

AN EVOLUTIONARY MULTIOBJECTIVE METHOD FOR LOW-RANK AND SPARSE MATRIX DECOMPOSITION

TAO WU*, YU LEI, JIAO SHI AND MAOGUO GONG

School of Electronics and Information
 Northwestern Polytechnical University
 127 West Youyi Road, Xi'an Shaanxi, 710072, China

ABSTRACT. This paper addresses the problem of finding the low-rank and sparse components of a given matrix. The problem involves two conflicting objective functions, reducing the rank and sparsity of each part simultaneously. Previous methods combine two objectives into a single objective penalty function to solve with traditional numerical optimization approaches. The main contribution of this paper is to put forward a multiobjective method to decompose the given matrix into low-rank component and sparse part. We optimize two objective functions with an evolutionary multiobjective algorithm MOEA/D. Another contribution of this paper, a modified low-rank and sparse matrix model is proposed, which simplifying the variable of objective functions and improving the efficiency of multiobjective optimization. The proposed method obtains a set of solutions with different trade-off between low-rank and sparse objectives, and decision makers can choose one or more satisfied decomposed results according to different requirements directly. Experiments conducted on artificial datasets and nature images, show that the proposed method always obtains satisfied results, and the convergence, stability and robustness of the proposed method is acceptable.

1. Introduction. Matrix decomposition can catch some characteristic information of matrix, in which low-rank and sparse components are two of the most interesting. In the past decades, the theory of matrix decomposition has been developed rapidly, especially low-rank or sparse matrix decomposition have been used for many popular research areas, such as machine learning, image/signal processing and network analysis etc [19, 20, 22, 24]. The problem of low-rank and sparse matrix decomposition (LRSMD) is highlighted and intensively studied recently in [4, 6], especially in robust principle component analysis (PCA).

As described in the literature [4], we present a matrix $D \in R^{m \times n}$ with $m \leq n$, and know that it can be decomposed as

$$D = L_0 + S_0, \quad (1)$$

where $L_0 \in R^{m \times n}$ has low rank and $S_0 \in R^{m \times n}$ is sparse. We need to get the low-rank matrix L_0 and sparse matrix S_0 without prior knowledge about rank information and sparse pattern, and let both the rank of L_0 and the sparsity of S_0

2010 *Mathematics Subject Classification.* 90C29, 92D15.

Key words and phrases. Matrix decomposition, evolutionary algorithm, multiobjective optimization, low-rank, sparse.

* Corresponding author: Tao Wu.

as smaller as possible. We can model the LRSMD problem as a two objectives optimization problem:

$$\begin{aligned} & \text{minimize } (rank(L_0), card(S_0)) \\ & \text{subject to } L_0 + S_0 = D. \end{aligned} \quad (2)$$

It is known that the LRSMD problem is in general ill-posed and NP-hard [4], and the easiest way is finding a convex optimization problem to approximate original problem. The heuristics of using the l_1 -norm as the proxy of sparsity and the nuclear norm as the surrogate of low-rank are widely used in many areas (see e.g. [5, 10, 17]). The authors of [4] proposed the model that converting the NP-hard problem into convex optimization problem:

$$\begin{aligned} & \text{minimize } \|L\|_* + \lambda \|S\|_1 \\ & \text{subject to } L + S = D, \end{aligned} \quad (3)$$

where $\|L\|_*$ denotes the nuclear norm of the matrix L , that is, the sum of the singular value of L , and $\|S\|_1$ denotes the l_1 -norm of S seen as a long vector in $R^{m \times n}$.

Nowadays, many methods based on different strategies have been proposed. Iterative thresholding (IT) algorithms [1, 2] regularize the problem (3) and construct a Lagrange function, and then two objective variables are updated alternately. The iterative form of IT algorithms is simple and convergent, but the rate of convergence is slow and it is difficult to select the appropriate length of step. Accelerated proximal gradient (APG) algorithm [12] is another iterative approach, which building partial quadratic approximation by using Lipschitz gradient to solve the Lagrange function, iterating and getting solution. Although it is similar as IT algorithms, it reduces the number of iterations greatly. The dual method [12] is proposed because the nuclear norm is dual to spectral norm, the dual problem of problem (3) is easier to solve. Compared with APG algorithm, it has better scalability because it do not need complete singular decomposition in each iteration. One of the most influential method is augmented Lagrange multipliers (ALM) method [13]. The ALM method constructs augmented Lagrange function firstly, and then using alternating iteration method to optimize the problem. It is called EALM when using an exact augmented Lagrange multipliers while it is named IALM with inexact multipliers. IALM is also known as alternating direction methods (ADM) [21]. ALM method is more efficient than APG method, and can achieve higher accuracy with lower storage space.

For low-rank and sparse matrix decomposition problem, most of methods consider the ratio of low-rank degree and sparsity as a certain parameter, however different given matrices or requirements are always with different parameter λ . There are some related works that use multiobjective evolutionary algorithms to directly optimize two objectives of some machine learning problems [15, 16]. In this paper, we model low-rank and sparse matrix decomposition as a multiobjective problem (MOP), and modify the multiobjective LRSMD (MOLRSMD), which has simple structure and is easy to optimize. In MOLRSMD, on the one hand, the measure of low-rank degree and sparsity are still used as the two objective functions, which ensure the results of decomposition are feasible. On the other hand, the optimized efficiency is improved by modifying the optimization function. We use the evolutionary multiobjective algorithms to optimize the MOLRSMD which obtain trade-off solutions in various situations. The decision maker only need to select the solution

from Pareto optimal set, and it is easy to select because the quality of solutions in Pareto optimal set can be judged directly.

