Toward Adaptive Knowledge Transfer in Multifactorial Evolutionary Computation

Lei Zhou, Liang Feng, Kay Chen Tan, Jinghui Zhong, Zexuan Zhu, Kai Liu, and Chao Chen

***Abstract*—A multifactorial evolutionary algorithm (MFEA) is a recently proposed algorithm for evolutionary multitasking, which optimizes multiple optimization tasks simultaneously. With the design of knowledge transfer among different tasks, MFEA has demonstrated the capability to outperform its single-task counterpart in terms of both convergence speed and solution quality. In MFEA, the knowledge transfer across tasks is realized via the crossover between solutions that possess different *skill factors*. This crossover is thus essential to the performance of MFEA. However, we note that the present MFEA and most of its existing variants only employ a single crossover for knowledge transfer, and fix it throughout the evolutionary search process. As different crossover operators have a unique bias in generating offspring, the appropriate configuration of crossover for knowledge transfer in MFEA is necessary toward robust search performance, for solving different problems. Nevertheless, to the best of our knowledge, there is no effort being conducted on the adaptive configuration of crossovers in MFEA for knowledge transfer, and this article thus presents an attempt to fill this gap. In particular, here, we first investigate how different types of crossover affect the knowledge transfer in MFEA on both single-objective (SO) and multiobjective (MO) continuous optimization problems. Furthermore, toward robust and efficient multitask optimization performance, we propose a new MFEA with adaptive knowledge transfer (MFEA-AKT), in which the crossover operator employed for knowledge transfer is self-adapted based on the information collected along the evolutionary search process. To verify the effectiveness of the proposed method, comprehensive empirical studies on both SO and MO multitask benchmarks have been conducted. The experimental results show that the proposed MFEA-AKT is able to identify the appropriate knowledge transfer crossover for different optimization problems and even at different optimization stages along the search, which thus leads to superior or competitive performances when compared to the MFEAs with fixed knowledge transfer crossover operators.**

***Index Terms*—Adaptive knowledge transfer, evolutionary multitasking, multifactorial evolutionary algorithm (MFEA).**

Manuscript received July 18, 2019; revised November 21, 2019 and January 13, 2020; accepted February 8, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61876025, Grant 61876162, Grant 61872049, Grant 61602181, and Grant 61871272, in part by the Venture & Innovation Support Program for Chongqing Overseas Returnees under Grant cx2019020 and Grant cx2018044, in part by the City University of Hong Kong Research Fund under Grant 9610397, in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20180307123637294 and Grant JCYJ20170302154328155, and the Research Grants Council of the Hong Kong under Grant CityU11202418 and Grant CityU11209219. This article was recommended by Associate Editor Y. S. Ong. *(Corresponding authors: Liang Feng; Kay Chen Tan.)*

Lei Zhou, Liang Feng, Kai Liu, and Chao Chen are with the Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400044, China, and also with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: liangf@cqu.edu.cn).

Kay Chen Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the City University of Hong Kong Shenzhen Research Institute, Shenzhen, China (e-mail: kaytan@cityu.edu.hk).

Jinghui Zhong is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

Zexuan Zhu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China.

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

Digital Object Identifier 10.1109/TCYB.2020.2974100

I. Introduction

MULTIFACTORIAL optimization (MFO) is a new evolutionary optimization paradigm proposed by Gupta *et al.* in 2016, for multitask optimization [1], [2]. In contrast to the traditional evolutionary algorithm (EA) which optimizes only one task in a single run, MFO optimizes multiple tasks at the same time. With the implicit knowledge transfer across tasks throughout the multitask optimization process, MFO has demonstrated to outperform its single-task counterpart in terms of both convergence speed and solution quality on a set of optimization problems, including continuous, discrete, and the mixture of continuous and discrete optimization problems [3]–[7].

Multifactorial EA (MFEA) was proposed together with MFO in [1], which serves as a particular realization of the MFO paradigm. Due to its simple implementation and strong search capability, it has attracted much research attention since it was proposed. For instance, Yuan *et al.* [8] proposed a permutation-based MFEA to efficiently solve combinatorial problems, such as a traveling salesman problem (TSP), a quadratic assignment problem (QAP), and a job-scheduling problem (JSP). Gupta *et al.* extended the MFEA for multitasking in the domain of multiobjective (MO) optimization [9] and bilevel optimization [7]. Further, Liaw and Ting [10] extended MFEA for solving the many task optimization problems, in which the number of tasks to be optimized is more than two. In [11], a linearized domain adaptation strategy was integrated into MFEA to transform the search space of a simple task to the search space of a similar but complex task so that negative transfer can be alleviated. Moreover, Ding *et al.* [12] proposed a generalized MFEA called G-MFEA to tackle the tasks with separated optimum and different variable dimensions. Gong *et al.* [13] and Wen and Ting [14] integrated resource reallocating strategies into MFEA, which allocated the computational resources according to the complexities of tasks. Tang *et al.* [15] proposed a group-based MFEA to group tasks of similar types and the genetic information can only be transferred within the same groups. More recently, an enhanced MFEA called MFEA-II was presented, which employs an online transfer parameter estimation scheme to dynamically control the extent of knowledge exchange across tasks [16]. In these multitasking algorithms, all the tasks are optimized by a single population of individuals, using a common solution representation that integrates distinct problem domains into a unified search space. Each individual is assigned with a *skill factor* which denotes the task where the individual has the best performance. The knowledge transfer across tasks happens implicitly when two individuals possessing different *skill factors* are selected for generating the offspring via crossover.

In the literature, for different optimization problems, many crossover operators with various search capabilities have been proposed [17]–[20]. For instance, the one- and two-point crossover [21], [22]; uniform crossover [23], [24]; simulated binary crossover (SBX) [25], [26]; etc., have been introduced to solve continuous optimization problems, while the order crossover (OX) [27], [28]; partially mapped crossover (PMX) [29], [30]; etc., have been designed for combinatorial optimization. It has been found that different crossovers have various capabilities for solving different problems. Particularly, in [31], it was reported that multipoint and uniform crossovers perform significantly better than one-point and two-point crossover on 12 benchmark functions [32] with various properties, such as unimodal, mulmodal, noisy, and discrete search spaces. Michalewicz *et al.* [33] showed that the geometrical crossover outperforms the arithmetical crossover on the majority of the unconstrained continuous problems. The authors also suggested that geometrical crossover performs well on problems where the global optimum lays on the boundary of the feasible region. In [25], the SBX was found to outperform other real-coded crossovers, such as BLX-α crossover, in difficult test functions like bimodal, unequal spread function, and pole problem. In these works, we can observe that different crossovers possess unique bias in generating offspring for guiding the evolutionary search. Therefore, these crossovers could also have different capabilities in conducting implicit knowledge transfer across tasks in evolutionary multitasking. Obviously, crossover determines the quality of knowledge transfer across tasks. As different crossovers possess unique bias in solution generation, the employment of a proper crossover for knowledge transfer across tasks is essential to the performance of evolutionary multitasking. However, it is worth noting that the existing evolutionary multitasking algorithms [8]–[16] employ a single- and fixed-crossover operator for knowledge transfer, which thus could impede the multitasking when the configured crossover does not fit the encountered problems well. Toward efficient and effective multitasking performance, it is desirable to enable MFEA with adaptive knowledge transfer capability, so that different crossovers for knowledge transfer can be employed on-demand for different problems. Nevertheless, to the best of our knowledge, there is no research effort conducted in the literature for adaptive MFEA design. This article thus presents an attempt to fill this gap.

In particular, in this article, we first embark on a study to investigate how different crossovers for knowledge transfer affect the multitask optimization performance of MFEA. By utilizing both the single-objective (SO) and MO multitask benchmarks in [34] and [35], it is observed that different crossovers achieved superior performance on different tasks. As indicated by the theory of "no free lunch" [36], there is no single crossover for knowledge transfer that can perform well on all the benchmark tasks. Furthermore, toward robust and efficient multitasking on different optimization tasks, we propose a new MFEA with adaptive knowledge transfer, called MFEA-AKT. In the proposed MFEA-AKT, the crossover operator for knowledge transfer across tasks is configured adaptively based on the information collected while the evolutionary search progresses online. Particularly, in MFEA-AKT, each individual is assigned with a *transfer crossover indicator* (T_{ci}) which is used to determine the crossover to be employed for knowledge transfer. The T_{ci} of each individual is updated based on the performance of the generated offspring across tasks in each generation. To verify the efficacy of the proposed MFEA-AKT, comprehensive experiments are conducted on the commonly used SO and MO multitask benchmarks [34], [35], and new SO and MO multitask problems constructed based on the CEC2014 SO test suites [37] and the LZ09 MO benchmarks [38], respectively. Experimental results show that with the proposed adaptive configuration of crossover for knowledge transfer, MFEA-AKT is able to obtain superior or competitive performances on both SO and MO optimization problems, when compared to the MFEAs with fixed knowledge transfer crossover operators.

The remainder of this article is organized as follows. Section II gives a brief introduction of MFEA and the review of representative crossover operators in the literature. Section III investigates the multitasking performance of MFEA when different crossovers are configured for knowledge transfer. Based on this investigation, the details of the proposed MFEA-AKT are then presented in Section IV. The experimental results of MFEA-AKT are presented and discussed in Section V. Finally, Section VI draws the conclusion of this article.

II. Preliminaries

In this section, we first present a brief introduction of the MFEA proposed in [1]. Next, representative crossover operators proposed in the literature for continuous optimization are reviewed and discussed.

A. Multifactorial Evolutionary Algorithm

The MFEA was first proposed by Gupta *et al.* [1], which aims to tackle multiple optimization problems simultaneously. In MFEA, different problems are optimized via a single population of individuals defined in a unified search space. To evaluate the individuals for multitasking, the following properties have been defined to each individual.

1) *Factorial Cost:* The factorial cost f_p of an individual p denotes its fitness or objective value evaluated on a particular task T_i. For K tasks, there will be a vector with length K, in which each dimension gives the fitness of p on the corresponding task.
2) *Factorial Rank:* The factorial rank r_p simply denotes the index of individual p in the list of population members sorted in ascending order with respect to their factorial costs on a specific task.
3) *Scalar Fitness:* The scalar fitness φ_p of an individual p is defined based on its best rank over all tasks, which is given by $\varphi_p = 1/[\min_{j\in\{1,\ldots,K\}} r_p^j]$. In MFEA, individual a is considered to dominate individual b if $\varphi_p > \varphi_b$.
4) *Skill Factor:* The skill factor τ_p of an individual p denotes the task among all other tasks, on which p is most effective, that is, $\tau_p = \operatorname{argmin}\{r_p^j\}$, where $j \in \{1, \ldots, K\}$.

With the above definitions, the workflow of *MFEA* can be summarized as follows.

