Cognizant Multitasking in Multiobjective Multifactorial Evolution: MO-MFEA-II

Kavitesh Kumar Bali, Abhishek Gupta, Yew-Soon Ong, *Fellow, IEEE*, and Puay Siew Tan

***Abstract*—Humans have the ability to identify recurring patterns in diverse situations encountered over a lifetime, constantly understanding relationships between tasks and efficiently solving them through knowledge reuse. The capacity of artificial intelligence systems to mimic such cognitive behaviors for effective problem solving is deemed invaluable, particularly when tackling real-world problems where speed and accuracy are critical. Recently, the notion of *evolutionary multitasking* has been explored as a means of solving multiple optimization tasks simultaneously using a single population of evolving individuals. In the presence of similarities (or even partial overlaps) between high-quality solutions of related optimization problems, the resulting scope for intertask genetic transfer often leads to significant performance speedup—as the cost of re-exploring overlapping regions of the search space is reduced. While multitasking solvers have led to recent success stories, a known shortcoming of existing methods is their inability to adapt the extent of transfer in a principled manner. Thus, in the absence of any prior knowledge about the relationships between optimization functions, a threat of predominantly *negative (harmful) transfer* prevails. With this in mind, this article presents a realization of a *cognizant* evolutionary multitasking engine within the domain of multiobjective optimization. Our proposed algorithm learns intertask relationships based on overlaps in the probabilistic search distributions derived from data generated during the course of multitasking—and accordingly adapts the extent of genetic transfers *online*. The efficacy of the method is substantiated on multiobjective benchmark problems as well as a practical case study of knowledge transfers from low-fidelity optimization tasks to substantially reduce the cost of high-fidelity optimization.**

***Index Terms*—Evolutionary multitasking, multifactorial optimization, multiobjective optimization, online similarity learning, probabilistic modeling.**

Manuscript received September 4, 2019; revised January 10, 2020; accepted March 3, 2020. This work was supported in part by the A*STAR Cyber-Physical Production Systems Research Project through IAF-PP under Grant A19C1a0018, and in part by the Singapore Institute of Manufacturing Technology-Nanyang Technological University (SIMTech-NTU) Joint Laboratory and Collaborative Research Programme on Complex Systems. This article was recommended by Associate Editor Q. Zhang. *(Corresponding author: Kavitesh Kumar Bali.)*

Kavitesh Kumar Bali is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: bali0001@e.ntu.edu.sg).

Abhishek Gupta and Puay Siew Tan are with the Singapore Institute of Manufacturing Technology, Agency for Science, Technology and Research, Singapore 138634 (e-mail: abhishek_gupta@simtech.a-star.edu.sg; pstan@simtech.a-star.edu.sg).

Yew-Soon Ong is with the Data Science and Artificial Intelligence Research Centre, School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, and also with the Agency for Science, Technology and Research, Singapore 138632 (e-mail: asysong@ntu.edu.sg).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

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.2020.2981733

I. Introduction

THE HUMAN brain is a self-learning system. Among various other things, it has the general capability of comprehending its surroundings—being *"aware of what is going on"* and *"figuring out"* what to do [1], [2]. This intrinsic feature enables humans to draw inferences from what one experiences, recognize recurring patterns (relationships) between diverse tasks, and share knowledge across related problems [3]. It is indeed this ability to leverage on what we already know to tackle related tasks that play a significant role in boosting our efficiency of problem solving and decision making [4].

The ability to learn from experience is considered invaluable for machines as well. Given modern technologies with large-scale data storage and seamless communication facilities, the development of machines that mimic human-like behaviors of autonomously extracting, processing, and transferring learned knowledge across related problems is surely an attractive proposition. In this regard, the concept of *multitasking* has drawn much interest in the field of computational intelligence [5]–[9]. Not surprisingly, present day artificial intelligence systems are emerging as powerful tools for multitasking, capable of simultaneously handling myriad tasks with greater speed and accuracy.

Evolutionary multitasking, a relatively new paradigm in the field of evolutionary computation, has recently been explored as a population-based search methodology for solving multiple optimization problems concurrently [7]. Notably, in the presence of underlying relationships between constituent problems, evolutionary multitasking leads to the possibility of fruitful knowledge transfers, thereby facilitating improved convergence characteristics. Many success stories have thus surfaced in recent years, encompassing the domains of discrete, continuous, single-objective and multiobjective optimization, including various applications in machine learning [10]–[22]. The practicality of evolutionary multitasking has also been demonstrated across a wide array of real-world applications spanning the manufacturing process design [14]; robot controller design [23]; vehicle routing [4], [7]; cloud computing [24]; and software engineering [25]; to name a few.

While practitioners have many algorithmic options to choose from, it is noted that the applicability and success of present day multitasking solvers depend strongly on task relatedness. For those cases where optimization problems share little in common, blindly transferring knowledge (in the form of encoded solutions) across them can lead to predominantly

negative (harmful) effects [14], [26], [27]. Susceptibility to negative intertask interactions has, in fact, been shown to impede overall convergence behavior [23]. Existing evolutionary multitasking algorithms, such as the multiobjective multifactorial evolutionary algorithm (MO-MFEA) [14], generally lack the cognitive capability of *situational awareness* [2], and thus, are unable to decipher and adapt to the degree of similarity between distinct tasks on the fly.

In light of the above, a matter of increasing importance has been the design of algorithms that are capable of processing incoming streams of data from multiple optimization tasks, and identifying relationships between them. Accordingly, in this article, we present a *cognizant* evolutionary multitasking approach, called MO-MFEA-II, which is adept at *online* intertask similarity learning and adaptive transfer. The MO-MFEA-II incorporates probabilistic modeling of the data generated online during the multitasking search to unveil recurring solution patterns (measured by the similarity in search distributions) across different tasks. The proposed *data-driven* approach showcases *awareness* as to *when* and *how much* knowledge is to be transferred—much like the conscious human mind that decides which tasks to focus on and when to suppress irrelevant information [28]. We note that the foundations of MO-MFEA-II encompass ideas described in a recently introduced multifactorial evolutionary algorithm with online learning capability (MFEA-II) [23]. While MFEA-II focuses purely on single-objective optimization, the MO-MFEA-II proposed herein analyzes how similar concepts can apply to the domain of multiobjective optimization as well. Thus, to the best of our knowledge, this article is among the first to utilize probabilistic modeling to capture intertask relationships between multiobjective optimization tasks in the context of evolutionary multitasking.

The efficacy of MO-MFEA-II is showcased first on a series of synthetic test functions. Further, we highlight a natural and practically useful application of the approach to the domain of multifidelity optimization; that is, leveraging on low-fidelity function approximations to accelerate the optimization performance of related high-fidelity tasks via multitasking.

The organization of the remainder of this article is as follows. Section II presents the backgrounds of multiobjective optimization and associated multiobjective evolutionary algorithms (MOEAs). In addition, we provide a brief review of related works in the literature and summarize the existing MO-MFEA framework. The MO-MFEA-II with online transfer adaptation capability is presented in Section III. The experimental results and analyses on a wide range of multiobjective benchmarks and a practical case study are provided in Sections IV and V, respectively. Section VI concludes this article with a discussion on future works.

II. Background

In this section, we first present preliminaries of multiobjective optimization and associated single-task evolutionary algorithms (EAs). Next, a brief review of the recent advances in evolutionary multitasking, particularly in the domain of multiobjective optimization is presented. The outlined contributions of this article are compared and contrasted against those available in the literature. Finally, the existing MO-MFEA is discussed and its potential limitations highlighted.

A. Multiobjective Optimization

The goal of solving multiobjective optimization problems (MOOPs) is to find multiple tradeoff solutions that offer an optimal compromise between conflicting objectives of interest. In what follows, we provide a general formulation for a MOOP.

Given a decision space $\boldsymbol{X} \subset \mathbb{R}^D$, a multiobjective minimization problem can be defined (without loss of generality) as follows:

$$\text{minimize } F(x) = (f_1(x), f_2(x), \ldots, f_M(x)) \tag{1}$$

where $f_1(x), f_2(x), \ldots, f_M(x)$ are M different minimization objectives, $F(x)$ is the overall objective vector, and $x \in \boldsymbol{X}$ is a decision vector. For the sake of brevity, additional constraint functions have been suppressed in (1). Following the concept of *Pareto dominance* [14], a solution x_1 is said to *dominate* another solution x_2 (i.e., $x_1 \prec x_2$) iff $\forall i \in \{1, 2, \ldots, M\} : f_i(x_1) \leq f_i(x_2)$, and $\exists j \in \{1, 2, \ldots, M\} : f_j(x_1) < f_j(x_2)$. Accordingly, a solution x^* is considered to be Pareto optimal if x^* is not dominated by any other solution in $\boldsymbol{X}$. The objective function values corresponding to all the Pareto-optimal solutions collectively form the Pareto front [29].

B. Multiobjective Evolutionary Algorithms

EAs are considered a popular choice for solving MOOPs. In particular, the implicit parallelism offered by a population enables simultaneous sampling, evaluation, and processing of multiple regions of the search space, leading to a reasonable representation of the entire set of Pareto-optimal solutions to be found in only a single run of the solver [9], [26]. In the literature, some examples of MOEAs that are commonly used today include nondominated sorting genetic algorithm (NSGA-II) [30], NSGA-III [31], [32], SPEA2 [14], [33], and MOEA/D [34]–[36], to name a few. While a plethora of MOEAs has been proposed over the years, the focus has mostly been to efficiently solve only one MOOP at a time. On the other hand, attempts to tackle multiple-related MOOPs simultaneously (in the spirit of evolutionary multitasking) have been relatively scarce.

