Multi-Objective Multifactorial Optimization in Evolutionary Multitasking

Abhishek Gupta, Yew-Soon Ong, Liang Feng, and Kay Chen Tan

***Abstract* — In recent decades, the field of multi-objective optimization has attracted considerable interest among evolutionary computation researchers. One of the main features that makes evolutionary methods particularly appealing for multi-objective problems is the implicit parallelism offered by a population, which enables simultaneous convergence towards the entire Pareto front. While a plethora of related algorithms have been proposed till date, a common attribute among them is that they focus on efficiently solving only a single optimization problem at a time. Despite the known power of implicit parallelism, seldom has an attempt been made to multitask, i.e., to solve multiple optimization problems simultaneously. It is contended that the notion of *evolutionary multitasking* leads to the possibility of automated transfer of information across different optimization exercises that may share underlying similarities, thereby facilitating improved convergence characteristics. In particular, the potential for automated transfer is deemed invaluable from the standpoint of engineering design exercises where manual knowledge adaptation and re-use are routine. Accordingly, in this paper, we present a realization of the evolutionary multitasking paradigm within the domain of multi-objective optimization. The efficacy of the associated evolutionary algorithm is demonstrated on some benchmark test functions as well as on a real-world manufacturing process design problem from the composites industry.**

***Index Terms* — Evolutionary Multitasking, Multi-objective Optimization, Memetic Computation.**

I. Introduction

MULTI-OBJECTIVE Optimization Problems (MOOPs) are ubiquitous in real-world decision making. It is generally the case that a decision maker must simultaneously account for multiple criteria while selecting a particular plan of action, with each criterion contributing a different objective to be optimized. When no *a priori* preference relationship between the criterions can be established, one is forced to abandon the standard engines of single-objective optimization.

Solving an MOOP involves obtaining a set of solutions that provide optimal trade-off among all the relevant objectives [1]. In other words, a solution is considered optimal in the multi-objective sense if an attempted improvement in any one of its objectives is necessarily accompanied by the deterioration of at least one other objective. This is often found to be the case when the criterions facing the decision maker are mutually competing. In recent decades, population-based optimization algorithms, such as those of Evolutionary Computation (EC), have emerged as the preferred choice for tackling MOOPs [2]-[10]. Some popular examples of Multi-objective Evolutionary Algorithms (MOEAs) commonly in use today include NSGA-II [11], NSGA-III [12], [13], SPEA2 [14], MOEA/D [15]-[17] and indicator-based MOEAs [18].

In the field of computational intelligence, Evolutionary Algorithms (EAs) are stochastic optimization engines that draw inspiration from Darwinian principles of *natural selection* or *survival of the fittest* [19], [20]. Thus, they are often described as computational analogues of biological evolution. Their increasing popularity in the fields of science, operations research, and engineering, can be attributed to the fact that EAs are derivative-free global optimizers that do not impose any continuity and/or differentiability requirements on the underlying objective function landscapes. Moreover, with regard to MOOPs, the population-based approach of MOEAs provides notable advantages over their mathematical counterparts. While classical methods typically solve for the set of optimal solutions in a sequential manner [1], MOEAs are able to harness the implicit parallelism of a population to synchronously obtain a diverse set of near optimal trade-off points. Encouraged by this observation, a central goal of the present study is to investigate and further exploit the potential benefits of implicit parallelism as made available by a population-based search strategy. To this end, it is observed that a common feature among all MOEAs is that they focus on efficiently solving only a single MOOP at a time. Seldom has an effort been made to multitask, i.e. to solve multiple MOOPs concurrently using a single population of evolving individuals.

