Convergence Acceleration for Multiobjective Sparse Reconstruction via Knowledge Transfer

1st Bai Yan *Institute of Laser Engineering*
Beijing University of Technology
Beijing, China
yanbai@emails.bjut.edu.cn
2nd Qi Zhao *College of Economics and Management*
Beijing University of Technology
Beijing, China
qzhao@emails.bjut.edu.cn
3rd Andrew J. Zhang *Global Big Data Technologies Centre*
University of Technology Sydney
Sydney, Australia
Andrew.Zhang@uts.edu.au
4th Yonghui Li *School of Electrical and Information Engineering*
The University of Sydney
Sydney, Australia
yonghui.li@sydney.edu.au
5th Zhihai Wang *Key Laboratory of Optoelectronics Technology, Ministry of Education*
Beijing University of Technology
Beijing, China
wangzhihai@bjut.edu.cn

Abstract

Multiobjective sparse reconstruction (MOSR) methods can potentially obtain superior reconstruction performance. However, they suffer from high computational cost, especially in high-dimensional reconstruction. Furthermore, they are generally implemented independently without updating and exploiting prior knowledge iteratively, leading to unnecessary computational consumption due to the re-exploration of similar search spaces. To address these problems, we propose a sparse-constraint knowledge transfer operator to accelerate the convergence of MOSR solvers by exploiting the knowledge from past problem-solving experiences. Firstly, we introduce the deep nonlinear feature coding method to extract the feature mapping for the search between the current problem and a previously solved MOSR problem. Through this mapping, we learn a set of knowledge-induced solutions which contain the search experience for the past problem. Thereafter, we develop and apply a sparse-constraint strategy to refine these learned solutions to guarantee their sparse characteristics. Finally, we inject the refined solutions into the iteration of the current problem to facilitate the convergence. To validate the efficiency of the proposed operator, comprehensive studies on extensive simulated signal reconstruction are conducted.

Index Terms

Sparse reconstruction, multiobjective evolutionary algorithm, learning, knowledge transfer

I. INTRODUCTION

In the compressed sensing (CS) theory [1][2], a sparse reconstruction problem is often considered:

$$\min_{\mathbf{x}} \ \|\mathbf{x}\|_0, \ s.t. \ \mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}, \tag{1}$$

where $\mathbf{x} \in \Re^n$ is a $k-$sparse signal (i.e., there are k nonzero values ($k < n$) in the signal), $\mathbf{A} \in \Re^{m\times n}(m \leq n)$ is the measurement matrix, $\mathbf{b} \in \Re^{m\times 1}$ is the measurement vector, and $\mathbf{e} \in \Re^{m\times 1}$ denotes the noise vector.

The problem in (1) can be rewritten as the following unconstrained optimization problem:

$$\arg\min_{\mathbf{x}} \ \lambda\|\mathbf{x}\|_p + \frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2, \ p \in [0, 1] \tag{2}$$

where λ is a pre-chosen positive regularization parameter being introduced to balance the two conflicting objective terms (the regularization term and measurement error). Unfortunately, there

The work of Bai Yan is supported by the China Scholarship Council under Grant 201706540025.*(Corresponding Author: Bai Yan)*

is no optimal rule for determining λ. Some heuristics methods are used, e.g., the Homotopy continuation methods [3][4], and the cross validation method [5].

To solve the optimization problem together with selecting an optimal λ, the multiobjective evolutionary algorithms (MOEAs) [6][7][8] are applied. MOEAs can simultaneously optimize all the objectives and obtain a number of nondominated solutions (termed as Pareto front, PF). In this regards, (2) is transformed into a MOSR problem as

$$f(\mathbf{x}) = \min_{\mathbf{x}}(\|\mathbf{x}\|_0, \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2). \tag{3}$$

The first solver to the MOSR problem is the soft-thresholding evolutionary multiobjective (StEMO) algorithm [9]. StEMO is based on the NSGA-II framework [7] and it uses the IST method [10] for local search. It was observed that the knee region can provide the best trade-off solution. To enhance the reconstruction precision, the LBEA is proposed in [11], which employed the improved linear Bregman-based local search operator in the differential evolution paradigm to accelerate the convergence. A two-phase evolutionary approach for sparse reconstruction is proposed in [12]. In phase 1, the statistical features of the nondominated solutions from MOEA/D [8] were extracted to generate new solutions. In phase 2, a forward-based selection method was designed for better locating the nonzero entries. An improved MOEA/D equipped with sparse preference-based local search, denoted as SPLS, was proposed in [13]. The knee region was exploited with preference. In [14], we proposed an adaptive decomposition-based evolutionary approach (ADEA). With the guidance of reference vectors, more search effort on the approximating knee region was executed by adaptively adding the reference vectors.

Although these MOSR solvers can achieve better reconstruction performance than conventional algorithms, they suffer from high computational cost, especially in high-dimensional reconstruction scenarios. When the signal is less sparse or there are fewer measurements, more iterations are needed and the computational cost is further increased.