C. Evolutionary Multitasking in Multiobjective Optimization

Since the conceptualization of evolutionary multitasking, associated algorithmic developments have been seen in the domain of multiobjective optimization as well [14], [37]. In particular, a handful of methods equipped with adaptive knowledge transfer capabilities has also come to the fore. For example, Feng *et al.* [8] proposed to learn optimal linear mappings between different continuous multiobjective tasks using a denoising autoencoder. In this method, the learned mappings serve as a bridge between tasks, providing a transformation of search spaces (and solutions) such that adaptive knowledge transfers can be conducted.

Further, in [38], a credit assignment-based technique was employed to adapt knowledge transfers between tasks in a manner that promotes exploration, thereby preventing solutions from obtaining trapped in local optima. More recently, Lin *et al.* [21] have utilized incremental Naive Bayes classifiers to select valuable solutions to be transferred during the multitasking search. In their proposed method, the authors also use a randomized mapping that enhances the exploration capacity of transferred solutions, with the aim of improving overall convergence behavior.

While there has been growing interest in designing adaptive multitask solvers, many existing methods are designed based on prior assumptions of task relatedness, such that solutions from a source may be directly transferable to a target task. Some other methods have also been proposed that learn a mapping between tasks with the goal of inducing a high (ordinal) correlation between their respective objective functions [27]. The learned mappings are then applied to transferred solutions, increasing the likelihood of them being beneficial to the target task. Nevertheless, it is noted that existing methods typically do not provide any explicit measures to guard against the threat of negative transfer; for example, when the learned mapping is inaccurate. In this regard, the key novelty of our proposed algorithm is that it takes a data-driven approach to explicitly guard against negative transfers. Our method is based on the simple idea that by learning the extent to which search distributions of different tasks overlap, the amount of knowledge to be transferred between them can be adapted.

D. Overview of the Existing MO-MFEA

The MO-MFEA [14] is a recently proposed method that allows efficient resolution of multiple MOOPs simultaneously by promoting *omnidirectional* knowledge transfers [10]. By this we mean that all tasks can benefit from each other via mutual knowledge sharing. For this to be possible, a *unified search space* encoding solutions to all constituent optimization tasks is first defined. At least in the case of box-constrained continuous optimization, such unification is achieved in a straightforward manner by a simple linear map between the task-specific solution space and the unified space [39]—which is also continuous and is defined in the range $[0, 1]^D$.

In MO-MFEA, the primary mode of knowledge transmission is through implicit genetic transfers during *intertask crossovers* between parent solutions belonging to different tasks. Essentially, the extent of genetic transfer is mandated by a user-defined and fixed scalar transfer parameter defined as the random mating probability (*rmp*). For further details about MO-MFEA, the reader is referred to [14].

1) Tuning the rmp Parameter: It is noteworthy that the *rmp* parameter in MO-MFEA is *manually* specified based on the intuition of a decision maker. That is, if the optimization tasks are known to have a high degree of intertask similarity, the *rmp* is usually set to a value close to 1. On the other hand, a low *rmp* value (closer to 0) is prescribed between optimization tasks that are known to bear low intertask similarity (LS). It is indeed patently clear that such an *offline rmp assignment* scheme is heavily dependent on the existence of prior knowledge in the mind of the practitioner about the relationships between different optimization problems. In practical settings, access to domain-specific knowledge about different optimization problems is often limited, particularly in general black-box optimization. Given the possible lack of prior knowledge about intertask relationships, the appropriate selection of *rmp* can become a significant hurdle for a practitioner. In fact, it has also been shown that inappropriate (blind) prescription of *rmp* values (via trial and error) risks the possibility of harmful genetic transfers, thereby leading to significant performance slowdowns [23].

III. Cognizant Multitasking MO-MFEA-II

This section provides details of the proposed MO-MFEA-II, which offers a data driven and cognitive enhancement of its predecessor MO-MFEA. Specifically, we would want the algorithm to be able to automatically decide *when* and *how much* knowledge to transfer such that the threat of negative transfers is minimized. To this end, the MO-MFEA-II learns the *rmp* parameter *online* (in place of an *offline rmp* assignment) and thus adapts the extent of genetic transfers between diverse tasks in a multitask setting. In order to develop an online *rmp* estimation technique with knowledge transfer adaptation capabilities, two key ideas have been incorporated in the existing MO-MFEA. These include as follows.

1) Utilizing an RMP matrix in place of a scalar *rmp* to effectively multitask across more than two tasks with possibly diverse intertask relationships. For example, given three different optimization tasks, the degree of similarity between the first and the second task may be different from that between the first and the third. Therefore, using the same *rmp* value for all task pairs is considered to be too restrictive.
2) Defining the MO-MFEA framework from a *probabilistic modeling perspective*, so as to allow capturing of intertask similarities based on the quantification of overlaps in search distributions. MO-MFEA-II is equipped to model the data generated online during the course of a multitasking search to unveil recurring solution patterns across different tasks.

A. RMP Matrix

Given K optimization tasks in a multitask setting, the RMP takes the form of a symmetric $K \times K$ *matrix* of pairwise *rmp* values represented as

$$\text{RMP} = \begin{bmatrix} \text{rmp}_{1,1} & \text{rmp}_{1,2} & \cdot & \cdot \\ \text{rmp}_{2,1} & \text{rmp}_{2,2} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \tag{2}$$

where $\text{rmp}_{k,j} = \text{rmp}_{j,k}$ reflects the extent of transfer between the kth and the jth task pair. Note that $\text{rmp}_{k,k} = 1 \ \forall k$. Unlike the scalar *rmp* of the present day MO-MFEA, the RMP matrix offers the distinct advantage of adapting the extent of knowledge transmissions between diverse task pairs with possibly nonuniform intertask similarities.

B. Latent Search Distributions in the MO-MFEA

Here, we briefly describe how knowledge transfers are actualized in the original MO-MFEA, through the *implicit mixture of search distributions drawn from different tasks*.

For $k \in \{1, 2, \ldots, K\}$ consider the parent subpopulation associated with the kth task T_k to be denoted as $P^k(t)'$ at iteration t. Further, let the subpopulations associated with each of the K tasks originate from the *true* (but unknown) underlying probability distributions $p^1(x, t), p^2(x, t), \ldots, p^K(x, t)$, respectively, that is, $P^k(t)' \sim p^k(x, t)$ for all $k \in \{1, 2, \ldots, K\}$.

As a consequence of the *intertask* interactions in the MO-MFEA (through intertask crossovers), the offspring population generated for the kth task at the tth iteration can be shown to be drawn from the following mixture distribution, under the simplifying assumption of *parent-centric* evolutionary operations (see supplementary material of [23])

$$p_c^k(x, t) = \left[1 - \frac{0.5}{K} \cdot \sum_{k \neq j} \text{rmp}_{k,j}\right] \cdot p^k(x, t) + \frac{0.5}{K} \sum_{j \neq k} \text{rmp}_{k,j} \cdot p^j(x, t). \quad (3)$$

The finite mixture $p_c^k(x, t)$ is a linear combination of all K task distributions, with the degree of mixing given by $\text{rmp}_{k,j}$'s. Note that $0 \leq \text{rmp}_{k,j} \leq 1$. Clearly, the most trivial way to prevent any harmful (negative) intertask interactions is to simply set the $\text{rmp}_{k,j}$ values to 0, leading to the cancellation of all knowledge transfers. In this case, (3) implies that $p_c^k(x, t) = p^k(x, t)$.

C. Preliminaries on Probabilistic Model-Based Search

For an ideal multitasking engine, the goal is to promote fruitful transfers whenever possible (i.e., by learning positive pairwise *rmp* values) while also minimizing negative intertask interactions. The proposed idea is thus to leverage on the data generated during the course of a multitask optimization run, and to learn an RMP matrix that leads to an effective mixture distribution in (3). For instance, if the search distributions of the respective tasks are far from each other in a unified search space, such that no relationship can be inferred between them, we would want the corresponding pairwise *rmp* values to be automatically very low (near zero). On the other hand, if the search distributions are overlapping, higher *rmp* values should ideally be prescribed. Such a learning strategy is expected to benefit related tasks while not hurting the performance of unrelated tasks.

In order to do this learning, we draw from existing theories in the field of *probabilistic model-based search algorithms*. These algorithms guide the optimization search by iteratively building probabilistic models of parent solutions, and subsequently generating new offspring by sampling these models. The typical steps can be summarized as follows.

Step 1: Start with a uniform distribution model $q(x, 0)$ over the search space.

Step 2: At iteration t, sample N offspring solutions from the current model $q(x, t-1)$.

Step 3: Based on the fitness of sampled solutions, *select* $n < N$ parent solutions.

Step 4: Build a probabilistic model $q(x, t)$ of the parent solutions.

Step 5: Increment iteration count and go to step 2, unless a termination condition is satisfied.

A well-known result that supports this class of algorithms is summarized in Theorem 1. Briefly, the result highlights the benefit of learning probabilistic models $q(x, t)$ that capture the true underlying parent distribution $p(x, t)$ accurately, in order to attain asymptotic convergence guarantees. It is this theorem that also serves as the basis of the mathematical program proposed in this article to learn the RMP matrix online, as shall be detailed in Section III-E.