It has been recently demonstrated in an introductory study on *Multifactorial Optimization* (MFO) that the process of *evolutionary multitasking* leads to implicit genetically-encoded information transfer across optimization tasks, which often facilitates improved convergence characteristics [21]. In the real-world, where manual knowledge adaptation across related tasks is commonplace, automated transfer of pertinent information is a particularly attractive proposition. In fact, ignoring such useful information that may lie outside the self-contained scope of a particular problem, as is generally the case with traditional optimization methods, is considered highly counterproductive. For instance, in complex engineering design exercises, the spontaneous refinement and transfer of relevant knowledge across similar design exercises can accelerate convergence towards near-optimal solutions of several optimization exercises at the same time, thereby significantly lowering the often exorbitant design time. Based on the aforesaid practical motivation, in this paper, we aim to push the envelope of existing EC methods by proposing an

This work was conducted within the Rolls-Royce@NTU Corporate Lab with support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

Abhishek Gupta and Yew-Soon Ong are affiliated to the Rolls-Royce@NTU Corporate Lab c/o, School of Computer Engineering, Nanyang Tech. University, Singapore. E-mail: {abhishekg, asysong}@ntu.edu.sg.

Liang Feng is with the College of Computer Science, Chongqing University, China. E-mail: liangf@cqu.edu.cn

Kay Chen Tan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. E-mail: {eletankc@nus.edu.sg}.

amalgamation of multi-objective optimization with the notion of evolutionary multitasking. The resultant paradigm is referred to as *Multi-Objective Multifactorial Optimization* (MO-MFO), wherein each constitutive MOOP contributes a distinct *factor* influencing the evolutionary search process.

While devising an EA with the capability of effective multitasking, it must be ensured that the population of navigating agents are appropriately steered through the multifactorial landscape. To this end, we turn to nature for inspiration. Specifically, we find the bio-cultural models of *multifactorial inheritance* [22], [23] to be well suited to our computational needs. These models have been widely studied by human geneticists for several years, and serve as a well-established means to explain the transmission of complex developmental traits to offspring through the interactions of genetic and cultural factors [24]. For our conceived analogy with the computational world, we consider the assortment of MOOPs in a multitasking environment to represent multiple cultural traits (or *memes* [25]) that coexist in a unified genotype space and intervene in the evolutionary process by interacting with the genetic mechanisms. The algorithmic manifestation of this phenomenon leads to the formulation of the *Multi-Objective Multifactorial Evolutionary Algorithm* (MO-MFEA). As the working of the MO-MFEA is based on the transmission of biological and cultural building blocks from parents to offspring, the algorithm is regarded as belonging to the realm of *memetic computation* [26]-[28].

In order to provide a thorough exposition of the ideas discussed so far, this paper has been organized as follows. In Section II, we present a brief overview of the preliminaries in multi-objective optimization and its extension to multitasking settings via MO-MFO, highlighting, in particular, the practical and theoretical distinctions between the two paradigms. Thereafter, in Section III, we delve into the details of MO-MFO whilst describing its associated EA. After having established the conceptual and algorithmic foundations of our work, we proceed to computational studies in Section IV where we carry out experiments on some synthetic benchmark functions. Then, in Section V, we present a real-world case study from the composites manufacturing industry that demonstrates the practical value of our proposed method in complex engineering design exercises. To conclude, the main contributions of the paper are summarized in Section VI together with important directions for future research work.

II. Multi-objective and Multifactorial Optimization

In this section, we present a review of the major concepts in multi-objective and multifactorial optimization that form the crux of our explorations in this paper. Distinctions between the two paradigms shall also be highlighted.

A. Multi-Objective Optimization

In a generic multi-objective minimization problem, one attempts to find all solutions $\boldsymbol{x} \in \boldsymbol{X}$ such that the vector-valued objective function $F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}),\ldots., f_M(\boldsymbol{x}))$ is minimized. Here, $\boldsymbol{X}$ represents the design space and M is the number of objective functions. Note that in real-world settings the design space is often subject to a variety of stringent constraints that must be satisfied for a solution $\boldsymbol{x}$ to be considered feasible.

While constructing an algorithm for the purpose of multi-objective optimization, it is important to devise a means of comparing candidate solutions. Following the principle of Pareto dominance [29] a feasible solution $\boldsymbol{x}_1$ is said to Pareto dominate another feasible solution $\boldsymbol{x}_2$ if and only if $f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$, $\forall i \in \{1, 2, \ldots, M\}$, with at least one strict inequality. Accordingly, a solution $\boldsymbol{x}^*$ is said to be optimal in multi-objective sense (or Pareto optimal) if $\boldsymbol{x}^*$ is feasible and is non-dominated with respect to all other feasible solutions in $\boldsymbol{X}$.

