Multispace Evolutionary Search for Large-Scale Optimization With Applications to Recommender Systems

Liang Feng, Qingxia Shang, Yaqing Hou, Member, IEEE, Kay Chen Tan, Fellow, IEEE, and Yew-Soon Ong, Fellow, IEEE

Abstract—Large-scale optimization is vital in today’s artificial intelligence (AI) applications for extracting essential knowledge from huge volumes of data. In recent years, to improve the evolutionary algorithms used to solve optimization problems involving a large number of decision variables, many attempts have been made to simplify the problem solution space of a given problem for the evolutionary search. In the literature, the existing approaches can generally be categorized as decomposition-based methods and dimension-reduction-based methods. The former decomposes a large-scale problem into several smaller subproblems, while the latter transforms the original high-dimensional solution space into a low-dimensional space. However, it is worth noting that a given large-scale optimization problem may not always be decomposable, and it is also difficult to guarantee that the global optimum of the original problem is preserved in the reduced low-dimensional problem space. This article, thus, proposes a new search paradigm, namely the multispace evolutionary search, to enhance the existing evolutionary search methods for solving large-scale optimization problems. In contrast to existing approaches that perform an evolutionary search in a single search space, the proposed paradigm is designed to conduct a search in multiple solution spaces that are derived from the given problem, each possessing a unique landscape. The proposed paradigm makes no assumptions about the large-scale optimization problem of interest, such as that the problem is decomposable or that a certain relationship exists among the decision variables. To verify the efficacy of the proposed paradigm, comprehensive empirical studies in comparison to five state-of-the-art algorithms were conducted using the CEC2013 large-scale benchmark problems as well as an AI application in e-commerce, i.e., movie recommendation.

Impact Statement—Today, optimization problems in intelligent applications could be very challenging, since the objective functions may have thousands, or even millions of inputs, and the function is often non-decomposable. To solve large-scale optimization problems, while existing evolutionary search methods mainly consider problem decomposition and problem transformation, the proposed paradigm in this paper provides a new perspective by creating multiple problem search spaces, and conducting evolutionary search on the created spaces simultaneously. The proposed paradigm is able to explore the useful traits embedded in different search spaces of the encountered large-scale optimization problems, and thus can provide enhanced problem-solving performance for large-scale optimization. Empirical studies on both commonly used large-scale optimization benchmarks as well as the popular AI application in e-commerce, i.e., movie recommendation, show that the proposed paradigm obtained significantly superior optimization performance over decomposition-based, transformation-based, and traditional recommendation algorithms.

Index Terms—Evolutionary search, knowledge transfer, large-scale optimization, multispace optimization.

I. INTRODUCTION

A N EVOLUTIONARY algorithm (EA) is a stochastic optimization search method that takes inspiration from the theory of natural biological evolution. It starts with a population of individuals that undergo reproduction, including crossover and mutation, to produce offspring. This procedure is executed iteratively and terminated when a predefined condition is satisfied. In contrast to traditional optimization approaches, such as calculus-based and enumerative strategies, an EA contains flexible procedures and is robust to changing problem circumstances. In recent decades, EAs have attracted significant research attention, and have been successfully applied in many complex applications, such as scheduling in logistics [1], [2], image processing [3], [4], and the architecture optimization of deep neural networks [5], [6].

Today, because of the exponential growth of the volume of data in big data applications, large-scale optimization problems (i.e., optimization problems with a large number of decision variables) have become ubiquitous in many AI applications [7]–[11]. Because increasing the decision variables not only leads to an exponential increase in the problem solution space, but also results in the growth of the computational cost involved.
in the solution search and evaluation process, the performance of an EA decreases significantly for large-scale optimization problems [12]–[15]. To improve the scalability of EA for solving problems with large-scale dimensionality, many research efforts have been conducted to simplify the search space of a large-scale optimization problem [16]–[21]. According to a recent survey [22], the existing approaches can generally be categorized into the following two groups: decomposition-based approaches and dimension-reduction-based approaches. Specifically, decomposition-based methods follow the principle of divide-and-conquer and decompose a problem into several relatively small subcomponents that are optimized concurrently. Furthermore, dimension-reduction-based methods reduce the number of decision variables by selecting a set of principal variables or transforming the high-dimensional space into a space with fewer dimensions. However, despite the successes enjoyed by these two classes of methods, it is worth noting here that because decomposition-based methods rely on the accurate detection of the relationships between decision variables, this type of method may fail when used for large-scale optimization problems that possess complex variable interactions, or are not even decomposable. Moreover, reducing the dimensionality of the search space could lose important information for optimization, and it is difficult to guarantee that the global optimum or high-quality solutions are preserved in the reduced solution space.

Evolutionary multitasking (EMT) is a recently emerging research topic in the field of evolutionary computation [23]–[26]. In contrast to a traditional single-task evolutionary search, EMT conducts an evolutionary search on multiple tasks, each corresponding to a particular optimization problem. It aims to improve the convergence characteristics of an evolutionary search across multiple optimization problems at once by seamlessly transferring knowledge among the tasks. In the literature, the efficacy of EMT has been verified on sets of continuous, discrete tasks, and mixtures of continuous and combinatorial tasks [24], [27]–[30]. Inspired by this, in the context of large-scale optimization, besides replacing the original problem space with a simplified space, the advantage of a simplified search space could also be obtained by configuring the constructed solution space as an auxiliary task of the original problem, under EMT. In this manner, as the solution space of the original problem serves as one task in EMT, there is no assumption and requirement for the relationships between decision variables, and the existence of global optimum or high-quality solutions is guaranteed.

Keeping the abovementioned in mind, this article proposes a new evolutionary search paradigm, namely the multispace evolutionary search, for solving large-scale optimization problems. In particular, for a given large-scale optimization problem, besides the original problem space, multiple simplified solution spaces are derived for the given problem, which possess unique landscapes. Furthermore, instead of conducting an evolutionary search on the problem space, evolutionary searches are concurrently performed on both the original and constructed simplified spaces of the given problem. By transferring useful traits while the search progresses online across different spaces, via EMT, an enhanced problem-solving process can be obtained. As the optimization of the large-scale optimization problem is maintained in the given problem space, and the simplified solution spaces serve to provide biases to guide the search in the original problem space, the proposed multispace evolutionary search paradigm provides more flexibilities of the construction of the simplified problem spaces, which does not rely on the assumption that a problem is decomposable or reducible. To evaluate the efficacy of the proposed paradigm for large-scale optimization, comprehensive empirical studies on both the CEC2013 large-scale optimization benchmarks and the AI recommendation application were conducted using five state-of-the-art representative methods for large-scale optimization, and the results were analyzed.