The remainder of this paper is organized as follows. Section 2 introduces the related background and motivation about the proposed method. In Section 3, the proposed model and algorithm are presented in detail. Experimental study is described in Section 4. Finally, concluding remarks are given in Section 5.

2. Related background and motivation. In this section, in order to make it easier to understand the proposed method, we will give a brief introduction to multiobjective optimization. Then we will describe the motivation of multiobjective low-rank and sparse matrix decomposition.

2.1. Multiobjective optimization. In general, it will be called multiobjective optimization (MOO), when the number of objective function is more than one and need to optimize simultaneously [7, 8, 14]. A MOP can be stated as:

$$\begin{aligned} \min F(x) &= (f_1(x), \dots, f_m(x))^T \\ \text{subject to } x &\in \Omega, \end{aligned} \quad (4)$$

where Ω is the decision(variable) space, $F : \Omega \rightarrow R^m$ consists of m real-value objective functions and R^m is called the objective space. The objectives are conflicting with each other in the most of time, it means no solution in Ω minimizes all the objectives simultaneously. The best trade-off among the objectives can be defined in terms of Pareto optimality.

Let $x_A, x_B \in \Omega$, x_A is said to dominate x_B if and only if

$$\begin{aligned} \forall i = 1, 2, \dots, m \quad & f_i(x_A) \leq f_i(x_B) \\ \exists j = 1, 2, \dots, m \quad & f_j(x_A) < f_j(x_B), \end{aligned} \quad (5)$$

and mark it as $x_A \succ x_B$. A solution $x^* \in \Omega$ is Pareto optimal if there is no solution $x \in \Omega$ such that $x \succ x^*$. The set of all the Pareto optimal solutions is called the Pareto set and the set of all the Pareto optimal objective vectors which corresponding to all Pareto optimal solution is the Pareto front. In order to illustrate the concepts of nondominated, dominated and Pareto front, we take a two objective optimization as example in Figure 1.

In practical applications, most objective functions have many or even infinite solutions, and we can not get all of them [23]. The purpose of multiobjective optimization is that finding a uniformly distributed Pareto front under a certain amount, which can represent the whole Pareto front. Evolutionary algorithm is one of the most popular methods for solving multiobjective optimization [9, 11, 18, 23, 25], such as VEGA [18], SPEA-II [25], NSGA-II [9] and MOEA/D [23]. Zhang and Li [23] proposed the MOEA/D which has a good performance in MOPs. MOEA/D decomposes the MOPs into a series scalar optimization subproblems with different weights. Due to the solution of each subproblem is similar to neighboring subproblems, the solution can be optimized by neighboring subproblems information. In this paper, due to the problem can be decomposed naturally with different parameter λ , we use MOEA/D to handling multiobjective low-rank and sparse matrix decomposition.

2.2. Motivation. For matrix decomposition with constraints of rank and sparsity, as described in problem 2, most of methods convert NP-hard problem into convex optimization. However these algorithms have to set a value of parameter λ for trade-off between reducing the rank of low-rank component and making sparse part

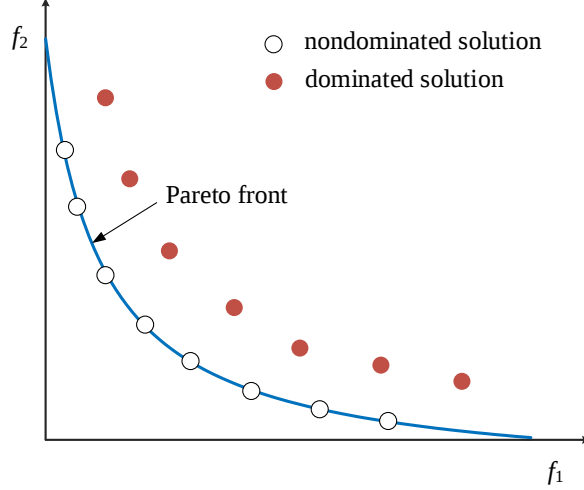


FIGURE 1. The distributions of nondominated solutions, dominated solutions and Pareto front in the objective space of the two objectives problem.

sparse as far as possible. Although [4] suggests $\lambda = 1/\sqrt{\max(m, n)}$, it is clear that different choices for λ in problem (3) will yield different optimal solutions and it is not easy to know the optimal λ , which will result in best decomposition of the given matrix. Figure 2 show a simple experiment to demonstrate how the parameter λ affects the result of sparse component recovering with ADM method. In Figure 2, the horizontal axis represents different possible choices of parameter λ , and the vertical axis means the error of recovering sparse matrix with different λ . We know that the best solution with the smallest error while parameter λ is about 0.22, and we can not recover a satisfied sparse component when λ is too small. The experiment emphasizes how important to choose a suitable parameter λ .

Multiobjective optimization is a popular way to avoiding the difficulty of parameter λ selection. Compared with traditional single objective penalty methods, multiobjective optimization takes apparent advantages on optimizing multi-objectives simultaneously and selecting solutions easily [3, 23]. Especially in solutions selection, MOO provides a set of trade-off solutions for decision maker while single objective methods have to try different parameters to find suitable solutions. In the most of time, single objective methods can not make a best decision without prior knowledge, even sometimes with prior knowledge, it is also.

For LRSMD, we focus on a more original problem, which thinks LRSMD as a multiobjective problem. The essence of the MOO is to optimize several conflicting objectives simultaneously. As described in problem (2), there are two existing conflicting objectives, which represent low-rank degree and sparsity respectively. We can obtain different solutions by a evolutionary multiobjective algorithm, such as MOEA/D. And each solution in the Pareto front represents a trade-off between reducing rank of low-rank component and decreasing the sparsity of sparse part, which can satisfy different requirements of decision makers.