Step 1: Generate an initial population consisting of *NP* individuals using a unified representation.

Step 2: Evaluate each individual on all the tasks by calculating its *factorial cost* f_p, *factorial rank* r_p, *scalar fitness* φ_p, and *skill factor* τ_p.

Step 3: Apply genetic operators, that is, crossover and mutation, on the current population via *assortative mating* to generate an offspring population.

Step 4: Evaluate offspring individuals on selected tasks based on *vertical cultural transmission*, and set the corresponding fitness values on the unevaluated tasks as infinity.

Step 5: Update the scalar fitness φ_p and skill factor τ_p of individuals in both parent and offspring population.

Step 6: Select the fittest *NP* individuals from both parent and offspring population to survive for the next generation.

Step 7: If the stopping criteria are not met, repeat Steps 3–6.

As can be observed, when two individuals with different *skill factors* are selected to generate the offspring via crossover, knowledge transfer is implicitly carried out from one task domain to the other. Therefore, the proper employment of crossover is essential to the quality of knowledge transfer across tasks in MFEA.

B. Representative Crossover Operators for Continuous Optimization

Generally, crossover denotes the reproduction operator which exchanges genetic materials between parents to generate offspring in EAs. In the last few decades, a number of crossover operators have been proposed in the literature for a wide range of optimization problems [39]–[41]. In this section, we briefly review existing representative crossover operators and discuss the differences of these crossovers for knowledge transfer. In particular, according to a survey [42], existing crossovers can be categorized into three classes, which are: 1) discrete crossover; 2) aggregation-based crossover; and 3) neighborhood-based crossover. Let p, c, and n denote the parent, offspring, and the dimension of the individuals, respectively. The review of representative existing crossover operators is given as follows.

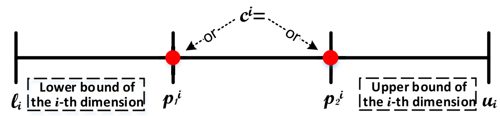


Fig. 1. Each dimension of the offspring inherits the exact knowledge from one of the parents.

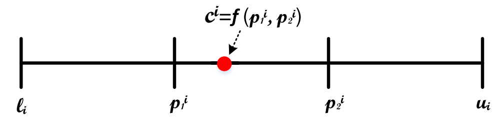


Fig. 2. Aggregation function f is utilized to merge the knowledge of two parents.

1) *Discrete Crossover Operators (DCOs):* In this category, each dimension of the offspring inherits the exact genetic materials from one of the parents, which is illustrated in Fig. 1. Popular crossovers in this group include single-point [43], two-point [43], uniform crossovers [44], etc. In this article, the two-point crossover and uniform crossover serving as the representative crossovers for investigation, are detailed as follows.
   a) *Two-Point Crossover:* Two positions $i, j \in \{1, 2, \ldots, n\}$ $(i < j)$ are randomly selected and the segments from i to j of the two selected parents p_1 and p_2 are exchanged to construct the offspring
   $$c_1 = \left(p_1^1, p_1^2, \ldots, p_2^i, p_2^{i+1}, \ldots, p_2^j, p_1^{j+1}, \ldots, p_1^n\right)$$
   $$c_2 = \left(p_2^1, p_2^2, \ldots, p_1^i, p_1^{i+1}, \ldots, p_1^j, p_2^{j+1}, \ldots, p_2^n\right).$$
   b) *Uniform Crossover:* The element at position i of the offspring is randomly chosen from one of the two selected parents with a uniform possibility
   $$c_1^i (or\ c_2^i) = \begin{cases} p_1^i, & \text{if } u = 0 \\ p_2^i, & \text{if } u = 1 \end{cases}$$
   where u is randomly assigned to 0 or 1.
2) *Aggregation-Based Crossover Operators (ABCOs):* Crossovers belonging to ABCOs transfer the combined genetic segments of the two parents to the offspring with an aggregation function, which is illustrated in Fig. 2. As the position of the offspring is fixed when the aggregation function is determined, the genetic inheritance is deterministic. Examples of ABCOs include arithmetical crossover [45], geometrical crossover [46], linear crossover (LX) [47], etc. In this article, the arithmetical crossover and the geometrical crossover are considered as the representative crossover operators in this group, which are given as follows.
   a) *Arithmetical Crossover:* The element at position i of the offspring is the linear combination of the two selected parents
   $$c_1^i = \lambda \cdot p_1^i + (1 - \lambda) \cdot p_2^i$$
   $$c_2^i = \lambda \cdot p_2^i + (1 - \lambda) \cdot p_1^i$$

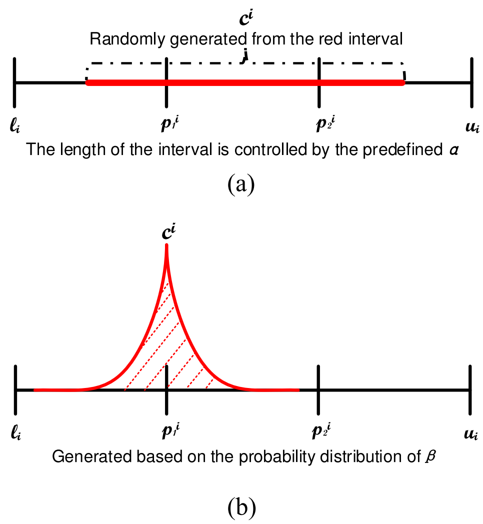


Fig. 3. Knowledge is extracted from the neighborhood of the parents with a predefined probability distribution. (a) BLX-α crossover. (b) SBX crossover.

where λ is a predefined coefficient, which ranges from 0 to 1.

b) *Geometrical Crossover:* The element at position i of the offspring is the exponential combination of the two selected parents

$$c_1^i = \cdot p_1^{i^w} \cdot p_2^{i^{1-\omega}}$$
$$c_2^i = \cdot p_2^{i^w} \cdot p_1^{i^{1-\omega}}$$

where ω is predefined by users, which has the range of [0,1].

3) *Neighborhood-Based Crossover Operators (NBCOs):* Crossovers in this group extract the genetic material from the interval between the parents with a predefined probability distribution. This interval is defined as the neighborhood of the parents. In contrast to the other two groups of crossover aforementioned, we can see the NBCOs give a more flexible way of generating offspring, which thus may introduce a higher diversity along the evolutionary search process. Representative crossovers in this group contain the fuzzy recombination (FR) [48], BLX-α [49], SBX [50], etc. Particularly, the BLX-α and SBX crossovers are employed here as the illustrative crossovers for investigation, which are based on the uniform and exponential probability distributions, respectively. As depicted in Fig. 3, the details of these two crossover operators are presented as follows.

a) *BLX-α Crossover:* The element at position i of the offspring is randomly generated from the interval $[P_{\min} - I \cdot \alpha, P_{\max} + I \cdot \alpha]$, where $P_{\max} = \max\{p_1^i,\ p_2^i\}$, $P_{\min} = \min\{p_1^i,\ p_2^i\}$, and $I = P_{\max} - P_{\min}$.

b) *SBX:* The element at position i of the offspring is generated from an exponential probability distribution

$$c_1^i = \frac{1}{2} \cdot \left[(1-\beta) \cdot p_1^i + (1+\beta) \cdot p_2^i\right]$$

$$c_2^i = \frac{1}{2} \cdot \left[(1+\beta) \cdot p_1^i + (1-\beta) \cdot p_2^i\right].$$

The distribution of β is defined by

$$\beta(u) = \begin{cases} (2u)^{1/(\eta_c+1)}, & \text{if } u \le \frac{1}{2} \\ [2(1-u)]^{-1/(\eta_c+1)}, & \text{if } u > \frac{1}{2} \end{cases}$$

where u is a random number generated in the range of [0, 1].

As can be observed, different crossovers have various forms that possess unique bias in generating offspring. Since crossover has also been employed in MFEA for implicit knowledge transfer across tasks, it may lead to different forms of knowledge transfer when different crossovers are configured, which could result in diverse multitask optimization performance.

III. Knowledge Transfer via Different Crossover Operators in MFEA

In this section, we investigate how different crossover operators for knowledge transfer affect the performance of MFEA, using the common SO and MO multitask optimization problems in [34] and [35].

In particular, the PILS and NIMS SO multitask problems, and the CIHS and NIMS MO multitask problems are studied in this section. More details of the multitask benchmarks and configurations of the crossovers will be presented later in Section V. Fig. 4 presents the averaged convergence graphs of MFEA with six different crossover operators for knowledge transfer on the four multitask problems, over 20 independent runs. In Fig. 4(a) and (b), the y-axis denotes the averaged fitness (objective value) in log scale, and the x-axis is the generation number. Further, in Fig. 4(c) and (d), the y-axis represents the averaged IGD value in log scale, and the x-axis denotes the generation number.

As can be observed in Fig. 4, the performance of MFEA varies when different crossover operators have been employed for knowledge transfer. The performance gap in terms of objective value and IGD can be observed clearly from Fig. 4. For instance, on NIMS2 of Fig. 4(d), the BLX-α crossover converges about two times faster than the SBX crossover since the 200th generation. Furthermore, the arithmetical and geometrical crossover achieved the best performance on both SO PILS and SO NIMS. Particularly, on PILS [see Fig. 4(a)], while other operators are all stagnated at the early optimization stage, the arithmetical and geometrical crossover converged fast and obtained much better solutions. However, on MO CIHS, these two crossovers degraded to the worst crossover operators, as shown in Fig. 4(c). Further, on the MO problems, the BLX-α crossover achieved the best IGD on MO PILS, while the geometrical crossover outperformed the others on MO NIMS2. These observations confirmed that different optimization problems may require different configurations of crossover for knowledge transfer across tasks, for efficient multitask optimization performance, and there is no single-crossover operator can perform well on all the SO and MO problems.