In addition, many optimization solvers, including MOSR, are implemented independently without exploiting the previous problem-solving knowledge. This causes unnecessary computational complexity due to the re-exploration of similar search spaces [15]. In fact, for practical artificial systems, the problems to be solved are rarely isolated, but may be repetitive or share domain-specific similarities. Some studies on evolutionary optimizers [16], [17], [18] have demonstrated the accelerated effect of reusing the prior information. This finding motivates us to exploit the

solution search knowledge from past solved problems for the MOSR problem, where similarities to the evolutionary optimizers exist in the problem form and the solution search process.

In this paper, we propose a sparse-constraint knowledge transfer operator to accelerate the convergence of MOSR solvers by exploiting the knowledge from a previous problem-solving process. Firstly, we introduce the deep nonlinear feature coding (DNFC) [19] to extract the feature mapping for the searching process between the current and a previously solved MOSR problems. Through the mapping, we learn a set of knowledge-induced solutions which provide the search experience for the past solved problem. We then propose a sparse-constraint strategy to refine these learned solutions to guarantee their sparsity characteristics. Finally, we inject the refined solutions into the iteration process for solving the current problem to facilitate the convergence.

The rest of this paper is organized as follows. Section II gives a brief introduction to the background techniques. Section III details the proposed sparse-constraint knowledge transfer operator. In Section IV, the performance of the proposed operator is examined using two baseline MOSR algorithms StEMO and ADEA. Finally, conclusions are described in Section VI.

II. Preliminaries

This section provides a brief introduction to the maximum mean discrepancy (MMD) [20] and marginalized denoising autoencoder (mDA) [21], which serve as basic techniques for the proposed knowledge transfer operator.

A. Maximum Mean Discrepancy

The MMD [20] is a distance estimation method, which measures the discrepancy between two distributions by comparing the difference of the mean values. Specifically, let $\mathbf{X}_s = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_s}]$ and $\mathbf{Y}_t = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{n_t}]$ denote the samples of two distributions on a domain χ, and Ω is a function: $\chi \rightarrow \Re$, then the MMD can be formulated as

$$\|\frac{1}{n_s}\sum_{i=1}^{n_s}\Omega(\mathbf{x}_i) - \frac{1}{n_t}\sum_{i=1}^{n_t}\Omega(\mathbf{y}_i)\|_{\chi}. \tag{4}$$

Further, the study [22] performed the MMD method in a Reproducing Kernel Hilbert Space (RKHS) to capture the nonlinear divergence between $\mathbf{X}_s$ and $\mathbf{Y}_t$. The function Ω is replaced by

a kernel-induced feature map $\Phi : \chi \to \mathcal{H}$ with $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ as the kernel of $\mathcal{H}$, (4) can be written as

$$\| \frac{1}{n_s} \sum_{i=1}^{n_s} \Phi(\mathbf{x}_i) - \frac{1}{n_t} \sum_{i=1}^{n_t} \Phi(\mathbf{y}_i) \|_{\mathcal{H}}. \tag{5}$$

B. Marginalized Denoising Autoencoder

The mDA [21] aims at learning a linear mapping $\mathbf{W}$ to reconstruct the original data from its corrupted versions. Assume $\overline{\mathbf{X}} = [\mathbf{X}, \mathbf{X}, \dots \mathbf{X}]$ is the union of r-times repeated $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_X}]$;, and $\widetilde{\mathbf{X}} = [\widetilde{\mathbf{X}^1}, \widetilde{\mathbf{X}^2}, \dots, \widetilde{\mathbf{X}^r}]$ is the combination of different r-times corrupted versions of $\mathbf{X}$ by random feature removal (i.e., each feature of $\mathbf{X}$ is corrupted to 0 with probability p), then the objective function of mDA can be modeled as

$$\min L(\mathbf{W}) = \frac{1}{2rn_X} tr[(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\mathbf{X}})^T (\overline{\mathbf{X}} - \mathbf{W}\widetilde{\mathbf{X}})] \tag{6}$$

where $tr(\cdot)$ is a function to compute the trace of a matrix.

III. MOSR VIA TRANSFER OPERATOR

In this section, we propose a novel sparse-constraint knowledge transfer operator to speed up the convergence of MOSR solvers. The motivation is that, although SR problems vary from each other, they are not isolated and may be repetitive or have some domain-specific similarities. Therefore, we design this operator to reuse the structural knowledge from previous search experiences, accelerating the convergence.

A. Framework

The workflow of the proposed operator in MOSR solvers is shown in Fig. 1, with the corresponding pseudo-code provided in Algorithm 1. For convenience, we denote the current MOSR problem as the *target problem* and name the previously solved problem which we want to transfer knowledge from as the *source problem*. $\mathbf{P}^t$ and $\mathbf{PS}^t$ are the solution sets to the target and source problems at generation t, respectively, and $\mathbf{PS}^{t_{max}}$ is the set of the optimized solutions to the source problem. As shown in Figure 1, in each iteration of the target problem, the recombination (i.e., crossover and mutation), local search, and selection steps of MOSR solvers are firstly executed. Subsequently, the proposed sparse-constraint knowledge transfer operator is implemented. This operator is depicted in the dotted box of Fig. 1 and detailed in the next subsection.