The rest of this article is organized as follows. Section II begins with a review of the literature on the existing decomposition- and dimension-reduction-based approaches for solving large-scale optimization problems. A brief introduction of EMT, which serves as the optimization engine of the proposed multispace evolutionary search, is also presented in Section II. Furthermore, Section III provides the details of the proposed multispace evolutionary search for solving large-scale optimization problems. Sections IV and V discusses the comprehensive empirical studies that were conducted on the CEC2013 large-scale optimization benchmarks and the AI recommendation application, using five state-of-art algorithms for large-scale optimization, respectively. Finally, Section VI concludes this article.

II. PRELIMINARY

This section first presents a review of the literature on the existing decomposition- and dimension-reduction-based approaches for solving large-scale optimization problems. In the following, a brief introduction of the EMT paradigm is provided.

A. Existing Approaches for Simplifying Search Space of Large-Scale Optimization Problems

According to recent surveys in the literature [8], [22], [31], the existing approaches to simplifying the search space of a given large-scale optimization problem can generally be categorized as decomposition-based approaches, and dimension-reduction-based methods. In particular, the decomposition-based approaches are also known as divide-and-conquer approaches in evolutionary computation and mainly involve cooperative co-evolution (CC) algorithms, which decompose a given large-scale optimization problem into several smaller subproblems and then optimize each subproblem separately using different EAs. Generally, decomposition-based approaches consist of three major steps. First, by considering the structure of the underlying decision variable interactions, the original $D$-dimensional problem is exclusively divided into $N$ $d_i$-dimensional subproblems, where $\sum_{i=1}^N d_i = D$. Next, each subproblem is solved by a particular EA. Finally, the $d$-dimensional solutions to these subproblems are merged to form the $D$-dimensional complete solution for the original problem. It is straightforward to see how the decomposition of the problem is essential to the performance of CC algorithms, and how an inappropriate decomposition of the
decision variables may even lead to a deteriorated optimization performance [31], [32]. Particular examples in this category include strategies that randomly divide the variables into groups without taking the variable interaction into consideration [33]–[35], approaches that make use of evolutionary information to learn variable interdependency and then divide variables into groups [36], [37], and static decomposition methods that are performed before conducting an evolutionary search based on the detection of variable interaction [32], [38]–[41].

On the other hand, instead of decomposing the solution space of the given problem, a dimension-reduction-based approach attempts to create a new solution space with lower dimensionality from the original solution space. The evolutionary search is then performed on the newly created low-dimension space, and the obtained solution is mapped back to the original space for evaluation. Generally, the existing approaches perform dimension reduction either by selecting a subset of the original decision variables or transforming the original solution space into a low-dimensional solution space. As can be observed, the preservation of important information for guiding the search toward high-quality solutions in the reduced solution space plays a key role in determining the performance of a dimension-reduction-based approach. Examples belonging to this class include the random matrix projection-based estimation of distribution algorithm, which introduced an ensemble of random projections to low dimensions of the set of fittest search points [42]; random embedding-based approach for large-scale optimization problems with low effective dimensions, which improve the scalability of the simultaneous optimistic optimization by projecting the problem space to a low-dimensional space via random embedding [43]; multiagent system assisted embedding for large-scale optimization that improved the reliability of random embedding via multiagent system for large-scale optimization [44]; solving large-scale multiobjective optimization via problem reformulation, which transformed the search in the original problem space into a low-dimensional single-objective optimization space [14], and the framework for large-scale optimization based on problem transformation, which optimizes the weight values of groups of decision variables instead of the high-dimensional decision variables directly [13].

Although the abovementioned methods have shown good performances in solving large-scale optimization problems, there are two main drawbacks with these two categories of methods. First, because decomposition-based methods rely heavily on the accurate detection of decision variable interactions, these methods may fail on large-scale optimization problems with complex variable interactions or that are not decomposable. Second, although dimension reduction may not rely on variable interaction, it is difficult to guarantee that the global optimum or high-quality solutions are preserved in the reduced space. However, because a simplified solution space can provide useful information for efficient and effective problem solving, it is desirable to develop new search paradigms for large-scale optimization that can leverage the advantage of simplified solution spaces without the limitations discussed earlier.

B. Evolutionary Multitasking

Consider a situation where \( K \) optimization tasks are to be performed. EMT has been defined in the literature as an optimization paradigm that solves multiple optimization tasks at the same time, with the aim of improving the problem-solving performance across tasks by seamlessly transferring knowledge between them [23]. In particular, as depicted in Fig. 1, let \( f_i : X_i \rightarrow \mathbb{R} \) be a global optimization task on a compact subset \( X_i \subset \mathbb{R}^D_i \), with objective \( x_i^* = \arg \min_{x_i \in X_i} f_i(x_i) \). The input of EMT is a set of optimization tasks \( IS \) (Input Set): \( \{f_1, \ldots, f_i, \ldots, f_K\} \), where \( K \) denotes the number of tasks. Please note that each task \( f_i \) may possess unique dimensionality \( D_i \). The output of EMT is then given by the set of optimized solutions output set (OS): \( \{x_1^*, \ldots, x_i^*, \ldots, x_K^*\} \).

In contrast to the traditional single-task optimization, EMT involves automatically exploiting and transferring the latent synergies between distinct (but possibly similar) optimization problems while the optimization progresses online, which could eventually lead to enhanced problem-solving on all the tasks. For large-scale optimization, if each task under EMT corresponds to a unique solution space for the given optimization problem, the useful traits found in different spaces could be transferred across these spaces via EMT, producing a more efficient and effective evolutionary search process. Inspired by this, a multispace evolutionary search paradigm is proposed for large-scale optimization, which will be discussed in detail in the following section.

III. PROPOSED MULTISPACE EVOLUTIONARY SEARCH FOR LARGE-SCALE OPTIMIZATION

This section presents the details of the proposed multispace evolutionary search for large-scale optimization. In particular, the outline of the proposed paradigm is presented in Fig. 2. For a given problem of interest, besides the original problem space, a simplified problem space for the given problem is first created. Next, the mapping between these two problem spaces is learned, which will be used for knowledge transfer across spaces during the evolutionary search process via EMT. Furthermore, by treating these two problem spaces as two tasks, evolutionary searches can be conducted on the tasks concurrently. As can be observed in the figure, knowledge transfer will be performed across tasks while the evolutionary search progresses online.
A. Construction of the Simplified Problem Space

Because the simplified problem space serves as an auxiliary task of a given problem of interest, there are generally no particular constraints on the construction of the simplified space. Therefore, the existing approaches proposed in the literature, such as random embedding [43], dimension reduction [45], or even search space decomposition [33], [37] could be employed for constructing the space.

