on the sparsity level k' of x'. No crossover is used in the SPLS algorithm. The truncation strategies used in SPLS could be the same as those in ITH/L0(hard), ITH/L0.5(half), and ITH/L1(soft) introduced in Section II. The thresholding search in Algorithm 3 is terminated if the number of iterations reaches the specified number of iterations ls.
- 3) *TruncateByMultiLevel*: The main idea of this step is to truncate the improved solution x'' at different sparsity levels in a subset of the *T*-neighborhood of k', i.e., $\{k_l, k_l + \delta, ..., k'' - \delta, k''\}$. As a result,



Fig. 2. Graphical illustration of the archiving set *E* in SPLS, which consists of two subsets E_1 and E_2 . Ideally, when $\beta = 0$ and $f_2^{min} = 0$, all solutions in E_1 satisfy y = Ax, while those in E_2 are strictly Pareto solutions of Problem (5). The circles being crossed stand for the nonpreferred solutions in selection.

Algorithm 3	ThresholdingSearch (x', k'')
	()

Step 1: Set n = 0 and $x^{(n)} = x'$.

Step 2: Generate a trial solution \tilde{x} using gradient descent method. That is

$$\tilde{x} = x^{(n)} + \mu A^T (y - A x^{(n)}).$$

Step 3: Sort the absolute values of components of \bar{x} :

$$|\tilde{x}_{i_1}| \ge |\tilde{x}_{i_2}| \ge \cdots \ge |\tilde{x}_{i_{k''}}| \ge \cdots \ge |\tilde{x}_{i_N}|$$

where $i_1, i_2, ..., i_N$ is a permutation of $\{1, 2, ..., N\}$. **Step 4:** Produce $x^{(n+1)}$ by the truncation of \tilde{x} using one of the thresholding operators in (6), (7), and (8) with $\bar{k} = k''$. **Step 5:** If $n \ge ls$, then **stop** and **output** $x'' = x^{(n+1)}$; otherwise set n := n + 1 and go to **Step 2**.

 $\lfloor (k'' - k_l)/\delta \rfloor + 1$ new solutions are obtained and added into \tilde{E} . The details of this step is given in *Algorithm* 4. Fig. 3 shows the distribution of multiple solutions obtained by truncating x'' for a number of times. To locate the exact position of k-sparse solution, δ is often set to 1.

Al	gorithm	4	TruncationBy	vMultiLevel	(x'')	k_l	k'	"
----	---------	---	--------------	-------------	-------	-------	----	---

Step 1: Set $\tilde{E} = \emptyset$, z = x'', and c = 1. **Step 2:** Add z into \tilde{E} . If $k'' - c\delta < k_l$, then **stop** and **output** \tilde{E} ;

Step 3: Set δ smallest nonzero components of z in absolute values as zeroes, c := c + 1 and go to Step 2.

4) UpdateArchive : All solutions in \tilde{E} obtained by thresholding search and multilevel truncation are used to update the archived population E. To focus on searching the knee part around the k-sparse solution Q, the size of E should be limited. Its members with the smallest sparsity levels or the largest sparsity levels could be removed with preference. To this end, the update of archived population E is implemented in Algorithm 5.



Fig. 3. Thresholding search method with multilevel truncation in SPLS. An initial solution x' is improved by thresholding search with a random sparsity level k'' in the right part of *T*-neighborhood with the center at k'. The improved solution x'' is truncated at multiple sparsity levels from k_l to k''. In this example, $x^{(1)}, \ldots, x^{(8)}$ are generated by thresholding search on x' at the sparsity level k''.

Steps 1.1 and 1.2 update the solutions in E_1 and E_2 , respectively. In Steps 2.2 and 2.3, the sizes of E_1 and E_2 are limited to T. The solutions with the largest sparsity levels in E_1 and those with the smallest sparsity levels in E_2 are not preferred in selection. That is, they should be removed from E shown in Fig. 2. In this way, the size of E is not larger than $2 \times T$. Note that the parameter β is decreased to max{ $\beta_1, 0.8\beta$ } if $|E_1| > T$. β_1 is the minimal value of β , which is much less than β_0 . The dynamical change of β is helpful on the diversity of the starting solution x'.

Algorithm 5 UpdateArchive (E, \tilde{E}, β, T)

Step 1: Update the external archive

for each $z \in \tilde{E}$

1.1 If $f_2(z) < \min_{x \in E} f_2(x) + \beta$, then

If $\not\exists z'$ in E with $||z'||_0 = ||z||_0$, then add z into E; otherwise, set z' := z if $f_2(z) < f_2(z')$.

1.2 If $f_2(z) \ge \min_{x \in E} f_2(x) + \beta$, then

Remove all solutions in E dominated by z. If no solution in E dominates z, then add z into E.

end for

Step 2: Reduce the size of the external archive

2.1 Separate E into E_1 and E_2 by (10) and (11).

2.2 If $|E_1| > T$, then remove $|E_1| - T$ solutions with the largest sparsity levels in E_1 , and set $\beta := \max\{\beta_1, 0.8\beta\}$;

2.3 If $|E_2| > T$, then remove $|E_2| - T$ solutions with the smallest sparsity levels in E_2 ;

Step 3: Set $E = E_1 \cup E_2$ and **output** it.

C. Computational Complexities of SPLS

In each step of ITH methods and SPLS, the gradient method is first applied to improve one certain solution regarding f_2 , and then, the components of the improved solution are sorted in terms of their absolute values with computational load $O(N \log(N))$. Compared with ITH methods, the extra computational complexities of SPLS are caused by Truncation-ByMultiLevel and UpdateArchive. In TruncationByMultiLevel, $(k'' - k_l)/\delta + 1 \ (\in [T, 2T])$ solutions should be generated and stored in \tilde{E} . Therefore, O(T) evaluations of loss function $f_2 = ||y - Ax||^2$ should be computed. In Step 1.2 of UpdateArchive, the update of E needs O(T) comparisons of Pareto dominance. To separate E into E_1 and E_2 , $2 \times T$ comparisons between $f_2(x)$ and $f_2^{\min} + \beta$ in Step 2.1 of UpdateArchive. To reduce the size of E, $O(T \times \log(T))$ comparisons of sparsity values for the solutions in E_1 and E_2 are required as shown in Steps 2.2 and 2.3 of UpdateArchive. According to the above analysis, it is a certain fact that the computational efficiency of SPLS depends on the settings of T and N.