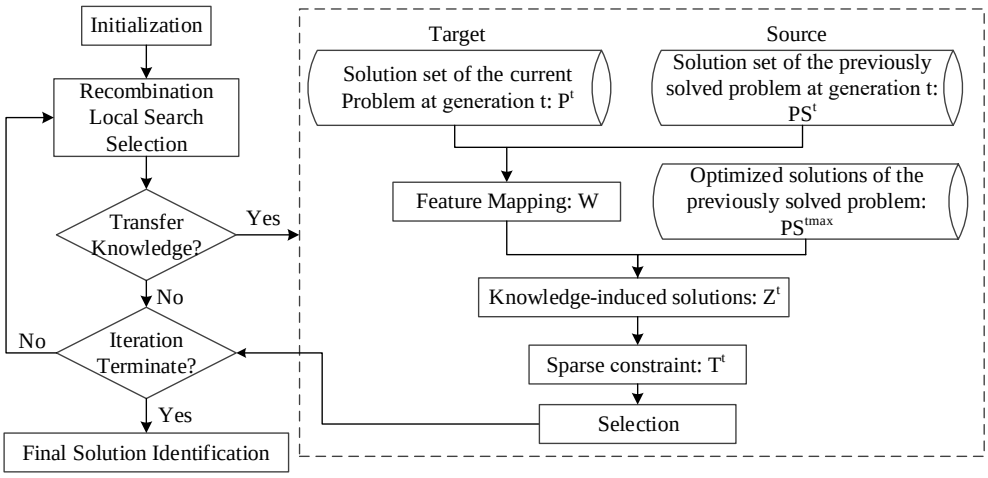

Fig. 1. Workflow of MOSR via knowledge transfer

Algorithm 1: Pseudo-code of MOSR via knowledge transfer

Input: $\mathbf{A}$, b, $\mathbf{PS}$, N
Output: $\mathbf{x}$

$\mathbf{P}^0 \leftarrow$ Initialization;
while *"the stopping criterion is not met"* **do**
  $\mathbf{Q}^t =$Recombination($\mathbf{P}^t$);
  $\mathbf{L}^t =$Local_Search($\mathbf{P}^t \cup \mathbf{Q}^t$);
  $\mathbf{P}^t =$Selection($\mathbf{P}^t \cup \mathbf{Q}^t \cup \mathbf{L}^t$);
  if *"knowledge transfer"* **then**
    $\mathbf{P}^t$=Sparse-Constraint_Knowledge_Transfer($\mathbf{P}^t, \mathbf{PS}^t$,$\mathbf{PS}^{t_{max}}$);
  end
  $t = t + 1$;
end
$\mathbf{x} \leftarrow$Final_ Solution_Identification($\mathbf{P}^t$);

B. Sparse-Constraint Knowledge Transfer Operator

In this subsection, we describe the four steps of the sparse-constraint knowledge transfer operator in detail. The pseudo-code is given in Algorithm 2.

1) Common feature extraction: Common feature extraction aims at finding a "connective bridge" $\mathbf{W}$ for the searching process between the source and target problems. DNFC [19] is an effective method for domain adaptation, which can minimize the distribution discrepancy between the source and target domains and provide a closed-form solution. We employ the single-layer form of DNFC, named NFC, for acquiring the feature mapping matrix $\mathbf{W}$.

The NFC incorporates the MMD (refer to II-A) and nonlinear coding by kernelization into

Algorithm 2: Sparse-constraint knowledge transfer operator

Input: $\mathbf{A}$, b, $\mathbf{P}^t$,$\mathbf{PS}^t$, $\mathbf{PS}^{t_{max}}$, θ
Output: $\mathbf{P}^{t+1}$

/*Common feature extraction*/
$\mathbf{W} = E[\mathbf{R}1](E[\mathbf{R}2] + \theta E[\mathbf{R}3])^{-1}$;
/*Acquiremen of knowledge-induced population*/
$\mathbf{K} = \Phi(\mathbf{X})^T \Phi(\mathbf{PS}^{t_{max}})$;
$\mathbf{Z} = \mathbf{WK}$;
/*Sparse constraint*/
for $i = 1 : N$ **do**
  $\mathbf{supp} = \{z | (\mathbf{P}_i^t)_z = 0\}$;
  $[\mathbf{T}^t]_{i,z} = \begin{cases} [\mathbf{Z}^t]_{i,z}, & z \notin \mathbf{supp}, \\ 0, & z \in \mathbf{supp}. \end{cases}$
end
/*Selection*/
$\mathbf{P}^{t+1}$ =Selection($\mathbf{P}^t \cup \mathbf{T}^t$);