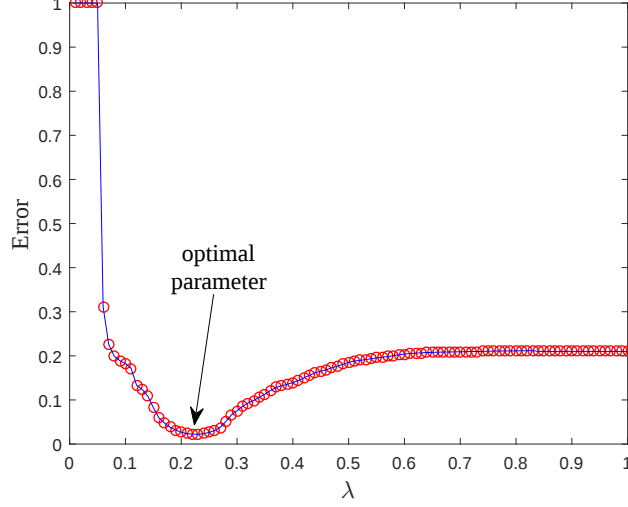


FIGURE 2. Error of sparse component recovering by the ADM method for LRSMD with different choices of the parameter λ

3. The proposed method. First of all, we introduce the model of multiobjective low-rank and sparse matrix decomposition based on matrix singular value decomposition. Then details of multiobjective low-rank and sparse matrix decomposition algorithm will be described, which show how the MOO approach is applied in matrix decomposition.

3.1. Modified multiobjective Low-rank and sparse matrix decomposition.

As is mentioned in Section 1, problem (2) is a standard multiobjective problem, however it is hard to optimize even with evolutionary algorithm. The reasons can summary as follow: 1) both two objective functions are NP-hard and the low-rank component and sparse component are tightly correlative. 2) the decision variable of problem (2) is complex whatever it is low-rank component L_0 or sparse part S_0 , it is difficult to obtain the optimal solution even if evolutionary algorithms are used, especially for the recovery of the low-rank component. A new multiobjective low-rank and sparse matrix decomposition model is modified by problem (2) and based on matrix singular value decomposition.

Let the singular value vector x as decision variable, f_1 represents the rank degree of low rank component, it is defined as

$$f_1(x) = \sum_{i=1}^n |x_i|. \quad (6)$$

Similarly, sparsity of sparse component is defined as

$$f_2(x) = \sum_{i=1}^m \sum_{j=1}^n |S_{ij}(x)|, \quad (7)$$

where $x \in R^n$, and $S(x) = D - U\Sigma(x)V^T$, $D \in R^{m \times n}$ is the given matrix and $\Sigma(x)$ is the diagonal matrix consisting of x . Both $U \in R^{m \times m}$ and $V \in R^{n \times n}$ are orthogonal matrix and they are fixed in the proposed method.

At this point, the two objective functions can be combined into a MOP. It has the following form:

$$\begin{aligned} \min_x F(x) &= \min_x (f_1, f_2) \\ \text{subject to } x &\in \Omega. \end{aligned} \quad (8)$$

For low rank component, it is hard to measure and optimize, but the singular value vector can approximate it. When U and V is suitable and fixed, every different low-rank matrix have only one singular value vector from the SVD theorem. The difficulty of sparse optimization is lower than low-rank, meanwhile, the complexity of sparse component as variable is harder than singular value vector. Based on the above mentioned, set the singular value vector as the variable of MOLRSMD is recommendable. The process of multiobjective LRSMD model is summarized in **Algorithm 1**.

Algorithm 1 Multiobjective LRSMD Model

Input: $D_{m \times n}$: matrix data.

Output: L, S : set of all possible decomposed results.

- 1: **Step 1)** $[U, \Sigma, V] \leftarrow \text{svd}(D)$
 - 2: **Step 2)** Generate a set of nondominated solutions (X) for problem (8) using Algorithm 2.
 - 3: **Step 3)** Recovery L and S .
 - 4: **for** $i = 1, 2, \dots, N$ **do**
 - 5: **Step 3.1)** $\Sigma_i \leftarrow X(i)$
 - 6: **Step 3.2)** $L_i \leftarrow U \Sigma_i V^T$
 - 7: **Step 3.2)** $S_i \leftarrow D - L_i$
 - 8: **end for**
-

3.2. Procedure of MOLRSMD based on MOEA/D. In this paper, the framework of MOEA/D [23] is applied in MOLRSMD because of the characteristic of MOEA/D which decomposing the MOP into several scalar optimization subproblems. In MOLRSMD we use Tchebycheff approach to decompose the multiobjective optimization problem (8), and the scalar optimization subproblem is in the form

$$\begin{aligned} \min g^{te}(x|w) &= \max\{w_1|f_1 - z_1^*|, w_2|f_2 - z_2^*|\} \\ \text{subject to } x &\in \Omega, \end{aligned} \quad (9)$$

where $w = (w_1, w_2)^T$ is a weight vector with $w_1, w_2 > 0$ and $w_1 + w_2 = 1$, and $z^* = (z_1^*, z_2^*)^T$ is the reference point, i.e., $z_1^* = \min\{f_1(x)|x \in \Omega\}$. We can obtain a set of subproblems like (9) by selecting different weight vector w . Each subproblem with different weight vector represents a certain trade-off between two objectives.