D. Superiority of SPLS Against MOEA/D and StEMO

The common feature of MOEA/D, StEMO, and SPLS is the overall framework, which combines EMO with thresholding search. However, the differences among them are substantial.

- 1) Decomposition and Preference: In both MOEA/D and SPLS, a number of single objective subproblems obtained by decomposition with a quadratic objective function $||y Ax||^2$ are optimized. The local knee region of Pareto front is examined with the preference of *k*-sparse solution in an online way. In StEMO, a Pareto-based NSGA-II is used to approximate the whole Pareto front, and the knee solution is obtained by a curve fitting method in a posterior way.
- 2) Thresholding: MOEA/D considers either ITH/L0 or ITH/L0.5 in local search, while StEMO uses ITH/L1 in local search, while StEMO uses iterative soft thresholding method (ITH/L1) in local search. In our proposed SPLS, any of ITH/L0, ITH/L1, and ITH/L0.5 can serve as a local search optimizer. An exclusive multiple truncation is used in the thresholding iteration of SPLS.
- 3) Steady-State or Generational Mode: MOEA/D and StEMO are two generational EMO algorithms, while SPLS is a steady-state approach. More precisely, SPLS only generates one solution by a preference-based local search at one iteration.

SPLS is superior to MOEA/D and StEMO due to the following reasons.

- SPLS spends most of its computational cost on approximating the knee region close to *k*-sparse solution, while the other two could waste a lot of computational cost on the area far from the *k*-sparse solution.
- 2) Compared with the ITH/L1 in StEMO and the ITH/L0 in MOEA/D, the ITH/L0.5 used in SPLS is more powerful for local improvement in thresholding search. It should be mentioned that both ITH/L0 and ITH/L0.5 cannot be used in StEMO due to its Pareto-based framework.

IV. COMPUTATIONAL EXPERIMENTS

In this section, we compare the performance of three versions of SPLS with three existing ITH methods, OMP, as well as two multiobjective evolutionary approaches on artificial

TABLE I Settings (N, M, k) of Six Artificial Problem Sets Used in Our Experiments

Problem	Configuration	Problem	Configuration
P1	(512, 330, 130)	P4	(512, 240, 130)
P2	(512, 300, 130)	P5	(512, 235, 130)
P3	(512, 270, 130)	P6	(512, 230, 130)

signals in terms of success ratio (SR) and solution precision. The parameter analysis of SPLS is also presented.

A. Noise-Free Artificial Sparse Signal

Similar to the method for generating the test problems of sparse optimization in [17], we artificially constructed six sets of test problems without noises in Table I. Each of them is associated with the configuration (N, M, k). Three problem sets P1–P3 involve 512-length optimal signals with the true sparsity level—130. The number of measurements M of P1–P3 (i.e., the row of sensing matrix A) is 330, 300, and 270, respectively. The other three problem sets P4–P6 have the same settings of N and k as those of P1–P3. But the M values of them, i.e., 240, 235, and 230, are smaller. For all problem sets, the range $[k_{\min}, k_{\max}]$ of the estimation sparsity levels is set to $[0.1 \times k, 2 \times k]$.

For each problem set, 100 random instances are generated in the following way.

- 1) The optimal sparse signal x^* with the given length and sparsity is generated with Gaussian distribution $\mathcal{N}(0, 2)$.
- 2) The entries in the sensing matrix A are sampled with the Gaussian distribution $\mathcal{N}(0, 1)$. A is orthogonalized and normalized.
- 3) The observation vector y equals to the multiplication of A and x^* , i.e., $y = Ax^*$.

To compare the performance of the sparse optimization methods under consideration, two indicators are considered in our experiments.

- 1) *Mean Square Error:* The MSE values between the true optimal sparse solution x^* and the approximate solutions obtained by the methods under consideration are used. In our SPLS algorithm, the solution in the final population with the minimal MSE value is considered in comparison with other ITH methods.
- 2) Success Ratio: If the final solution found by a certain method has the MSE value below a given successful MSE level τ , the corresponding run of this method is denoted as a successful run. The ratio of the successful runs on 100 random instances in each problem set is used to measure the reconstruction ability of sparse optimization methods. Empirically speaking, the sparse signal is successfully reconstructed if the MSE value of the final solution is below 10^{-6} . Therefore, τ is set to 10^{-6} in this paper.

1) Comparison Between Three SPLS Variants and four State-of-the-Art Sparse Optimizers: In this section, the performances of three versions of SPLS (i.e., SPLS/L0.5, SPLS/L0, and SPLS/L1) are compared with those of the three related ITH algorithms (i.e., ITH/L0.5, ITH/L0, and ITH/L1), as well



Fig. 4. Linear scale (left) and log scale (right) plots of the weakly Pareto-optimal solutions found by SPLS/L0.5 in a successful run on two small scale problem sets (i.e., P1 and P6).



Fig. 5. Linear scale (left) and log scale (right) plots of the weakly Pareto-optimal solutions found by SPLS/L0.5 in a successful run on 20 times of P2 (10240, 6000, 2600).

as OMP on P1–P6. All algorithms were implemented in MATLAB on the PC with the Intel Xeon CPU at 3.20-GHz and 32-GB memory running Windows 7 operating system.