the mDA (refer to II-B), in which the MMD enables the extracted features to have a small distribution discrepancy while the kernelization ensures the nonlinearity relationship between the domains to be well exploited. Let $\mathbf{X} = \mathbf{PS}^t \cup \mathbf{P}^t$ and its r-times copies $\overline{\mathbf{X}} = [\mathbf{X}, \mathbf{X}, \dots \mathbf{X}]$. As a function of the feature mapping matrix $\mathbf{W}$, the objective function is formulated as

$$\begin{aligned} \Gamma(\mathbf{W}) =& tr[(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\Phi(\mathbf{X})})^T(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\Phi(\mathbf{X})})] \\ &+ \|\frac{1}{n_s}\sum_{i=1}^{n_s}\Phi(\widetilde{\mathbf{x}_i^r}) - \frac{1}{n_t}\sum_{i=n_s+1}^{n_t}\Phi(\widetilde{\mathbf{x}_i^r})\|^2 \\ =& \underbrace{tr[(\overline{\mathbf{X}} - \mathbf{W}_k\widetilde{\mathbf{K}})^T(\overline{\mathbf{X}} - \mathbf{W}_k\widetilde{\mathbf{K}})]}_{\text{mDA}} + \\ & \theta\underbrace{tr(\mathbf{W}_k\widetilde{\mathbf{K}}\widetilde{\mathbf{G}}\widetilde{\mathbf{K}}^T\mathbf{W}_k^T)}_{\text{MMD}} \end{aligned} \tag{7}$$

where $\mathbf{W} = \mathbf{W}_k\Phi(\mathbf{X})^T$, $\Phi(\mathbf{X})$ is the mapped $\mathbf{X}$ in the RKHS; $\mathbf{K} = \Phi(\mathbf{X})^T\Phi(\mathbf{X})$ is the corresponding kernel matrix; $\widetilde{\mathbf{K}}$ is the corrupted kernel matrix with a corruption probability p; $\mathbf{G} = [\mathbf{G}_{i,j}]_{(n_s+n_t)\times(n_s+n_t)}$ with $G_{i,j} = 1/n_s^2$ if $\mathbf{X}_{i,j} \in \mathbf{PS}^t$, $G_{i,j} = 1/n_t^2$ if $\mathbf{X}_{i,j} \in \mathbf{P}^t$, $G_{i,j} = -1/(n_s n_t)$ otherwise; θ is the balancing parameter. Applying the weak law of large numbers and computing the expectations when $r \to \infty$, a closed-form solution for $\mathbf{W}_k$ that

minimizes (7) can be obtained as

$$\mathbf{W}_k = E[\mathbf{R}1](E[\mathbf{R}2] + \theta E[\mathbf{R}3])^{-1} \tag{8}$$

with

$$\begin{aligned} E[\mathbf{R}1] &= (1-p)\mathbf{X}\mathbf{K}^T \\ E[\mathbf{R}2]_{i,j} &= \begin{cases} (1-p)^2\mathbf{K}\mathbf{K}^T, & i \neq j \\ (1-p)\mathbf{K}\mathbf{K}^T, & i = j \end{cases} \\ E[\mathbf{R}3]_{i,j} &= \begin{cases} (1-p)^2\mathbf{K}\mathbf{G}\mathbf{K}^T, & i \neq j \\ (1-p)^2\mathbf{K}\mathbf{G}\mathbf{K}^T + p(1-p)\mathbf{K}\mathbf{F}\mathbf{K}^T, & i = j, \end{cases} \end{aligned} \tag{9}$$

where $\mathbf{F}$ is a diagonal matrix having the same diagonal elements with $\mathbf{G}$.

2) Acquirement of knowledge-induced solutions: With $\mathbf{W}_k$, the search experience from the optimized solutions to the source problem $\mathbf{PS}^{t_{max}}$ can be transferred into the current iterations for resolving the target problem to improve the solution quality. As $\mathbf{W}_k$ is a connective mapping between $\mathbf{PS}^t$ and $\mathbf{P}^t$, we can obtain the knowledge-induced solution set $\mathbf{Z}^t$ as

$$\begin{aligned} \mathbf{Z}^t &= \mathbf{W}\Phi(\mathbf{PS}^{t_{max}}) \\ &= \mathbf{W}_k\Phi(\mathbf{X})^T\Phi(\mathbf{PS}^{t_{max}}) \\ &= \mathbf{W}_k\mathbf{K}(\mathbf{X}, \mathbf{PS}^{t_{max}}). \end{aligned} \tag{10}$$

3) Sparse constraint: To guarantee the sparsity characteristics of the acquired knowledge-induced solutions, we propose a sparse constraint strategy by finding the positions of zero elements in $\mathbf{P}^t$ and setting the elements at the same position in Z^t to be zero. An example is depicted in Fig. 2. Specifically, we firstly find the locations of all zero elements in $\mathbf{P}_i^t$: $\mathbf{supp} = \{z|[\mathbf{P}^t]_{i,z} = 0\}$, where $[\cdot]_{i,z}$ represents the element in the $i-$th row, $z-$th column. Then, the sparsified knowledge-induced solution set $\mathbf{T}^t$ can be obtained as

$$[\mathbf{T}^t]_{i,z} = \left\{ \begin{array}{cc} [\mathbf{Z}^t]_{i,z}, & z \notin \mathbf{supp}, \\ 0, & z \in \mathbf{supp}. \end{array} \right\}, \quad i = \{1, 2, \ldots N\}. \tag{11}$$

$\mathbf{T}^t$ can not only possess the valuable knowledge extracted from the search experience for the past solved problem, but also inherit the sparse structure of $\mathbf{P}^t$.