In this article, for simplicity, the dimension reduction is considered for constructing a simplified problem space, $P_s$, in the proposed multiscpace evolutionary search paradigm. In particular, to generate initial population $PoP_s$ of the evolutionary search in $P_s$, initial population $PoP$ is first sampled in original problem space $P$, which is routine [46]. Next, the obtained $PoP$ in $P$ will undergo dimension reduction\(^1\) with dimension $d_s$ to generate $PoP_s$ for the evolutionary search in $P_s$.

B. Learning of Mapping Across Problem Spaces

Once the simplified problem space has been constructed, the mappings across simplified problem space $P_s$ and original problem space $P$ have to be learned, to allow the useful traits found in each space to be transferred across spaces toward efficient and effective problem solving for large-scale optimization.

In this article, the mappings across $P_s$ and $P$ are learned using labeled data from each space via supervised learning. In particular, as discussed in Section III-A, $PoP_s$ is generated by performing dimension reduction of $PoP$. Therefore, each solution in $PoP_s$ has a unique corresponding solution in $PoP$. This correspondence, thus, provides the label information to connect spaces $P_s$ and $P$. Taking this cue, by configuring $T$ and $S$ as $PoP$ and $PoP_s$, respectively, the mapping $M_{P_s \rightarrow P}$: $\mathcal{R}^{d_s} \rightarrow \mathcal{R}^{d}$ ($M_{P_s \rightarrow P}$ is a $d \times d_s$ matrix; $d_s$ and $d$ are the dimensions of the simplified and original problem spaces, respectively,) from simplified space $P_s$ to original problem space $P$ can then be approximated by minimizing the squared reconstruction loss\(^2\), which is given by the following:

$$L_{sq}(M) = \frac{1}{2N} \sum_{i=1}^{N} ||p_i - M \times q_i||^2$$  \hspace{1cm} (1)

where $N$ denotes the number of solutions in $S$ and $T$\(^3\), $q_i$ is the solution in $S$, and $p_i$ gives the solution in $T$, which corresponds to $q_i$.

Furthermore, to simplify the notation, it is assumed that a constant feature is added to the input, that is, $p_i = [p_i; 1]$ and $q_i = [q_i; 1]$, and an appropriate bias is incorporated within the mapping. $M = [M, b]$. The loss in (1) is then reduced to the matrix form

$$L_{sq}(M) = \frac{1}{2N} \text{tr}((T - M \times S)^T(T - M \times S))$$  \hspace{1cm} (2)

where $\text{tr}(\cdot)$ and $T$ denote the trace operation and transpose operation of a matrix, respectively. The solution of (2) can be expressed as the well-known closed-form solution for the ordinary least squares [47], which is given by the following:

$$M = (T \times S^T)(S \times S^T)^{-1}.$$  \hspace{1cm} (3)

Finally, it is straightforward to see that the mapping $M_{P \rightarrow P_s}$, \( (a \times d_s \times d_m) \) \text{ matrix from space } $P$ to $P_s$ can also be learned via (3) by configuring $T$ and $S$ as $PoP_s$ and $PoP$, respectively.

\(^1\)Without lose of generality, any dimension reduction method can be applied here.
\(^2\)In this article, as $S$ is generated via dimension reduction using $T$, this mapping could be directly obtained in the dimension reduction process. However, the proposed learning method is general in cases when only solutions in both the simplified and the original problem spaces are given.
\(^3\)S and T are, thus, represented by a $d_s \times N$ matrix and a $d \times N$ matrix, respectively.
C. Knowledge Transfer Across Problem Spaces

With the learned mappings $M_{P_s \rightarrow P}$ and $M_{P \rightarrow P_s}$ across the simplified and original problem spaces, a knowledge transfer across these two spaces can easily be conducted by the simple operation of matrix multiplication. In particular, for a knowledge transfer from $P_s$ to $P$, suppose this process occurs every generation $G_s$. First, the $Q$ best solutions$^4$ are selected in terms of the fitness values from the population of the simplified problem space, denoted by $S_s$, which is a $d_s \times Q$ matrix. Next, the transferred solutions, $TS_{P_s \rightarrow P}$, are obtained by $M_{P_s \rightarrow P} \times S_s$. Finally, the solutions in $TS_{P_s \rightarrow P}$ are injected into the population of the original problem space to undergo natural selection for the next generation.

On the other hand, every generation $G_t$ knowledge transfer also occurs from the original problem space to the simplified problem space. In particular, the $P$ best solutions in terms of fitness values are first selected from the population of the original problem space, which are labeled as $S'_t$, and are a $d \times P$ matrix. Subsequently, the transferred solutions $TS_{P \rightarrow P_s}$ can be obtained by $M_{P \rightarrow P_s} \times S'_t$. Furthermore, the solutions in $TS_{P \rightarrow P_s}$ are inserted into the population of the simplified problem space with natural selection.

Moreover, after the knowledge transfer process, the updated population of the simplified problem space is further transformed back to the original problem space, and archived in $A_s$. The repeated solutions in $A_s$ are removed. As can be observed, $A_s$ preserves the search traces in the present simplified problem space, which will be used for the reconstruction of a new simplified space. That is discussed in detail in the following section.

D. Reconstruction of the Simplified Space

To explore the usefulness of diverse auxiliary tasks for large-scale optimization, instead of using one fixed simplified problem space, the proposal is made to build multiple simplified problem spaces periodically while the evolutionary search progresses online. In particular, if the reconstruction of the simplified problem space occurs every generation $G_r$, the dimension reduction used in Section III-A is considered here again to reconstruct a simplified problem space with a new set of solutions in the original problem space. Furthermore, in order to preserve the useful traits found in the last simplified space, the solutions in archive $A_s$ are used as the new set of solutions and subjected to the dimension reduction to construct a new $P_s$. Subsequently, the solutions in $A_s$ and corresponding mapped solutions in $P_s$ will be used to learn mappings $M_{P_s \rightarrow P}$ and $M_{P \rightarrow P_s}$ across problem spaces $P$ and $P_s$ as discussed in section III-B.

Finally, the population of the simplified problem space is also reinitialized in the new $P_s$, which is shown in detail in Algorithm 1.