The procedure of the proposed algorithm for low-rank and sparse matrix decomposition is given in **Algorithm 2**. The MOLRSMD problem can be decomposed into N scalar optimization subproblems by (9). Therefore we defined a series of weight vectors $w^1, w^2, \dots, w^N$ with uniform spread and calculate the Euclidean distance between any two weight vectors, and then work out the T closest weight vectors to each weight vector. The neighborhood of each subproblem consists of all the subproblems with T closest weight vectors. The initial population of N individuals $x^1, x^2, \dots, x^N \in \Omega$ is randomly generated, where x^i is singular value vector of i -th low-rank component. During iteration, some specific genetic operators and problem specific heuristic are used to generate and improve new individuals. The

Algorithm 2 Procedure of MOLRSMD based on MOEA/D

Input: stopping criterion: t_{max} is the maximum number of generations; N : the number of subproblems; T : the size of neighborhood; pc : the probability of crossover; pd : the differential multiplier; pm : the probability of mutation; ps : the probability of mutation strategies selection.

Output: Pareto front solution; matrix decomposition results.

- 1: **Step 1) Initialization**
- 2: **Step 1.1)** Generate a uniform distributed of N weight vectors $w^1, w^2, \dots, w^N$ and computer the Euclidean distances between any two weight vectors and then work out the T closest weight vectors to each weight vector. For each $i = 1, \dots, N$, set $B(i) = \{i_1, \dots, i_T\}$, where $w^{i_1}, \dots, w^{i_T}$ are the T closest weight vector to w^i .
- 3: **Step 1.2)** Generate an initial population $x^1, x^2, \dots, x^N$ randomly.
- 4: **Step 1.3)** Initialize two reference points $z^l = (z_1^l, z_2^l)^T$ and $z^u = (z_1^u, z_2^u)^T$ represent the best and the worst values of each objective function in objective space respectively.
- 5: **Step 1.4)** For each $i = 1, \dots, N$, calculate f_1, f_2 and $g^{te}(x^i|w^j, z^l, z^u)$.
- 6: **Step 2)** Set $iter = 0$.
- 7: **Step 3) Update**
- 8: **for** $i = 1, 2, \dots, N$, **do**
- 9: **Step 3.1)** Randomly select two indexes k, l from $B(i)$, and then generate a new individual solution y from x^i, x^k and x^l by using genetic operator with pc, pd, pm and ps .
- 10: **Step 3.2)** Apply a thresholding method to improve y .
- 11: **Step 3.3)** For each $j = 1, 2$, if $z_j^l > f_j(y)$, then set $z_j^l = f_j(y)$; and if $z_j^u < f_j(y)$, then set $z_j^u = f_j(y)$.
- 12: **Step 3.4)** For each index $j \in B(i)$, calculate $g^{te}(y|w^j, z^l, z^u)$, if $g^{te}(y|w^j, z^l, z^u) \leq g^{te}(x^i|w^j, z^l, z^u)$, then set $x^i = y$ and $g^{te}(x^i|w^j, z^l, z^u) = g^{te}(y|w^j, z^l, z^u)$.
- 13: **end for**
- 14: **Step 4) Stopping criteria:** If $iter < t_{max}$, then $iter = iter + 1$ and go to **Step 3**, otherwise, stop the algorithm and output.

neighboring solutions are updated by $g^{te}(x|w)$ with each subproblem. Finally, we get a set of solutions with different trade-off relationship between the low-rank degree and sparsity of each component.

In this algorithm, some ideas of Tchebycheff decomposition and genetic operators are used which make solutions distribute more uniform and algorithm convergence faster. Next, some details about these ideas will be introduced.

3.2.1. Objective function normalization. In this paper, due to the difference of orders of magnitude between two objective functions value, the strategy of objective functions normalization is employed. In general, the value of objective function is large especially the range of two objectives have a huge difference, which make Pareto front is not uniform and most of points cluster in corner. A simple function normalization method formed as follow

$$\bar{f}_i = \frac{f_i - z_i^l}{z_i^u - z_i^l} \quad (10)$$

where $i = 1, 2$, $z^l = (z_1^l, z_2^l)^T$ is the lowest point and $z^u = (z_1^u, z_2^u)^T$ is the highest point in objective space. For scalar optimization subproblems, the objective function of Tchebycheff decomposition is

$$\begin{aligned}\overline{g^{ws}}(x|w, z^l, z^u) &= \max\{w_1|f_1 - z_1^*|, w_2|f_2 - z_2^*|\} \\ &= \max\{w_1|\bar{f}_1 - \bar{z}_1^*|, w_2|\bar{f}_2 - \bar{z}_2^*|\} \\ &= \max\{w_1|\frac{f_1 - z_1^l}{z_1^u - z_1^l}|, w_2|\frac{f_2 - z_2^l}{z_2^u - z_2^l}|\},\end{aligned}\tag{11}$$

3.2.2. Genetic operator. Genetic operator is the key of evolutionary algorithm, with genetic operator combined characteristic of optimization problem, the convergence speed of algorithm can be improved. In this paper, we use a difference strategy and double local mutation method to speed up algorithm.

In crossover operator, difference optimization strategy is applied. We select two individuals randomly from the neighborhood of x^i , noted as x^k and x^l , and new individual can generate by

$$y_j = \begin{cases} x_j^i + F(x_j^k - x_j^l) & \text{if } r < pc \\ x_j^i & \text{otherwise} \end{cases}\tag{12}$$

where x_j , y_j mean the j -th value of x and y and $j = 1, 2, \dots, N$.