Toward robust and efficient multitask optimization performance when different problems are encountered, here,

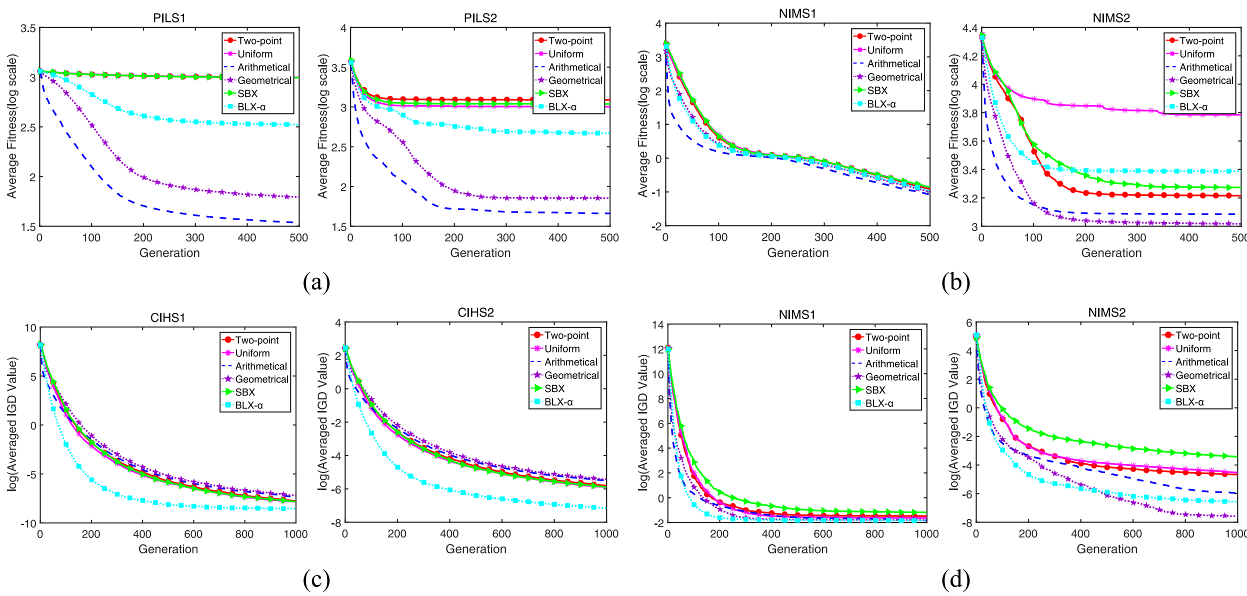


Fig. 4. Convergence traces of MFEA with six different knowledge transfer crossovers on PILS, NIMS of the SO multitask benchmarks and CIHS, NIMS of the MO multitask benchmarks, respectively. y-axis: log(Averaged results); x-axis: Generation. (a) Convergence traces of SO *PILS*. (b) Convergence traces of SO *NIMS*. (c) Convergence traces of MO *CIHS*. (d) Convergence traces of MO *NIMS*.

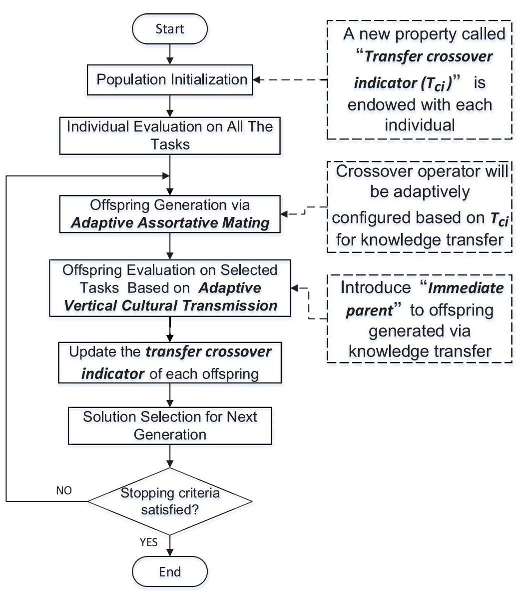


Fig. 5. Outline of the proposed *MFEA-AKT*.

we thus propose a new MFEA with an adaptive configuration of crossover for knowledge transfer, called MFEA-AKT, which will be presented in the next section.

IV. Proposed MFEA With Adaptive Knowledge Transfer

This section presents the details of the proposed MFEA-AKT. In particular, the framework of MFEA-AKT is illustrated in Fig. 5, where the main differences between MFEA-AKT and the original MFEA are highlighted in the dashed boxes. In the proposed MFEA-AKT, we first introduce three new definitions, which are given as follows.

Definition 1 (Transfer Crossover Indicator): The *Transfer crossover indicator* (T_{ci}) of an individual is an integer in the range of $[1, m]$, where m is the number of crossover operators available for knowledge transfer. Particularly, $T_{\mathrm{ci}} = i$ indicates the individual prefers to take the ith crossover operator for sharing knowledge with other individuals.

Definition 2 (Transferred Offspring): The *transferred offspring* refers to the solution which is generated by the parents with different *skill factors* via crossover.

Definition 3 (Immediate Parent): The parent who has the same *skill factor* with the *transferred offspring* is defined as the offspring's *immediate parent*.

In MFEA-AKT, toward adaptive knowledge transfer, the *transfer crossover indicator* of each individual is randomly assigned initially. Next, the *adaptive assortative mating* kicks in to adaptively configure the crossover operator for knowledge transfer across tasks based on the T_{ci} of the mated individuals. The T_{ci} of each individual is updated according to the information collected along the evolutionary search process. Further, the T_{ci} of each offspring individual is obtained via the *adaptive vertical cultural transmission*. Last but not least, other procedures of MFEA-AKT, such as initialization, solution evaluation, and selection, are kept the same as MFEA in [1].

A. Adaptive Assortative Mating and Adaptive Vertical Cultural Transmission

Algorithm 1 presents the details of the *adaptive assortative mating* procedure. In contrast to the original MFEA, which fixes the crossover for knowledge transfer across tasks, MFEA-AKT contends to adaptively employ the appropriate knowledge transfer crossovers for different individuals. In particular, as depicted in Algorithm 1, first, two parents p_1 and p_2 are randomly selected for reproduction. If the parents hold

Algorithm 1: Adaptive Assortative Mating

Input: Two randomly selected parents p_1 and p_2, and their *skill factor* τ_1 and τ_2; A random mating probability *rmp*.
Output: The generated offspring, c_1 and c_2.
Generate a random number $rand \in [0, 1]$;
if ($\tau_1 == \tau_2$) or ($rand < rmp$) **then**
  if ($\tau_1 == \tau_2$) **then**
    Generate c_1, c_2 via the SBX crossover with p_1 and p_2;
    c_1, c_2 inherit the T_{ci} from p_1 and p_2, respectively;
  else
    Randomly select the T_{ci} of p_1 or p_2 as the activated *transfer crossover indicator* (T_{ci}^a);
    Select the crossover associated with T_{ci}^a for knowledge transfer;
    Generate c_1, c_2 with p_1 and p_2 via the selected crossover;
    Configure the T_{ci} of both c_1 and c_2 as T_{ci}^a;
    Mark c_1, c_2 as *transferred offspring*;
  end if
else
  c_1 = mutate(p_1);
  c_2 = mutate(p_2);
  c_1, c_2 inherit the T_{ci} from p_1 and p_2, respectively;
end if

the same *skill factor*, the SBX crossover is performed for the offspring generation as in MFEA.[1] Otherwise, with a random mating probability (rmp), individuals with different *skill factors* are mated via an adaptively configured crossover for knowledge transfer. Specifically, T_{ci} of p_1 or p_2 is randomly selected as the activated *transfer crossover indicator* (T_{ci}^a), and the associated crossover operator of T_{ci}^a is then used with p_1 and p_2 for knowledge transfer. Further, the generated individuals, that is, *transferred offspring*, take T_{ci}^a as their *transfer crossover indicator*. If the two offsprings c_1 and c_2 are generated via crossover without knowledge transfer (or mutation), c_1 and c_2 inherit the *transfer crossover indicator* from p_1 and p_2, respectively.

Next, the *adaptive vertical cultural transmission* procedure is described in Algorithm 2. In particular, if an offspring has two parents p_1 and p_2, it imitates the *skill factor* of either p_1 or p_2 with equal probability. Otherwise, the offspring imitates the *skill factor* of the parent after mutation. If the offspring is a *transferred offspring*, the parent which has the same *skill factor* with the offspring is set as the offspring's *immediate parent*. This immediate parent will be used in the adaptation of *transfer crossover indicator* of the offspring, which is detailed in the next section.

B. Adaptation of Transfer Crossover Indicators

After the generation and evaluation of the offspring population, the *transfer crossover indicator* of each offspring will be updated according to the information collected along the evolutionary search process, which is detailed in Algorithm 3. In particular, first, the *transfer crossover indicator* of the *transferred offspring* which achieved the largest improvement ratio (IR) in the current generation is selected as the best *transfer crossover indicator* (T_{ci}^b).[2] The IR here refers to the percentage of the *scalar fitness* improvement of a *transferred offspring* s against its *immediate parent* p_s, which is calculated as follows[3]:

$$\mathrm{IR} = \frac{f(s) - f(p_s)}{|f(p_s)|} \tag{1}$$

where $f(*)$ is the *scalar fitness* of a solution and $|f(p_s)|$ gives the absolute value of $f(p_s)$. Without loss of generality, other metrics proposed in the literature to measure the performance of a solution could also be employed here.

Next, for each *transferred offspring* in the offspring population, if it is worse than its *immediate parent* in terms of *scalar fitness*, its *transfer crossover indicator* will be replaced

Algorithm 2: Adaptive Vertical Cultural Transmission

Input: An offspring c which is generated from crossover (using parents p_1 and p_2) or mutation (using parent p);
Output: The evaluated c with updated *skill factor*.
if (c has two parents) **then**
  Generate a random number $rand \in [0, 1]$;
  if ($rand < 0.5$) **then**
    c imitates the *skill factor* of p_1;
  else
    c imitates the *skill factor* of p_2;
  end if
  if (c is a *transferred offspring*) **then**
    Set the parent solution which possess with the same skill factor with c as *immediate parent*.
  end if
else
  c imitates the *skill factor* of p;
end if
Evaluate c on the task associated with its *skill factor*;

Algorithm 3: Adaptation of Transfer Crossover Indicators

Input: The offspring population.
Output: Offspring with updated *transfer crossover indicator*.
Calculate the best *transfer crossover indicator* T_{ci}^b according to Eq. (1).
for each offspring c in the offspring population **do**
  if (c is a *transferred offspring*) **then**
    if (c is worse than its *immediate parent*) **then**
      Set the T_{ci} of c as T_{ci}^b;
    end if
  else
    Generate a random number $rand \in [0, 1]$;
    if ($rand < 0.5$) **then**
      Set the T_{ci} of c as T_{ci}^b;
    else
      Set the T_{ci} of c as a randomly generated integer in range of $[1, m]$;
    end if
  end if
end for

[1] In order to investigate the effect of crossover in knowledge transfer, we kept the crossover for offspring generation without knowledge transfer the same as that in MFEA. However, other crossovers can also be applied here.

[2] In the cases that a single *transfer crossover indicator* has transferred offspring possessing different *skill factor*s, the largest IR obtained will be assigned to this *transfer crossover indicator*.