The major parameters in the SPLS algorithm are T, (β_0, β_1) , and ls, which are set to 10, $(1.0, 10^{-6})$, and 20, respectively. The total number of thresholding iterations is 3000 in the first six algorithms. The minimal interval between two consecutive sparsity levels δ is set to 1 for P1–P6.

Fig. 4 shows the distribution of the weakly Pareto solutions found by SPLS/L0.5 in one of its successful run on P1 and P6, which are the easiest instance and the hardest instance among P1–P6, respectively. It is very clear that the Pareto solution



Fig. 6. Errors of signals reconstructed by SPLS/L0, SPLS/L0, SPLS/L1, ITH/0.5, ITH/L0, and ITH/L1 on a typical instance from P1 (left) and P6 (right). The original sparse signals of both instances are plotted on the top.



Fig. 7. Errors of signals reconstructed by OMP on P1 (left) and P6 (right).

with true sparsity k = 130 can be visually identified. In the log scale of this figure, there is a jump at the position of the true sparse solution. This result shows that SPLS/L0.5 is effective for detecting the exact position of the true sparsity. It should be mentioned that SPLS/L0.5 also fails to detect the sparsity on the instances from P4–P6 with certain probabilities. The reason is that the ratio of observations in these problems could be too small to make a successful reconstruction by the variants of SPLS or ITH methods.

The SPLS algorithm is also applicable to the long-length sparse signal. To move the population toward the knee part faster, the parameter δ is set to 10 with the probability 0.9 and 1 with the probability of 0.1. In this way, the interval $[k_l, k_r]$ of sparsity values in the multilevel thresholding search is 200 in most cases. The weakly Pareto solutions found by SPLS/L0.5 on 20 times of (512, 300, 130) in P2 is shown in Fig. 5. It is evident that SPLS/L0.5 still has the ability to detect the large sparsity value 2600 for 10 240-length signals.

Apart from the detection of sparsity, SPLS can also serve as an optimizer for sparse optimization. The solution at the knee position can be treated as the recovered solution. To show the difference between this solution and the original solution, we plot the error values between them on two instances from P1 and P6 in Figs. 6 and 7. It can be observed from these two figures that the small error values are found by all seven algorithms, including OMP on the easy problem P1. For the hard problem P6, SPLS/L0.5 is the only optimizer obtaining an approximate solution with very small error values.

Fig. 8 shows the SRs of seven algorithms for the instances from P1–P6 at different MSE values in between 10^{-34} and 1. From the results in this figure, we have the following observations.

- For the instances from three problems P1–P3, SPLS/L0.5, SPLS/L0, and ITH/L0.5 clearly outperform three other algorithms (i.e., ITH/L0, SPLS/L1, and ITH/L1) in terms of the SR at various MSE levels. The successful runs of SPLS/L0.5, SPLS/L0, and ITH/L0.5 are 100 or close to 100. It can also be observed from Fig. 8 that each ITH method is evidently outperformed by its counterpart version of SPLS algorithm. For example, the SR of ITH/L0 on the instances from P3 is only 6%, while that of SPLS/L0 is 96%. This indicates that the performance of ITH/L0 can be significantly improved if it is integrated into the framework of SPLS.
- 2) Fig. 8 also shows the SRs of three SPLS algorithms and three ITH methods on P4–P6, which have fewer number of observations (240 or 235 or 230). These results indicate that SPLS/L0.5 performs best among all six algorithms. It is also evident that the SRs of all algorithms on P4–P6 are clearly worse than those of all algorithms on P1–P3. This is because P4–P6 are more challenging for sparse optimization algorithms due to the smaller value of M.
- 3) Among three SPLS variants, SPLS/L0.5 is clearly superior to other two, while SPLS/L1 has the worst SRs on all problem instances. SPLS/L0 succeeds in detecting the true sparsity levels of P1–P6 in some runs with the MSE values below 10⁻⁶. In contrast, SPLS/L1 is only effective in sparsity detection for P1. Moreover, although ITH/L0 is outperformed by ITH/L0.5 on P1–P6, its SPLS counterpart method, i.e., SPLS/L0, clearly performs better than ITH/L0.5

LI et al.: PREFERENCE-BASED MULTIOBJECTIVE EVOLUTIONARY APPROACH FOR SPARSE OPTIMIZATION



Fig. 8. SRs of SPLS/L0.5, SPLS/L0, SPLS/L1, ITH/L0.5, ITH/L0, ITH/L1, and OMP at different MSE levels on P1-P6.

on P4–P6. This result also proves the advantage of SPLS.

- 4) As for the solution precision, it is also easy to see from Fig. 8 that SPLS/L0.5 is competitive to ITH/L0.5 in terms of the mean MSE values. In the most successful runs of SPLS, its mean MSE values have the magnitude of 10^{-30} .
- 5) The comparison among the SPLS variants and OMP is also shown in Figs. 7–8. From these results, we can observe that OMP is clearly outperformed by SPLS/L0.5 and SPLS/L0 on P1–P6 regarding SRs, and it only performs better than SPLS/L1. Compared with three ITH methods, OMP is clearly inferior to ITH/L0.5, and superior to ITH/L0 and ITH/L1. These results are consistent to those report in [17].

Fig. 9 shows the SRs of SPLS/L0.5, ITH/L0.5, and OMP on two large scale problem sets, i.e., 20 times of P2 and P4. From the experimental results in this figure, we can observe that: 1) all three algorithms have 100% SRs on the 20 times of P2, but OMP has the poorest performance in solution precision and 2) the SR of SPLS/L0.5 is still 100% on the 20 times of P4, while those of ITH/L0.5 and OMP are zero. Compared with the results for P4 in Fig. 8, where the SRs of SPLS/L0.5 and ITH/L0.5 are about 65% and 26%, respectively, the performance of SPLS/L0.5 has significant improvement on the large scale version of P4, while that of ITH/L0.5 gets deteriorated dramatically. The results indicate that SPLS/L0.5 has very good potential to reconstruct long-length sparse signals with low ratio (M/N) of observations.

