MOEA/D with Iterative Thresholding Algorithm for Sparse Optimization Problems

Hui Li[1], Xiaolei Su[1], Zongben Xu[1], and Qingfu Zhang[2]

[1] Institute for Information and System Sciences & Ministry of Education Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, Shaanxi,710049, China
{lihui10,zbxu}@mail.xjtu.edu.cn, suxl062641@stu.xjtu.edu.cn

[2] School of Computer Science & Electronic Engineering,University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK
qzhang@essex.ac.uk

Abstract. Currently, a majority of existing algorithms for sparse optimization problems are based on regularization framework. The main goal of these algorithms is to recover a sparse solution with k non-zero components(called k-sparse). In fact, the sparse optimization problem can also be regarded as a multi-objective optimization problem, which considers the minimization of two objectives (i.e., loss term and penalty term). In this paper, we proposed a revised version of MOEA/D based on iterative thresholding algorithm for sparse optimization. It only aims at finding a local part of trade-off solutions, which should include the k-sparse solution. Some experiments were conducted to verify the effectiveness of MOEA/D for sparse signal recovery in compressive sensing. Our experimental results showed that MOEA/D is capable of identifying the sparsity degree without prior sparsity information.

Keywords: sparse optimization, multi-objective optimization, hard/half thresholding algorithm, evolutionary algorithm.

1 Introduction

Compressive sensing (CS) is a novel sampling theory for reconstructing sparse signals or images from incomplete information. In recent years, it has found numerous applications, such as signal recovery, image processing as well as medical imaging [1]. A fundamental problem in CS is to find a sparse solution for underdetermined linear systems, which generally have infinite number of solutions. A sparse solution is often defined as the one with the minimal number of nonzero components among all solutions. Finding sparse solution involves the following NP-hard sparse optimization problem [2]:

$$\min \|x\|_0, \quad \text{s.t. } Ax = y \tag{1}$$

where $x \in R^N$ is a N-dimensional signal vector, A is a $M \times N$ measurement matrix with $M \ll N$, $y \in R^M$ is a measurement vector, and $\|x\|_0$ stands for the number of nonzero components of x.

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part II, LNCS 7492, pp. 93–101, 2012.

In the area of sparse optimization, greedy strategies and regularization methods are two commonly-used methods for finding sparse solutions [3–6]. The well-known greedy methods include matching pursuit (MP) [3] and orthogonal matching pursuit (OMP) [5]. In both algorithms, a k-sparse solution is iteratively constructed component by component in a greedy manner until k nonzero components are determined. Greedy strategies only provide approximate solutions for sparse optimization problems. In contrast, sparse optimization methods based on regularization frameworks, such as ℓ_0, ℓ_1 and $\ell_{0.5}$ [7], are more efficient to recover k-sparse solutions since they can recover the exact sparse solution. Iterative hard thresholding algorithm (iHardT) [8] and iterative half thresholding algorithm (iHalfT) [7] are two representative thresholding algorithms based on regularization frameworks.

The main difficulty in previous sparse optimization methods lies in the fact that the sparsity degree k is unknown in many real applications. To overcome this problem, an estimate of sparsity degree is often used both in greedy methods and in thresholding algorithms. In fact, a sparse optimization problem can also be modeled as a multi-objective optimization problem. Two conflicting objectives - the loss term ($\|Ax - y\|$) and the penalty term ($\|x\|_0$) should be minimized simultaneously. The solutions balancing both objectives are called trade-offs. In both greedy methods and regularization methods, the main goal is only to find one k-sparse solution, which belongs to the set of trade-off solutions for the multi-objective optimization problem. So far, not much work has been done for solving sparse optimization problems by multi-objective methods.