E. Summary of Proposed Multispace Evolutionary Search

A summary of the proposed multispace evolutionary search for large-scale optimization is presented in Algorithm 2. As can be observed, the proposed algorithm starts with the construction of simplified problem spaces ($P_s$), and the learning of mappings (i.e., $M_{P_s \rightarrow P}$ and $M_{P \rightarrow P_s}$) across the simplified ($P_s$) and original problem ($P$) spaces. Next, the reproduction operators are conducted to generate offspring solutions in $P_s$ and $P$, respectively. The evaluations of solutions in both $P_s$ and $P$ are performed using the given problem objective function in the original problem space $P$ (see lines 8–9 in Algorithm 2). Moreover, while the EMT progresses online, the knowledge transfer across $P_s$ and $P$ occurs in every $G_t$ generation (see lines 10–13 in Algorithm 2), and the reconstruction of a new simplified problem space $P_s$ is performed in every $G_r$ generations (see lines 15–18 in Algorithm 2). The proposed EMT search process proceeds iteratively until a certain stopping criteria satisfied. Furthermore, archive $A_s$ on lines 12–13 of Algorithm 2 is used to store the nonrepeating search traces in the simplified problems with solution representation in the original problem space. Without loss of generality, the volume of $A_s$ can be configured as needed. However, in this article, for simplicity, the volume of $A_s$ is configured as $5 \times NP$, where NP denotes

$^4$The fitness values of solutions in $P_s$ are evaluated by transforming these solutions back to the original problem space and using the given problem objective function.

$^5$In this study, for simplicity, the dimension of $P_s$ ($d_s$) was kept unchanged.
the population size of the evolutionary search. Once the search traces exceed the volume of \( A_s \), only the latest search traces are archived.

### IV. Empirical Study

This section discusses the results of comprehensive empirical studies that were conducted to evaluate the performance of the proposed multispace evolutionary search paradigm on the commonly used large-scale optimization benchmarks, compared to several state-of-the-art algorithms proposed in the literature.

#### A. Experimental Setup

In this study, the commonly used CEC2013 large-scale optimization benchmark [48], which contains 15 functions with diverse properties, was used to investigate the performance of the proposed multispace evolutionary search. As summarized in Table I, according to [48], the benchmark consists of both unimodal and multimodal minimization functions, which can be generally categorized into the following five classes:

1. fully separable functions;
2. partially additive separable functions I;
3. partially additive separable functions II;
4. overlapping functions;
5. fully nonseparable functions.

Furthermore, except for functions F13 and F14, all the functions have a dimensionality of 1000. Because of the overlapping property, functions F13 and F14 both have 905 decision variables. For more details on the CEC2013 large-scale optimization benchmark, interested readers can refer to [48].

Next, to verify the efficacy of the proposed multispace evolutionary search (referred to as MSES hereafter) for large-scale optimization, five state-of-the-art methods for addressing large-scale optimization, including decomposition-based CC and nondecomposition-based approaches, were considered as the baseline algorithms for comparison. In particular, the CC approaches included the recursive decomposition methods proposed by Sun et al. (called RDG) [40] and (called RDG3) [41], and an improved variant of the differential grouping algorithm introduced by Omidvar et al., which is called DG2 [49]. The nondecomposition-based approaches included the level-based learning swarm optimizer proposed by Yang et al. (called DLLSO) [50], and the random embedding-based method proposed by Hou et al. (called MeMAO) [44]. Furthermore, in these compared algorithms, it should be noted that different evolutionary search methods were used as the basic optimizer. For example, RDG, RDG3, and DG2 employed the self-adaptive differential evolution with neighborhood search (SaNSDE) [40], [49] as the optimizer, while MeMAO considered the classical differential evolution (DE) method as the optimizer [44]. Rather than using differential evolution, DLLSO used the particle swarm optimizer as the basic search method [50]. For a fair comparison to the different baseline algorithms, the optimizer for each space in the proposed MSES was kept consistent with the optimizer used in the compared algorithm. Furthermore, the basic optimizers performed in the original problem space are also included as baseline algorithms for comparison, which serves as the ablation studies. Lastly, the parameter and operator settings of all the compared algorithms and the proposed MSES were kept the same as those in [40], [41], [49], [50], and [44], which are summarized as follows:

1. population size: population size \( \text{NP} = 50, 100, \) and \( 500 \) for optimizers SaNSDE, DE, and DLLSO, respectively;
2. independent number of runs: \( \text{runs} = 25 \) for all compared algorithms;
3. maximum number of fitness evaluations: \( \text{Max_FEs} = 3E + 06 \);

\[ \text{Algorithm 2: Pseudo Code of the Proposed Multi-Space Evolutionary Search for Large-Scale Optimization.} \]

**Input:** \( \text{P} \): Given the problem space of interest; \( d_s \): dimensionality of the simplified problem space; \( G_i \): interval of knowledge transfer across problem spaces; \( G_r \): interval of simplified space reconstruction

**Output:** \( s^* \): optimized solution of the given problem

1: Begin
2: Construct the simplified problem space \( \text{P}_s \) with dimension \( d_s \) of given problem \( \text{P} \);
3: Learn the mappings \( \text{M}_{\text{P}_s \rightarrow \text{P}} \) and \( \text{M}_{\text{P} \rightarrow \text{P}_s} \) across \( \text{P}_s \) and \( \text{P} \);
4: \( \text{gen} = 1; A_s = \emptyset \)
5: while terminate condition is not satisfied do
6: \( \text{gen} = \text{gen} + 1 \);
7: Perform reproduction operators (e.g., crossover and mutation) for \( \text{P}_s \) and \( \text{P} \), respectively;
8: Transform solutions in \( \text{P}_s \) back to \( \text{P} \);
9: Perform natural selection using the given problem objective function for both \( \text{P}_s \) and \( \text{P} \) in problem space \( \text{P} \);
10: if \( \text{mod}(\text{gen}, G_i) = 0 \) then
11: Perform knowledge transfer across \( \text{P}_s \) and \( \text{P} \);
12: Transform the population of \( \text{P}_s \) back to \( \text{P} \), and archive the population in \( A_s \);
13: Remove the repeated solutions in \( A_s \);
14: end if
15: if \( \text{mod}(\text{gen}, G_r) = 0 \) then
16: Reconstruct the new simplified problem space \( \text{P}_s \) with dimension \( d_s \) of solutions in \( A_s \);
17: Learn the new mappings \( \text{M}_{\text{P}_s \rightarrow \text{P}} \) and \( \text{M}_{\text{P} \rightarrow \text{P}_s} \) across the new constructed \( \text{P}_s \) and \( \text{P} \);
18: Re-initialize the populations in \( \text{P}_s \) using the new learned \( \text{M}_{\text{P} \rightarrow \text{P}_s} \);
19: end if
20: end while
21: End
II