To compare the efficiency of all seven algorithms quantitatively, the computational time in 100 runs of nine algorithms is shown in Table II. The results in this table indicate that OMP is the fastest method for sparse reconstruction among all algorithms. The computational time consumed by OMP is less than 0.05 s for P1–P6. However, the advantage of OMP



Fig. 9. SRs of SPLS/L0.5, ITH/L0.5, and OMP on two large scale problem sets—20 times of P2 (top) and 20 times of P4 (bottom).

in computational efficiency depends on the scale of sparse optimization problem. In our experiments, we also compared OMP with SPLS/L0.5 on two large scale problems, i.e., the 20 times of P2 and P4. To finish one run, OMP spends about 360 s for P2 and 293 s for P4, while SPLS/L0.5 needs about 80 s for P2 and 65 s for P4. Therefore, SPLS/L0.5 has better scalability on the scale of sparse optimization problem than OMP.

IADLE II						
AVERAGE RUNNING TIME (IN SECONDS)	CONSUMED BY NINE ALGORITHMS	ON THE PROBLEMS P1–P6				

Problem	SPLS/L0.5	SPLS/L0	SPLS/L1	ITH/L0.5	ITH/L0	ITH/L1	OMP	MOEA/D	StEMO
P1	0.91	0.59	0.62	0.61	0.29	0.28	0.049	0.68	7.03
P2	0.89	0.56	0.62	0.60	0.27	0.26	0.047	0.67	6.65
P3	0.89	0.56	0.61	0.59	0.25	0.25	0.046	0.66	5.97
P4	0.89	0.55	0.61	0.58	0.23	0.24	0.044	0.66	2.60
P5	0.90	0.56	0.60	0.59	0.23	0.24	0.044	0.66	2.46
P6	0.90	0.55	0.59	0.57	0.22	0.23	0.044	0.67	2.54

From the results in Table II, it can be found that the computational time of SPLS algorithms is only about 1.5–3 times of their counterpart ITH algorithms. Compared with three ITH methods, such computational efficiency of the populationbased SPLS algorithms is quite promising. However, this comparison is unfair for the SPLS algorithms, since the true sparsity k is directly used in the ITH algorithms. In practice, the cross validation is often adopted for the estimation of k in the ITH methods. This means part of or all sparsity levels in $[k_{\min}, k_{\max}]$ should be considered one by one. In the case of k = 130, if ten sparse levels in $[0.5 \times 130, 2 \times 130]$ are considered in cross validation, the total computational time consumed by the ITH methods is at least 3 times of the SPLS algorithms. In this sense, the efficiency of SPLS is much better than that of ITH methods.

2) Comparison Between SPLS and MOEAs: In this paper, we also discussed two other multiobjective methods-StEMO and MOEA/D in the comparison with our proposed method. It should be pointed out that StEMO aims at finding the whole Pareto front, while MOEA/D attempts to approximate the knee area of the weakly Pareto front. In this paper, we also consider three different versions of StEMO (i.e., StEMO/L0.5, StEMO/L0, and StEMO/L1), where the solution with sparsity level closest to k in the final population is further improved by ITH/L0.5 or ITH/L0 or ITH/L1. It should be mentioned that the population size of StEMO is set to 50, since it aims at approximating the whole Pareto front. In MOEA/D, the population size is set to 10, and the number of iterations in thresholding search (i.e., ls) is set to 20. The maximal number of iterations in StEMO and MOEA/D is 3000, while those of StEMO/L0.5, StEMO/L0, and StEMO/L1 are 6000.

Fig. 10 shows the SRs of six MOEAs on P2 and P4. These results show that SPLS/L0.5 and StEMO/L0.5 clearly perform better than the other four MOEAs on both problem sets. The main reason is that both algorithms use ITH/L0.5 as local search for intensifying the search near the true sparse solution. Between SPLS/L0.5 and StEMO/L0.5, the former performs much better than the latter on P4. Both of them have 100% SRs on P2. Together with results in Fig. 8, we can observe that StEMO/L0.5 performs slightly better than ITH/L0.5. Although MOEA/D also uses ITH/L0.5 to optimize its subproblems, its performance is clearly worse than SPLS/L0.5 and StEMO/L0.5 on both P2 and P4. From the above results, we can conclude that the combination of MOEAs with ITH/L0.5 can provide better ability for reconstructing sparse signals in the case of fewer observations. From the computational times reported in Table II, StEMO is clearly less efficient than any other methods.



Fig. 10. SRs of SPLS/L0.5, MOEA/D(L0.5), and StEMO with L0, L1, and L0.5 on P2 (top) and P4 (bottom).

3) Performances of SPLS on Different Values of $(\mathcal{N}, \mathcal{M}, k)$: From the above experimental results, it is clear that the overall performance of SPLS is significantly better than that of three thresholding methods and two MOEAs. Note that all six problems P1–P6 only differ in the setting of \mathcal{M} . To further show the advantages of SPLS against others, we also compared SPLS/L0.5 and ITH/L0.5 on 25 problems with 512 signal length and different combinations of $M \in$ {210, 240, 270, 300, 330} and $k \in$ {110, 120, 130, 140, 150}. Each problem was tested by both SPLS/L0.5 and ITH/L0.5 for 100 times. The SRs found by these two algorithms are shown in Fig. 11. From this figure, the following observations can be made as follows.

- 1) Case 1 (M = 300 or M = 330): It can be observed from this figure that the SRs of SPLS are 100% for all five values of sparsity k. Note that the SRs of ITH/L0.5 are also 100%. Therefore, the performances of two algorithms have no difference when the number of observations is large enough.
- 2) Case 2 (M = 270): SPLS/L0.5 has 100% of SRs on all values of k except its smallest value 110.





Fig. 11. SRs of SPLS/L0.5 (top) and ITH/L0.5 (bottom) on 25 problems of $N = 512, M \in \{210, 240, 270, 300, 330\}$, and $k \in \{110, 120, 130, 140, 150\}$.