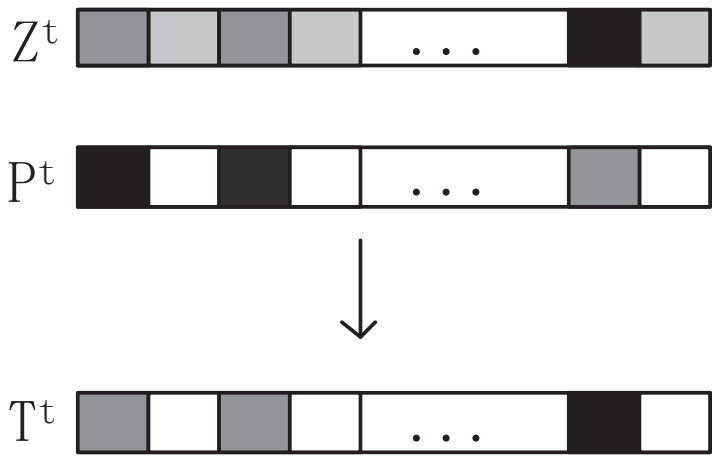

Fig. 2. Illustration of sparse constraint. Each lattice represents an entry of the estimated signal, where the white ones denote zero values, and the uncolored are the nonzero entries.

4) Selection: The selection operator of the MOSR solver is implemented in $\mathbf{P}^t \cup \mathbf{T}^t$ to select N elitism solutions for the next generation. If the knowledge transfer is beneficial, some of the sparsified learning-induced solutions survive in the selection procedure; otherwise, solutions in $\mathbf{T}^t$ will not be sent to the next generation, which avoids negative transfer.

IV. Experiments and Discussions

A. Experimental Settings

1) Test problems: We artificially generate a series of simulated signals as test problems as follows. Firstly, a k-sparse signal $\mathbf{x}$ is produced, in which the nonzero elements are sampled from a Gaussian distribution $\mathcal{N}(0,1)$. Then, a Gaussian matrix $\mathbf{A}$ is yielded and the measurement vector $\mathbf{b}$ is obtained by $\mathbf{b} = \mathbf{A}\mathbf{x}$. Lastly, the measurement $\mathbf{b}$ is corrupted by additive white Gaussian noise with elements from the normal distribution $N(0, 0.01)$. Each test problem involves three key parameters: (n, m, k). To better explore the effects of knowledge transfer, six complex test problems are randomly generated, with different parameters specified in Table I.

2) Settings of the sparse-constraint transfer learning operator: In this operator, the corruption probability p is set between 0.5 and 0.9 with an interval of 0.1 by doing the cross-validation on the population of a past problem in the first generation. The balancing parameter θ and the kernel function are suggested to be 10^3 and 'RBF' respectively according to [19]. The knowledge transfer is executed every five generations.

TABLE I
A LIST OF TEST PROBLEMS

Problem	(n, m, k)
P1	(1000, 200, 60)
P2	(1000, 300, 60)
P3	(1000, 300, 100)
P4	(1200, 250, 60)
P5	(1200, 350, 60)
P6	(1200, 350, 120)

3) MOSR solvers: Here, StEMO and a variant of ADEA (i.e., the ADEA without the reference vector adaptation, denoted as DEA) are employed as baseline solvers. We denote Δ as a MOSR solver, then its three versions are compared in this paper: the first version is the original solver Δ; the second and third versions are both equipped with the sparse-constraint knowledge transfer operator but receive the past experience from the source problems "P1-P3" and "P4-P6" respectively. We use different settings in the source problems because "P1-P3" and "P4-P6" have different dimensions and are heterogeneous. For convenience, we denote the second and third version as Δ-tr1 and Δ-tr2, respectively. For these solvers, their basic parameters are set as suggested in their original versions [9][14]. The population size of StEMO and ADEA is set to 50.

4) Terminate criterion: For a fair comparison, all methods stop running when the maximum function evaluations reach 5000 times. Each algorithm runs 15 times in each test case.

5) Evaluation criterion: All the versions of MOSR solvers are evaluated by hypervolume (HV) [23], which is the only parameter to measure the quality of a solution set. The larger HV values, the better the reconstruction quality achieved. For all scenarios, the reference sets for HV computation are all set to $(1000, 1.2)$, and the obtained HV results are normalized. In addition, the average reconstruction error (RE) and the RE variance under each test case are also compared, where $RE = \|\mathbf{x} - \mathbf{x}_G\|_2 / \|\mathbf{x}_G\|_2$, $\mathbf{x}$ and $\mathbf{x}_G$ are the estimated and ground-truth signals. Smaller RE can lead to better reconstruction quality.