Theorem 1: Consider a continuous optimization task T with objective function f and global optimum f^*. The prior distribution $q(x, 0)$ from which initial solutions are sampled is assumed to be positive and continuous everywhere in the search space $\boldsymbol{X}$, with $N \to \infty$. With this, at every iteration "t" of a probabilistic model-based EA, if the learned model $q(x, t)$—from which offspring are sampled—is identical to the true parent distribution $p(x, t)$, then the search asymptotically converges to a distribution $p(x, \infty)$ such that $\int_{\boldsymbol{X}} f(x) \cdot p(x, \infty) \cdot dx = f^*$.

The proof of the above can be found in [40]. We note that this result originally pertains to the case of single-objective optimization, and has recently been used in [23] for learning optimized mixture models in the context of multitasking with single-objective optimization tasks. In contrast, in this article, we investigate how the incorporation of the same result can lead to performance gains in MO-MFEA as well. Details of the resultant MO-MFEA-II are presented next.

D. Probabilistic Mixture Modeling in MO-MFEA-II

At any iteration t of MO-MFEA-II, we start by building probabilistic models $q^k(x, t)$ for parent subpopulations $P^k(t)' \in \{P^1(t)', P^2(t)', \ldots, P^K(t)'\}$. The model $q^k(x, t)$ is seen as a limited data approximation of the *true* parent distribution $p^k(x, t)$. Given the learned models above, (3) can equivalently be written as

$$q_c^k(x, t) = \left[1 - \frac{0.5}{K} \cdot \sum_{k \neq j} \text{rmp}_{k,j}\right] \cdot q^k(x, t) + \frac{0.5}{K} \sum_{j \neq k} \text{rmp}_{k,j} \cdot q^j(x, t) \quad (4)$$

where $q_c^k(x, t)$ is a mixture model approximating the *true* $p_c^k(x, t)$ in (3).

From Theorem 1, we know that in this probabilistic interpretation, asymptotic convergence is guaranteed in the limit $N \to \infty$ if the offspring distribution is equal to the parent distribution. This result suggests that learning the RMP matrix such that mixture model $q_c^k(x, t)$ in (4) accurately replicates $p^k(x, t)$, for all $k \in \{1, 2, \ldots, K\}$, shall implicitly decrease the tendency of negative intertask interactions.

E. Learning the RMP Matrix Online in MO-MFEA-II

To this end, given the probabilistic models $q^k(x,t)$ for all tasks $k \in \{1,2,\ldots,K\}$, the following mathematical program is formulated to learn the RMP matrix:

$$\max_{\text{RMP}} \sum_{k=1}^{K} \sum_{i=1}^{n} \log q_c^k(x_{ik},t) \tag{5}$$

where x_{ik} is the ith sample (individual) in parent subpopulation $P^k(t)'$. The rationale behind this formulation can be seen from the following theorem.

Theorem 2: Maximizing the likelihood function in (5), in the limit $n \to \infty$, is equivalent to minimizing the Kullback–Leibler divergence $\text{KL}(p^k(x,t)||q_c^k(x,t))$ averaged across all tasks $\{T_1, T_2, \ldots, T_k, \ldots, T_K\}$.

The proof of the above can be found in [23], where the Kullback–Leibler divergence is used as a standard measure of the discrepancy between probability distributions. With regard to the computational tractability of optimizing the RMP matrix, we note that the mathematical program in (5) is convex upward. Therefore, in practice, the RMP matrix can be learned at little computational overhead (as long as K is not large) using classical convex optimization solvers.

It is also worth highlighting the importance of the type of constituent probabilistic models $q^k(x,t)$ to be chosen for approximating the parent distributions $p^k(x,t)$. If we choose to build a complex (highly expressive) model on a typically small subpopulation dataset $P^k(t)'$, then this may cause model *overfitting*—which in turn can be shown to result in the cancellation of intertask knowledge transfers in (4) as the trivial solution of $rmp_{k,j} = 0 \; \forall k \neq j$ optimizes the program in (5). To alleviate this overfitting issue, we propose to use simple (less expressive) and fast probabilistic models, for example, univariate marginal distributions ignoring variable dependencies, to approximate the parent distributions; thus, allowing all other models in the multitask setting to be mobilized to fill in the distribution gaps and actualize knowledge transfers. An additional step that actively prevents the cancellation of intertask transfers is to inject some random noise into the available subpopulation datasets. Specifically, a simple modification that is found to work well in practice is to add a small amount of uniformly sampled noise to the dataset $P^k(t)' \; \forall k$, resulting in a corrupted version $P^k(t)''$. Thereafter, building the model $q^k(x,t)$ on the corrupted dataset $P^k(t)''$ naturally prevents overfitting to the original dataset $P^k(t)'$ [41].

F. Summary of MO-MFEA-II

Given all the enhancements above, we can now summarize the MO-MFEA-II algorithm.

1) Preliminary Definitions: Consider the case of solving K multiobjective optimization tasks $\{T_1, T_2, \ldots, T_K\}$ concurrently using a single population P of evolving individuals, encoded in a unified space (denoted as $\boldsymbol{Y}$). Let the subpopulation associated with the kth task be denoted as P^k. An approach for comparing the fitness of candidate solutions in such a multitask setting is formulated by defining the following properties for each individual in P.

Algorithm 1: Pseudocode of MO-MFEA-II

```
1  Randomly sample N · K individuals in Y to form initial population P(0)
2  for every individual p_i in P(0) do
3      Assign skill factor τ_i =  mod (i, K) + 1, for the case of K tasks
4      Evaluate p_i for task τ_i
5  Compute scalar fitness φ_i ∀ p_i via lexicographic ranking of NF_i, CD_i
6  Set t = 1
7  while stopping conditions are not satisfied do
8      Configure offspring population P_c(t) = ∅
9      P(t)' = Tournament Selection of n · K parent solutions from P(t)
10     Learn RMP(t) as per Eq. (5)
11     while offspring generated for each task < N do
12         Sample two individuals x_i and x_j at random from P(t)'
13         if τ_i == τ_j then
14             [x_a, x_b] ← Intra-task crossover + mutate (x_i, x_j)
15             Assign offspring x_a and x_b skill factor τ_i
16         else if rand ≤ rmp_{τ_i,τ_j} then
17             [x_a, x_b] ← Inter-task crossover + mutate (x_i, x_j)
18             Each offspring is randomly assigned skill factor τ_i or τ_j
19         else
20             Randomly select x'_i with skill factor τ_i
21             [x_a] ← Intra-task crossover + mutate (x_i, x'_i)
22             Assign offspring x_a skill factor τ_i
23             Randomly select x'_j with skill factor τ_j ;
24             [x_b] ← Intra-task crossover + mutate (x_j, x'_j)
25             Assign offspring x_b skill factor τ_j
26         Evaluate [x_a, x_b] for their assigned skill factors only
27         P_c(t) = P_c(t) ∪ [x_a, x_b]
28     R(t) = P_c(t) ∪ P(t)
29     Update scalar fitness of all individuals in R(t);
30     Select N · K fittest members from R(t) to form P(t + 1)
31     t = t + 1
```

1) *Definition 1 (Skill Factor):* The *skill factor* τ_i corresponds to the *one* task (out of the available K tasks) the ith individual is associated with in the multitasking environment.
2) *Definition 2 (Scalar Fitness):* The *scalar fitness* of the ith individual is defined as $\varphi_i = 1/r^i_{\tau_i}$, where $r^i_{\tau_i}$ corresponds to the rank of the ith individual on task τ_i. Note that $\max\{\varphi_i\} = 1$.

2) Algorithm Summary: Algorithm 1 provides the pseudocode for MO-MFEA-II. The algorithm is built upon the NSGA-II [30], which utilizes lexicographic ranking based on each candidate solution's *nondominated front (NF)* and *crowding distance (CD)* to guide the population as a whole through complex multiobjective search spaces. For the sake of brevity, further details of the NSGA-II have not been reproduced in this article.

The MO-MFEA-II evolves a single population of $N \cdot K$ individuals encoded in a unified space $\boldsymbol{Y}$. A uniform resource allocation scheme is adopted whereby each task $T_k \in \{T_1, T_2, \ldots, T_K\}$ is equally assigned N individuals to comprise its subpopulation P^k. Lines 7–31 in Algorithm 1 form the main evolutionary multitasking loop, describing the steps associated with offspring generation, task-specific fitness evaluation, and environmental selection.

For offspring creation, a pool of parent candidates $P(t)'$ is first selected through binary tournament selection (line 9). The selection is based on the scalar fitness of each of the individuals such that $P^1(t)' \cup P^2(t)' \cup, \ldots, \cup P^K(t)' = P(t)'$. The selected parent subpopulations are then used to build probabilistic models and learn the RMP matrix according to (5).

Notice that the RMP matrix is currently learned and adapted online at every iteration t (line 10).

Apart from the conventional *intratask crossover* between parent solutions belonging to the same task (lines 13–15), *intertask crossover* (lines 16–18) also takes place between parent solutions belonging to different tasks (i.e., parents having different skill factors). In particular, the *extent* of implicit genetic transfer in MO-MFEA-II is governed by the pairwise *rmp* values in the learned RMP matrix–which mandates the probability of intertask crossover between two solutions x_i and x_j that may have different skill factors. Clearly, if the learned RMP is the identity matrix, then no knowledge transfer will be actualized.