[3] The performance metric used in MFEA is positively correlated to the performance, that is, a large value indicates superior performance. If a metric which is negatively correlated to the performance is employed, the calculation of IR should be $\mathrm{IR} = [f(p_s) - f(s)]/|f(p_s)|$.

by T^b_{ci}. Otherwise, the offspring remains its *transfer crossover indicator*. For other offspring (i.e., offspring that are not generated via knowledge transfer), they either inherit T^b_{ci} or randomly reinitialize T_{ci} with equal probability. In this way, the proposed MFEA-AKT gradually increases the probability of the crossover with the best performance to be adopted for knowledge transfer, and at the same time, encourages the diversity of the transfer crossover equipped on each of the individuals in the population.

In summary, in MFEA-AKT, each crossover operator has the same probability to be selected for knowledge transfer at the initial optimization stage. As the evolutionary search progresses over time, the crossover which continuously achieves good performance in knowledge transfer will be identified. Meanwhile, the diversity of the crossover operators is also considered so that other crossovers may also be selected for knowledge transfer along the evolutionary search process. Further, it is straightforward to see that the proposed adaptive approach can also be easily integrated into other MFEA variants. For the sake of generality, the realization of the adaptive approach is based on the original MFEA [1] in this article.

V. Empirical Study

In this section, empirical studies have been conducted to investigate the performance of the proposed MFEA-AKT against MFEA with fixed crossovers for knowledge transfer, using the multitask benchmarks. Furthermore, new complex multitask problems are also developed to verify the effectiveness of the proposed method. Finally, discussions on the available crossovers for knowledge transfer and the employ of different crossovers for generating offspring in the proposed MFEA-AKT are also presented.

A. Experimental Setup

In the experiment, the commonly used SO and MO multitask benchmarks proposed in [34] and [35] are adopted. In these benchmarks, several classical functions, such as *Sphere*, *Rastrigin*, and *Griewank*, with various properties like dimension, landscape, etc., are paired based on the task similarity and the degree of global optima's intersection. In particular, each of the SO and MO multitask benchmark contains nine multitask problems, that is, CIHS, CIMS, CILS, PIHS, PIMS, PILS, NIHS, NIMS, and NILS. The nine problems can be divided into three categories based on the degree of task intersection, that is, complete intersection (CI), partial intersection (PI), and no intersection (NI). Moreover, by considering the task similarity, the problems in each category can be further classified as high similarity (HS), medium similarity (PS), and low similarity (LS), respectively. For more details of each benchmark, interested readers are referred to [34] and [35].

Next, as the adaptive approach is realized on the original MFEA in this article, MFEA is also utilized to integrate with the six crossover operators described in Section II-B for comparison. The crossovers investigated are the two-point, uniform, arithmetical, geometrical, BLX-α, and SBX crossover. Particularly, it is worth noting that these crossover operators will only be applied for knowledge transfer across tasks in MFEA. For generating offspring without knowledge transfer, according to [1], the SBX crossover is used for a fair comparison. Furthermore, the single-task solvers which use the same search operators as MFEA are compared, that is, genetic algorithm (GA) is employed for SO optimization and NSGAII for MO optimization. Finally, to further confirm the efficacy of the MFEA-AKT, the MFEA with random knowledge transfer (denoted as MFEA-RKT in the following), in which the crossover for knowledge transfer is randomly selected, is also included in the comparison. For a fair comparison, other experimental settings are kept consistent with [34] and [35], which are summarized as follows.

1) *Population Size:*
   a) *SO Optimization:* 100 for both MFEA and GA.
   b) *MO Optimization:* 200 for MFEA and 100 for NSGAII.
2) *Maximum Generations:*
   a) *SO Optimization:* 1000 for MFEA and 500 for GA.
   b) *MO Optimization:* 1000 for both MFEA and NSGAII.
3) *Independent Run Times:* 20.
4) *Evolutionary Operators and Parameters:*
   a) *SBX Crossover:* $\eta_c = 2$ for SO [34] and $\eta_c = 20$ for MO [35].
   b) *Arithmetical Crossover:* $\lambda = 0.25$ [42].
   c) *Geometrical Crossover:* $\omega = 0.25$ [42].
   d) *BLX-α Crossover:* $\alpha = 0.3$ [42].
   e) *Polynomial Mutation:* $\eta_m = 5$ for SO [34] and $\eta_m = 20$ for MO [35].
5) *rmp in MFEA:* 0.3 for SO [34] and 0.9 for MO [35].

B. Performance Metric

To compare the performance of different algorithms on the multitask problems, we adopt the performance metric proposed in [34]. Suppose there are N algorithms to be compared on one multitask problem consisting of optimization tasks $T_1, T_2, \ldots, T_K$, each of which is executed for L times. For each algorithm A_i, the *performance score* is formulated as follows:

$$\text{Score}_i = \frac{1}{L}\sum_{k=1}^{K}\sum_{l=1}^{L}\frac{I_k^l - \mu_k}{\delta_k} \tag{2}$$

where I_k^l is the best result obtained by A_i in the lth run on task T_k, and μ_k and δ_k are the mean and the standard deviation of I_k obtained by the N algorithms in all execution times, respectively. In the context of minimization problem, a smaller score corresponds to a better performance.

C. Results and Discussions

1) Common Multitask Benchmarks: To assess the efficiency of the proposed method, the averaged convergence graphs obtained by MFEA-AKT, MFEA-RKT, and MFEA with different knowledge transfer crossover on representative SO and MO multitask problems over 20 independent runs are depicted in Figs. 6 and 7, respectively. In the figures,

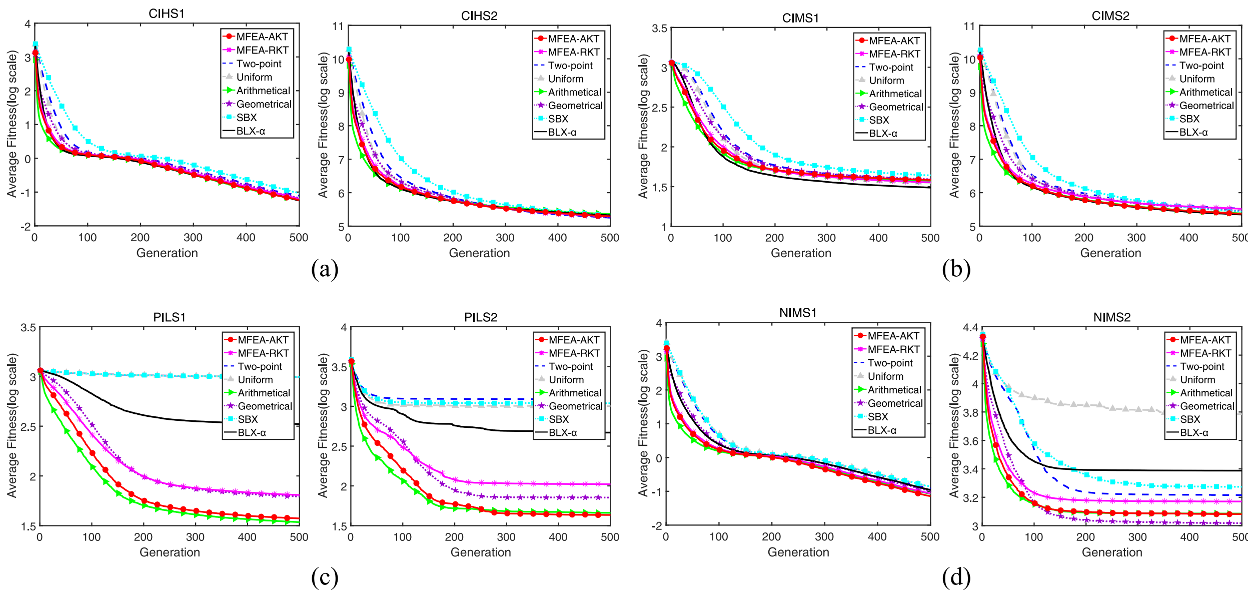


Fig. 6. Convergence traces of the averaged fitness obtained by MFEA-AKT, MFEA-RKT, and MFEA with six different fixed knowledge transfer crossovers on representative problems of the SO multitask benchmarks over 20 independent runs. *y*-axis: log(Averaged fitness); *x*-axis: Generation. (a) Convergence traces of *CIHS*. (b) Convergence traces of *CIMS*. (c) Convergence traces of *PILS*. (d) Convergence traces of *NIMS*.

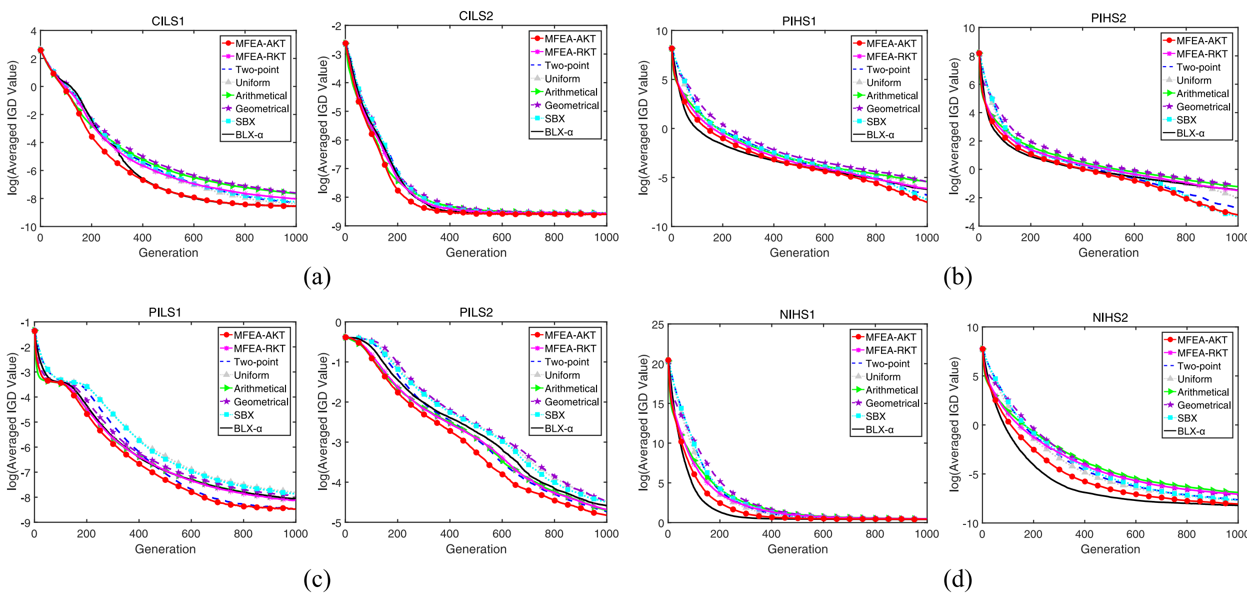


Fig. 7. Convergence traces of the averaged IGD obtained by MFEA-AKT, MFEA-RKT, and MFEA with six different fixed knowledge transfer crossovers on representative problems of the MO multitask benchmarks over 20 independent runs. *y*-axis: log(Averaged IGD value); *x*-axis: Generation. (a) Convergence traces of *CILS*. (b) Convergence traces of *PIHS*. (c) Convergence traces of *PILS*. (d) Convergence traces of *NIHS*.