We apply a scale controllable mutation with two different mutations to balance diversity and precision of population. In the paper, we select Gauss mutation and polynomial mutation. Gauss mutation is simple than polynomial mutation and well done in global searching. Polynomial mutation for local search can improve the quality of solution. Therefore, if a random number $r < pm$, do

$$y' = \begin{cases} Gauss(y, \sigma) & \text{if } rand < ps \\ Polynomial(y) & \text{otherwise} \end{cases}\tag{13}$$

where $Gauss(y, \sigma)$ is a Gauss function with y, σ , and $Polynomial(y)$ is a polynomial function with variable y .

In this section, setting singular value vector as decision variable, makes problem simpler and easier to optimize. On the one hand, the complexity of variable is $O(n)$ while traditional method with $O(mn)$. On the other hand, there are few SVD process in multiobjective LRSMD model while most traditional penalty methods need more SVD process, and the most of time cost in iteration spend on SVD process. Furthermore, multiobjective LRSMD model can obtain a set of trade-off solutions which can be chosen directly by decision makers with requirements.

4. Experimental study. In this section, we will represent experiments on artificial generated datasets and some nature images to evaluate the performance of the proposed method. The experiment focus on how the multiobjective low-rank and sparse matrix decomposition algorithm works with different type test data, and analyzing the performance with the convergence, stability and robustness.

4.1. Experimental datasets and experimental setting. Let $D = L^* + S^*$ be the available data, where L^* and S^* are, the original low-rank and sparse matrices, respectively, that need to be recovered. The datasets of experiment are generated with different combinations of the rank of L^* and the sparsity of S^* . In addition, the size of original matrix is also different in experiment and all matrices are generated by Gauss random number. We also have experiments on datasets with noise to evaluate the robustness of the proposed method. The experimental

datasets are not only artificial generated datasets as described above but also some nature images. The nature images are consisted of *Lena* and some images from *orl-faces* datasets.

In order to illustrate the result of experiment intuitively, we define some indexes to measure results and the performance of the proposed method. The relative error to the original low-rank matrix L^* is defined by

$$ErrLR := \frac{\|L - L^*\|_F}{\|L^*\|_F}. \quad (14)$$

Similarly, the relative error to the original sparse matrix S^* is defined by

$$ErrSP := \frac{\|S - S^*\|_F}{\|S^*\|_F}. \quad (15)$$

The total error to original matrix is

$$ErrT := \frac{\|L + S - D^*\|_F}{\|D^*\|_F}. \quad (16)$$

Due to the characteristic of multiobjective matrix decomposition method, there are many different decomposition results and some results are widely different with original matrices. On the one hand, in order to analysis the accuracy of matrix decomposition in artificial datasets, we just focus on solution with smallest recovery error and the distribute of Pareto front. On the other hand, we could select solution with actual demand when all decomposed solutions would be represented with images in nature images.

The parameters in the proposed method are given in Table 1.

TABLE 1. The Parameters in the MOLRSMD used in the Experiments

Parameter	Meaning	Value
N	The number of subproblems	100
T	The number of neighbors	20
t_{max}	The maximum of generations	300
pc	The probability of crossover	0.8
pd	The differential multiplier	0.5
pm	The probability of mutation	0.2
ps	The probability of mutation selection	0.85

4.2. Experimental results.

4.2.1. *Artificial datasets.* For artificial generated datasets, we select three different size test matrices, which the sizes are 20×20 , 50×50 and 100×100 , respectively. In addition, the rank and sparsity are also different. From Figure 3, it illustrates the results with three different sizes and represents MOLRSMD work well in this datasets. The first column is three Pareto front for MOLRSMD in different data, which the biggest different is matrix size. The Pareto front is with approximate uniform distribution for all of three different data, which means MOLRSMD has a good convergence. Each figure in the second column of Figure 3 is two objective functions corresponding to the solutions. The blue points represent low-rank function values while the red points represent sparse function value. The value of low-rank function is always in a small range compared with sparse function. The

last column is three box-plot about $ErrLR$ and $ErrSP$ by running 30 times independent experiments. The box-plot shows the statistical results of the change of $ErrLR$ and $ErrSP$. From box-plot we can see that low-rank recovering error is lower than sparse and do not change appreciably with different times running. It means the proposed method has a great stability.