Since the early 1990s, evolutionary multi-objective algorithms (MOEAs) have received a lot of research interests [9]. The well-known representatives are NSGA-II [10] (Pareto-based), MOEA/D [11](decomposition-based), and IBEA [12](indicator-based). The main advantages of MOEAs lie in (i) the ability of finding multiple trade-off solutions with even spread in a single run, and (ii) the high possibility of finding global optima. In this work, we proposed a revised version of MOEA/D with thresholding algorithm for sparse optimization. In the proposed algorithm, the sparse multi-objective optimization problem is decomposed into multiple single objective subproblems. Each subproblem is associated with one sparsity level and a trade-off solution. It is optimized by existing thresholding algorithms in each generation. Moreover, the sparsity levels of subproblems are adaptively changed during the search. In our experiments, we tested the performance of the revised MOEA/D for sparse signal recovery in CS.

The remainder of this paper is organized as follows. In Section 2, the sparse optimization problem in CS is introduced. Section 3 gives an overview on two well-known iterative thresholding algorithms. MOEA/D with iterative thresholding algorithm for sparse optimization is presented in Section 4. The experimental results are reported and analyzed in Section 5. The final section concludes the paper.

2 Sparse Multi-objective Optimization

2.1 Sparse Optimization

A general sparse optimization problem in the CS can be formulated as the following bi-objective optimization problem

$$\min\{\|y - Ax\|^2, J(x)\} \tag{2}$$

where $\|y - Ax\|^2$ is the loss function. $J(x)$ is the penalty term for sparsity. The typical examples of $J(x)$ are $\|x\|_0$, $\|x\|_1 = \sum_{i=1}^N |x_i|$ and $\|x\|_{0.5}^{0.5} = (\sum_{i=1}^N |x_i|^{\frac{1}{2}})^2$, which correspond to three well-known regularization frameworks, denoted by ℓ_0, ℓ_1, and $\ell_{0.5}$, respectively.

Over the last a few years, thresholding algorithms based on regularization have been widely used in sparse optimization. In these algorithms, a regularization optimization problem is obtained by combining the loss function and the sparsity function of (2) in a linear manner:

$$\min \|y - Ax\|^2 + \lambda J(x) \tag{3}$$

where A and y are the same as above. $\lambda > 0$ is the regularization parameter, which is very sensitive to the performance of thresholding algorithms. The larger the value of λ, the solution of (3) is more sparse.

Among the aforementioned regularization frameworks, the solutions of ℓ_0 regularization problem are sparsest. But ℓ_0 regularization problem is difficult to solve since it is a NP-hard combinatorial optimization problem. To overcome this difficulty, ℓ_1 regularization, the relaxation of ℓ_0 regularization, was suggested [13]. Since ℓ_1 regularization problem is a convex quadratic optimization problem, there exist efficient algorithms for sparse solutions. $\ell_{0.5}$ regularization, a special case of $\ell_q(0 < q < 1)$, is the other popular framework for sparse optimization, which allows fast method for sparse solutions as they can be analytically expressed. Compared with ℓ_1 regularization, $\ell_{0.5}$ thresholding algorithms need less measurements to recover sparse signals, but it is more difficult to solve .

2.2 Pareto Optimality

As shown in (2), a sparse optimization problem is in nature a bi-objective optimization problem, which should have many trade-off solutions. In this work, we focus on the following bi-objective sparse optimization problem:

$$\min_{x \in R^N} \{(f_1(x), f_2(x))\} \tag{4}$$

where $f_1(x) = \|x\|_0$ and $f_2(x) = \|y - Ax\|^2$. A and y are the same as in (1).

In the context of multi-objective optimization, the optimality of solutions is defined in terms of *Pareto dominance*. For any two solutions $x^{(1)}$ and $x^{(2)}$ in R^N, $x^{(1)}$ is said to *dominate* $x^{(2)}$ if and only if $f_i(x^{(1)}) \leq f_i(x^{(2)})$ for all $i \in \{1, 2\}$, and there exists at least one index $j \in \{1, 2\}$ such that $f_j(x^{(1)}) < f_j(x^{(2)})$. A solution

x^* is said to be *Pareto-optimal* if there doesn't exist such a solution in R^N which dominates x^*. The set of all Pareto-optimal solutions in the objective space is called Pareto-optimal front. A solution x^* is said to be *weakly Pareto-optimal* if no solution in R^N is strictly better than x^* regarding all objectives.