3) Selective Evaluations in MO-MFEA-II: Each of the offspring individuals is evaluated with respect to only one task (its assigned skill factor) amongst all other tasks in the multitasking environment (line 26). This salient feature combats the issue of otherwise exhaustively evaluating every individual for every task, which would often be computationally demanding. Post evaluation, the fittest $N \cdot K$ individuals survive for the next iteration of the evolutionary search (line 30).

The aforementioned steps (lines 7–31 in Algorithm 1) repeat until some stopping condition is met.

IV. Experimental Study

In this section, experimental results are presented that showcase the efficacy of the proposed MO-MFEA-II in the domain of multiobjective optimization. The performance of MO-MFEA-II is compared against its predecessor, the MO-MFEA, and its baseline algorithm, that is, NSGA-II performing a single task at a time. For the case of synthetic benchmarks tackling two MOOPs at a time (Sections IV-D1 and IV-D2), the evolutionary multitasking via explicit autoencoding (EMEA) [8] algorithm is considered as an additional state-of-the-art multitasking baseline for comparison. Essentially, the EMEA leverages on the search biases from two different MOEAs (i.e., NSGA-II and SPEA2) acting upon two distinct tasks, respectively. During knowledge transfers, candidate solutions are first transformed via a linear mapping before being transferred from one task to the other.

A. Experimental Configuration

The experimental setup is outlined in what follows. To ensure consistency, the population size N (per task) is kept the same for all algorithms. For all the test problems under consideration, the following general settings are applied.

1) Continuous unified search space $[0, 1]^D$.
2) Evolutionary operators used for MO-MFEA-II, MO-MFEA, and NSGA-II are:
   a) SBX crossover with probability $(p_c) = 1$ and distribution index $\eta_c = 10$;
   b) polynomial mutation with probability $(p_m) = 1/D$ and distribution index $\eta_m = 10$.
3) *Probability Model in MO-MFEA-II:* Normal distribution.
4) Population size (N):
   a) for synthetic benchmarks: 50;
   b) for practical study: 20.
5) Maximum number of generations: 250.
6) RMP is learned online for MO-MFEA-II. For MO-MFEA, *rmp* values set as either 0.9, 0.6, or 0.3.
7) Transfer interval in the EMEA: ten generations.

Note that no uniform crossover like variable swap is applied between offspring to ensure parent-centric crossover operations with minimum chromosome disruptions [42], thus allowing us to clearly observe the effects of intertask implicit genetic transfers.

B. Specification of Benchmark MOOPs

We highlight that the following benchmarks are created based on prior knowledge of the relationships between tasks in multitask settings. The relationship is measured in terms of the overlap between their optimal solutions and the rank (ordinal) correlations between their function landscapes [26]. Given such prior knowledge, evolutionary multitasking is naturally expected to benefit from the scope for genetic transfers between tasks that share a degree of intertask similarity. On the other hand, it may be less effective to conduct knowledge exchange across tasks that have little in common, as this could potentially lead to negative transfers and impede the optimization process. In the latter scenario, an ideal multitasking algorithm should be able to automatically reduce intertask interactions, thereby performing no worse than its baseline single-tasking algorithm.

In this particular study, we consider a variety of MOOP benchmarks proposed for evolutionary multitask optimization [14], [26]. These problems are characterized by different properties, including the type of the Pareto front, multimodality, and separability.

1) Problem 1: In this multiobjective problem, $g(\boldsymbol{x})$ takes the form of the sphere function

$$\begin{aligned} &\min\ f_1(\boldsymbol{x}) = g(\boldsymbol{x}) \cos\left(\frac{\pi x_1}{2}\right) \\ &\min\ f_2(\boldsymbol{x}) = g(\boldsymbol{x}) \sin\left(\frac{\pi x_1}{2}\right) \\ &g(\boldsymbol{x}) = 1 + \sum_{i=2}^{D} {x_i}^2 \\ &x_1 \in [0, 1], \quad x_i \in [-100, 100], \quad i = 2, 3, \ldots, D. \end{aligned} \tag{6}$$

2) Problem 2: This particular MOOP is defined as follows:

$$\begin{aligned} &\min\ f_1(\boldsymbol{x}) = x_1 \\ &\min\ f_2(\boldsymbol{x}) = g(\boldsymbol{x}) \left(1 - \left(\frac{x_1}{g(\boldsymbol{x})}\right)^2\right) \\ &g(\boldsymbol{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{D} |x_i| \\ &x_1 \in [0, 1], \quad x_i \in [-100, 100], \quad i = 2, 3, \ldots, D. \end{aligned} \tag{7}$$

3) Problem 3: Here, $g(\boldsymbol{x})$ takes the form of the Griewank function [43]

$$\begin{aligned} &\min\ f_1(\boldsymbol{x}) = g(\boldsymbol{x}) \cos\left(\frac{\pi x_1}{2}\right) \cos\left(\frac{\pi x_2}{2}\right) \\ &\min\ f_2(\boldsymbol{x}) = g(\boldsymbol{x}) \cos\left(\frac{\pi x_1}{2}\right) \sin\left(\frac{\pi x_2}{2}\right) \\ &\min\ f_3(\boldsymbol{x}) = g(\boldsymbol{x}) \sin\left(\frac{\pi x_1}{2}\right) \end{aligned}$$

TABLE I
SUMMARY OF TEST PROBLEMS

	Pareto Front	Properties
Problem 1	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	concave,unimodal, separable
Problem 2	$f_2 = 1 - f_1^2$ $0 \leq f_1 \leq 1$	concave,unimodal, separable
Problem 3	$\sum_{i=1}^{3} f_i^2 = 1$ $f_i \geq 0, i = 1, 2, 3.$	concave,multimodal, nonseparable
Problem 4	$f_2 = 1 - f_1^2$ $0 \leq f_1 \leq 1$	concave,multimodal, nonseparable
Problem 5	$f_2 = 1 - \sqrt{f_1}$ $0 \leq f_1 \leq 1$	convex,multimodal, nonseparable

$$
\begin{aligned}
& g(\boldsymbol{x}) = 2 + \frac{1}{4000}\sum_{i=3}^{D}(z_i)^2 - \prod_{i=3}^{D}\cos\left(\frac{z_i}{\sqrt{i-2}}\right) \\
& (z_3, z_4, \ldots, z_D)^T = (x_3, x_4, \ldots, x_D)^T - \boldsymbol{s} \\
& x_1 \in [0, 1], \quad x_2 \in [0, 1], \quad x_i \in [-50, 50] \\
& i = 3, 4, \ldots, D
\end{aligned} \tag{8}
$$

where $\boldsymbol{s} = (20, 20, \ldots, 20)$ represents a shift vector.

4) Problem 4: Here, $g(\boldsymbol{x})$ takes the form of the Ackley function [44]

$$
\begin{aligned}
& \min \ f_1(\boldsymbol{x}) \frac{1}{2}(x_1 + x_2) \\
& \min \ f_2(\boldsymbol{x}) = g(\boldsymbol{x})\left(1 - \left(\frac{x_1 + x_2}{2 \cdot g(\boldsymbol{x})}\right)^2\right) \\
& g(\boldsymbol{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D-2}\sum_{i=3}^{D}x_i^2}\right) \\
& -\exp\left(\frac{1}{D-2}\sum_{i=3}^{D}\cos(2\pi x_i)\right) + 21 + e \\
& x_1 \in [0, 1], \quad x_2 \in [0, 1], \quad x_i \in [-100, 100] \\
& i = 3, 4, \ldots, D.
\end{aligned} \tag{9}
$$

5) Problem 5: In this case, $g(\boldsymbol{x})$ takes the form of the Rastrigin function [45]

$$
\begin{aligned}
& \min \ f_1(\boldsymbol{x}) = x_1 \\
& \min \ f_2(\boldsymbol{x}) = g(\boldsymbol{x})\left(1 - \sqrt{\frac{x_1}{g(\boldsymbol{x})}}\right) \\
& g(\boldsymbol{x}) = 1 + 10(D-1) + \sum_{i=1}^{D-1} z_i^2 - 10\cos(4\pi z_i) \\
& (z_2, z_3, \ldots, z_D)^T = M \cdot (x_2, x_3, \ldots, x_D)^T \\
& x_1 \in [0, 1], \quad x_i \in [-5, 5], \quad i = 2, 3, \ldots, D
\end{aligned} \tag{10}
$$

where M is a randomly generated $(D-1) \times (D-1)$ rotation matrix. The properties of these MOOPs are summarized in Table I. In all cases, the dimensionality of the search space is set to $D = 10$. The inverted generational distance (IGD) measure [46] is used for performance comparison between all the algorithms under consideration.

C. Measuring Complementarity Between Constitutive Tasks

In [26], two quantities have been proposed to measure the complementarity between different multiobjective optimization tasks in a multitask setting. First, the degree of overlap of the global optima of the $g(\boldsymbol{x})$ functions of constitutive tasks is considered. A scenario in which there is complete overlap between the respective global optima in the unified space is defined as *complete intersection*, and is denoted as CI. On the other hand, the global optima of task pairs may also *not intersect*, which we denote as NI. Second, Spearman's rank correlation (R_s) measure is utilized to capture the similarity between the overall trends of the fitness landscape of tasks. The task pairs for which the Spearman rank correlation $R_s < 0.2$ are characterized as having LS; the task pairs in the range $0.2 \leq R_s < 0.8$ are defined as having medium similarity (MS); and $R_s \geq 0.8$ implies high intertask similarity (HS).