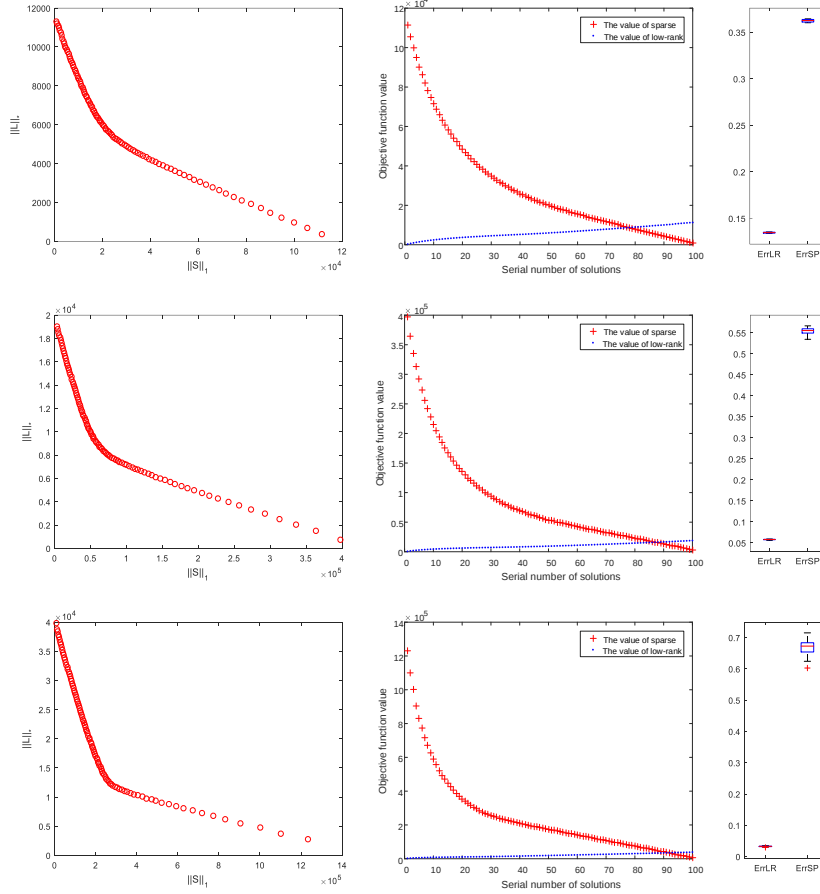


FIGURE 3. The Pareto front, two objective functions corresponding to the solutions and box-plot for $ErrLR$ and $ErrSP$ by running 30 times independent experiments. Three rows represent experiment on the data with size: 20×20 , rank: 5, sparsity: 0.2, size: 50×50 , rank: 5, sparsity: 0.2 and size: 100×100 , rank: 5, sparsity: 0.5 respectively.

Furthermore, we also do experiments on some data with noise to test the robustness of the proposed method. For data with size: 20×20 , rank: 1 and sparsity: 0.1, we add Gauss noise randomly. The types of noise determined by two factors and can be described as follow: number of noise points can choose from 20 and 50; σ of Gauss noise: 1, 5 and 15. We randomly combine two factors of noise and test all of them except 50 points, $\sigma = 1$.

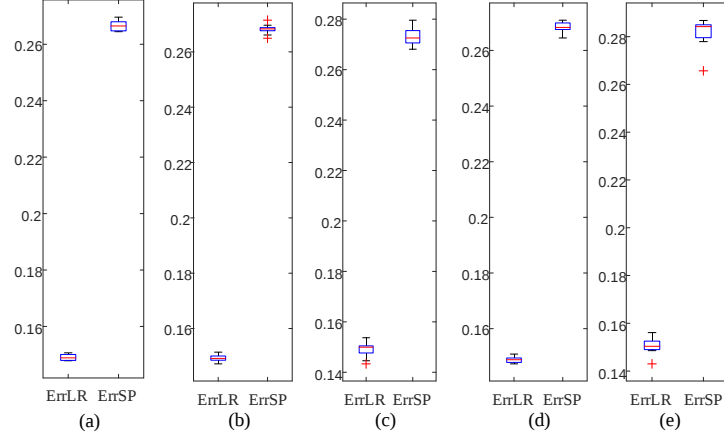


FIGURE 4. Box-plot about $ErrLR$ and $ErrSP$ for data with different noise. (a): number of noise points is 20, $\sigma = 1$. (b): number of noise points is 20, $\sigma = 5$. (c): number of noise points is 20, $\sigma = 15$. (d): number of noise points is 50, $\sigma = 5$. (e): number of noise points is 50, $\sigma = 15$.

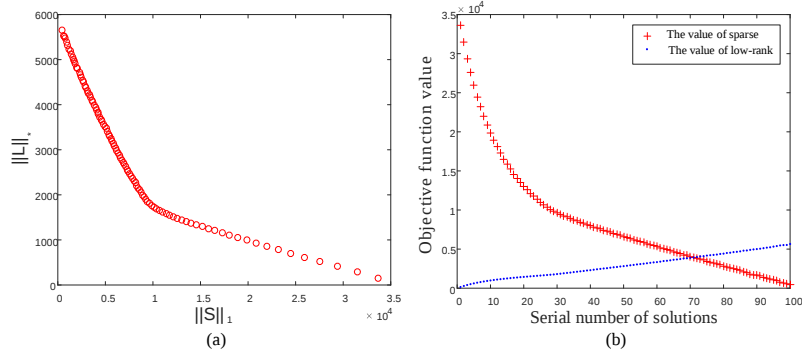


FIGURE 5. The Pareto front and two objective functions corresponding to the solutions for noise data. Noise type: number of noise points is 50 and $\sigma = 15$.

Figure 4 displays the box-plot of recovering error in five noisy situations, which is described as above. We can know that not only the error of low-rank recovering but also sparse recovering is changed in a small range with different types noise. With the improved of number of noise points or σ of Gauss function, the changed range of error is improved. However, the range of difference between different noise is reasonable. In a word, the robustness of the proposed method is acceptable. Figure 5 shows that the Pareto front is also smooth and uniform distribution with the worst noise, which number of noise points is 50 and σ of Gauss function is 15.

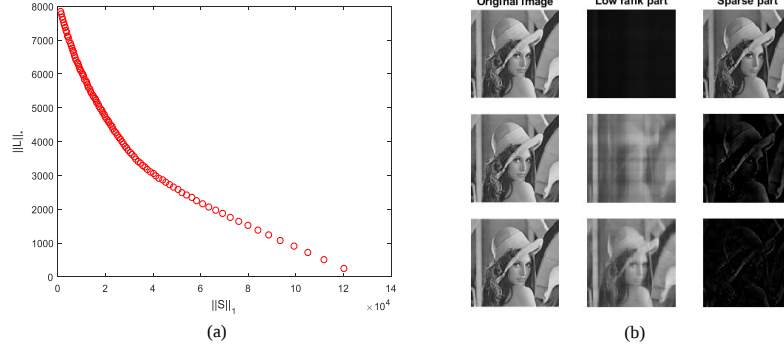


FIGURE 6. The Pareto front and decomposed results with image lena. (a) is Pareto front of MOLRSMD and (b) is three different decomposed results selected from Pareto front, and they locate at the top, middle and bottom of Pareto front, respectively.