From the description above, it follows that for $M > 1$ there will exist a set of Pareto optimal solutions, especially when the objectives are conflicting in nature (i.e., an improvement in one is accompanied by the deterioration of another). The image of all the Pareto optimal solutions in the objective space is said to constitute the Pareto front (PF) [30].

B. Multi-Objective Multifactorial Optimization

The Multi-Objective Multifactorial Optimization (MO-MFO) paradigm serves as a means of fully unlocking the potential of implicit parallelism of population-based search. We begin the formulation by considering a theoretical scenario where K distinct optimization tasks (that are traditionally treated separately as self-contained MOOPs) are to be solved concurrently. Without loss of generality, all tasks are assumed to be minimization problems. The j^{th} task, denoted as T_j, has design space $\boldsymbol{X}_j$ on which the vector-valued objective function is defined as F_j: $\boldsymbol{X}_j \rightarrow \mathbb{R}^{M_j}$, where M_j is the number of elements in the objective function vector. Then, we define MO-MFO as an evolutionary multitasking approach that aims to simultaneously navigate the design space of all tasks, assisted by the potential for fruitful genetic transfer, so as to efficiently deduce $argmin\{F_1(\boldsymbol{x}), F_2(\boldsymbol{x}), \ldots., F_K(\boldsymbol{x})\}$. Since each F_j presents an added *factor* influencing the evolutionary processes of a single population, the combined problem may also be termed as a K-factorial environment.

With the end goal of developing an effective evolutionary solver for MO-MFO, it is important to formulate a standard methodology for comparing candidate solutions during multitasking. In order to achieve this, we first define a set of properties for describing every individual p_i, where $i \in \{1, 2, \ldots, |P|\}$, in a population P. Note that every individual is encoded into a unified space $\boldsymbol{Y}$ that encompasses $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K$, and can be translated into a task-specific solution with respect to any of the K optimization tasks.

Definition 1 (*Factorial Rank*)*:* The factorial rank r_j^i of p_i for task T_j is the index of p_i in the list of population members sorted in decreasing order of preference with respect to T_j.

Definition 2 (*Skill Factor*)*:* The skill factor τ_i of p_i is the one task, amongst all other tasks in a K-factorial environment, with which the individual is associated. If p_i is evaluated for all tasks then $\tau_i = argmin_j\{r_j^i\}$, where $j \in \{1, 2, .., K\}$.

Definition 3 (*Scalar Fitness*)*:* The scalar fitness of p_i in a multitasking environment is given by $\varphi_i = 1/r_{\tau_i}^i$.

With regard to Definition 1, we realize that prescribing an order of preference among individuals of a population is not trivial when the task is an MOOP. One possible approach for determining a meaningful order of preference, as has been employed in the Multi-Objective Multifactorial Evolutionary Algorithm (MO-MFEA), shall be presented in Section III.

After the fitness of each individual is scalarized as per Definition 3, performance comparisons can be performed in a simplistic manner. To demonstrate, an individual p_1 will be considered to dominate some other individual p_2 in multifactorial sense simply if $\varphi_1 > \varphi_2$.

C. Distinguishing the Two Paradigms

As multi-objective optimization and evolutionary multitasking (which is manifested as MO-MFO in this paper) are both concerned with optimizing a set of objective functions, conceptual similarities may be seen to exist between them. However, it must be noted that while evolutionary multitasking aims to leverage upon the implicit parallelism of population-based search to exploit latent complementarities between essentially separate (but possibly similar) tasks, multi-objective optimization deals with efficiently resolving conflicts among competing objectives of the same task. An illustration summarizing the statement is depicted in Fig. 1. The main feature distinguishing the two paradigms is the simultaneous existence of multiple heterogeneous design spaces in the case of multitasking. On the other hand, for the case of multi-objective optimization, there typically exists a single design space for a given task, with all objective functions depending on variables contained within that space. As a point of particular interest, note that a multitasking environment could in fact include a multi-objective optimization task as one among many other concurrent tasks. This aspect shall be thoroughly investigated in the subsequent sections of the paper. Most importantly, it highlights the greater generality of the multi-objective multifactorial optimization paradigm.