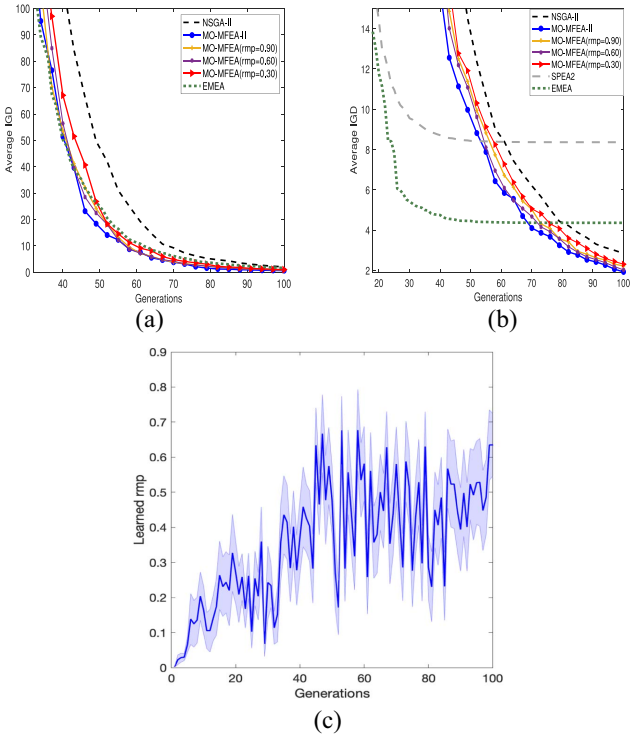

Fig. 1. Top: Convergence trends (averaged over 30 independent runs) achieved by all the algorithms in the CI-HS composite MOOP benchmark. Below: Pairwise *rmp's* learned between tasks T_1 and T_2 over successive generations. The shaded regions span one half standard deviation on either side of the mean. (a) Convergence trends of T_1 in CI-HS. (b) Convergence trends of T_2 in CI-HS. (c) Learned intertask *rmp's of MO-MFEA-II*.

D. Experimental Results

1) Complete Intersection and High Similarity (CI-HS): We begin by considering a composite MOOP set with complete intersection and high similarity. In this setup, task T_1 is represented by Problem 1 (6) while task T_2 is represented by Problem 2 (7). The intertask similarity is measured to be $R_s = 0.97$. The convergence trends (IGD metric averaged over 30 independent runs) for the first 100 generations are shown in Fig. 1.

Since the composite MOOPs share high degrees of intertask similarities, the effect of knowledge transfers in all the multitasking algorithms is observed to be beneficial. The results indicate that both MO-MFEA-II and MO-MFEA outperform

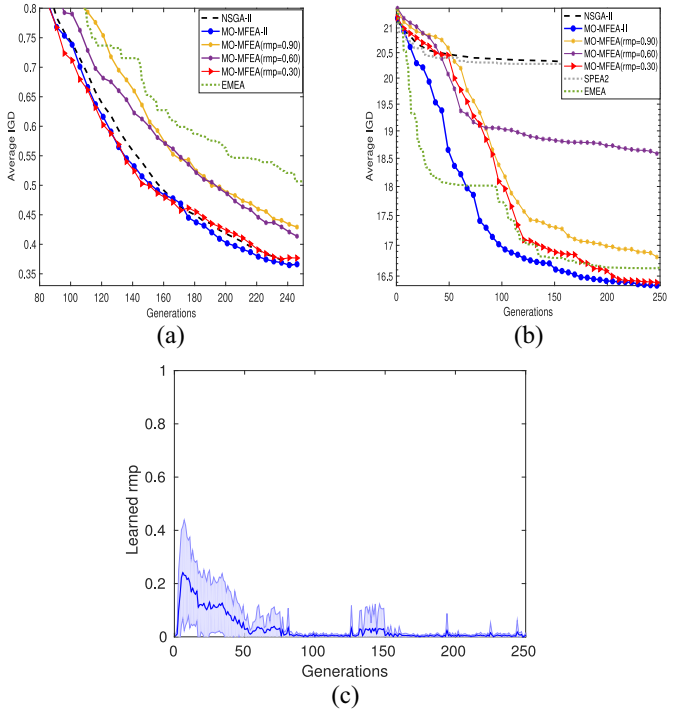

Fig. 2. Top: Convergence trends of all the algorithms in the NI-LS composite MOOP benchmark. Bottom: Pairwise learned *rmp's*. (a) Convergence trends of T_1 in NI-LS. (b) Convergence trends of T_2 in NI-LS. (c) Learned intertask *rmp's of MO-MFEA-II*.

the single-tasking NSGA-II. Fig. 1(a) and (b) shows that MO-MFEA variants with high specified *rmp* values (i.e., 0.9 and 0.6, inducing high frequency of genetic transfers) showcase better convergence trends than MO-MFEA with a lower prescribed *rmp* value of 0.3. As such, note that the MO-MFEA requires careful *rmp* tuning in order to achieve the desired performance. In contrast, the MO-MFEA-II alleviates the hurdle of manual parameter tuning. By learning the pairwise *rmp* values online and adapting knowledge transfers on the fly, MO-MFEA-II has recorded improved convergence characteristics for both tasks T_1 and T_2. Fig. 1(c) reveals that the learning module of MO-MFEA-II is able to deduce *rmp* values that agree well with the *a priori* known intertask similarity; high *rmp* values are learned throughout the optimization process implying high frequency of transfers.

We further compare MO-MFEA-II with the results achieved by the EMEA algorithm for the CI-HS case. Recall that the EMEA employs two different solvers to tackle the two different tasks; T_1 is optimized by NSGA-II while T_2 is optimized by SPEA2. As observed from Fig. 1, the EMEA has shown similar performance behavior as the MO-MFEA-II and other MO-MFEA variants. It leverages on the scope of knowledge transfers to improve its baseline algorithms for both the tasks.

2) No Intersection and Low Similarity (NI-LS): Next, we consider the extreme case of tackling two unrelated MOOPs with no intersection of the global optima and having LS. In this setup, task T_1 is represented by Problem 3 (8) while task T_2 is represented by Problem 4 (9). The intertask similarity is measured to be $R_s = 0.08$. The experimental results are reported in Fig. 2.

TABLE II
INTERTASK SIMILARITIES R_s

	T_1	T_2	T_3	T_4	T_5
T_1	1	0.970	0.298	0.778	0.402
T_2	-	1	0.299	0.753	0.391
T_3	-	-	1	0.08	0.130
T_4	-	-	-	1	0.817
T_5	-	-	-	-	1

The lack of intertask similarity in this scenario suggests that there exists little scope for crossover-based intertask knowledge transfers. However, consider the convergence trends of the MO-MFEA variants with higher frequency of transfers (i.e., rmp $= 0.6$ and rmp $= 0.9$). While these configurations seem to have improved the performance of task T_2 in comparison to NSGA-II [Fig. 2(b)], we observe that their performance deteriorates significantly on task T_1 [Fig. 2(a)]. This may be a counter effect of the *omnidirectional* transfer mechanism, which, if not controlled, could lead to undesired negative transfers for one of the tasks. This conjecture is also supported by the results of EMEA on T_1, which is found to be hampered by negative transfers as well. As such, the MO-MFEA variants are seen to improve one task at the expense of worsening the other, but, with a decreased rmp value (0.3), such counter effects are minimized. On the other hand, MO-MFEA-II has recorded superior convergence trends for both tasks. As shown by the learned *rmp* curve in Fig. 2(c), the frequency of transfer is low, thus making the algorithm less prone to the impact of negative transfers.

3) Study With More Than Two Tasks: Studies focusing on many tasking have been carried out in the domain single-objective optimization [13], [47]. Along similar lines, we extend our current experimental study to more than two multiobjective tasks. In particular, we attempt to analyze the effects of omnidirectional transfers in the context of pooling several tasks (up to five MOOPs) in the same multitasking environment. For this set of experiments, all the five problems defined in Section IV-B are considered. Tasks $T_1, T_2, \ldots, T_5$ are represented by *problems:* $1, 2, \ldots, 5$, respectively. The intertask similarities are given in Table II.

The experimental results are reported in Table III and Fig. 3. According to the results, MO-MFEA-II records the overall best performance. More important, no harmful effects of negative transfer are seen for MO-MFEA-II in any of the five tasks. Clearly, such superior performance is attributed to the fact that MO-MFEA-II incorporates an RMP matrix, capable of capturing diverse intertask relationships between various task pairs and adapting the extent of transfers accordingly.

In contrast, the MO-MFEA variants that employ a user-specified scalar *rmp* are found to suffer. Since the *rmp* is fixed, the extent of transfer remains the same (uniform) across all task pairs (irrespective of the existence of nonuniform intertask synergies reported in Table II). In such a restrictive approach, the threats of negative transfers are observed to be predominant. For instance, while analyzing the convergence trends in Fig. 3, it can be observed that prescribing a relatively high *rmp* (e.g., 0.6) may bolster performance in a few tasks while hampering others (in most cases, the conventional MO-MFEA performs worse than single-tasking NSGA-II).