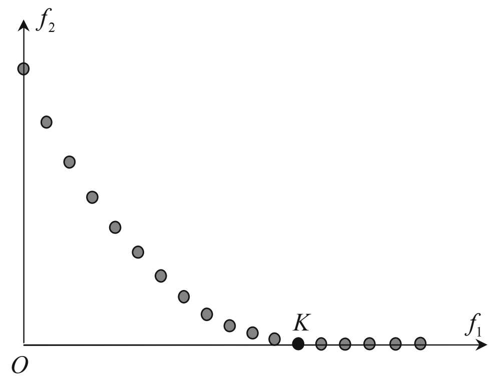


Fig. 1. Weakly Pareto-optimal solutions in sparse optimization

Fig. 1 illustrates the distribution of weakly Pareto-optimal solutions in the sparse optimization problem (4). Note that the number of these solutions is finite because the first objective $\|x\|_0$ takes integer numbers in $[0, N]$. Note that part of trade-off solutions are not Pareto-optimal but weakly Pareto-optimal. For example, all points on the right side of point K in Fig. 1 are only weakly Pareto-optimal. In many existing sparse optimization methods, the goal is to find the 'knee' Pareto-optimal solution K (k-sparse) with $y = Ax$. Unfortunately, the value of sparsity k is usually unknown.

3 Iterative Thresholding Algorithms

In this section, we briefly introduce two efficient iterative thresholding algorithms for sparse optimization - iterative hard thresholding algorithm (iHardT) [14] and iterative half thresholding algorithm [7] (iHalfT), which are based on ℓ_0 regularization and $\ell_{0.5}$ regularization respectively.

– In iHardT, an iterative procedure is performed as follows:

$$x^{(n+1)} = H_k(x^{(n)} + A^T(y - Ax^{(n)})) \tag{5}$$

where $H_k(\cdot)$ is a nonlinear thresholding operator that retains the largest k components of a vector in magnitude and sets others as zeros. Note that $A^T(Ax^{(n)} - y)$ is actually the gradient vector of $\|Ax - y\|^2$.

– In iHalfT, the important components of a vector are retained by a more complex rule as follows:

$$x^{(n+1)} = H_{\lambda,\mu,0.5}(x^{(n)} - \mu A^T(y - Ax^{(n)})) \tag{6}$$

where $H_{\lambda,\mu,0.5}(x) = (\rho_{\lambda,\mu,0.5}(x_1)), \rho_{\lambda,\mu,0.5}(x_2), \cdots, \rho_{\lambda,\mu,0.5}(x_N)))$ is the thresholding operator. The details of $\rho_{\lambda,\mu,1/2}(\cdot)$ is referred to the literature [7].

4 MOEA/D for Sparse Optimization

4.1 Motivations

In most existing regularization methods, the bi-objective optimization problem (2) is often converted into a single objective regularization optimization problem, which could be NP-hard, or have many local optimal solutions. Due to these characteristics, these methods may suffer from being local optimizers and having sensitivity to regularization parameter, and replying on the estimation of sparsity degree. To determine the sparsity degree, regularization methods should be repeatedly applied to solve regularization problems with various sparsity levels. The set of all resultant solutions should be weakly Pareto-optimal and may contain the 'knee' Pareto-optimal solution shown in Fig. 1.

Since sparse optimization problem is a multi-objective optimization problem, the use of MOEAs to find multiple weakly Pareto-optimal solutions should be a straightforward choice. Among all MOEAs, a recent popular algorithm - MOEA/D is very suited for multi-objective sparse optimization problem due to the fitness assignment based on decomposition. The main idea in MOEA/D is to optimize multiple subproblems. Each subproblem is associated with one Pareto-optimal solution. For sparse optimization, the objective functions of subproblems can be defined by:

$$\begin{aligned} &\min g^{(s)}(x) = \|y - Ax\|^2, \\ &\text{s.t.} \quad \|x\|_0 = s, \end{aligned} \tag{7}$$

where the sparsity level s ranges from 0 to N. Note that only the optimal solutions of $g^{(s)}$ with sparsity level s close to sparsity degree k are preferred.