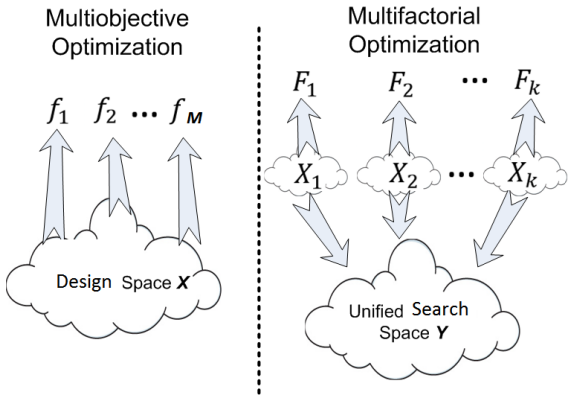

Fig. 1. While multi-objective optimization generally comprises a single design space for all objective functions, evolutionary multitasking unifies multiple heterogeneous design spaces.

On observing Fig. 1, it may be argued that once a unified search space encompassing all tasks has been defined, it is conceivable for standard MOEAs to be adapted for the purpose of evolutionary multitasking. In this regard, it must be noted that the concept of Pareto dominance, which forms the crux of most MOEAs, is not an explicit ingredient in the prescribed scope for multifactorial optimization. To be precise, the purpose of evolutionary multitasking is to optimize each constitutive task absolutely, instead of having to establish any kind of trade-off between individual tasks. Thus, the evolutionary selection pressure that is imparted to a population during multi-objective optimization may not be entirely well suited for the purpose of multitasking in optimization.

We illustrate the effect of the selection pressure by considering the hypothetical 2-factorial environment in Fig. 2 comprising a pair of 'mono-objective' tasks. From the notion of Pareto dominance in multi-objective optimization, it follows that individuals $\{p_2, p_3, p_4, p_5\}$ belong to the first non-dominated front while $\{p_1, p_6\}$ belong to the second front. In other words, individuals $\{p_2, p_3, p_4, p_5\}$ are considered incomparable to each other and are always preferred over $\{p_1, p_6\}$ as they represent superior convergence to the true PF. In contrast, working out the scalar fitness of the individuals according to the definitions in Section II-B, we find that individuals p_2 and p_5 have scalar fitness = 1 (as they minimize task 1 and task 2, respectively), individuals p_1 and p_6 have scalar fitness = 0.5, and finally, individuals p_3 and p_4 have scalar fitness = 0.33. Thus, based on the fitness assignment scheme in multitasking, the evolutionary selection pressure favors individuals $\{p_2, p_5\}$ over $\{p_1, p_6\}$, which are in turn favored over $\{p_3, p_4\}$. As is therefore clear, there emerges a disagreement between the outcomes deduced from the principles of multi-objective and multifactorial optimization.

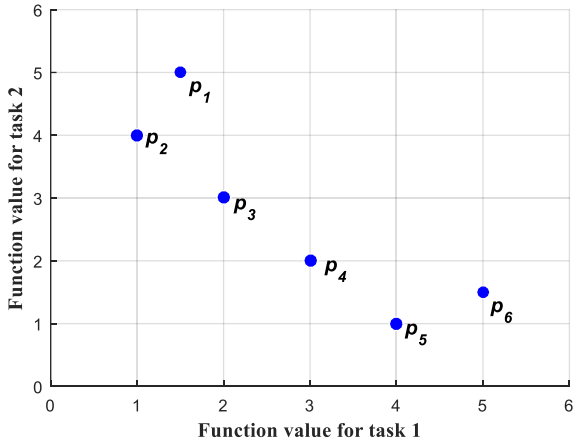

Fig. 2. Sample points in the combined objective space of two hypothetical mono-objective tasks in a 2-factorial environment.