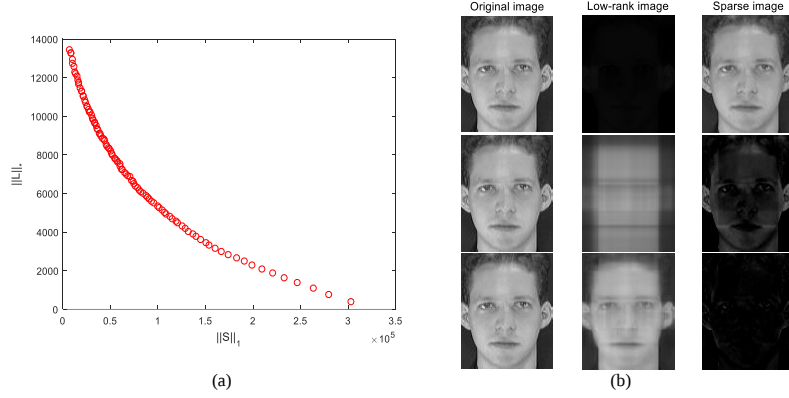


FIGURE 7. The Pareto front and decomposed results with image face. (a) is Pareto front of MOLRSMD and (b) is three different decomposed results selected from Pareto front, and they locate at the top, middle and bottom of Pareto front, respectively.

4.2.2. Nature Datasets. In the nature images, the proposed method also works well. The images that we used is consisted of *Lena* and *orl-faces* datasets. Due to no standard decomposed results of datasets, we just analysis the Pareto front and practical results to evaluate performance of the proposed method. Some results are displayed in Figure 6 and Figure 7. Figure 6 is about image *Lena* and Figure 7 is result in one random face image from *orl-faces*. We can see that the Pareto front of the proposed method is smooth and equally distributed, which means the method converges to PF stably. The (b) of Figure 6 and 7 show three different decomposed results with different locations in the PF. The first row of part (b) means sparse component has an absolute advantages, on the contrary, the third row shows low-rank part with more advantage. The middle one is a balance between low-rank and sparse components.

4.2.3. Comparison Study. The performance of the proposed MOLRSMD have been examined in the previous subsection. In this subsection, simple comparison with some advanced LRSMD algorithms is tested on artificial datasets. ADM [21] and ALM [13] are two mainstream and efficient algorithms in exiting research. Both of them need set parameter λ as prior information which balances low-rank and sparse components. For all the algorithms, the parameter λ is varied from 0 to 1, with a step size of 0.01 in order to compare with MOLRSMD fairly. The results of the proposed MOLRSMD as compared with ADM and ALM are presented in Figure 8 and Table. 2.

Each row of Figure 8 presents the Pareto front by different algorithms. The Pareto front of MOLRSMD is complete and smooth as compared with ADM and ALM. The results of ADM and ALM are with uneven distribution, most of Pareto optimal solutions cluster in low sparsity region even the region with sparsity close to zero. In particular, from the second row of Figure 8, solutions of ADM and ALM with a sudden change and most of solutions cluster in two regions (the value of two objective functions are about 7500, 2500 and 4500, 0 respectively) while solutions of the proposed MOLRSMD with uniform and smooth distribution. Table 2 represents the smallest recovery error from 100 solutions of the proposed MOLRSMD, ADM and ALM on artificial datasets (size: 50×50 , rank: 5, sparsity: 0.2). We can see that errors of three algorithms are similar. We have a discovery from above all, the influence of parameter λ is nonlinear and hard to described. Sometimes it is hard to find a suitable parameter λ . For example, if the solution in region with low rank and high sparsity in this case, the difficulty of find suitable λ is very hard even impossible. In contrary, the proposed MOLRSMD avoids the selection of parameter λ . It is easier to find optimal solution compared with ADM and ALM.

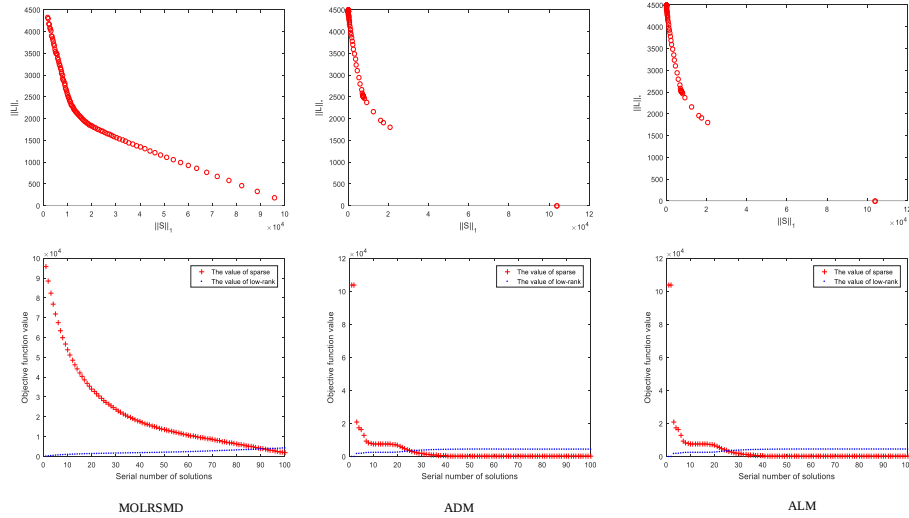


FIGURE 8. Results of the proposed MOLRSMD compared with ADM and ALM. The test dataset size is 50×50 , rank equals 5 and sparsity is 0.2.