TABLE II

AVERAGED OBJECTIVE VALUES AND STANDARD DEVIATIONS OBTAINED BY PROPOSED MSES AND COMPARED BASELINE ALGORITHMS

| Problems | MSES <span class="subscript">comp</span> | SaNSDE | DGI | RDG | RDG3 |
|----------|---------------------------------|--------|-----|-----|-----|
| F1       | 1.360±0.125×0.06               | 4.706±0.27×0.05 | 5.209±0.13×0.03 | 2.160±0.13×0.03 | 6.570±0.12×0.03 |
| F2       | 1.800±0.135±0.02               | 5.040±0.17×0.03 | 5.240±0.13×0.03 | 2.040±0.13×0.03 | 4.250±0.14×0.03 |
| F3       | 2.000±0.12×0.03                | 2.060±0.17×0.03 | 2.160±0.13×0.02 | 2.140±0.13×0.02 | 2.130±0.13×0.02 |
| F4       | 1.290±0.14×0.03                | 5.670±0.27×0.05 | 5.190±0.13×0.03 | 2.040±0.13×0.03 | 4.250±0.14×0.03 |
| F5       | 4.520±0.48×0.05                | 2.790±0.42×0.05 | 1.830±0.48×0.05 | 2.000±0.48×0.05 | 2.930±0.48×0.05 |
| F6       | 1.000±0.51×0.03                | 1.050±0.51×0.03 | 1.060±0.51×0.03 | 1.060±0.51×0.03 | 1.060±0.51×0.03 |
| F7       | 1.800±0.13×0.03                | 6.830±0.13×0.03 | 6.550±0.13×0.03 | 6.490±0.13×0.03 | 6.510±0.13×0.03 |
| F8       | 7.140±0.54×0.12               | 5.715±1.11×0.13 | 4.590±1.11×0.13 | 4.160±1.11×0.13 | 4.170±1.11×0.13 |
| F9       | 6.480±0.30×0.07                | 2.670±0.84×0.07 | 4.940±0.30×0.07 | 4.930±0.30×0.07 | 4.930±0.30×0.07 |
| F10      | 9.040±7.54×0.02               | 9.290±7.54×0.02 | 9.450±7.54×0.02 | 9.450±7.54×0.02 | 9.450±7.54×0.02 |
| F11      | 5.150±8.35×0.06               | 5.080±8.35×0.06 | 6.170±8.35×0.06 | 5.500±8.35×0.06 | 5.920±8.35×0.06 |
| F12      | 1.680±0.18×0.03               | 4.190±0.79×0.03 | 2.630±0.18×0.03 | 4.380±0.18×0.03 | 1.930±0.79×0.03 |
| F13      | 1.510±7.28×0.06               | 5.880±8.07×0.06 | 1.820±9.38×0.06 | 3.020±9.38×0.06 | 9.490±2.92×0.06 |
| F14      | 6.850±7.52×0.06               | 6.940±8.50×0.06 | 5.260±8.50×0.06 | 3.730±8.19×0.06 | 3.360±10.39×0.06 |

4) number of solutions to be transferred across spaces in MSES: \( P = Q = 0.2 \times NP \);
5) Interval of knowledge transfer across problem spaces: \( G_t = 1 \);
6) interval of simplified space reconstruction: \( G_r = 10 \);
7) dimensionality of simplified problem space: \( d_s = 600 \);
8) dimension reduction method: principal component analysis (PCA) [51];
9) size of \( A_s \): \( A_{s\text{size}} = 5 \times NP \).

B. Results and Discussion

This section presents and discusses the performance of the proposed MSES in comparison to those of the existing state-of-the-art approaches on the CEC2013 large-scale benchmark functions in terms of the solution quality and search efficiency.

1) Solution Quality: Table II tabulates the results with respect to the averaged objective values and standard deviations obtained by all the compared algorithms over 25 independent runs. In particular, based on the evolutionary solver employed for the search (e.g., SaNSDE, PSO, and DE), the comparison results are divided into three groups, with each group sharing the same evolutionary solver. The best performance in each comparison group is highlighted using a bold font in the table. Furthermore, in order to obtain a statistical comparison, a Wilcoxon rank sum test with a 95% confidence level was conducted on the experimental results, where \( +, \) \( - \), and \( \approx \) denote that the compared algorithm is statistically significantly better than, significantly worse than, or similar to the proposed MSES, respectively. As can be observed in the table, in all three comparison groups, when using different evolutionary search methods as the optimizer, the proposed MSES obtained a superior solution quality in terms of the averaged objective value on most of the problems compared to the other algorithms. In the comparison groups using SaNSDE and PSO as the optimizers, the proposed approach, that is, MSES\textsubscript{SaNSDE} and MSES\textsubscript{PSO}, lost to the compared algorithms on large-scale benchmarks F1 and F2. Table I shows that F1 and F2 are fully separable functions. Moreover, F1 is based on a unimodal “Elliptic” function, and the search space of F2 is only within the range of \([-5, 5]\), which indicates the simplicity of the search spaces for these two functions. Therefore, the reason behind the obtained performance of the proposed method could be the simple use of dimension reduction for constructing the simplified problem space of these two different problems, which may destroy the separable information between decision variables. However, on the other more complex large-scale benchmarks, such as partially additive separable, overlapping, and fully nonseparable problems, where greater appropriate guidance is required for an effective search, the proposed MSES\textsubscript{SaNSDE} and MSES\textsubscript{PSO} achieved superior and competitive averaged objective values.
in contrast to DG2/RDG/RDG3 and DLLSO, respectively. On benchmarks F11, F13, and F14, only the proposed method was able to consistently find solutions with objective values of approximately $e + 07$ in both of these comparison groups. On 15 large-scale benchmarks, the proposed MSES$_{SaNSDE}$ and MSES$_{DLLSO}$ achieved significantly better averaged objective values on 13, 12, and 9 problems in contrast to DG2/RDG, RDG3, and DLLSO, respectively.

Furthermore, in the comparison group of that used DE as the optimizer, the proposed MSES$_{DE}$ obtained superior or competitive averaged objective values on all the large-scale benchmarks compared to MeMAO. In particular, on benchmarks such as F4 and F8, MSES$_{DE}$ achieved improvements of orders of magnitude in contrast to MeMAO. The objective values achieved on these benchmarks were even superior to those obtained using SaNSDE and PSO as the optimizers. On 15 large-scale benchmarks, the proposed MSES$_{DE}$ achieved significantly better averaged objective values on 13 problems in contrast to MeMAO.