it can be observed that MFEA-AKT achieved faster convergence speed and high-quality solutions on these problems when compared to the other algorithms. For instance, MFEA-AKT obtained the best results among all the algorithms on SO CIHS [Fig. 6(a)] and MO PILS [Fig. 7(c)]. Next, the self-adaptation of crossover for knowledge transfer in MFEA-AKT has been observed. In particular, on the one hand, MFEA-AKT identified the appropriate knowledge transfer crossover for different optimization problems. As shown in Figs. 6(c) and 7(c), the arithmetical crossover and the two-point crossover appeared to be the best crossover operator on SO PILS and MO PILS, respectively. It is observed that the proposed MFEA-AKT successfully identified these two crossovers and achieved competitive performance on both problems. On the other hand, MFEA-AKT is able to adaptively configure proper crossover at different optimization stages of the same problem. For instance, MFEA-AKT performed similar to the BLX-α crossover on MO PIHS at first [Fig. 7(b)]. However, as the convergence of the BLX-α crossover slowed down at around 500th generation, MFEA-AKT adapted to the SBX crossover which converged much faster than the BLX-α crossover. Further, as shown in Figs. 6(c) and Fig. 7(b), the convergence traces of MFEA-AKT and MFEA-RKT are almost overlapped in the first few generations. This is because

TABLE I
AVERAGED PERFORMANCE SCORE OBTAINED BY MFEA-AKT, MFEA-RKT, AND MFEA WITH SIX DIFFERENT FIXED KNOWLEDGE TRANSFER CROSSOVERS AND GA ON THE NINE COMMON SO MULTITASK BENCHMARKS OVER 20 INDEPENDENT RUNS

Problem	Two-Point	Uniform	Arithmetical	Geometrical	BLX-α	SBX	MFEA-AKT	MFEA-RKT	GA
CIHS	-0.59	-0.56	-0.46	**-0.70**	-0.62	-0.19	**-0.70**	-0.67	4.50
CIMS	0.23	-0.22	-0.39	-0.03	**-1.05**	0.27	-0.34	-0.03	1.56
CILS	**-0.69**	-0.28	-0.14	-0.21	-0.09	-0.43	-0.33	-0.34	2.51
PIHS	**-1.07**	0.37	-0.55	-0.25	0.32	0.41	-0.86	-0.33	1.97
PIMS	-0.67	-1.02	-0.51	-0.31	**-1.12**	0.13	-0.97	-0.53	4.98
PILS	2.40	2.16	**-1.86**	-1.53	0.39	2.25	-1.85	-1.36	-0.61
NIHS	-0.43	-0.38	**-0.93**	-0.62	-0.65	-0.29	-0.77	-0.70	4.76
NIMS	-0.57	1.56	-1.17	-1.19	-0.15	-0.28	**-1.29**	-0.95	4.05
NILS	**-0.40**	0.26	-0.13	0.12	-0.03	0.59	-0.20	-0.25	-0.16
NSum	1.81	3.21	1.04	1.62	1.82	3.64	**0.70**	1.22	7.53

TABLE II
AVERAGED PERFORMANCE SCORE OBTAINED BY MFEA-AKT, MFEA-RKT, AND MFEA WITH SIX DIFFERENT FIXED KNOWLEDGE TRANSFER CROSSOVERS AND NSGAII ON THE NINE COMMON MO MULTITASK BENCHMARKS OVER 20 INDEPENDENT RUNS

Problem	Two-Point	Uniform	Arithmetical	Geometrical	BLX-α	SBX	MFEA-AKT	MFEA-RKT	NSGAII
CIHS	-0.26	-0.53	1.04	1.48	**-2.48**	-0.49	-1.87	-0.39	3.52
CIMS	-0.15	0.18	0.83	-0.26	-0.65	-0.21	**-0.73**	-1.00	1.98
CILS	-0.45	**-0.96**	0.33	-0.40	-0.59	-0.37	-0.69	-0.41	3.55
PIHS	-1.22	-0.24	2.39	2.62	0.56	-1.69	**-1.75**	0.60	-1.26
PIMS	-0.59	**-1.50**	1.41	0.84	0.05	-0.72	-0.79	-0.27	1.57
PILS	-1.03	0.37	-0.30	0.02	-0.27	0.22	**-1.04**	-0.41	2.45
NIHS	-0.55	-0.68	1.30	0.35	**-1.27**	-0.59	-1.12	0.72	1.85
NIMS	-0.19	-0.32	-0.67	-0.69	**-0.70**	0.71	-0.69	-0.34	2.91
NILS	0.29	0.67	**-1.75**	0.33	-0.24	0.64	-0.15	-0.28	0.49
NSum	2.40	2.77	4.43	4.48	2.07	2.95	**1.19**	2.93	8.04

the crossover operator for knowledge transfer in MFEA-AKT is also randomly selected at first. However, with the appropriate knowledge transfer crossover being identified in MFEA-AKT, the proposed MFEA-AKT searched significantly faster than MFEA-RKT in the following generations.

In addition, it is also observed that, on the multitask problems which share high intersections or similarities, for example, SO CIHS, SO CIMS, MO CILS, etc., different crossovers for knowledge transfer do not affect the optimization performance much. However, on the multitask problems with diverse properties, such as SO PILS, SO NIMS, and MO PILS, the proper employ of crossover for knowledge transfer plays an important role in determining the performance of evolutionary multitasking. In both cases, the proposed MFEA-AKT demonstrated robust and effective adaptation capability in selecting the appropriate crossover for knowledge transfer along the evolutionary multitasking process.

Moreover, the averaged *performance score* obtained by MFEA-AKT, MFEA-RKT, and MFEA with six different knowledge transfer crossovers and the single-task EAs on the common SO and MO benchmarks over 20 independent runs are summarized in Tables I and II, respectively. In particular, the *NSum* in the last row represents the overall performance of an algorithm on the entire benchmark set. To calculate the *NSum*, the scores of all compared algorithms for the same problem are first normalized via the min–max normalization method [51] and the normalized scores of each algorithm on all the test problems are then accumulated. The normalization operation is applied to scale the scores obtained on each problem into $[0, 1]$ so that the value of *NSum* will not be influenced by extremely large (or small) scores of one or two problems.

First, in the tables, it can be observed that MFEA achieved superior solution quality on all the multitask problems over single-task GA and NSGAII, no matter which crossover is employed for knowledge transfer. This again confirmed the effectiveness of evolutionary multitasking which is able to leverage the useful traits across optimization problems. However, it is also found that the performance of each crossover operator varies on different problems.

Furthermore, MFEA-AKT is observed to achieve the best overall performance, that is, *NSum*, against the other algorithms on both SO and MO benchmarks. Particularly, MFEA-AKT outperformed MFEA with the arithmetical crossover, which obtained the best overall performance on the common SO benchmarks among the MFEA variants with fixed crossover for knowledge, on six out of nine benchmarks. For MO benchmarks, MFEA-AKT achieved competitive or superior performance on six out of nine MO multitask problems than MFEA with BLX-α crossover for knowledge transfer. Furthermore, compared with MFEA-RKT, MFEA-AKT obtained superior solution quality on 14 out of 18 multitask problems. Particularly, due to the inappropriate selection of crossover operator for knowledge transfer, it is observed that the results of MFEA-RKT on MO PIHS is even worse than the single-task NSGAII. Since MFEA-AKT and MFEA-RKT share the same crossover operators to be selected for knowledge transfer, and the only difference is the adaptive approach incorporated in MFEA-AKT, the superior performance obtained by MFEA-AKT confirmed the efficacy of the proposed adaptive knowledge transfer approach.

Finally, to provide a deeper insight of the performance obtained by different crossover operators for knowledge transfer, we present the adaptation of T^a_{ci} and T^b_{ci} on representative multitask problems over generations in Fig. 8. In the figures, the x-axis denotes the number of generations, while the y-axis gives the ratio of each crossover employed for knowledge transfer and the identified best knowledge transfer crossover via (1) in a generation in the T^a_{ci} and T^b_{ci} figures, respectively. As can be observed, at the beginning of the evolutionary multitasking process, different crossovers have been employed for knowledge transfer across tasks. However, over generations, with more information being obtained along the optimization process, the most useful crossover is then identified as the best crossover (i.e., see the figures for T^b_{ci} adaptation), which is widely employed for knowledge transfer over generations (i.e., see the crossover which has the highest ratio in the figures for T^a_{ci} adaptation).

Moreover, we define *transfer distance* as the average Euclidean distance between all the *transferred offspring* created in a generation and their parents which hold different *skill factors*. The *transfer distance* estimates the fidelity of the knowledge from one task (carried by the parents) transferred to the other. A small *transfer distance* indicates the transferred knowledge is of high fidelity, while a large *transfer distance* implies that more variation could be introduced in the transferred knowledge.

The averaged *transfer distance* in the first 500 generations on all the SO and MO multitask problems is illustrated in Fig. 9(a) and (b), respectively. It is observed that the arithmetical crossover obtained the smallest *transfer distance* comparing with the other crossover operators. As it is well

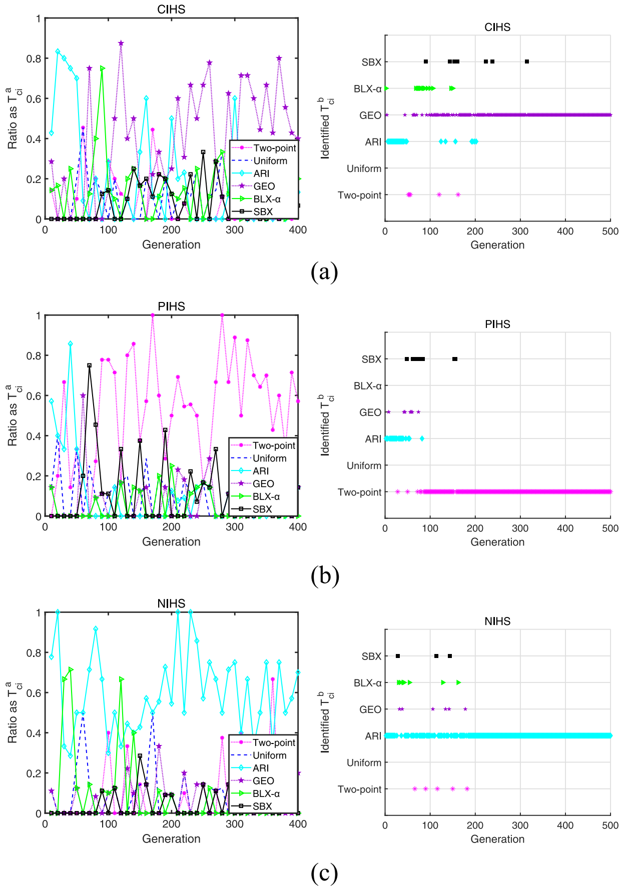


Fig. 8. Adaptation of T^a_{ci} and T^b_{ci} on representative multitask problems over generations. (a) Adaptation of T^a_{ci} (left figure) and T^b_{ci} (right figure) on *CIHS*. (b) Adaptation of T^a_{ci} (left figure) and T^b_{ci} (right figure) on *PIHS*. (c) Adaptation of T^a_{ci} (left figure) and T^b_{ci} (right figure) on *NIHS*.