TABLE 2. Errors of the proposed MOLRSMD compared with ADM and ALM.

Methods	$ErrLR$	$ErrSP$	$ErrT$
MOLRSMD	0.0100	0.1986	0
ADM	0.0299	0.1331	1.9135e-17
ALM	0.0148	0.0661	2.2412e-10

5. Conclusion. In this paper, an evolutionary multiobjective low-rank and sparse matrix decomposition method has been proposed, for finding optimal trade-off solutions between the rank of low-rank component and the sparsity of sparse part. We focus on more original problem of low-rank and sparse matrix decomposition, with the theorem of matrix singular value decomposition, modifying the model by replacing decision variable with singular value vector. We have experiments on different type datasets, including artificial datasets, datasets with noise and nature images, to indicate that the proposed MOLRSMD is always with good convergence, reliable stability and acceptable robustness. It also show that it can be applied to some practical problem.

Future work will be attention on decreasing the accuracy of recovering. In the proposed method, the orthogonal matrices U and V is fixed, however different matrix with different orthogonal matrices in the most of time. Fixed orthogonal matrices is always increasing the error. Next, we will explore how to find the best U and V while singular value vector changed.

REFERENCES

- [1] A. Beck and M. Teboulle, [A fast iterative shrinkage-thresholding algorithm for linear inverse problems](#), *SIAM Journal on Imaging Sciences*, **2** (2009), 183–202.
- [2] J.-F. Cai, E. J. Candès and Z. Shen, [A singular value thresholding algorithm for matrix completion](#), *SIAM Journal on Optimization*, **20** (2010), 1956–1982.
- [3] Z. Cai and Y. Wang, [A multiobjective optimization-based evolutionary algorithm for constrained optimization](#), *IEEE Transactions on Evolutionary Computation*, **10** (2006), 658–675.
- [4] E. J. Candès, X. Li, Y. Ma and J. Wright, [Robust principal component analysis?](#), *Journal of the ACM (JACM)*, **58** (2011), Art. 11, 37 pp.
- [5] E. J. Candès and B. Recht, [Exact matrix completion via convex optimization](#), *Foundations of Computational Mathematics*, **9** (2009), 717–772.
- [6] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo and A. S. Willsky, [Rank-sparsity incoherence for matrix decomposition](#), *SIAM Journal on Optimization*, **21** (2011), 572–596.
- [7] C. A. C. Coello, D. A. Van Veldhuizen and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic Algorithms and Evolutionary Computation, 5. Kluwer Academic/Plenum Publishers, New York, 2002.
- [8] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Ltd., Chichester, 2001.
- [9] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, [A fast and elitist multiobjective genetic algorithm: NSGA-II](#), *IEEE Transactions on Evolutionary Computation*, **6** (2002), 182–197.
- [10] M. Fazel, H. Hindi and S. P. Boyd, [A rank minimization heuristic with application to minimum order system approximation](#), in *Proceedings of the American Control Conference*, IEEE, **6** (2001), 4734–4739.
- [11] M. Gong, L. Jiao, H. Du and L. Bo, [Multiobjective immune algorithm with nondominated neighbor-based selection](#), *Evolutionary Computation*, **16** (2008), 225–255.
- [12] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen and Y. Ma, Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix, *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, vol. 61, 2009.

- [13] Z. Lin, M. Chen and Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, arXiv preprint, arXiv:1009.5055, 2010.
- [14] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston, MA, 1999.
- [15] C. Qian, Y. Yu and Z.-H. Zhou, Pareto ensemble pruning, in *AAAI*, (2015), 2935–2941.
- [16] —, Subset selection by pareto optimization, in *Advances in Neural Information Processing Systems*, (2015), 1774–1782.
- [17] B. Recht, M. Fazel and P. A. Parrilo, [Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization](#), *SIAM Review*, **52** (2010), 471–501.
- [18] J. D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in *Proceedings of the 1st international Conference on Genetic Algorithms*. L. Erlbaum Associates Inc., (1985), 93–100.
- [19] J. L. Starck, M. Elad and D. L. Donoho, [Image decomposition via the combination of sparse representations and a variational approach](#), *IEEE Transactions on Image Processing*, **14** (2005), 1570–1582.
- [20] J. Yan, J. Liu, Y. Li, Z. Niu and Y. Liu, [Visual saliency detection via rank-sparsity decomposition](#), in *IEEE International Conference on Image Processing*, IEEE, (2010), 1089–1092.
- [21] X. Yuan and J. Yang, Sparse and low-rank matrix decomposition via alternating direction methods, *Pacific Journal of Optimization*, **9** (2013), 167–180.
- [22] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, and S. Ma, [Image classification by non-negative sparse coding, low-rank and sparse decomposition](#), in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2011), 1673–1680.
- [23] Q. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation*, **11** (2007), 712–731.
- [24] M. Zibulevsky and B. A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, *Neural Computation*, **13** (2001), 863–882.
- [25] E. Zitzler, M. Laumanns and L. Thiele *et al.*, SPEA2: Improving the strength pareto evolutionary algorithm, in *Eurogen*, **3242** (2001), 95–100.

E-mail address: tao_woe@mail.nwpu.edu.cn

E-mail address: leiy@nwpu.edu.cn

E-mail address: Jiaoshi@nwpu.edu.cn

E-mail address: gong@ieee.org