TABLE III
AVERAGE IGD VALUES ACHIEVED BY NSGA-II, MO-MFEA-II, AND MO-MFEA (WITH A RANGE OF *rmp* VALUES) FOR ALL $K = 5$ TASKS. BOLD FACE ENTRIES ARE THE BEST (BY WILCOXON TEST WITH SIGNIFICANCE LEVEL 0.05)

		Tasks				
		T_1	T_2	T_3	T_4	T_5
	NSGA-II	$7.75e-02$ $(3.31e-02)$	$3.46e-01$ $(1.45e-01)$	$1.90e+01$ $(4.88e+00)$	$3.97e-01$ $(5.65e-02)$	$1.70e+00$ $(8.21e-01)$
	MO-MFEA-II	$\mathbf{1.98e-02}$ $(5.70e-03)$	$\mathbf{2.27e-01}$ $(6.94e-02)$	$\mathbf{1.67e+01}$ $(7.40e+00)$	$\mathbf{3.74e-01}$ $(5.14e-02)$	$\mathbf{1.58e+00}$ $(7.88e-01)$
MO-MFEA	$(rmp = 0.90)$	$1.24e-01$ $(5.92e-02)$	$8.11e-01$ $(2.47e-01)$	$2.04e+01$ $(5.14e-02)$	$8.86e-01$ $(1.98e-01)$	$1.27e+01$ $(3.24e+00)$
	$(rmp = 0.60)$	$5.24e-02$ $(2.31e-02)$	$5.07e-01$ $(1.30e-01)$	$1.90e+01$ $(7.40e+00)$	$5.60e-01$ $(5.68e-02)$	$4.75e+00$ $(1.93e+00)$
	$(rmp = 0.30)$	$3.57e-02$ $(1.39e-02)$	$3.91e-01$ $(1.06e-01)$	$1.88e+01$ $(4.77e+00)$	$4.38e-01$ $(5.44e-02)$	$2.63e+00$ $(1.04e+00)$

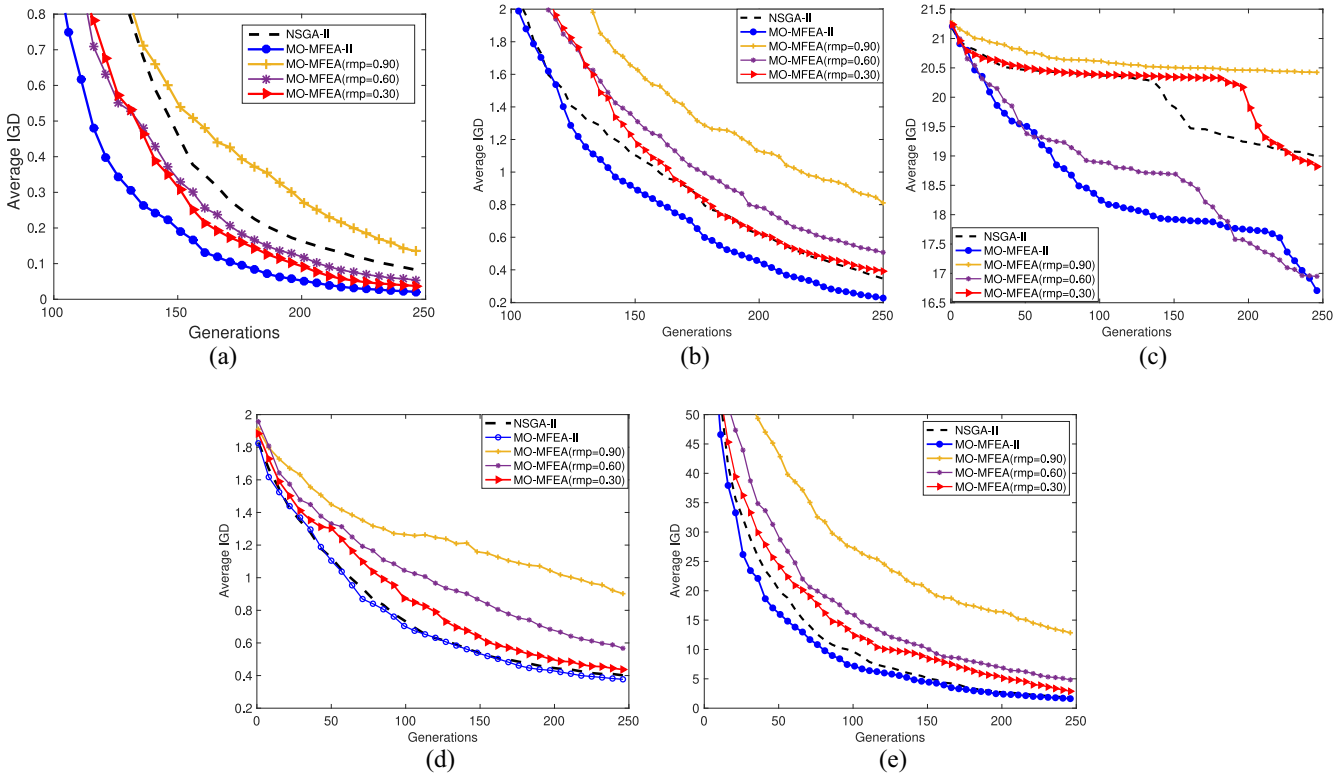

Fig. 3. Convergence trends (averaged over 30 independent runs) achieved by all the algorithms on five tasks. (a) Convergence trends of T_1. (b) Convergence trends of T_2. (c) Convergence trends of T_3. (d) Convergence trends of T_4. (e) Convergence trends of T_5.

Furthermore, Tables IV and V show numerical results for pooling $K = 4$ and $K = 3$ tasks (MOOPs) together, respectively. Performance trends similar to Table III are observed.

V. EVOLUTIONARY MULTITASKING FOR HIGH-FIDELITY OPTIMIZATION: PRACTICAL STUDY

One of the key beneficiaries of our proposed MO-MFEA-II framework is expected to be in the domain of high-fidelity optimization settings. The fact that real-world high-fidelity numerical simulations can take anywhere between several minutes to several hours for a single run, the use of traditional single-task optimization approaches still remains too costly. On the other hand, knowledge exploitation from cheaper optimization problems to cut down the computational costs of (related) expensive problems has shown to be promising [6], [48]. In this section, we demonstrate the efficacy of MO-MFEA-II in leveraging knowledge transfers from low-fidelity tasks to speed up the performance of higher fidelity tasks.

In particular, we consider multiobjective optimization of the unmanned aerial vehicle (UAV) path planning problem. Before we proceed to the details of our experimental settings, we present a brief overview of the UAV path planning procedure. UAVs have emerged as promising tools for autonomous inspection of large geographical areas and performing missions in hazardous environments, such as military surveillance, nuclear power plant operations, and outer space exploration [49], [50]. Autonomous navigation by UAVs through unknown environments is a challenging task that requires safe and efficient path planning strategies, such that operational risks (in the presence of hazards, including obstacles or environmental factors) are minimized. The risks associated

TABLE IV
AVERAGE IGD VALUES FOR ALL $K = 4$ TASKS

		Tasks			
		T_1	T_2	T_3	T_4
	NSGA-II	$7.75e-02$ $(3.31e-02)$	$3.46e-01$ $(1.45e-01)$	$1.90e+01$ $(4.88e+00)$	$3.97e-01$ $(5.65e-02)$
	MO-MFEA-II	$\mathbf{2.44e-02}$ $(1.12e-02)$	$\mathbf{2.83e-01}$ $(9.23e-02)$	$\mathbf{1.83e+01}$ $(6.34e+00)$	$\mathbf{3.92e-01}$ $(7.46e-02)$
MO-MFEA	$(rmp = 0.90)$	$1.21e-02$ $(1.12e-02)$	$7.46e-01$ $(1.77e-01)$	$2.04e+01$ $(1.30e-01)$	$9.00e-01$ $(8.57e-02)$
	$(rmp = 0.60)$	$4.38e-02$ $(1.42e-02)$	$4.75e-01$ $(1.03e-01)$	$1.87e+01$ $(5.24e+00)$	$5.30e-01$ $(1.27e-01)$
	$(rmp = 0.30)$	$3.43e-02$ $(8.30e-03)$	$3.42e-01$ $(7.47e-02)$	$2.03e+01$ $(2.19e-02)$	$4.44e-01$ $(6.97e-02)$

TABLE V
AVERAGE IGD VALUES FOR ALL $K = 3$ TASKS

		Tasks		
		T_1	T_2	T_3
	NSGA-II	$7.75e-02$ $(3.31e-02)$	$3.46e-01$ $(1.45e-01)$	$1.90e+01$ $(4.88e+00)$
	MO-MFEA-II	$\mathbf{2.27e-02}$ $(6.20e-03)$	$\mathbf{2.75e-01}$ $(7.35e-02)$	$\mathbf{1.31e+01}$ $(9.45e+00)$
MO-MFEA	$(rmp = 0.90)$	$4.19e-02$ $(1.27e-02)$	$4.26e-01$ $(7.50e-02)$	$1.88e+01$ $(4.87e+00)$
	$(rmp = 0.60)$	$3.85e-02$ $(1.77e-02)$	$4.02e-01$ $(1.03e-01)$	$1.86e+01$ $(5.67e+00)$
	$(rmp = 0.30)$	$2.74e-02$ $(8.80e-03)$	$3.04e-01$ $(9.67e-02)$	$2.04e+01$ $(6.56e-02)$