In contrast, ITH/L0.5 fails to obtain 100% of SRs on k = 110, 120, 130.

3) Case 3 (M = 210 or M = 240): The SRs of SPLS/L0.5 are less than 100% on all ten problems except M = 240 and k = 150. In contrast, the corresponding SRs of ITH/L0.5 are significantly worse than those of SPLS/L0.5 except three combinations of (M, k), i.e., (240, 110), (210, 110), and (210, 120), which are difficult to solve due the small values of (M, k) under the 512-length signal.

From the above analysis, the performance of SPLS/L0.5 is more robust than that of ITH/L0.5 when M is not sufficiently large.

As we have reported, the performance of sparse optimizations may have different performances on small scale problems and large scale problems. To study the performance of sparse optimization methods on the scalability of problem settings, we tested both SPLS/L0.5 and ITH/L0.5 on the 2, 4, 6, 8, and 10 times of P4 (512, 240, and 130). The SRs are shown in Fig. 12. It can be seen from this figure that the performance of SPLS/L0.5 get worse as the scale of P2 is increased to its four times. When the scale of P2 is further increased to its ten times, i.e., 5120-length signal, the SR of SPLS/L0.5 is changed from around 40%-96%. As reported before, the SR of SPLS/L0.5 on 20 times of P4, i.e., 10240-length signal, is 100%. In contrast, the performance of ITH/L0.5 becomes worse as the scale of P2 is increased. These results show that SPLS/L0.5 is more suitable than ITH/L0.5 on large scale problems.



Fig. 12. SRs of SPLS/L0.5 (top) and ITH/L0.5 (bottom) on 1, 2, 4, 6, 8, and 10 times of P4 (512, 240, 130).



Fig. 13. Plots of the nondominated solutions found by SPLS/L0.5 on P1 with noise level $\sigma = 0.01, 0.03$, and 0.1.

B. Artificial Sparse Signals With Noises

In many real applications, sparse signals often involve with noises. In this paper, we assume that the observation is mixed with Gaussian white noise, i.e., $\epsilon \sim N(0, \sigma^2)$. It is easy to deduce that the true sparse signal x^* does not satisfy the constraint y = Ax, i.e., $f_2(x^*) = ||y - Ax^*||_2^2 > 0$. In this case, all optimal solutions of Problem (5) are strictly Pareto optimal. Therefore, it is nontrivial to detect the exact position of k. Instead, the approximate value of k can be obtained if β is properly set. To deal with the instances with noise, we set $(\beta_0, \beta_1) = (1.0, 0.1)$. All other parameters are the same as those in the subsection IV-A. The experimental results are summarized as follows.

- 1) Fig. 13 shows the solutions found by SPLS/L0.5 on P1 with $\sigma = 0.01, 0.03$, and 0.1. From this figure, we can observe that all solutions have the sparsity values close to the true sparsity 130. For each noisy level, the maximal sparsity level among these solutions can be treated as a good approximation of the true sparse level. The related solution is also the approximation of the true sparse solution.
- 2) Fig. 14 shows the true sparse signals and the errors of the approximate solutions found by ITH/L0.5, SPLS/L0.5, and OMP on one instance of P1 and P4 with noise level $\sigma = 0.01$. It is evident that all three algorithms can reconstruct the sparse signals with very small errors on P1. For P4 with $\sigma = 0.01$, SPLS/L0.5 performs better than the other two algorithms.

optimal optimal 10 10 i filiti feler fiele 0 0 Statifier Seis Charl -10 -10 200 300 400 500 100 200 300 400 500 SPLS/L0.5 SPLS/L0.5 4 2 0 -2 4 2 0 -2 -4 400 100 200 300 500 100 200 300 400 500 TH/L0.5 ITH/L0.5 20-2 300 100 300 400 500 500 200 100 200 400 OMP OME 4 2 0 -2 renzi le li l -2 100 200 300 400 500 100 200 300 400 500

Fig. 14. Plots of the original optimal signal (on the top) and the errors of the approximate solutions found by SPLS/L0.5, ITH/L0.5, and OMP on P1 (left) and P4 (right) with a noise level of 0.01.



Fig. 15. SRs found by SPLS/L0.5 on P2 (top) and P4 (bottom) at different values of (β_0, β_1) .

Overall, SPLS/L0.5 can approximate sparse signals with noise well to some extent when the noise level is not very big.

C. Analysis of the Parameter Sensitivities in SPLS

The performance of SPLS algorithm mainly relies on the setting three major parameters β , *T*, and *ls*. In this section, we investigate the sensitivities of these parameters in SPLS/L0.5. Some experimental results are summarized as follows.

1) Fig. 15 shows the SRs found by SPLS/L0.5 on P2 and P4 with different combinations of (β_0, β_1) , where β_0 takes the value from {1, 3, 5, 10, 15, 20} and β_1 takes the value from {1, 0.1, 10⁻², 10⁻³, 10⁻⁴, 10⁻⁵, 10⁻⁶}. It can be seen from these results that SPLS/L0.5 with

IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS



Fig. 16. Plots of the distributions of solutions found by SPLS/L0.5 on P1 with $\sigma = 0.01$ at different values of β_1 .



Fig. 17. SRs of SPLS/L0.5 on 100 instances of P6 with different settings of T.

 $\beta_1 = 1.0$ fails to reconstruct the sparse signals in any of its runs on both P2 and P4. The poor performance of SPLS/L0.5 with a large value of β_1 can be explained by the fact that the solutions stored in E1 may have large violation on the constraints y = Ax. Moreover, it can also be observed that the performance of SPLS/L0.5 on SRs is very stable for any setting of β_0 in [1, 20].