In this work, we suggested a revised version of MOEA/D based on thresholding algorithms to find part of weakly Pareto-optimal solutions near the 'knee' solution. Unlike previous variants of MOEA/D, this version evolves all subproblems such that the corresponding solutions are in the neighborhood of the preferred 'knee' solution. The detail of MOEA/D is described in the following subsection.

4.2 MOEA/D with Thresholding Algorithm

In MOEA/D, a set of *pop* subproblems are defined by $g^{(s_i)}(x), i = 1, \ldots, pop$, where s_i is an integer number between an estimation interval $[s_{min}, s_{max}]$ including sparsity degree k. For each subproblem $g^{(s_i)}(x)$, a solution $x^{(i)}$ is associated and maintained. The general framework of MOEA/D for sparse optimization is outlined in Algorithm 1. The following are the detailed illustrations for the major steps in MOEA/D.

Algorithm 1. MOEA/D for Sparse Optimization

Input *pop*- population size, $\#ls$- maximal no. of steps in local search
Output P - sparse solutions, S- sparsity levels
Step 1: Initialization
    initialize $S = \{s_1, \ldots, s_{pop}\}$ and $P = \{x^{(1)}, \ldots, x^{(pop)}\}$ randomly.
Step 2: Variation and Local Search
for all $i \in \{1, \ldots, pop\}$ **do**
    perturb $x^{(i)}$ via mutation;
    apply iterative thresholding algorithm to improve $x^{(i)}$ w.r.t sparsity level s_i.
end for
Step 3: Non-dominated Sorting
    determine all non-dominated solutions in P and save them into Q.
Step 4: Sparsity Update
    Step 4.1 sort S in an increasing order, i.e., $s_{i_1} < s_{i_2} < \ldots < s_{i_{pop}}$.
    Step 4.2 determine the set of new potential sparsity levels $\tilde{S} = \{\lfloor s_{i_j} + 0.5(s_{i_{j+1}} - s_{i_j})\rfloor | s_{i_{j+1}} - s_{i_j} \geq 2, j = 1, \ldots, pop - 1\}$
    Step 4.3 insert sparsity level in $\tilde{S}$ if $|\tilde{S}| \neq 0$:
        Step 4.3.1 $s_{i_1} \leftarrow$ a level in $\tilde{S}$ randomly if $|Q| > 0.5pop$.
        Step 4.3.2 $s_{i_{pop}} \leftarrow$ a level in $\tilde{S}$ randomly if $|Q| \leq 0.5pop$.
    Step 4.4 offset sparsity level if $|\tilde{S}| = 0$:
        Step 4.4.1 $s_{i_1} \leftarrow \min\{s_{i_{pop}} + 0.5pop, s_{max}\}$ if $|Q| = pop$.
        Step 4.4.2 $s_{i_{pop}} \leftarrow \max\{s_{i_1} - 0.5pop, s_{min}\}$ if $|Q| = 1$.
    Step 4.5 replace $x^{(i_1)}$ or $x^{(i_{pop})}$ by one solution in the half best in P if updated.
Step 5: Stopping Criteria
    If stopping criteria is fulfilled, then output P and S; otherwise go to **Step 2**.