B. Experimental Results and Discussions

To evaluate the convergence performance of the proposed knowledge transfer operator, the median HV values obtained by different versions of DEA and StEMO, across 15 independent runs with 5000 function evaluations are depicted in Fig. 3 and Fig. 4. In these two figures, the

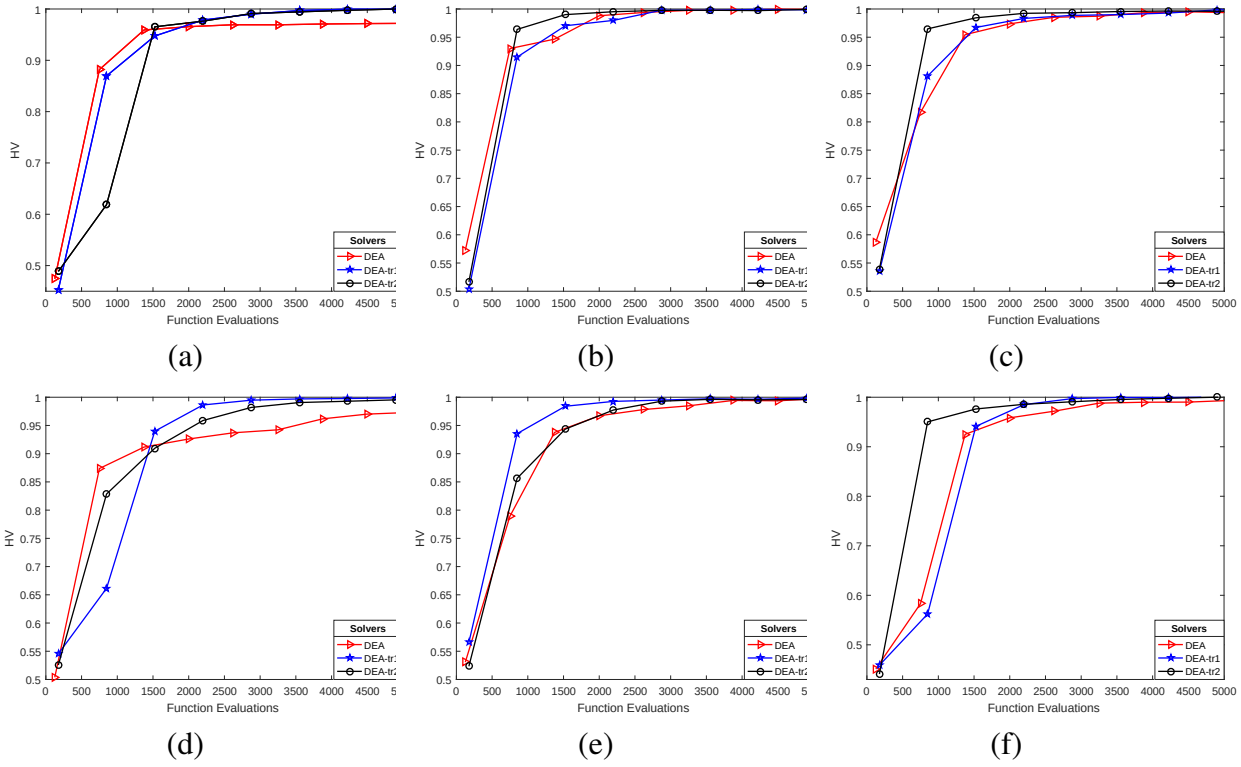

Fig. 3. Mean HV median values obtained by DEA for various source problems: “none”, “P1-P3” and “P4-P6”. (a)∼(f) are the target problems to be solved: (a) P1. (b) P2. (c) P3. (d) P4. (e) P5. (f) P6.

sub-figures (a)∼(f) correspond to the six target problems P1∼P6 respectively; the red, blue and black curve in each sub-figure indicates the original MOSR solver, the solvers with knowledge transfer from “P1-P3” and “P4-P6” respectively.

As can be seen from Fig. 3, DEA-tr1, and DEA-tr2 achieve higher accuracy or faster convergence than its original version in most problems. For P1 and P4, DEA-tr1 and DEA-tr2 obtain much larger HV values than DEA when the function evaluations arrive at 5000. For P3, P5 and P6, compared with DEA, the convergence of DEA-tr1 and DEA-tr2 is significantly faster, more than 500 function evaluations ahead. Except that DEA-tr1 converges slightly slower than DEA when solving P2, DEA-tr2 spends only 1500 function evaluations to generate the same HV values with those by DEA which takes about 2500 function evaluations.

Fig. 4 shows similar observations when StEMO is employed as a baseline solver. Except for P1, StEMO-tr1 and StEMO-tr2 converge much faster than StEMO, with a save of at least 500 function evaluations. When solving P1, three versions of StEMO have almost the same convergence speed, however, StEMO-tr1 and stEMO-tr2 provides larger HV values.