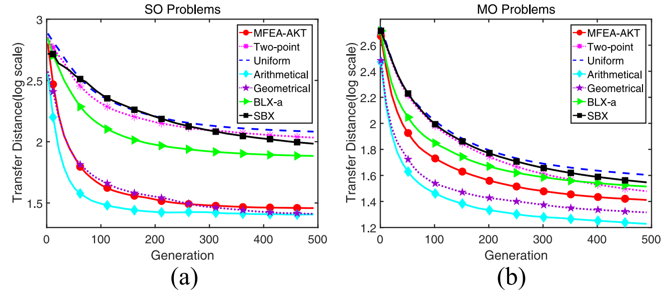


Fig. 9. Averaged transfer distance of MFEA-AKT and MFEA with six different fixed knowledge transfer crossovers in the first 500 generations on all the multitask problems. (a) SO multitask problems. (b) MO multitask problems.

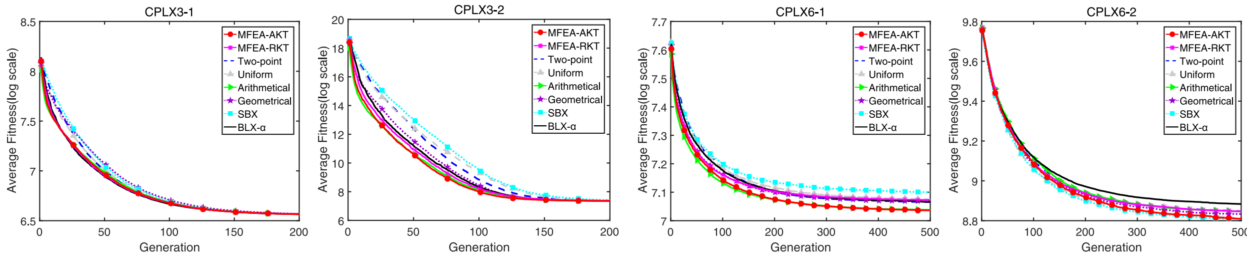
TABLE III
SUMMARY OF THE TEN COMPLEX SO AND MO MULTITASK PROBLEMS GENERATED IN THIS ARTICLE

Multitask Problems	Task No.	$CPLX1$	$CPLX2$	$CPLX3$	$CPLX4$	$CPLX5$	$CPLX6$	$CPLX7$	$CPLX8$	$CPLX9$	$CPLX10$
CEC2014 SO Functions	T1	$F_7(x)$	$F_7(x)$	$F_7(x)$	$F_8(x)$	$F_9(x)$	$F_9(x)$	$F_{10}(x)$	$F_{11}(x)$	$F_{13}(x)$	$F_{14}(x)$
	T2	$F_8(x)$	$F_{12}(x)$	$F_{15}(x)$	$F_{15}(x)$	$F_{10}(x)$	$F_{11}(x)$	$F_{16}(x)$	$F_{13}(x)$	$F_{14}(x)$	$F_{16}(x)$
LZ09 MO Functions	T1	$F1$	$F1$	$F2$	$F2$	$F3$	$F3$	$F4$	$F5$	$F6$	$F7$
	T2	$F2$	$F7$	$F4$	$F9$	$F6$	$F9$	$F5$	$F7$	$F9$	$F8$

established that the SO problem is relatively simple than the MO problem, if a solution of task A happens to be good to task B, to transfer the knowledge of the solution in a high fidelity to task B could be more helpful to the optimization of task B in the context of SO optimization. As a result, MFEA with the arithmetical crossover for knowledge transfer obtained superior performance in terms of convergence speed and solution quality on the SO multitask benchmark, as shown in Table I and Fig. 6. In contrast, as a set of Pareto-optimal solutions are required in MO, more diversity is required for solving MO problems. As illustrated in Table II, the performance of the arithmetical crossover degraded dramatically on the MO benchmark. Therefore, crossovers with larger *transfer distance*, such as BLX-α and two-point crossovers, obtained much better results than the arithmetical crossover. With respect to the MFEA-AKT, it is observed that the *transfer distance* of MFEA-AKT is close to arithmetical and geometrical crossover on SO problems [see Fig. 9(a)], while becoming large (close to the trace of the BLX-α crossover) on MO problems [see Fig. 9(b)]. This explains the efficient and robust performance obtained by the proposed MFEA-AKT on both SO and MO multitask problems.

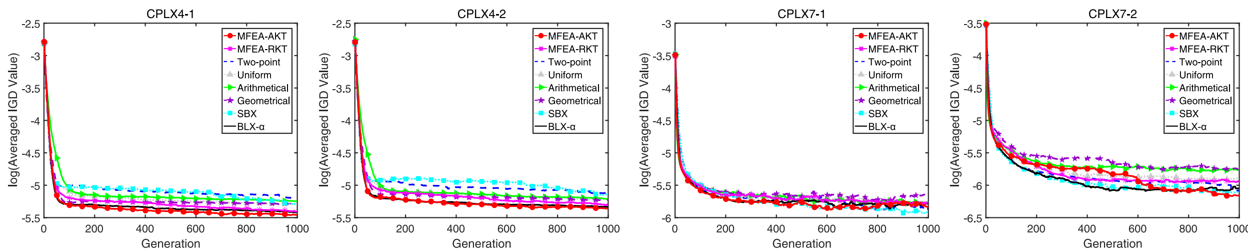
2) Complex Multitask Problems: In this part, the effectiveness of the proposed MFEA-AKT is further investigated on ten complex SO and ten complex MO multitask problems, respectively. Particularly, to construct the new complex multitask problems, the individual functions $F_7(x) - F_{16}(x)$ of the CEC2014 SO test suites [37] and $F1 - F9$ of the LZ09 MO benchmarks [38], which are more complex than tasks in the common multitask benchmarks, are utilized in this article. The multitask problems are then constructed by randomly pairing the functions. The details of the new multitask problems (e.g., both SO and MO *CPLX*1 to SO and MO *CPLX*10) are summarized in Table III.

The averaged convergence graphs and *performance score* obtained by MFEA-AKT, MFEA-RKT, and MFEA with six different knowledge transfer crossovers on the complex SO and MO multitask problems over 20 independent runs are illustrated in Tables IV and V and Figs. 10 and 11, respectively. It is found that the superiority of MFEA-AKT can also be observed in these complex multitask problems. First, in Tables IV and V, we can see that MFEA-AKT obtained the best overall performance on both SO and MO complex multitask problems. Next, as shown in Figs. 10 and 11, MFEA-AKT achieved a robust and efficient search performance on all the complex problems when compared to the baseline algorithms. It is observed that MFEA-AKT is able to identify the appropriate knowledge transfer crossover for different optimization problems. For instance, in Figs. 10(b) and 11(a), MFEA-AKT achieved superior performance as MFEA with the arithmetical crossover and BLX-α crossover on SO CPLX6 and MO CPLX4, respectively. Furthermore, it is again observed that MFEA-AKT is able to identify the proper knowledge transfer crossover at different optimization stages. In particular, in Fig. 11(b), MFEA-AKT performed competitively over the MFEA with the arithmetical crossover on CPLX7-2, initially. However, as the arithmetical crossover stagnated around the 200th generation, MFEA-AKT gradually adapted to the SBX crossover and converged to the solution with superior fitness. Finally, compared to MFEA-RKT, on 20 complex multitask problems, MFEA-AKT obtained superior



(a) (b)

Fig. 10. Convergence traces of the averaged fitness obtained by MFEA-AKT, MFEA-RKT, and MFEA with six different fixed knowledge transfer crossovers on representative complex SO multitask problems *CPLX*3 and *CPLX*6 over 20 independent runs. *y*-axis: log(Averaged fitness); *x*-axis: Generation. (a) Convergence traces of *CPLX*3. (b) Convergence traces of *CPLX*6.



(a) (b)

Fig. 11. Convergence traces of the averaged IGD obtained by MFEA-AKT, MFEA-RKT, and MFEA with six different fixed knowledge transfer crossovers on representative complex MO multitask problems *CPLX*4 and *CPLX*7 over 20 independent runs. *y*-axis: log(Averaged IGD value); *x*-axis: Generation. (a) Convergence traces of *CPLX*4. (b) Convergence traces of *CPLX*7.

TABLE IV
AVERAGED PERFORMANCE SCORE OBTAINED BY MFEA-AKT, MFEA-RKT, AND MFEA WITH SIX DIFFERENT FIXED KNOWLEDGE TRANSFER CROSSOVERS AND GA ON THE TEN COMPLEX SO MULTITASK PROBLEMS OVER 20 INDEPENDENT RUNS

Problem	Two-Point	Uniform	Arithmetical	Geometrical	BLX-α	SBX	MFEA-AKT	MFEA-RKT	GA
CPLX1	-0.67	-0.69	-0.64	-0.69	**-0.77**	-0.67	-0.75	-0.70	5.59
CPLX2	-0.27	-0.38	-0.27	**-0.73**	-0.02	-0.60	-0.41	-0.19	2.86
CPLX3	-0.07	-0.06	-0.43	**-0.64**	-0.39	-0.13	-0.26	-0.05	2.04
CPLX4	-0.51	-0.35	**-0.71**	-0.09	-0.26	-0.23	-0.52	-0.17	2.84
CPLX5	**-0.55**	-0.08	-0.51	-0.36	-0.16	0.29	-0.37	-0.51	2.25
CPLX6	0.01	-0.11	-0.66	-0.07	0.35	0.52	**-1.03**	0.17	0.84
CPLX7	-0.32	-0.29	**-0.98**	-0.61	-0.05	-0.51	-0.95	-0.31	4.01
CPLX8	-0.07	0.25	0.15	0.06	0.06	0.35	**-0.16**	0.27	-0.90
CPLX9	-0.59	0.24	**-0.86**	-0.85	-0.38	0.20	-0.19	-0.57	3.00
CPLX10	-0.70	0.39	**-0.71**	-0.43	0.01	0.74	-0.52	-0.27	1.48
NSum	1.84	2.94	1.28	1.75	2.70	3.54	**1.21**	2.54	9.00