- 2) Fig. 16 shows the nondominated solutions found by SPLS/L0.5 with different values of β_1 on P1 with noise $\sigma = 0.01$. From this figure, we can observe that the setting of β_1 has great influence on the position of nondominated solutions. When $\beta_1 = 10^{-1}$, SPLS/L0.5 finds the sparsity levels close to the true sparsity 130. SPLS/L0.5 with either too large value of β_1 or too small value of β_1 fails to approximate the true sparsity.
- 3) Fig. 17 shows the SRs of SPLS/L0.5 on P6 at different values of T. It is clear that SPLS/L0.5 has the best performance when T = 5. The performance of SPLS/L0.5 with T = 1 or T = 40 is worst in terms of SR. In the case of T = 1, the population of SPLS will lose diversity, and the solutions move toward the true sparsity 130 very slowly. On the other hand, the solutions will move toward the right side of the true sparsity very quickly when T is set to 40. In either case,

LI et al.: PREFERENCE-BASED MULTIOBJECTIVE EVOLUTIONARY APPROACH FOR SPARSE OPTIMIZATION



SRs of SPLS/L0.5 on 100 instances of P6 with different settings Fig. 18. of ls.

SPLS/L0.5 fails to find the sparsity levels close to the true sparsity.

4) Fig. 18 shows the results of SPLS/L0.5 with different settings of ls on P6. It is evident from this figure that SPLS/L0.5 with ls = 20 or ls = 40 has the best performance in terms of SRs. We can also observe that SPLS/L0.5 with ls = 40 performs better than the one with ls = 20 at the MSE levels larger than 10^{-15} , but worse at the MSE levels less than 10^{-15} . This can be explained by the fact that the large value of *ls* for the thresholding search on each sparsity level k'' will decrease the computational cost on searching the knee part when the total number of thresholding iterations is limited.

The setting of the minimal gap δ between two neighboring sparsity levels is often set to 1 for reconstruct the shortlength sparse signals. To speed up the convergence toward the knee part, a large value of δ is very helpful. This has been verified by the experimental results of SPLS for reconstructing the long-length signals. Moreover, the total number of iterations can be reduced in a successful run if the search is terminated when the population is well converged. In fact, if SPLS/L0.5 is terminated when x' is not improved by Algorithm3-ThresholdingSearch in five consecutive iterations, the total number of iterations will be decreased from 3000 to the number in [1000, 1500].

V. APPLICATION OF SPLS IN MRI IMAGE SIGNAL RECONSTRUCTION

The sparse reconstruction of MRI images is one of the most important applications of compressive sensing [50]. Most previous work on sparse optimization for MRI images considered the use of ITH methods. In this paper, we studied the extension of SPLS for sparse reconstruction on MRI images, where the sampling trajectory with radial lines is used. In the following, we state the formulation of L0 problem for MRI images. Assume that the following holds.

- 1) $X \in \mathbb{R}^{d \times d}$ represents a 2-D image.
- 2) $W \in \mathbb{R}^{d \times d}$ is a sparse transformation, and W^{-1} is its inverse transformation.



Fig. 19. Plots $(||Z||_0 \text{ versus } ||Y - A(Z)||^2)$ of the nondominated solutions found by SPLS/L0.5 on the first MRI image.



Fig. 20. Comparison between the images recovered by SPLS/L0.5 and ITH/L0.5 with 60 radical lines for observations.

- 3) F_T ∈ R^{d×d} is a Fourier transformation matrix.
 4) S ∈ R^{d×d} is a sampling matrix. The sampling trajectory with radial lines is used in this paper.

The L0 regularization model for sparse image recovery can be written as

$$\min_{Z \in \mathbb{R}^{d \times d}} \|Y - A(Z)\|_F^2 + \lambda \|Z\|_0$$
(12)

where Z is the representation of X under sparse transformation W, i.e., Z = WX. The mapping $A : \mathbb{R}^{d \times d} \to \mathbb{R}^{d \times d}$ is defined by: $[A(Z)]_{ij} = S_{ij} \times [F_T W^{-1} Z]_{ij}; Y \in \mathbb{R}^{d \times d}$ is an observation matrix in frequency domain.

In our experiments, five 128×128 grayscaling images from the benchmark MRI library shown in the left of Fig. 20 are

IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS

considered. The sparsity of all MRI images is considered under the wavelet transformation with Haar basis. The number of nonzero components in the representation Z of five images are 5716, 6294, 7059, 7422, and 6885, respectively. Note the minimum and the maximum of sparsity ratio (k/N) in all five MRI images ranged from (5716/16384) = 0.349to (7422/16384) = 0.453. This means that all MRI images are not very sparse. To make them more sparse, we carry out the following three steps: 1) the sparse transformation Wis applied to convert a given image X into Z; 2) a sparse solution Z' is obtained by keeping k' largest (in absolute value) components of Z unchanged and setting all others as zeros; and 3) the inverse transformation W^{-1} is used to convert Z' into a new image X' with the sparsity level k'. In this way, we obtained the modified versions of five MRI images with sparsity values-3612, 3397, 3414, 3502, and 3468 by truncating the smallest components of the original MRI images. Then, the sparsity ratios of the modified MRI images range from 0.183 to 0.224.

According to our experimental analysis in Section IV, SPLS/L0.5 and ITH/L0.5 are two best algorithms for reconstructing artificial sparse signals. Therefore, we only compared them in our experiments for MRI imaging reconstruction. Similar to the use of ITH/L0.5 for artificial signals, the version of ITH/L0.5 for MRI image suggested in [17] is used as local search in SPLS. The estimation interval of sparsity in SPLS/L0.5 is set to $[0.5 \times k, 2 \times k]$, i.e., $k_{\min} = 0.5 \times k$ and $k_{\text{max}} = 2 \times k$. The number of radial lines used in sparsity sampling is set to 60 in both SPLS/0.5 and ITH/L0.5. All other parameters of SPLS/L0.5 are the same as those used for artificial signals. Our experimental results show that the detected sparsity values by SPLS/L0.5 on five MRI images are 3330, 3021, 3059, 3180, and 3194, which are the number of nonzero components with the absolute values larger than 10^{-10} . In Fig. 19, the distribution of the weakly Pareto-optimal solutions found by SPLS/L0.5 on the first MRI image is plotted. This result indicates that SPLS/L0.5 can detect the majority of large components in the sparse MRI images.