TABLE VI
PRACTICAL STUDY EXPERIMENTAL CONFIGURATIONS. DIFFERENT TASK INSTANCES ARE GENERATED BY DEFINING DIFFERENT DISCRETIZATION STEP SIZES (d_{step})

	Tasks	
	Main task	Helper task(s)
Configuration 1	$T_1 : d_{step} = 500m$	$T_2 : d_{step} = 1000m$
Configuration 2	$T_1 : d_{step} = 500m$	$T_2 : d_{step} = 750m$ $T_3 : d_{step} = 1000m$

with UAVs may vary from collisions to loss of control. Collisions are usually caused by static or dynamic obstacles while loss of control can result from system failure, navigating beyond signal range (e.g., radio, WiFi or GSM networks), or adverse environmental conditions (e.g., bad weather or bad GPS reception) [51]. In our experimental study, we consider a realistic scenario with multiple UAVs operating in a specified region in the southwest of Singapore. The operational environment is characterized by different types of hazards and environmental factors (e.g., signal strength and weather conditions). For our simulations, a data-driven multitask Gaussian process (GP) model of the environment is created. The probabilistic predictions from such a GP model are then used to define a *path-integral* risk metric, which gives the probability of an unsafe outcome along a UAV's traversed path. Background details of safety risk assessment in UAVs, GP modeling of the environment, and the mathematical definitions of the path-integral risk metric are not reproduced herein for the sake of brevity, but can be found in [51].

A. Multiobjective Risk-Averse UAV Path Planning

The problem of minimizing operational risk for multiple UAVs in uncertain environments can be defined in a multiobjective optimization framework, addressing the trade-off between the path-integral risk measure and path efficiency (travel distance). The considered MOOP can be represented as follows:

$$\text{minimize} \quad (\text{Risk}, \text{Distance}) \tag{11}$$

where the first objective corresponds to minimizing the probability of an unsafe event occurring for at least one UAV, while the second objective accounts for the total distance traveled by all the UAVs from their respective origins to destinations.

With regard to the risk measure, the path integral over a continuous path for a particular UAV is evaluated through a discretization of its trajectory. We note that the runtime of evaluating the risk metric along a finite discretized path is highly dependent on the discretization step size. To elaborate, a smaller step size implies numerical integrations of high-fidelity, making the optimization process computationally expensive. Such high-fidelity simulations, albeit more accurate, can take up several minutes or even hours. On the other hand, lower fidelity simulations with larger discretization step sizes can be much faster, although they do not offer the same level of accuracy in the risk estimations.

In this demonstration, we consider five UAVs operating in a region of about 10×7 km^2 in the southwest of Singapore. We introduce two environmental factors that translate into hazards: 1) signal strength (4G signal) and 2) the weather conditions (defined by different wind speeds). Further, the region includes geofences which UAVs cannot enter, and there exist a couple of other constraints such as maintaining a minimum separation distance between UAVs and flying within specified altitude boundaries. The trajectories of UAVs are defined using cubic spline interpolation [52]. For full experimental details, the reader is referred to [51].

Table VI shows two different experimental configurations used to demonstrate the efficacy of MO-MFEA-II for high-fidelity multicriteria optimization of UAV trajectories.

As shown, different instances of tasks (representing different fidelity levels) are generated using different discretization step sizes (d_{step}) associated with the path-integral risk measure. In both configurations, the high-fidelity task T_1 represents the main task of interest, while the low-fidelity tasks T_2 and T_3 are

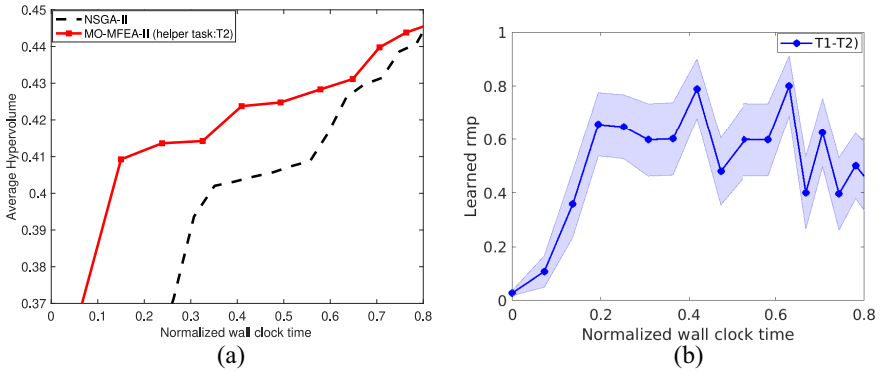

Fig. 4. Left: Convergence trends (averaged over ten independent runs) achieved by NSGA-II and MO-MFEA-II for experimental Configuration 1. Right: Pairwise *rmp's* learned between tasks T_1 and the helper task T_2 over total run time. The shaded regions span one half standard deviation on either side of the mean. (a) Convergence trends of T_1. (b) Learned intertask *rmp's of MO-MFEA-II*.

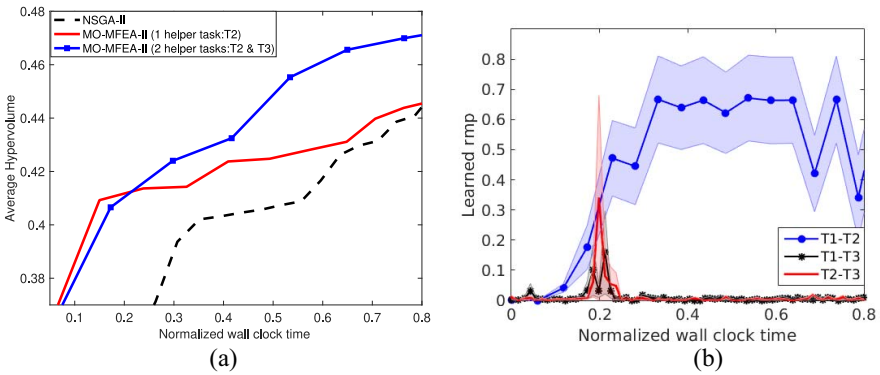

Fig. 5. Left: Convergence trends achieved by NSGA-II and MO-MFEA-II for experimental Configuration 2. MO-MFEA-II employs two helper tasks, respectively. Right: Pairwise *rmp's* learned between tasks T_1 and the helper tasks T_2 and T_3. The shaded regions span one half standard deviation on either side of the mean. (a) Convergence trends of T_1. (b) Learned intertask *rmp's of MO-MFEA-II*.

considered as helper tasks. Since all the specified tasks essentially solve the same MOOP, but with different fidelity levels, there inherently exists underlying complementarities that can be exploited via multitasking.

While the baseline NSGA-II solves the main tasks in isolation, the MO-MFEA-II pools together the main and helper task(s) for simultaneous optimization. The hypervolume metric [53] is used as an indicator to measure the quality of solutions obtained by the two different algorithms.

Fig. 4(a) shows the convergence trends of T_1 mapped against the total run time for experimental Configuration 1. As is seen, the MO-MFEA-II leads to a significant boost in the optimization search during the initial stages of the main task. Specifically, to achieve a hypervolume measure of 0.415 for the high-fidelity task T_1, MO-MFEA-II takes 33% less time compared to NSGA-II. The corresponding learned *rmp* values are depicted in Fig. 4(b). As shown, T_1 experiences a high frequency of transfers from its lower fidelity helper task T_2, thereby boosting the overall convergence trend.

Next, in experimental Configuration 2, we demonstrate the effects of introducing an additional low-fidelity helper task T_3. As shown in Fig. 5(a), the overall convergence trend of the high-fidelity task T_1 (blue curve) is further boosted relative to the former experimental setup with only one helper task [shown by the red curve in Fig. 5(a)]. The performance enhancement is attributed to the learned *rmp* curves given by Fig. 5(b). Therein, T_1 is observed to be experiencing knowledge transfer from both helper tasks T_2 and T_3, with the learned rate of transfer from T_2 being much more frequent; this agrees with our intuition from Table VI that T_1 is more similar to T_2 than it is to T_3. Notably, from Fig. 5(b), it is also seen that T_3 helps T_2 (see $T2-T3$ *rmp* curve), which in turn helps boost the optimization performance of the high-fidelity task T_1.

VI. Conclusion

In this article, we presented a cognizant multitasking multiobjective multifactorial evolution algorithm, namely, MO-MFEA-II. The salient feature of MO-MFEA-II is its *self awareness*, constantly modeling diverse relationships in the optimization environment and adapting its own mechanisms accordingly. We demonstrated the efficacy of MO-MFEA-II experimentally on various benchmark MOOPs. The experimental results show that the proposed method is capable of learning intertask relationships between different MOOPs, demonstrating the algorithm's ability to decipher *when* and *how much* knowledge to transfer across different tasks. As opposed to the restrictive scalar transfer parameter in the present day MO-MFEA, the flexibility of MO-MFEA-II to capture diverse intertask relationships is enhanced through an RMP matrix representation.