- In **Step 2**, each solution $x^{(i)}$ is first perturbed by a mutation operator, and then improved by a local search procedure, i.e., iterative thresholding algorithm. In this work, we use either iHardT in (5) or iHalfT in (6) for this purpose. The parameter $\#ls$ is used to control the maximal number of iterations allowed in each step of local search. $x^{(i)}$ is used as the starting solution and then updated by the improved solution obtained in the previous step. To perturb the starting solution, one of its non-zero component is set to zero randomly. Then, we use the greedy constructive strategy in OMP to complete the solution until s_i non-zero components are determined.
- In **Step 3**, the set of all non-dominated solutions in P are determined and saved in Q. Note that weakly Pareto-optimal solutions for sparse optimization problem are excluded in Q. The size of Q will tell us if the sparsity k is among S.
- **Step 4** is the core step of MOEA/D for sparse optimization, which adaptively updates the set S of *pop* sparsity levels. **Step 4.1** and **Step 4.2** determine the set $\tilde{S}$ of candidate sparsity levels. Each of them is located in the middle of adjacent two sparsity levels in S.
- In **Step 4.3** and **Step 4.4**, the minimum s_{i_1} or the maximum $s_{i_{pop}}$ among all sparsity levels in S are updated by the candidate sparsity levels randomly

chosen from $\tilde{S}$. As the search progresses, subproblems with *pop* consecutive sparsity levels are expected to obtain.

- In **Step 4.4**, we also increase the maximal sparsity level $s_{i_{pop}}$ or decrease the minimal sparsity level s_{i_1} since all sparsity levels in S may locate on one side of the sparsity degree k. If all members of S are on the left side of k, then we need to increase the maximal sparsity level. In this case, all members of population are non-dominated. Otherwise, the minimal sparsity level should be decreased if only one is non-dominated.
- **Step 4.5** updates the solutions of the selected candidate sparsity level by the member of P with the better value of f_2.

5 Computational Experiments

5.1 Experimental Settings

In our experiments, we considered to recover noiseless real-valued signals. The elements in sensor matrix $A_{M\times N}$ and a k-sparse signal x^* are randomly generated. MOEA/D was tested on four small instances with the length of signal 512 and sparsity degree 130, and four large instances the length of signal 1024 and sparsity degree 130. All instances are named by N-M-k. The initial range of sparsity degree is assumed to be $[50, 250]$. For all instances, *pop* in MOEA/D is set to 10. In local search, $\#ls$ used in iHardT or iHalfT is set to 20. The total number of iterations is set to 10000 for the small instances and 20000 for the large instances. MOEA/D was implemented by C++ and tested on the operating system Windows XP with Intel Quad CPU 2.66 GHz.

5.2 Experimental Results

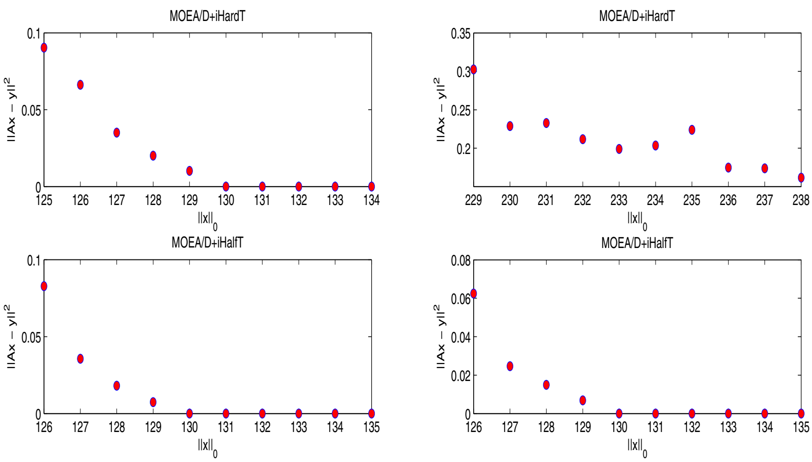
Table 1 summarizes the average values of mean square error (MSE) between the sparse signals and the recovered signals found by MOEA/D with two thresholding algorithms in 20 runs. The comparison of MOEA/D with iHardT and iHalfT was also provided. From these results, we can see that MOEA/D with both thresholding algorithms can find 130-sparse solutions for the first three

Table 1. Comparison of MOEA/D with iHardT and iHalfT for 8 instances with sparsity level 130 in terms of average mse to the true sparse solution