The recovered MRI images corresponding to the solutions at the knee position of the weakly Pareto front found by SPLS/L0.5 and ITH/L0.5 with 60 radical lines are shown in Fig. 20. From this figure, it is easy to see that all images recovered by ITH/L0.5 have big errors to the corresponding original MRI images, while SPLS/L0.5 is still able to reconstruct high-quality MRI images for images 2–5. Unlike ITH/L0.5, SPLS/L0.5 is free to set the regularization parameter for MRI imaging reconstruction, which needs to estimate the true sparsity value of MRI imaging signals.

VI. CONCLUSION

The majority of sparse optimization algorithms reconstruct sparse signals by means of regularization. The setting of the regularization parameter and the estimation of the true sparsity are two difficult tasks in these algorithms. In this paper, we proposed a sparse preference-based multiobjective evolutionary approach, i.e., SPLS, for signal reconstruction, which use the preference information of the *k*-sparse solution to guide the search toward the knee part of the weakly Paretooptimal front. The existing ITH methods are integrated as local optimizers. A new truncation strategy based on multiple sparsity levels was suggested to increase the diversity of the population. One major advantage of SPLS is that it does not need to set the regularization parameter. Our experimental results showed that SPLS is superior to other ITH methods, OMP, as well as two MOEAs, i.e., MOEA/D and StEMO, in terms of SRs on the artificial sparse signals with or without noises. Particularly, SPLS performs much better than other sparse optimizers when the length of signals is long or the number of observations is relatively low. Compared with the ITH methods with cross validation, the SPLS algorithm consumed less computational time to find the solution in the knee part. Furthermore, the SPLS algorithm was also successfully applied to reconstruct MRI images with sparsity under wavelet basis. The majority of large components in the signals of five MRI images were detected by the SPLS algorithm with success. Our future work will focus on the extension of the SPLS algorithm to solve more challenging sparse optimization problems in various applications.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions on the original manuscript.

REFERENCES

- H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, Jun. 2006.
- [2] R. Tibshirani, "Regression shrinkage and selection via the lasso," J. Roy. Statist. Soc., B (Methodol.), vol. 58, no. 1, pp. 267–288, 1996.
- [3] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.
- [4] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.
- [5] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. New York, NY, USA: Springer, 2010.
- [6] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [7] B. K. Natraajan, "Sparse approximation to linear systems," SIAM J. Comput., vol. 24, no. 2, pp. 227–234, 1995.
- [8] S. G. Mallat and Z. Zhang, "Matching pursuits with timefrequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [9] J. Tropp and A. C. Gilbert, "Signal recovery from partial information via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
- [10] K. Bredies and D. Lorenz, "Iterated hard shrinkage for minimization problems with sparsity constraints," *SIAM J. Sci. Comput.*, vol. 30, no. 2, pp. 657–683, Jan. 2008.
- [11] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, Nov. 2009.
- [12] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 255–261, Jan. 2006.
- [13] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 298–309, Apr. 2010.
- [14] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.
- [15] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Appl. Comput. Harmon. Anal.*, vol. 10, no. 3, pp. 234–253, May 2001.

- [16] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
- [17] Z. Xu, X. Chang, F. Xu, and H. Zhang, "L_{1/2} regularization: A thresholding representation theory and a fast solver," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1013–1027, Jul. 2012.
- [18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York, NY, USA: Springer, 2001.
- [19] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," Annu. Statst., vol. 35, no. 6, pp. 2313–2351, 2007.
- [20] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. London, U.K.: Chapman & Hall, 2015.
- [21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [22] M. A. Lukas, "Comparisons of parameter choice methods for regularization with discrete noisy data," *Inverse Problems*, vol. 14, no. 1, p. 161, 1998.
- [23] M. E. Kilmer and D. P. O'Leary, "Choosing regularization parameters in iterative methods for ill-posed problems," *SIAM J. Matrix Anal. Appl.*, vol. 22, no. 4, pp. 1204–1221, 2001.
- [24] F. Bauer and M. A. Lukas, "Comparingparameter choice methods for regularization of ill-posed problems," *Math. Comput. Simul.*, vol. 81, no. 9, pp. 1795–1841, 2011.
- [25] A.-K. Seghouane, "Model selection criteria for image restoration," *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1357–1363, Aug. 2009.
- [26] H. Chen and X. Yao, "Regularized negative correlation learning for neural network ensembles," *IEEE Trans. Neural Netw.*, vol. 20, no. 12, pp. 1962–1979, Dec. 2009.
- [27] C.-S. Leung and J. P.-F. Sum, "RBF networks under the concurrent fault situation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1148–1155, Jul. 2012.
- [28] C. G. Sentelle, G. C. Anagnostopoulos, and M. Georgiopoulos, "A simple method for solving the svm regularization path for semidefinite kernels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 709–722, Apr. 2015.
- [29] Z. Dong and W. Zhu, "Homotopy methods based on 10-norm for compressed sensing," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2017.2658953
- [30] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proc. 2nd Int. Symp. Inf. Theory*, 1973, pp. 267–281.
- [31] G. Schwarz, "Estimating the dimension of a model," Ann. Statist., vol. 6, no. 2, pp. 461–464, 1978.
- [32] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 260–270, Aug. 1995.
- [33] H. Ishibuchi, T. Murata, and I. B. Türkşen, "Single-objective and twoobjective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy Sets Syst.*, vol. 89, no. 2, pp. 135–150, 1997.
- [34] X. Llorà and D. E. Goldberg, "Bounding the effect of noise in multiobjective learning classifier systems," *Evol. Comput.*, vol. 11, no. 3, pp. 279–298, 2003.
- [35] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 1, pp. 45–65, Feb. 2013.
- [36] H. Li, X. L. Su, Z. B. Xu, and Q. Zhang, "MOEA/D with iterative thresholding algorithm for sparse optimization problems," in *Proc. 12th Int. Conf. Parallel Problem Solving Nature (PPSN)*, 2012, pp. 93–101.
- [37] L. Li, X. Yao, R. Stolkin, M. G. Gong, and S. He, "An evolutionary multiobjective approach to sparse reconstruction," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 827–845, Dec. 2014.
- [38] G. Greenwood, X. Hu, and J. D'Ambrosio, "Fitness functions for multiple objective optimization problems: Combining preferences with Pareto rankings," in *Proc. Found. Genetic Algorithms*, 1997, pp. 437–455.
- [39] J. Branke and K. Deb, "Integrating user preferences into evolutionary multi-objective optimization," in *Knowledge Incorporation in Evolutionary Computation* (Studies in Fuzziness and Soft Computing), vol. 167. Springer, 2005, pp. 461–477.