In addition to the above, a practically useful application is found to be in the domain of multifidelity optimization. The simulation results highlight the ability of MO-MFEA-II to exploit relevant knowledge from low-fidelity (cheaper) tasks to quickly optimize high-fidelity tasks, thereby leading to significant cost-saving benefits. Further, the empirical evidence in this article shows the potential benefits of incorporating multiple low-fidelity tasks for performance enhancement of the high-fidelity task.

Although MO-MFEA-II is adept at capturing intertask relationships across several tasks at once, we however note that the associated learning procedure becomes computationally demanding as the number of tasks grows increasingly large. In the literature, there have however been some recent works to efficiently handle *many-task* optimization [13], [18]. While the direct scalability of MO-MFEA-II to many tasking can be achieved through simplifications such as reduced frequency of RMP learning, we shall take up a more rigorous study of this issue in our future work.

REFERENCES

[1] L. S. Gottfredson, "Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography," *Intelligence*, vol. 24, no. 1, pp. 13–23, 1997.

[2] C. D. Wickens, "Situation awareness: Review of Mica Endsley's 1995 articles on situation awareness theory and measurement," *Human Factors*, vol. 50, no. 3, pp. 397–403, 2008.

[3] K. Tirri and P. Nokelainen, *Measuring Multiple Intelligences and Moral Sensitivities in Education*, vol. 5. New York, NY, USA: Springer, 2012.

[4] 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.

[5] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[6] K. Swersky, J. Snoek, and R. P. Adams, "Multi-task bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2004–2012.

[7] 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.

[8] L. Feng *et al.*, "Evolutionary multitasking via explicit autoencoding," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3457–3470, Sep. 2019.

[9] A. Gupta and Y.-S. Ong, "Back to the roots: Multi-x evolutionary computation," *Cognitive Computation*, vol. 11, pp. 1–17, Jan. 2019.

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

[11] X. Zheng, A. Qin, M. Gong, and D. Zhou, "Self-regulated evolutionary multitask optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 16–28, Feb. 2020.

[12] B. Zhang, A. K. Qin, and T. Sellis, "Evolutionary feature subspaces generation for ensemble classification," in *Proc. ACM Genet. Evol. Comput. Conf.*, 2018, pp. 577–584.

[13] R.-T. Liaw and C.-K. Ting, "Evolutionary manytasking optimization based on symbiosis in biocoenosis," in *Proc. 33rd AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4295–4303.

[14] 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.

[15] Y.-W. Wen and C.-K. Ting, "Learning ensemble of decision trees through multifactorial genetic programming," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2016, pp. 5293–5300.

[16] R. Chandra, A. Gupta, Y.-S. Ong, and C.-K. Goh, "Evolutionary multitask learning for modular knowledge representation in neural networks," *Neural Process. Lett.*, vol. 47, pp. 993–1009, Oct. 2017.

[17] E. O. Scott and K. A. De Jong, "Automating knowledge transfer with multi-task optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 2252–2259.

[18] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, early access, doi: 10.1109/TETCI.2019.2916051.

[19] G. Li, Q. Lin, and W. Gao, "Multifactorial optimization via explicit multipopulation evolutionary framework," *Inf. Sci.*, vol. 512, pp. 1555–1570, Feb. 2020.

[20] S. Huang, J. Zhong, and W. Yu, "Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization," *IEEE Trans. Emerg. Topics Comput.*, early access, doi: 10.1109/TETC.2019.2945775.

[21] J. Lin, H. Liu, B. Xue, M. Zhang, and F. Gu, "Multi-objective multi-tasking optimization based on incremental learning," *IEEE Trans. Evol. Comput.*, early access, doi: 10.1109/TEVC.2019.2962747.

[22] J. Mo, Z. Fan, W. Li, Y. Fang, Y. You, and X. Cai, "Multi-factorial evolutionary algorithm based on m2m decomposition," in *Proc. Asia–Pacific Conf. Simulat. Evol. Learn.*, 2017, pp. 134–144.

[23] 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.

[24] L. Bao *et al.*, "An evolutionary multitasking algorithm for cloud computing service composition," in *Proc. World Congr. Services*, 2018, pp. 130–144.

[25] M. Iqbal, W. N. Browne, and M. Zhang, "Reusing building blocks of extracted knowledge to solve complex, large-scale Boolean problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 465–480, Aug. 2014.

[26] Y. Yuan *et al.*, "Evolutionary multitasking for multiobjective continuous optimization: Benchmark problems, performance metrics and baseline results," 2017, [Online]. Available: https://arxiv.org/abs/1706.02766.

[27] 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.

[28] D. T. Stuss and D. F. Benson, "Neuropsychological studies of the frontal lobes," *Psychol. Bull.*, vol. 95, no. 1, p. 3, 1984.

[29] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. New York, NY, USA: Springer, 2007.

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

[31] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2013.

[32] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[33] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Dept. Elect. Eng., Swiss Federal Inst. Technol., Zürich, Switzerland, TIK-Rep. 103, 2001.

[34] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[35] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, Oct. 2015.

[36] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, Feb. 2016.

[37] S. Yao, Z. Dong, X. Wang, and L. Ren, "A multiobjective multifactorial optimization algorithm based on decomposition and dynamic resource allocation strategy," *Inf. Sci.*, vol. 511, pp. 18–35, Feb. 2020.

[38] H. T. T. Binh, N. Q. Tuan, and D. C. T. Long, "A multi-objective multi-factorial evolutionary algorithm with reference-point-based approach," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 2824–2831.

[39] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA J. Comput.*, vol. 6, no. 2, pp. 154–160, 1994.

[40] Q. Zhang and H. Muhlenbein, "On the convergence of a class of estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 127–136, 2004.

[41] A. Gupta and Y.-S. Ong, *Memetic Computation: The Mainspring of Knowledge Transfer in a Data-Driven Optimization Era*, vol. 21. New York, NY, USA: Springer, 2018.

[42] E. A. Williams and W. A. Crossley, "Empirically-derived population size and mutation rate guidelines for a genetic algorithm with uniform crossover," in *Soft Computing in Engineering Design and Manufacturing*. London, U.K.: Springer, 1998, pp. 163–172.
[43] M. Locatelli, "A note on the Griewank test function," *J. Glob. Optim.*, vol. 25, no. 2, pp. 169–174, 2003.
[44] D. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, vol. 28. New York, NY, USA: Springer, 2012.
[45] H. Mühlenbein, M. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," *Parallel Comput.*, vol. 17, nos. 6–7, pp. 619–632, 1991.
[46] S. Jiang, Y.-S. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2391–2404, Dec. 2014.
[47] 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.*, 2018, pp. 3870–3876.
[48] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized multitasking for evolutionary optimization of expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 44–58, Feb. 2019.
[49] M. Jun and R. D'Andrea, "Path planning for unmanned aerial vehicles in uncertain and adversarial environments," in *Cooperative Control: Models, Applications and Algorithms*. Boston, MA, USA: Springer, 2003, pp. 95–110.
[50] M. Monwar, O. Semiari, and W. Saad, "Optimized path planning for inspection by unmanned aerial vehicles swarm with energy constraints," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.
[51] J. Rubio-Hervas, A. Gupta, and Y.-S. Ong, "Data-driven risk assessment and multicriteria optimization of UAV operations," *Aerospace Sci. Technol.*, vol. 77, pp. 510–523, Jun. 2018.
[52] C. H. Reinsch, "Smoothing by spline functions," *Numerische Mathematik*, vol. 10, no. 3, pp. 177–183, 1967.
[53] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2007, pp. 862–876.

Kavitesh Kumar Bali received the M.Sc. degree in computer science from the University of the South Pacific, Suva, Fiji, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

His current research interests include evolutionary computation, transfer optimization, and machine learning.

Abhishek Gupta received the Ph.D. degree in engineering science from the University of Auckland, Auckland, New Zealand, in 2014.

He currently serves as a Scientist with the Singapore Institute of Manufacturing Technology, Agency for Science, Technology and Research, Singapore. He has diverse research experience in the field of computational science, ranging from numerical methods in engineering physics to topics in computational intelligence. His current research focus is in the development of memetic computation and probabilistic model-based algorithms for automated knowledge extraction and transfer across optimization problems, with applications in cyber-physical production systems and engineering design.

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 a President's Chair Professor of computer science with Nanyang Technological University (NTU), Singapore, and the Chief Artificial Intelligence Scientist with the Agency for Science, Technology and Research (A*star), Singapore. At NTU, he currently serves as the Director of the Data Science and Artificial Intelligence Research Center and the Director of the Singtel-NTU Cognitive and Artificial Intelligence Joint Lab. His current research interests are in artificial and computational intelligence.

Dr. Ong has also received several IEEE outstanding paper awards, listed as a Thomson Reuters Highly Cited Researcher and among the World's Most Influential Scientific Minds. He is the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and an Associate Editor of several IEEE TRANSACTIONS.

Puay Siew Tan received the Ph.D. degree in context-aware B2B collaboration from Nanyang Technological University, Singapore, in 2010.

She leads the Manufacturing Control Tower as the Programme Manager which is responsible for the setup of Model Factory with the Singapore Institute of Manufacturing Technology, Singapore. Her research has been in the cross-field disciplines of computer science and operations research for cyber physical production system collaboration, in particular sustainable complex manufacturing and supply chain operations. She has been active in using context-aware and services techniques. Her current research interests include complex systems, specifically quantitative techniques for understanding disruptions propagation in networked supply chains and mitigation of risks caused by these disruptions.