Instance	MOEA/D+iHardT	MOEA/D+iHalfT	iHardT	iHalfT
512-350-130	3.23255e-014	3.98776e-014	3.17919e-014	3.74779e-014
512-320-130	4.82847e-014	4.83276e-014	4.39184e-014	4.21504e-014
512-290-130	4.79143e-014	5.60058e-014	3.25566e-014	4.80787e-014
512-260-130	N/A	7.26465e-014	N/A	N/A
1024-500-130	5.81117e-014	6.25171e-014	6.53619e-014	5.54127e-014
1024-400-130	8.05214e-014	8.43726e-014	8.79100e-014	8.54657e-014
1024-350-130	7.95822e-014	8.87359e-014	N/A	6.20355e-014
1024-300-130	N/A	N/A	N/A	N/A

instances since the MSE values of the solutions found by MOEA/D are quite small. This indicates the obtained solutions are very close to the sparse solutions (less than 10^{-13}). However, for the instance 512-260-130, all four algorithms except iHalfT failed to find the 130-sparse solution. When M is small, the sparse optimization problem becomes too difficult to solve. This also happened for the instance 1024-300-130, where all four algorithms failed to find 130-sparse solution. Overall, MOEA/D with iHalfT works better than MOEA/D with iHardT.

Fig. 5.2 plots the weakly solutions found by MOEA/D with two thresholding algorithms for two instances 512-350-130 and 512-260-130 in one of 20 runs. It can be seen from Fig. 5.2 (a) that the 130-sparse solution is also the 'knee' solution along the weakly Pareto front for the instance 512-350-130. Fig. 5.2(b) shows that MOEA/D with iHalfT still found that 130-sparse solution for the instance 512-260-130 while MOEA/D with iHardT failed. This indicates that iHalfT is superior to iHardT in MOEA/D for the instances with less measurements. This observation is also consistent with the results in Table 1.



.

(a) (b)

Fig. 2. Weakly Pareto-optimal solutions found by MOEA/D for the instance 512-350-130 (a) and the instance 512-260-130 (b).

6 Conclusions

In this work, we suggested a revised version of MOEA/D for sparse signal recovery in CS. It attempts to find a local part of Pareto front, which should include the k-sparse solution. Our experimental results showed that MOEA/D with both iHardT and iHalfT is effective for sparse optimization without prior sparsity information. In the future work, we plan to apply the proposed algorithm to deal with nonlinear sparse optimization problems.

Acknowledgment. The authors would like to thank the anonymous reviewers for their insightful comments. This work was supported by National Natural Science Foundation of China (NSFC) grants 61175063 and 11131006 .

References

1. Donoho, D.: Compressed sensing. IEEE Trans. on Information Theory 52, 1289–1306 (2006)
2. Natarajan, B.K.: Sparse Approximate Solutions to Linear Systems. SIAM Journal on Computing 24, 227–234 (1995)
3. Mallat, S., Zhang, Z.: Matching pursuits with time-frequency dictionaries. IEEE Trans. on Signal Processing 41, 3397–3415 (1993)
4. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Annals of Statistics 23, 407–499 (2004)
5. Tropp, J., Gilbert, A.: Signal recovery from partial information via orthogonal matching pursuit. IEEE Trans. on Information Theory 53, 4655–4666 (2006)
6. Candés, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure Applied Mathematics 59, 1207–1223 (2006)
7. Xu, Z., Chang, X., Xu, F., Zhang, H.: L1/2 Regularization: A Thresholding Representation Theory and A Fast Solver. Technical report, Xi'an Jiaotong University (2010)
8. Bredies, K., Lorenz, D.: Iterated hard shrikage for minimization problems with sparsity constraints. SIAM Journal of Scientific Computing 30, 657–683 (2008)
9. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1(1), 32–49 (2011)
10. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation 6(2), 182–197 (2002)
11. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Trans. Evolutionary Computation 11(6), 712–731 (2007)
12. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN VIII. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
13. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society 46, 431–439 (1996)
14. Blumensath, T., Davies, M.: Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. 27, 265–274 (2009)