TABLE V
AVERAGED PERFORMANCE SCORE OBTAINED BY MFEA-AKT, MFEA-RKT, AND MFEA WITH SIX DIFFERENT FIXED KNOWLEDGE TRANSFER CROSSOVERS AND NSGAII ON THE TEN COMPLEX MO MULTITASK PROBLEMS OVER 20 INDEPENDENT RUNS

Problem	Two-Point	Uniform	Arithmetical	Geometrical	BLX-α	SBX	MFEA-AKT	MFEA-RKT	NSGAII
CPLX1	-0.18	0.54	0.25	-0.14	0.95	-0.07	**-0.57**	0.04	-0.82
CPLX2	1.02	0.77	-1.79	**-1.86**	0.73	1.10	-1.68	-1.78	3.48
CPLX3	-0.27	-0.32	0.24	1.49	0.13	**-0.48**	-0.31	-0.17	-0.32
CPLX4	0.65	-0.03	0.16	-0.03	-0.86	0.20	**-1.07**	-0.70	1.69
CPLX5	-0.22	-0.19	1.43	0.14	-0.26	-0.20	**-0.62**	-0.25	0.17
CPLX6	0.09	0.25	**-0.71**	0.20	-0.35	-0.86	-0.24	-0.70	2.32
CPLX7	-0.44	0.23	0.70	1.26	0.02	**-0.77**	-0.45	0.29	-0.84
CPLX8	-0.21	-0.07	0.56	0.55	-0.21	**-0.68**	0.19	0.52	-0.65
CPLX9	-0.29	0.18	0.35	0.64	-0.68	0.17	**-0.78**	-0.30	0.70
CPLX10	-0.31	0.41	0.68	0.94	-0.18	-0.64	**-0.59**	-0.29	-0.01
NSum	3.24	4.59	5.81	6.42	3.36	2.32	**1.38**	3.08	4.89

performance in terms of overall performance score on 16 benchmarks.

D. Other Issues on MFEA-AKT

In this section, empirical studies are presented to investigate the following two aspects of MFEA-AKT.

1) How does MFEA-AKT behave with different number of crossover operators available for knowledge transfer?
2) How does MFEA-AKT behave if other crossovers are employed for generating offspring without knowledge transfer?

For the first question, MFEA-AKT and MFEA-RKT with two, four and six crossovers for knowledge transfer are investigated on the common SO and MO multitask benchmarks over 20 independent runs, which are summarized in Tables VI and VII,[4] respectively. In these two tables, A and R denote the MFEA-AKT and MFEA-RKT, respectively. The number next to A and R indicates the number of crossover operators employed for knowledge transfer. As can be observed from Table VI, in terms of overall performance, that is, *NSum*, A2, A4, and A6 have won the third-, second-, and first-place among all the six algorithms on the SO problems, respectively. Furthermore, among A2, A4, and A6, with the increased number of crossover operators, the obtained corresponding *NSum* value has been improved from 3.11 of A2 to 1.25 of A6. However, among R2, R4, and R6, as the crossover for knowledge transfer is randomly configured along the multitask optimization process, the increase of number of crossover operators will decrease the probability to select the proper one for knowledge transfer. Therefore, it is observed that the obtained *NSum* value become worse from 4.20 of R2 to 8.01 of R6, accordingly. For the investigation on the MO multitask benchmarks in Table VII, similar observation can be obtained. From these observations, we can conclude that, first, to increase the number of available crossovers could improve the multitasking performance of MFEA-AKT,

[4]The results of A6 in these two tables are corresponding to the results presented in Tables I and II, respectively. However, as the performance scores calculated via (2) are related to the compared algorithms, the obtained score values in these two tables are different from those in Tables I and II.

TABLE VI
AVERAGED PERFORMANCE SCORE OBTAINED BY MFEA-AKT AND MFEA-RKT WITH DIFFERENT NUMBER OF CROSSOVER OPERATORS EMPLOYED FOR KNOWLEDGE TRANSFER ON THE NINE PROBLEMS OF THE SO MULTITASK BENCHMARKS OVER 20 INDEPENDENT RUNS

Problem	A2	R2	A4	R4	A6	R6
CIHS	-0.25	-0.18	0.04	0.23	**-0.58**	0.74
CIMS	-0.05	0.10	-0.29	0.13	**-0.67**	0.78
CILS	0.09	0.12	-0.01	-0.34	**-0.43**	0.56
PIHS	0.15	0.16	**-0.50**	0.17	-0.26	0.29
PIMS	0.12	-0.01	**-0.62**	0.65	-0.48	0.35
PILS	**-0.55**	-0.21	-0.16	0.33	-0.52	1.11
NIHS	-0.44	0.91	**-0.48**	0.39	-0.23	-0.15
NIMS	-0.17	-0.19	-0.29	0.56	**-0.56**	0.66
NILS	-0.29	**-0.39**	0.21	0.19	0.02	0.26
NSum	3.11	4.20	2.53	6.06	**1.25**	8.01

TABLE VII
AVERAGED PERFORMANCE SCORE OBTAINED BY MFEA-AKT AND MFEA-RKT WITH DIFFERENT NUMBER OF CROSSOVER OPERATORS EMPLOYED FOR KNOWLEDGE TRANSFER ON THE NINE PROBLEMS OF THE MO MULTITASK BENCHMARKS OVER 20 INDEPENDENT RUNS

Problem	A2	R2	A4	R4	A6	R6
CIHS	-0.85	2.33	-1.79	0.98	**-1.48**	0.81
CIMS	0.68	**-0.31**	0.31	-0.09	-0.08	-0.51
CILS	-0.87	1.13	-0.65	0.73	**-0.93**	0.60
PIHS	-1.42	1.41	**-1.49**	1.95	-1.45	1.00
PIMS	0.40	0.20	0.05	0.01	**-0.61**	-0.05
PILS	0.57	0.57	0.01	0.31	**-1.38**	-0.08
NIHS	-1.25	2.36	-1.84	1.11	**-1.86**	1.48
NIMS	-0.19	-0.04	-0.06	0.54	**-0.48**	0.23
NILS	0.47	0.93	0.28	0.50	**-1.69**	-0.49
NSum	4.53	7.25	3.36	6.85	**0.45**	5.27

TABLE VIII
NSum VALUES OBTAINED BY MFEA-AKT AND MFEA WITH SIX DIFFERENT KNOWLEDGE TRANSFER CROSSOVERS ON THE COMMON SO AND MO BENCHMARKS OVER 20 INDEPENDENT RUNS, WHERE THE TWO-POINT, ARITHMETICAL, AND SBX CROSSOVERS ARE EMPLOYED FOR GENERATING OFFSPRING WITHOUT KNOWLEDGE TRANSFER, RESPECTIVELY

Variant	Two-Point	Uniform	Arithmetical	Geometrical	BLX-α	SBX	MFEA-AKT
SO-TP	4.93	6.04	4.91	3.42	2.94	4.61	**1.68**
SO-ARI	4.50	5.92	4.35	4.38	5.89	2.83	**2.70**
SO-SBX	3.77	4.56	2.52	3.50	2.81	8.25	**1.33**
MO-TP	4.37	5.20	5.67	5.24	2.97	4.78	**1.44**
MO-ARI	**0.57**	1.72	8.87	4.42	1.38	1.45	1.26
MO-SBX	3.19	4.04	6.26	6.49	2.32	4.59	**1.12**

since it increased the diversity of knowledge transfer. Second, increasing the number of available crossovers without adaptive design may even impede the multitasking performance, since it reduced the probability of identifying the appropriate crossover for knowledge transfer.

Furthermore, for the second question, the *NSum* values obtained by MFEA-AKT and MFEA with six different knowledge transfer crossovers on the common SO and MO multitask benchmarks over 20 independent runs are reported in Table VIII. In the first column of Table VIII, "SO" and "MO" indicate the type of the benchmarks, while "TP," "ARI," and "SBX" denote the two-point, arithmetical, and SBX crossovers that have been employed for generating offspring without knowledge transfer, respectively. In the table, it can be observed that the setting of different crossovers for generating offspring without knowledge transfer resulted in diverse multitasking performance. For instance, for SO-ARI, where the arithmetical crossover is employed only for generating offspring, MFEA with the SBX crossover outperformed MFEA using BLX-α for knowledge transfer. However, in the case of SO-SBX, using BLX-α for knowledge transfer in MFEA achieved better *NSum* value than SBX. Nevertheless, we can observe that with the proposed approach for adaptively configuring the crossover for knowledge transfer, the MFEA-AKT achieved the best overall performance on 5 out of 6 scenarios, which again confirmed the effectiveness of the proposed adaptive approach for efficient and robust evolutionary multitasking performance.

VI. CONCLUSION

In this article, MFEA with different crossover operators for knowledge transfer has been investigated. It is found that each crossover operator has unique knowledge transfer capability in solving the given multitask optimization problems. To achieve an efficient and robust evolutionary multitasking performance, a new MFEA with adaptive knowledge transfer called MFEA-AKT has been proposed. In contrast to the original MFEA and most of its variants which employ a fixed crossover for knowledge transfer, MFEA-AKT is able to adaptively configure the crossover operator for knowledge transfer across tasks based on the information collected along the evolutionary search process. To validate the efficacy of the proposed MFEA-AKT, comprehensive studies have been conducted on commonly used SO and MO multitask benchmarks as well as ten new complex SO and ten new complex MO multitask problems. Experimental results have shown that the proposed MFEA-AKT is able to identify the appropriate knowledge transfer crossover for different optimization problems and different optimization stages along the search, which thus leads to superior or competitive performances on both SO and MO optimization problems when compared to the MFEAs with fixed knowledge transfer crossover operators.

For future work, toward a unified adaptation framework, we would like to extend the proposed adaptation algorithm for genetic crossover and integrate the crossover adaptation of knowledge transfer and genetic evolutionary into a common framework. Moreover, in this article, the adaptation of the crossover for knowledge transfer is from the perspective of individuals. However, the adaptation of knowledge transfer could also be conducted from the task perspective. In the future, we would also like to conduct a deeper analysis following this direction. Finally, a straightforward extension of this article is to extend the proposed adaptation of knowledge transfer for many task optimization (i.e., in cases having more than two tasks). In this case, the transfer crossover indicator T_{ci} and the best transfer crossover indicator T_{ci}^{b} will be defined as a vector or a matrix, where each element gives the crossover and best crossover for knowledge transfer between two specific tasks, respectively.

REFERENCES

[1] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, Jun. 2016.

[2] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: A computer science view of cognitive multitasking," *Cogn. Comput.*, vol. 8, no. 2, pp. 125–142, 2016.