In summary, because the proposed method used the same optimizer as the compared algorithms in each comparison group, and only differed in the search spaces, the superior solution quality observed in Table II confirmed the effectiveness of the proposed multispace evolutionary search for large-scale optimization.

2) Search Efficiency: This section presents the convergence graphs obtained by all the compared algorithms on all the large-scale benchmarks to assess the search efficiency of the proposed multispace evolutionary search for large-scale optimization. In particular, Figs. 3 and 4 show the obtained convergence graphs obtained on the fully separable functions, partially additive separable functions, overlapping functions, and fully nonseparable functions. In these figures, the $Y$-axis denotes the averaged objective values in log scale, while the $X$-axis gives the respective computational effort required in terms of the number of fitness evaluations. It can be observed from Figs. 3 and 4 that on benchmarks F1 and F2, the compared algorithm DLLSO obtained the best convergence performance. This indicates that on benchmarks with a relatively simple decision space, a search on the original problem space can efficiently find high-quality solutions using properly designed search strategies.

Moreover, on the other functions of the CEC 2013 benchmarks with more complex decision spaces (e.g., partially additive separable functions, overlapping functions, and fully nonseparable functions), where greater appropriate guidance is required for an efficient search for high-quality solutions, the proposed MSES obtained the best and competitive convergence performances in all three comparison groups. In particular, even on the fully separable function F3, because it is based on the complex “Ackley” function, the proposed MSES$_{SaNSDE}$ and MSES$_{DLLSO}$ obtained faster convergence over the compared algorithms that shared the same EA solvers. In addition, for functions, such as F8, F13, and F14, regardless of which EA was considered as the search optimizer, the proposed MSES obtained the best convergence performance in contrast to the baseline algorithms in all three comparison groups. Because the proposed MSES used the same search optimizers as the compared algorithms, the superior search speed obtained confirmed the efficiency of the proposed multispace evolutionary search for large-scale optimization.

Furthermore, to investigate the computational cost of the proposed multispace evolutionary search in terms of central processing unit (CPU) time, we also depict the CPU time and convergence graphs with respect to CPU time of all the compared algorithms on representative benchmark problems. In particular, Figs. 5 and 6 present the averaged CPU time and the convergence curves obtained by all the compared algorithms on benchmark F8 and F14, respectively. It can be observed from the figures that, as the proposed paradigm does not include processes like...
problem analysis and problem decomposition involved in exiting cooperative coevolutionary approaches, the CPU time consumed by the proposed multispace evolutionary search paradigm will not increase a lot when compared to the baseline evolutionary search algorithms.

Finally, to provide deeper insights into the superior performance obtained by the proposed MSES, considering the three different search algorithms (SaNSDE, PSO, and DE) as the optimizers, the transferred solutions from the simplified space and the best solutions in the population of the original problem space on the representative benchmarks are plotted in Fig. 7. As can be observed in the figure, solutions were transferred across the problem spaces during the evolutionary search process. In particular, in Fig. 7(a), compared to the best solution in the original problem space at different stages of the search, both inferior and superior solutions in terms of the objective value were transferred across the spaces. The former could be eliminated via natural selection, while the latter survived and efficiently guided the evolutionary search in the
Fig. 7. Illustration of transferred solutions and the best solution in the population on representative benchmarks of different comparison groups. (a) Comparison 1: SaNSDE as EA solver. (b) Comparison 2: PSO as EA solver. (c) Comparison 3: DE as EA solver.

original problem space toward promising areas of high-quality solutions, which led to the enhanced search performance of the proposed MSES, as observed in Table II, Figs. 3, and 4. Similar observations can also be made in the cases of using PSO and DE as the optimizers, as depicted in Fig. 7(b) and (c), respectively. These also confirmed that useful traits could be embedded in the different spaces of a given problem, and concurrently conducting an evolutionary search on multiple spaces can lead to efficient and effective problem-solving for large-scale optimization.

3) Sensitivity Study: Five parameters were used in the proposed multispace evolutionary search: the size of $A_s$ ($A_{s\text{size}}$), dimension of the simplified problem space ($d_s$), interval for reconstructing the simplified problem space ($G_r$), and interval and number of solutions transferred from the simplified to the original problem space ($G_t$ and $Q$, respectively). This section presents and discusses how these parameters affect the performance of the proposed multispace evolutionary search.

In particular, Figs. 8–10 present the averaged objective values obtained by the proposed MSES and DLLSO on the representative benchmarks across 25 independent runs with various configurations of $A_{s\text{size}}$, $d_s$, $Q$, $G_t$, and $G_r$. In the figures, the $X$-axis gives different benchmark functions, while the $Y$-axis denotes the normalized averaged objective value obtained by each compared configuration. Specifically, the obtained objective values on each benchmark are normalized by the worst (largest) objective obtained by all the compared algorithms on each benchmark.
the benchmark. Therefore, values close to 0 and 1 denote the best and worst performances, respectively. Furthermore, because DLLSO was observed to obtain a superior solution quality and search speed in contrast to the other compared algorithms in Sections IV-B1 and IV-B2, it is considered as the baseline algorithm here. For a fair investigation, DLLSO was also used as the optimizer in the proposed MSES.

Out of these parameters, $A_{s\text{size}}$, $d_s$, and $G_r$ were involved in the simplified problem space construction. $A_{s\text{size}}$ defined the number of solutions for constructing the simplified problem space, while $d_s$ gave the dimensionality of the constructed space. Furthermore, $G_r$ determined the frequency for reconstructing the simplified problem space. Generally, small numbers of $A_{s\text{size}}$ and $d_s$ simplified the problem space to a large extent, and large numbers of these two parameters could make the constructed space close to the original problem space. In addition, small and large values for $G_r$ reconstructed the low-dimensional space frequently and infrequently during the evolutionary search, respectively. As can be observed in Fig. 8, on the partially additive separable functions, for example, F5 and F8, NP (population size) number of solutions are already able to construct a useful problem space that can improve the search in the original problem space (see the superior objective values achieved by MSES with $A_{s\text{size}} = \text{NP}$, $A_{s\text{size}} = 5 * \text{NP}$, and $A_{s\text{size}} = 10 * \text{NP}$). However, on the more complex functions, for example, F12 and F15, a larger $A_{s\text{size}}$ may be required to provide more information for constructing a useful problem space in MSES. Furthermore, for the dimensionality of the simplified problem space, as can be observed in Fig. 9, neither a small nor a large number for $d_s$ is good for building a useful simplified problem space because a very low dimensionality could lose important information for efficient evolutionary search, and a space with a dimensionality close to the original problem cannot play a complementary role to the original problem space for the proposed MSES.\footnote{Dimension reduction is only one of the possible ways to construct the simplified problem space, and different dimension reduction approaches may possess various properties. For instance, the PCA can only provide linear dimension reduction, while the self-organizing map is able to conduct nonlinear dimension reduction. Prior knowledge or analysis on the given optimization, thus, could be helpful in selecting proper approaches for constructing the simplified space in the proposed multisphere evolutionary search.}

Lastly, as depicted in Fig. 10, the frequency of reconstructing the simplified space did not significantly affect the performance of MSES on the considered large-scale benchmarks.