III. Multi-objective Multifactorial Evolution

In MO-MFO, a matter of significant importance is the prescription of a meaningful order of preference among candidate solutions to a constitutive MOOP. Doing so is necessary for determining the factorial rank (see Definition 1), and consequently, the scalar fitness (see Definition 3) of an individual in the Multi-Objective Multifactorial Evolutionary Algorithm (MO-MFEA). Thus, we begin this section by first describing a simple approach for achieving the above. Thereafter, based on the prescribed ordering scheme, we present details of the MO-MFEA.

A. Ordering Population Members in MOOPs

For the sake of brevity, the concepts of Non-dominated Front (NF) and Crowding Distance (CD) in constrained multi-objective optimization are directly adopted from the literature

[11], [31]. We do not elaborate on their interpretations in this paper since these concepts have been well-established over several years of multi-objective optimization research. For an overview of the topic, the reader is referred to [32] and [33].

Notice that for ordering individuals in a population, it is sufficient to define a preference relationship between two individuals and show that the binary relationship satisfies the properties of irreflexivity, asymmetry, and transitivity. To this end, let us consider a pair of individuals p_1 and p_2 with non-dominated fronts NF_1 and NF_2 and crowding distances CD_1 and CD_2, respectively. With the aim of obtaining a diverse distribution of points along the PF, we prescribe individual p_2 to be preferred over p_1 (i.e., $p_2 \succ p_1$) if any one of the following conditions holds:

- $NF_2 < NF_1$
- $NF_2 = NF_1$ and $CD_2 > CD_1$

For the aforementioned preference relationship, the satisfaction of the necessary properties can be simply shown as below.

Property 1 (Irreflexivity): $p_i \not\succ p_i$, for all $p_i \in P$.

Proof: Suppose $p_i \succ p_i$. Then, either (a) $NF_i < NF_i$ or (b) $NF_i = NF_i$ and $CD_i > CD_i$. However, since $NF_i = NF_i$ and $CD_i = CD_i$, the supposition leads to a contradiction.

Property 2 (Asymmetry): If two individuals p_1 and p_2 satisfy $p_2 \succ p_1$, then $p_1 \not\succ p_2$.

Proof: Suppose $p_1 \succ p_2$. Then, either (a) $NF_1 < NF_2$ or (b) $NF_1 = NF_2$ and $CD_1 > CD_2$. However, according to $p_2 \succ p_1$, we have either (a) $NF_2 < NF_1$ or (b) $NF_2 = NF_1$ `and $CD_2 > CD_1$. Thus, the supposition leads to a contradiction.

Property 3 (Transitivity): If $p_2 \succ p_1$ and $p_3 \succ p_2$, then it must also be the case that $p_3 \succ p_1$.

Proof: We are given that p_2 is preferred over p_1 according to the conditions stated earlier. Since $p_3 \succ p_2$ we also have either (a) $NF_3 < NF_2$ or (b) $NF_3 = NF_2$ and $CD_3 > CD_2$. If condition (a) is true then it implies $NF_3 < NF_1$. If condition (b) is true then it implies either (b.1) $NF_3 < NF_1$ or (b.2) $NF_3 = NF_1$ and $CD_3 > CD_1$. Therefore, if p_2 is preferred over p_1, and p_3 is preferred over p_2, then p_3 must also be preferred over p_1.

B. The MO-MFEA

Having set out the basic concepts of MO-MFO, we proceed to the development of the MO-MFEA. The pseudocode of the MO-MFEA is presented in Algorithm 1. The algorithm has been built upon the popular Non-dominated Sorting Genetic Algorithm (NSGA-II) [11]. In fact, in the special case of $K = 1$, the algorithm takes the exact form of NSGA-II. The interesting feature of the MO-MFEA is that it attempts to combine biological and cultural building blocks for the purpose of effective evolutionary multitasking. Accordingly, the algorithm has been attributed to the emerging area of memetic computation.