The corresponding average REs and the variances of all solver versions for P1$\sim$p6 are given in Table II and Table III respectively. In these tables, the symbols "$\approx$", "$+$" and "$-$" represent that the average RE of the corresponding solver is similar, smaller and larger than that of its baseline solver respectively. As can be observed from Table II, DEA-tr1 and DEA-tr2 have much better reconstruction quality when solving more complex problems (P1, P3, P4, and P6). For the remaining problems, compared with the baseline solver DEA, the RE results of DEA-tr1 and DEA-tr2 are statistically similar or slightly larger. These results are consistent with the median HV results in Fig. 3.

Similar results can be seen in Table III which presents the comparison results for the StEMO algorithms. Except for in P5, StEMO-tr1 and StEMO-tr2 achieve better reconstruction than the original StEMO solver, thanks to the knowledge transfer. The inferior result of StEMO-tr2 on P5 is probably caused nearby suboptimal solutions in identifying the knee points.

V. Conclusion

We presented a scheme for accelerating the convergence of MOSR solvers by introducing a sparse-constraint knowledge transfer operator that exploits the search experience from a previously solved problem. We employ the NFC technique to extract the feature mapping between the source and target problems, and then apply it to generate a set of knowledge-induced solutions for the target problem. A sparse constraint strategy is then proposed for sparsifying the obtained knowledge-induced solutions to reserve the sparse characteristics. Using StEMO and ADEA as the baseline MOSR solvers for several experimental problems, we demonstrate that the proposed operator can significantly improve the convergence performance of MOSR solvers by transferring knowledge across either homogeneous or heterogeneous problems.

TABLE II
Comparisons of average REs and variance obtained by ADEA for various source problems within 12000 function evaluations

Problems	DEA	DEA-tr1	DEA-tr2
P1	0.3421 ($\pm$1.76E-2)	0.2249 ($\pm$3.46E-3) $+$	0.2359 ($\pm$2.00E-3) $+$
P2	0.1307 ($\pm$1.43E-5)	0.1302 ($\pm$1.56E-5) $\approx$	0.1418 ($\pm$1.63E-5) $-$
P3	0.5261 ($\pm$7.67E-4)	0.4248 ($\pm$7.83E-4) $+$	0.4319 ($\pm$9.03E-4) $+$
P4	0.2725 ($\pm$7.77E-2)	0.2287 ($\pm$3.82E-3) $+$	0.2265 ($\pm$2.97E-3) $+$
P5	0.2027 ($\pm$1.87E-4)	0.2016 ($\pm$2.00E-4) $\approx$	0.2124 ($\pm$1.85E-4) $-$
P6	0.3031 ($\pm$5.07E-3)	0.2889 ($\pm$8.98E-4) $+$	0.2968 ($\pm$8.23E-4) $+$