On the other hand, parameters $Q$ and $G_t$ defined the amount and frequency of knowledge sharing across problem spaces. Generally, a small value of $Q$ and large value of $G_t$ significantly reduced the amount and frequency of solution transfer across spaces, while a large value of $Q$ and small value of $G_t$ greatly increased the amount and frequency of knowledge sharing across problem spaces. It can be observed from Figs. 11 and 12, with different configurations of $Q$ and $G_t$ values, superior solution qualities were obtained by the proposed MSES compared to DLLSO on most of the benchmarks. However, while the optimal confirmations of these parameters were generally problem-dependent, the configuration considered in the empirical study, as discussed earlier, was found to provide noteworthy results across a variety of larger-scale optimization problems.

V. AI APPLICATION ON RECOMMENDER SYSTEM

Recommender systems provide users with personalized online recommendations of products or information and have shown great potential to help users find relevant items from an information overload space [52]. The recommender system has become a vital part of e-commerce and has been widely adopted in a variety of online applications, ranging from social network, tourism, and education, to healthcare, etc. Keeping this
in mind, to illustrate the efficacy of the proposed multispace evolutionary search in solving real-world large-scale optimization problem of movie recommendation.

In the experiments, we consider movie recommendation by seeking the movies that are most relevant to users’ preferred type (or genre), which is essentially a single objective large-scale optimization problem. Formally, a minimization problem is investigated here, and the objective function is formulated as [53], [54]

$$\min f(x) = \sqrt{\frac{1}{\sum_{i \in R} p_u \cdot x(i)} + 1}$$

(4)

where $R$ represents the recommendation item set, $p_u$ is the latent embedding vector associated with the target user $u$. For evolutionary search, following [55], we use $x = \{x_1, x_2, \ldots, x_k\}$ to present a chromosome, as depicted in Fig. 13, the embedding vector of $i$th item $x(i)$ denotes as the $i$th gene in the chromosome. The length of a chromosome is the multiplication of embedding dimension and the length of item list. In other terms, each chromosome indicates a movie list recommended to the user, which is represented by $R = (m_1, m_2, \ldots, m_k)$.

The proposed MSES is applied to movie recommendation and the performances are compared with three large-scale optimization methods as discussed in Section IV, which are DLLSO, RDG3, and MeMAO. The traditional recommendation algorithm, i.e., Bayesian Personalized Ranking (labeled as BPR) [52], is also considered as a baseline algorithm. Moreover, according to [55], the baseline algorithm BPR is employed to learn the latent embedding of each candidate movie in this study, which is further normalized into a bounded latent space $[0, 100]^{10}$. The experiments are conducted on the publicly available Movielens-1 M dataset, which contains movie ratings collected from the MovieLens web site. The ratings larger than 3 are considered as positive feedback (575 169 ratings for 3457 movies by 6034 users), and the dataset is randomly split into two nonoverlapping sets. Empirically, 80% of the ratings are used for training, and the remaining 20% are used for testing [54]. For a fair comparison, the algorithm configurations of the large-scale optimization method are kept consistent with that in Section IV, and the setting of BPR is referred to [52].

![Fig. 13. Structure of a solution with chromosome encoding in solving real-world large-scale optimization problem of movie recommendation.](image)

![Fig. 14. Convergence curves of average fitness values (over all users) obtained by MSES and compared algorithms on movie recommendation. (Y-axis: averaged objective value in log scale; X-axis: number of fitness evaluation).](image)

The convergence curves of the average fitness values over all users obtained by all the compared algorithms on movie recommendation are presented in Fig. 14. It can be observed from the figure that the proposed MSES$_{DLLSO}$ and MSES$_{SaNSDE}$ achieved faster convergence on the optimization of movie recommendation than the compared algorithms (DLLSO and RDG3), which use the same EA solver, respectively. Moreover, the final optimal averaged objective value of MSES$_{DLLSO}$ obtained superior performance than that of BPR (the blue pentagram) and the other compared algorithms. This again confirmed the efficacy of the proposed multispace evolutionary search for solving large-scale optimization problem.

VI. CONCLUSION

This article proposed a multispace evolutionary search paradigm for large-scale optimization. In particular, it presented the details of the problem space construction, learning of the mapping across problem spaces, and knowledge transfer across problem spaces. In contrast to existing methods, the proposed paradigm conducts an evolutionary search on multiple solution spaces derived from the given problem, each possessing a unique landscape. More importantly, the proposed paradigm makes no assumptions about the given large-scale optimization problem, such as that the problem is decomposable or that a certain relationship exists among the decision variables. To validate the performance of the proposed paradigm, comprehensive empirical studies on both the CEC2013 large-scale benchmark problems and an AI recommendation application were conducted. The results were compared to those of recently proposed large-scale EAs and traditional recommendation approach, which confirmed the efficacy of the proposed multispace evolutionary search for large-scale optimization.

Moreover, in this article, we simply used dimension reduction to construct the simplified problem space, which may not suitable for any arbitrary large-scale optimization problem encountered (e.g., F1 and F2 as depicted in Table II). Therefore, future work will further explore effective approaches for constructing simplified problem spaces in the proposed multispace evolutionary search for efficient problem solving in large-scale optimization. The design of adaptive parameter configurations...
as well as the developing more general mapping across problem structures in the proposed paradigm is also a promising research direction for improving the generality of a multipurpose evolutionary search for large-scale optimization.