The MO-MFEA begins by generating a random initial population of N individuals in a unified search space $\boldsymbol{Y}$. Further, every individual in the population is assigned a specific skill factor (refer to Definition 2). The assignment technique must ensure that every task is uniformly represented. Note that, in the MO-MFEA, the skill factor of an individual is seen as a computational depiction of its assigned cultural trait. The importance of such an assignment is mainly to guarantee that every individual is evaluated with respect to only one task (i.e., its assigned skill factor) amongst all other tasks in the multitasking environment. This step is practically motivated as exhaustively evaluating every individual for every task will often be computationally too demanding, particularly when K is large (many-tasking).

Algorithm 1: Pseudocode of the MO-MFEA

1. Generate N individuals in $\boldsymbol{Y}$ to form initial population P_0
2. **for every** p_i in P_0 **do**
   Assign skill factor τ_i
   Evaluate p_i for task τ_i only
3. **end for**
4. Compute scalar fitness φ_i for every p_i based on NF and CD
5. Set $t = 0$
6. **while** (stopping conditions are not satisfied) **do**
   P'_t = Binary Tournament Selection(P_t)
   C_t = Offspring(P'_t) → Refer Algorithm 2
   for every c_i in C_t **do**
   Determine skill factor τ_i → Refer Algorithm 3
   Evaluate c_i for task τ_i only
   end
   $R_t = C_t \cup P_t$
   Update scalar fitness of all individuals in R_t.
   Select N fittest members from R_t to form P_{t+1}.
   Set $t = t + 1$
7. **end while**

1) Offspring creation

For producing offspring, a pool of parent candidates is first created via binary tournament selection. The selection is performed purely based on the scalar fitness values of the individuals (see Definition 3), regardless of their respective skill factors. In other words, two individuals with different skill factors are considered comparable simply based on their scalar fitness values. Individuals in the parent pool are to undergo crossover and/or mutation in the unified search space $\boldsymbol{Y}$, thereby passing down their genetic material to a new generation of offspring. At this juncture, in accordance with the phenomenon of *assortative mating* in multifactorial inheritance [22], [23], we establish a set of conditions that must be satisfied for two randomly selected parent candidates (from the pool) to undergo crossover. If these conditions are not satisfied, then the two candidate parents simply undergo mutation separately, producing two mutant offspring.

Algorithm 2: Offspring creation via assortative mating

Consider candidate parents p_1 and p_2 in P'_t

1. Generate a random number *rand* between 0 and 1.
2. **if** $\tau_1 == \tau_2$ or *rand* < *rmp* **then**
   (c_1, c_2) = Crossover+Mutate(p_1, p_2)
3. **else**
   c_1 = Mutate(p_1), c_2 = Mutate(p_2)
4. **end if**

The principle of assortative mating states that individuals prefer to mate with those belonging to a similar cultural background. Accordingly, in the MO-MFEA, parents having the same skill factor (i.e., having identical cultural traits) can crossover freely, while cross-cultural parents may only crossover with a prescribed random mating probability (*rmp*). An overview of the steps are presented in Algorithm 2.

2) Offspring evaluation

Besides inheriting the genetic material of their parents, offspring are also culturally influenced by them, as is explained by the natural phenomenon of *vertical cultural transmission* [24]. In gene-culture co-evolutionary theory [24], vertical cultural transmission is regarded as a type of inheritance that acts jointly with genetics and causes the phenotype of offspring to be affected by that of their parents. In the proposed framework, the aforementioned notion is algorithmically realized via a *selective imitation strategy*. To elaborate, the strategy mimics the natural tendency of offspring to imitate the cultural traits of their parents. Accordingly, an offspring created in the MO-MFEA randomly imitates the skill factor of any one of its parents. Thus, it is enforced that an individual can only be evaluated for one task with which at least one of its parents is associated. A summary of the proposed steps is provided in Algorithm 3. Recall that selective evaluation has a critical role in reducing the computational expense of the MO-MFEA, especially with increasing number of tasks to be handled at the same time.