[3] A. Gupta, Y.-S. Ong, B. Da, L. Feng, and S. D. Handoko, "Landscape synergy in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2016, pp. 3076–3083.

[4] B. Da, A. Gupta, Y. S. Ong, and L. Feng, "The boon of gene-culture interaction for effective evolutionary multitasking," in *Proc. Aust. Conf. Artif. Life Comput. Intell.*, 2016, pp. 54–65.

[5] A. Gupta and Y.-S. Ong, "Genetic transfer or population diversification? Deciphering the secret ingredients of evolutionary multitask optimization," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, 2016, pp. 1–7.

[6] L. Zhou, L. Feng, J. Zhong, Y.-S. Ong, Z. Zhu, and E. Sha, "Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, 2016, pp. 1–8.

[7] A. Gupta, J. Mańdziuk, and Y. S. Ong, "Evolutionary multitasking in bi-level optimization," *Complex Intell. Syst.*, vol. 1, nos. 1–4, pp. 83–95, 2015.

[8] Y. Yuan, Y.-S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2016, pp. 3157–3164.

[9] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1652–1665, Jul. 2017.

[10] R.-T. Liaw and C.-K. Ting, "Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 2266–2273.

[11] K. K. Bali, A. Gupta, L. Feng, Y.-S. Ong, and T. P. Siew, "Linearized domain adaptation in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 1295–1302.

[12] J. Ding, C. Yang, Y. Jin, and Y. Chai, "Generalized multi-tasking for evolutionary optimization of expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 44–58, Feb. 2019.

[13] M. Gong, Z. Tang, H. Li, and J. Zhang, "Evolutionary multitasking with dynamic resource allocating strategy," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 858–869, Oct. 2019.

[14] Y. Wen and C.-K. Ting, "Parting ways and reallocating resources in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 2404–2411.

[15] J. Tang, Y. Chen, Z. Deng, Y. Xiang, and C. P. Joy, "A group-based approach to improve multifactorial evolutionary algorithm," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 3870–3876.

[16] K. K. Bali, Y. S. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 69–83, Feb. 2020.

[17] G. Pavai and T. V. Geetha, "A survey on crossover operators," *ACM Comput. Surveys*, vol. 49, no. 4, p. 72, 2017.

[18] M. Coli, G. Gennuso, and P. Palazzari, "A new crossover operator for genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1996, pp. 201–206.

[19] J. C. Bongard, "A probabilistic functional crossover operator for genetic programming," in *Proc. 12th Annu. Conf. Genet. Evol. Comput.*, 2010, pp. 925–932.

[20] X. Qiu, K. C. Tan, and J.-X. Xu, "Multiple exponential recombination for differential evolution," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 995–1006, Apr. 2017.

[21] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, 1994.

[22] R. Poli and W. B. Langdon, "Genetic programming with one-point crossover," in *Proc. Soft Comput. Eng. Design Manuf.*, 1998, pp. 180–189.

[23] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proc. 3rd Int. Conf. Genet. Algorithms*, 1989, pp. 2–9.

[24] X. Hu and E. D. Paolo, "An efficient genetic algorithm with uniform crossover for air traffic control," *Comput. Oper. Res.*, vol. 36, no. 1, pp. 245–259, 2009.

[25] R. B. Agrawal, K. Deb, and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[26] K. Deb and H. G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evol. Comput.*, vol. 9, no. 2, pp. 197–221, 2001.

[27] K. Deep and H. Mebrahtu, "New variations of order crossover for travelling salesman problem," *Int. J. Comb. Optim. Problems Informat.*, vol. 2, no. 1, pp. 2–13, 2011.

[28] J. R. Radley and L. W. Robinson, "Improved base-stock approximations for independent stochastic lead times with order crossover," *Manuf. Service Oper. Manag.*, vol. 7, no. 4, pp. 319–329, 2005.

[29] D. E. Goldberg and R. Lingle, "AllelesLociand the traveling salesman problem," in *Proc. 1st Int. Conf. Genet. Algorithms*, 1985, pp. 154–159.

[30] O. Kramer and P. Koch, "Self-adaptive partially mapped crossover," in *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, 2007, p. 1523.

[31] G. Lin and X. Yao, "Analysing crossover operators by search step size," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1997, pp. 107–110.

[32] T. Bäck, *A User's Guide to Genesys 1.0*, Univ. Dortmund, Dortmund, Germany, 1992.

[33] Z. Michalewicz, G. Nazhiyath, and M. Michalewicz, "A note on usefulness of geometrical crossover for numerical optimization problems," *Evol. Program.*, vol. 5, no. 1, pp. 305–312, 1996.

[34] B. Da *et al.*, " Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results," 2017. [Online]. Available: arXiv:1706.03470.

[35] Y. Yuan *et al.*, "Evolutionary multitasking for multiobjective continuous optimization: Benchmark problems, performance metrics and baseline results," 2017. [Online]. Available: arXiv:1706.02766.

[36] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[37] J. Liang, B. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Rep., 2013.

[38] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[39] N. Durand and J. M. Alliot, "Genetic crossover operator for partially separable functions," in *Proc. 3rd Annu. Conf. Genet. Program.*, 1998, pp. 487–494.

[40] J.-J. Lin, S.-C. Huang, and M.-K. Jiau, "An evolutionary multiobjective Carpool algorithm using set-based operator based on simulated binary crossover," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3432–3442, Sep. 2019.

[41] S. Hui and P. N. Suganthan, "Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 64–74, Jan. 2016.

[42] F. Herrera, M. Lozano, and A. M. Sánchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *Int. J. Intell. Syst.*, vol. 18, no. 3, pp. 309–338, 2003.

[43] T. Jones, "Crossover, macromutation, and population-based search," in *Proc. 6th Int. Conf. Genet. Algorithms*, 1995, pp. 73–80.

[44] E. Falkenauer, "The worth of the uniform [uniform crossover]," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 1, 1999, pp. 776–782.

[45] Z. Michalewicz and S. J. Hartley, "Genetic algorithms + data structures = Evolution programs," *Math. Intell.*, vol. 18, no. 3, p. 71, 1996.

[46] J. Feder, E. L. Hinrichsen, K. J. Måløy, and T. Jøssang, "Geometrical crossover and self-similarity of DLA and viscous fingering clusters," *Physica D Nonlin. Phenom.*, vol. 38, nos. 1–3, pp. 104–111, 1989.

[47] A. H. Wright, "Genetic algorithms for real parameter optimization," *Found. Genet. Algorithms*, vol. 1, pp. 205–218, Jun. 2014.

[48] H. M. Voigt, H. Mühlenbein, and D. Cvetkovic, "Fuzzy recombination for the breeder genetic algorithm," in *Proc. 6th Int. Conf. Genet. Algorithms*, 1995, pp. 104–113.

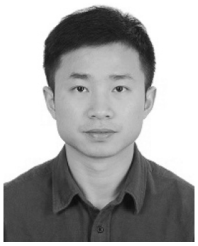
[49] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," *Found. Genet. Algorithms*, vol. 2, pp. 187–202, Jun. 2014.

[50] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, 2007, pp. 1187–1194.

[51] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," 2015. [Online]. Available: arXiv:1503.06462.

Lei Zhou received the B.E. degree from the School of Computer Science and Technology, Shandong University, Jinan, China, in 2014, and the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2019.

He is currently a Postdoctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include evolutionary computations, memetic computing, as well as transfer learning and optimization.



Liang Feng received the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2014.

He was a Postdoctoral Research Fellow with the Computational Intelligence Graduate Lab, Nanyang Technological University. He is currently an Assistant 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, as well as transfer learning.

Prof. Feng's research work on evolutionary multitasking won the 2019 IEEE Transactions on Evolutionary Computation Outstanding Paper Award. He is an Associate Editor of the *IEEE Computational Intelligence Magazine*, *Memetic Computing*, and *Cognitive Computation*. He is also the Founding Chair of the IEEE CIS Intelligent Systems Applications Technical Committee Task Force on "Transfer Learning & Transfer Optimization."

Kay Chen Tan received the B.Eng. degree (Hons.) in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published over 200 refereed articles and six books.

Prof. Tan was the Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2010 to 2013, the Editor-in-Chief of the IEEE Transactions on Evolutionary Computation, and currently serves as the Editorial Board Member of over ten journals. He was an Elected Member of the IEEE CIS AdCom from 2017 to 2019.

Jinghui Zhong received the Ph.D. degree from the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, in 2012.

He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. From 2013 to 2016, he worked as a Postdoctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include evolutionary computation (e.g., genetic programming and differential evolution), machine learning, and agent-based modeling.

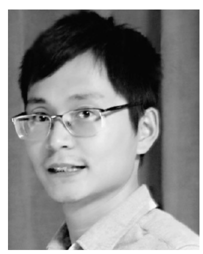
Zexuan Zhu received the B.S. degree in computer science and technology from Fudan University, Shanghai, China, in 2003, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2008.

He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include computational intelligence, machine learning, and bioinformatics.

Prof. Zhu is an Associate Editor of the IEEE Transactions on Evolutionary Computation and the IEEE Transactions on Emerging Topics in Computational Intelligence, and serves as an Editorial Board Member for *Memetic Computing* and *Soft Computing*. He is also the Chair of the IEEE CIS, Emergent Technologies Task Force on Memetic Computing.

Kai Liu received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2011.

From 2010 to 2011, he was a Visiting Scholar with the Department of Computer Science, University of Virginia, Charlottesville, VA, USA. From 2011 to 2014, he was a Postdoctoral Fellow with Nanyang Technological University, Singapore; the City University of Hong Kong; and Hong Kong Baptist University, Hong Kong. He is currently a Professor with the College of Computer Science, Chongqing University, Chongqing, China. His research interests include Internet of Vehicles, mobile computing, pervasive computing, and big data.



Chao Chen received the B.Sc. and M.Sc. degrees in control science and control engineering from Northwestern Polytechnical University, Xi'an, China, in 2007 and 2010, respectively, and the Ph.D. degree from Pierre and Marie Curie University, Paris, France, and Institut Mines-Télécom/Télécom SudParis, Évry, France, in 2014.

He is a Full Professor with the College of Computer Science, Chongqing University, Chongqing, China. He has published over 80 papers, including 20 ACM/IEEE transactions. His work on taxi trajectory data mining was featured by *IEEE Spectrum* in 2011 and 2016, respectively. In 2009, he worked as a Research Assistant with Hong Kong Polytechnic University, Hong Kong. His research interests include pervasive computing, mobile computing, urban logistics, data mining from large-scale GPS trajectory data, and big data analytics for smart cities.

Prof. Chen was the Winner of the Best Paper Runner-Up Award at MobiQuitous 2011.