REFERENCES

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st ed. Boston, MA, USA: Addison-Wesley, 1989.
[2] R. Chiong, T. Weise, and Z. Michalewicz, Variants of Evolutionary Algorithms for Real-World Applications. Berlin, Germany: Springer, 2011.
[3] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford, U.K.: Oxford Univ. Press, 1996.
[4] D. Sneyers and Y. P. et al., "Image processing optimization by genetic algorithm with a new coding scheme," Pattern Recognit. Lett., vol. 16, no. 8, pp. 843–848, 1995.
[5] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, “Evolving deep convolutional neural networks for image classification,” IEEE Trans. Evol. Comput., vol. 24, no. 2, pp. 394–407, Apr. 2020.
[6] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lu, “Automatically designing CNN architecture among the genetic algorithm for image classification,” IEEE Trans. Cybern., vol. 50, no. 9, pp. 3840–3854, Sep. 2020.
[7] U. A. Khan, S. Sar, and J. M. F. Moura, “Higher dimensional consensus: Learning in large-scale networks,” IEEE Trans. Signal Process., vol. 58, no. 5, pp. 2836–2849, May 2010.
[8] P. Yang, K. Tang, and X. Yao, “A parallel divide-and-conquer-based evolutionary algorithm for large-scale optimization,” IEEE Access, vol. 7, pp. 163105–163118, 2019.
[9] Z. Sheng, H. D. Tuan, H. H. Nguyen, and M. Debbah, “Optimal training sequences for large-scale MIMO-OFDM systems,” IEEE Trans. Signal Process., vol. 65, no. 13, pp. 3329–3343, Jul. 2017.
[10] Y. Tian, X. Zhang, C. Wang, and Y. Jin, “An evolutionary algorithm for large-scale sparse multiobjective optimization problems,” IEEE Trans. Evol. Comput., vol. 24, no. 2, pp. 380–393, Apr. 2020.
[11] Z. Li, Q. Zhang, X. Lin, and H. Zhen, “Fast covariance matrix adaptation for large-scale black-box optimization,” IEEE Trans. Cybern., vol. 50, no. 5, pp. 2073–2083, May 2020.
[12] R. Cheng and Y. Jin, “A competitive swarm optimizer for large scale optimization,” IEEE Trans. Cybern., vol. 45, no. 2, pp. 191–204, Feb. 2015.
[13] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, “A framework for large-scale multiobjective optimization based on problem transformation,” IEEE Trans. Evol. Comput., vol. 22, no. 2, pp. 260–275, Apr. 2018.
[14] C. He et al., “Accelerating large-scale multiobjective optimization via problem reformulation,” IEEE Trans. Evol. Comput., vol. 23, no. 6, pp. 940–961, Dec. 2019.
[15] Z. Wang et al., “Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling,” IEEE Trans. Cybern., vol. 50, no. 6, pp. 2715–2729, Jun. 2020.
[16] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, “A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization,” IEEE Trans. Evol. Comput., vol. 22, no. 1, pp. 97–112, Feb. 2018.
[17] W. Chen, Y. Jia, F. Zhao, X. Luo, X. Jia, and J. Zhang, “A cooperative co-evolutionary approach to large-scale multisource water distribution network optimization,” IEEE Trans. Evol. Comput., vol. 23, no. 5, pp. 842–857, Oct. 2019.
[18] X. Peng, Y. Jin, and H. Wang, “Multimodal optimization enhanced cooperative coevolution for large-scale optimization,” IEEE Trans. Cybern., vol. 49, no. 9, pp. 3507–3520, Sep. 2019.
[19] Q. Yang et al., “A distributed swarm optimizer with adaptive communi- cation for large-scale optimization,” IEEE Trans. Cybern., vol. 50, no. 7, pp. 3393–3408, Jul. 2020.
[20] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, “Test problems for large-scale multiobjective and many-objective optimization,” IEEE Trans. Cybern., vol. 47, no. 12, pp. 4108–4121, Dec. 2017.
[21] R. Lan, L. Zhang, Z. Tang, Z. Liu, and X. Luo, “A hierarchical sort- ing swarm optimizer for large-scale optimization,” IEEE Access, vol. 7, pp. 40625–40635, 2019.
[22] J. Tian, Z. Zhan, and J. Zhang, “Large-scale evolutionary optimization: A survey and experimental comparative study,” Int. J. Mach. Learn. Cybern., vol. 11, no. 3, pp. 729–745, 2020.
Liang Feng received the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2014. He is currently a Professor with the College of Computer Science, Chongqing University, Chongqing, China. His research interests include computational and artificial intelligence, memetic computing, big data optimization and learning, and transfer learning. He is an Associate Editor for the IEEE Computational Intelligence Magazine, IEEE Transactions on Evolutionary Computation Outstanding Paper Award for his research work on evolutionary multitasking.

Qingxia Shang received the B.E. degree in 2012 from the Department of Computer Science and Information Engineering, Anyang Institute of Technology, Anyang, China, and the M.S. degree in 2017 from Southwest University, Chongqing, China, where she is currently working toward the Ph.D. degree with the College of Computer Science, Chongqing University, Chongqing, China. Her research interests include large-scale evolutionary optimization, multitask optimization, and transfer learning.

Yaqing Hou (Member, IEEE) received the Ph.D. degree in artificial intelligence from the Interdisciplinary Graduate School, Nanyang Technological University, Singapore, in 2017. He was a Postdoctoral Research Fellow with the Data Science and Artificial Intelligence Research Centre, Nanyang Technological University. He is currently an Associate Professor with the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests include computational and artificial intelligence, memetic computing, multiagent reinforcement learning, and transfer learning and optimization.

Dr. Hou is currently an Associate Editor for the Memetic Computing.

Kay Chen Tan (Fellow, IEEE) received the B.Eng. (First Class Hons.) and Ph.D. degrees from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively. He is currently a Chair Professor of computational intelligence with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He has authored or coauthored 300 refereed articles and seven books. Prof. Tan is currently the Vice-President (Publications) of IEEE Computational Intelligence Society, USA. He was the Editor-in-Chief of the IEEE Computational Intelligence Magazine from 2010 to 2013 and the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2015 to 2020. He is currently an Editorial Board Member for more than ten journals. He is currently an IEEE Distinguished Lecturer Program (DLP) Speaker and the Chief Co-Editor of Springer Book Series on Machine Learning: Foundations, Methodologies, and Applications.

Yew-Soon Ong (Fellow, IEEE) received the Ph.D. degree in artificial intelligence in complex design from the University of Southampton, Southampton, U.K., in 2003. He is currently the President Chair Professor of computer science with Nanyang Technological University (NTU), Singapore, and is also a Chief Artificial Intelligence Scientist of the Agency for Science, Technology and Research Singapore, At NTU, he is currently a Co-Director of the Singtel-NTU Cognitive and Artificial Intelligence Joint Lab. His research interest include artificial and computational intelligence. He is a founding Editor-in-Chief of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE and an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, and others. Dr. Ong was the recipient of several IEEE outstanding paper awards and was listed as a Thomson Reuters highly cited Researcher and among the World’s Most Influential Scientific Minds.