- [40] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multiobjective optimization," *Evol. Comput.*, vol. 17, no. 3, pp. 411–436, 2009.
- [41] A. López-Jaimes and C. A. C. Coello, "Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems," *Inf. Sci.*, vol. 277, pp. 1–20, Sep. 2014.
- [42] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [43] J. Zeng, Z. Xu, B. Zhang, W. Hong, and Y. Wu, "Accelerated L_{1/2} regularization based SAR imaging via BCR and reduced Newton skills," *Signal Process.*, vol. 93, no. 7, pp. 1831–1844, 2013.
- [44] W. Cao, J. Sun, and Z. B. Xu, "Fast image deconvolution using closed-form thresholding formulas of Lq(q=12,23) regularization," J. Vis. Commun. Image Represent., vol. 24, no. 1, pp. 31–41, 2013.
- [45] C. Liu et al., "The L_{1/2} regularization method for variable selection in the Cox model," Appl. Soft Comput., vol. 14, pp. 498–503, Jun. 2014.
- [46] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1998.
- [47] C. A. C. Coello, "Constraint-handling using an evolutionary multiobjective techniques," *Civil Eng. Environ. Syst.*, vol. 17, no. 4, pp. 319–346, 2000.
- [48] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 658–675, Dec. 2006.
- [49] S. Venkatraman and G. G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 9, no. 4, pp. 424–435, Aug. 2005.
- [50] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.



Hui Li (M'16–SM'17) received the B.Sc. and M.Sc. degrees in applied mathematics from the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China, in 1999 and 2002, respectively, and the Ph.D. in computer science from the University of Essex, Colchester, U.K., in 2008.

From 2007 to 2010, he was a post-Doctoral Research Associate with the School of Computer Science, University of Nottingham, Nottingham, U.K. He is currently an Associate Professor with the School of Mathematics and Statistics, Xi'an Jiaotong

University. His current research interests include evolutionary computation, multiobjective optimization, and machine learning.

Dr. Li was a recipient of the 2010 IEEE TRANSACTIONS ON EVOLU-TIONARY COMPUTATION Outstanding Paper Award as one of the inventors for MOEA/D.



Qingfu Zhang (M'01–SM'06–F'17) received the B.Sc. degree in mathematics from Shanxi University, Taiyuan, China, in 1984, the M.Sc. degree in applied mathematics and the Ph.D. degree in information engineering from Xidian University, Xi'an, China, in 1991 and 1994, respectively.

He is currently a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong, and a Changjiang Visiting Chair Professor with Xidian University, China. He is also leading the Metaheuristic Optimization Research Group,

City University of Hong Kong. MOEA/D, a multiobjective optimization algorithmic framework, developed in his group, is one of the most widely used and researched multiobjective evolutionary algorithms. His current research interests include evolutionary computation, optimization, neural networks, data analysis, and their applications.

Dr. Zhang is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the *IEEE Transactions on Cybernetics*. He is also an Editorial Board Member of three other international journals. He was a recipient of the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award. He was a 2016 Web of Science highly cited researcher in computer science.



Jingda Deng received the B.Sc. and M.Sc. degree from the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree in computer science, City University of Hong Kong, Hong Kong.

His current research interests include evolutionary computation and multiobjective optimization.



Zong-Ben Xu received the Ph.D. degree in mathematics from Xi'an Jiaotong University, Xi'an, China, in 1987.

In 1998, he was a Post-Doctoral Researcher with the Department of Mathematics, University of Strathclyde, Glasgow, U.K. He was a Research Fellow with the Department of Computer Science and Engineering from 1992 to 1994 and from 1996 to 1997 with The Chinese University of Hong Kong. He was a Visiting Professor with the University of Essex, Colchester, U.K., and University of Naples

Federico II, Naples, Italy, in 2001 and 2002, respectively. He has been with the School of Mathematics and Statistics at Xi'an Jiaotong University since 1982, where he was the Department Head, the Dean, and the VP. He currently serves as Director of the National Key Laboratory of Big Data Analytics, Xi'an Jiaotong University. He has been appointed as a Chief Scientist of the National Basic Research Program of China (973 Project). He has authored over 270 academic papers on nonlinear analysis, sparse modeling, machine learning, optimization and information processing, most of which are in international journals. His current research interests include big data analytics and machine learning.

Dr. Xu holds the title Owner of Chinese Ph.D. Degree Having Outstanding Achievements received from the Chinese State Education Commission and the Academic Degree Commission of the Chinese Council in 1991. He was a recipient of the National Natural Science Award of China in 2007, and the CSIAM Su Buchin Applied Mathematics Prize in 2008. He was an invited speaker (45 min) in the 26th International Congress of Mathematicians in 2010.