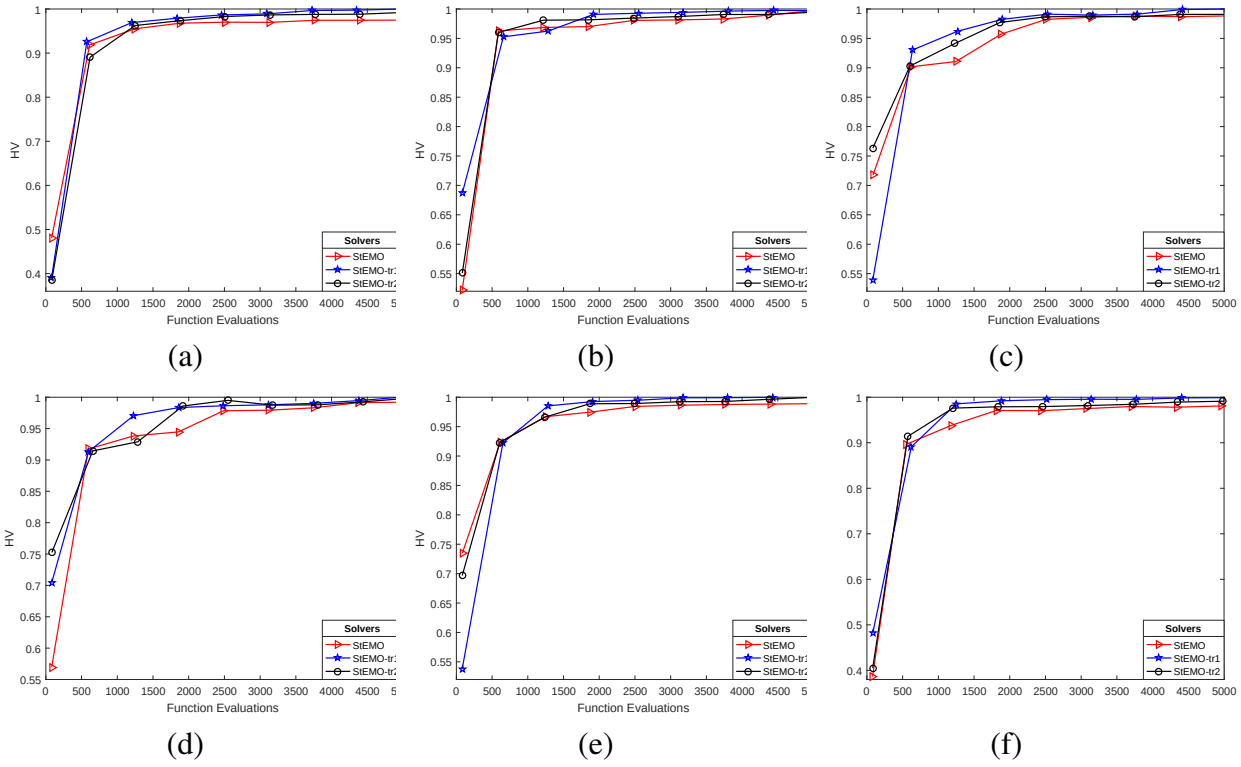

Fig. 4. Mean HV median values obtained by StEMO for various source problems: "none", "P1-P3" and "P4-P6". (a)∼(f) are the target problems to be solved: (a) P1. (b) P2. (c) P3. (d) P4. (e) P5. (f) P6.

TABLE III
COMPARISONS OF AVERAGE REs AND VARIANCE OBTAINED BY STEMO FOR VARIOUS SOURCE PROBLEMS WITHIN 12000 FUNCTION EVALUATIONS

Problems	StEMO	StEMO-tr1	StEMO-tr2
P1	0.7201 (±1.73E-2)	0.5665 (±1.01E-2) +	0.5923 (±9.76E-3) +
P2	0.4754 (±9.07E-3)	0.4304 (±3.90E-4) +	0.4699 (±3.01E-4) ≈
P3	0.6632 (±9.82E-3)	0.6201 (±1.30E-2) +	0.6329 (±9.93E-3) +
P4	0.6995 (±6.10E-2)	0.5794 (±9.82E-3) +	0.6280 (±8.82E-3) +
P5	0.5791 (±2.85E-4)	0.5227 (±1.19E-4) +	0.5921 (±1.97E-3) −
P6	0.5997 (±6.11E-3)	0.5706 (±5.23E-3) +	0.5992 (±7.84E-3) ≈

REFERENCES

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[3] Y. Jiao, B. Jin, and X. Lu, "Iterative soft/hard thresholding with homotopy continuation for sparse recovery," *IEEE Signal Processing Lett.*, vol. 24, no. 6, pp. 784–788, 2017.

[4] Z. Dong and W. Zhu, "Homotopy methods based on $l\_\{0\}$-norm for compressed sensing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1132–1146, 2018.

[5] 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, 2016.

[6] C. Liu, Q. Zhao, B. Yan, S. Elsayed, T. Ray, and R. Sarker, "Adaptive sorting-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, 2018.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[8] 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, 2007.

[9] L. Li, X. Yao, R. Stolkin, M. Gong, and S. He, "An evolutionary multiobjective approach to sparse reconstruction," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 827–845, 2014.

[10] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model Sim.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[11] B. Yan, Q. Zhao, Z. Wang, and X. Zhao, "A hybrid evolutionary algorithm for multiobjective sparse reconstruction," *Signal Image Video P.*, pp. 1–8, 2017.

[12] Y. Zhou, S. Kwong, H. Guo, X. Zhang, and Q. Zhang, "A two-phase evolutionary approach for compressive sensing reconstruction," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2651–2663, 2017.

[13] H. Li, Q. Zhang, J. Deng, and Z.-B. Xu, "A preference-based multiobjective evolutionary approach for sparse optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1716–1731, 2018.

[14] B. Yan, Q. Zhao, Z. Wang, and J. A. Zhang, "Adaptive decomposition-based evolutionary approach for multiobjective sparse reconstruction," *Inform. Sciences*, vol. 462, pp. 141–159, 2018.

[15] A. Gupta, Y.-S. Ong, and L. Feng, "Insights on transfer optimization: Because experience is the best teacher," *IEEE Trans. Emerg. Topics Comput.Intell.*, vol. 2, no. 1, pp. 51–64, 2018.

[16] L. Feng, Y.-S. Ong, S. Jiang, and A. Gupta, "Autoencoding evolutionary search with learning across heterogeneous problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 760–772, 2017.

[17] M. Jiang, Z. Huang, Q. Liming, W. Huang *et al.*, "Transfer learning based dynamic multiobjective optimization algorithms," *IEEE Trans. Evol. Comput.*, 2017.

[18] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, 2011.

[19] P. Wei, Y. Ke, and C. K. Goh, "Deep nonlinear feature coding for unsupervised domain adaptation." in *IJCAI*, 2016, pp. 2189–2195.

[20] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Conf. Neural Inf. Process. Syst.*, 2007, pp. 513–520.

[21] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," *Proc. 29th Int. Conf. Mach. Learn.*, 2012.

[22] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J. Mach. Learn. Res.*, vol. 2, no. Nov, pp. 67–93, 2001.

[23] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.