Algorithm 3: Vertical cultural transmission via selective imitation

Consider offspring $c \in C_t$

1. Generate a random number *rand* between 0 and 1.
2. **if** c = Crossover+Mutate(p_1, p_2) and *rand* ≤ 0.5 **then**
   c imitates skill factor of p_1
3. **else if** c = Crossover+Mutate(p_1, p_2) and *rand* > 0.5 **then**
   c imitates skill factor of p_2
4. **else if** c = Mutate(p_1)
   c imitates skill factor of p_1
5. **else**
   c imitates skill factor of p_2
6. **end if**

C. Constructing the Unified Search Space

For efficient inter-task *implicit genetic transfer* [21] in an evolutionary multitasking environment, it is essential to first describe a unified genotype space that encompasses the heterogeneous design spaces of all constitutive optimization tasks. In fact, the unification can be viewed as a higher-order abstraction [34] or *meme space* wherein genetic building blocks of encoded knowledge [35] are processed and shared across tasks. Thus, the schemata corresponding to different tasks are combined into a unified pool of genetic material, which enables the MO-MFEA to process them in parallel. With this background, consider a multitask setting where the design space dimensionality of task T_j is D_j, for $j \in \{1, 2, \ldots, K\}$. Herein, the unified search space $\boldsymbol{Y}$ is constructed such that $D_{multitask} = max_j\{D_j\}$. In other words, the chromosome $\boldsymbol{y}$ of an individual in $\boldsymbol{Y}$ is simply a vector of $D_{multitask}$ *random-keys* [36], [37]. While addressing the j^{th} task, we extract D_j variables (random-keys) from $\boldsymbol{y}$ and decode them into a relevant solution representation. In many cases of practical interest where some a priori domain knowledge is available, an informed selection of D_j task-specific variables from the list of $D_{multitask}$ variables can significantly improve the utilization of shared knowledge. For instance, for the real-world case study in Section V, variables belonging to different tasks but bearing the same phenotypic meaning are associated with the same keys in $\boldsymbol{y}$. However, in many naive cases with no prior domain knowledge, extracting the *first* D_j variables from the chromosome can also be a viable alternative [21].

Based on the above, we now describe a sample decoding procedure for a chromosome $\boldsymbol{y}$ to be translated into a task-specific solution representation. For the case of continuous optimization, a straightforward linear mapping of the random-keys from the genotype space to the box-constrained design space of the optimization task suffices. For example, consider task T_j for which the i^{th} variable (x_i) is box-constrained as [L_i, U_i]. Assuming the i^{th} random-key of $\boldsymbol{y}$ to be $y_i \in [0, 1]$, the decoding may simply be achieved as follows,

$$x_i = L_i + (U_i - L_i) \cdot y_i. \tag{1}$$

D. The Mechanics of Implicit Genetic Transfer

In any EA, 'knowledge' exists in the form of a population of genetically encoded solutions. Thus, for transfer of knowledge to occur across tasks in multitasking environments, an interesting proposition is the implicit transfer of genetic material between candidate solutions belonging to different tasks. In the MO-MFEA, this is achieved by two components of the algorithm acting in conjunction, namely, (a) the random mating probability, which allows individuals with distinct skill factors to crossover, and (b) the fact that their offspring can randomly select a parental skill factor for imitation.

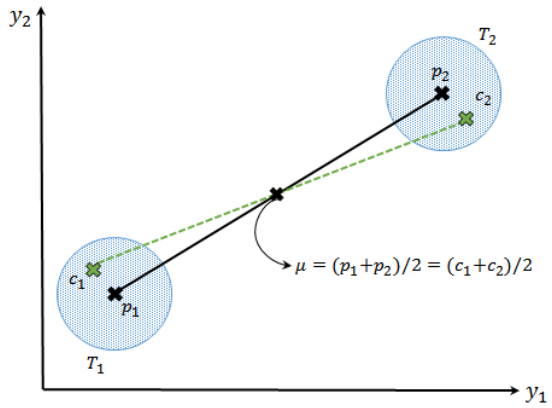

Fig. 3. The SBX crossover produces offspring (c_1 and c_2) that are located near their parents (p_1 and p_2) with high probability. If the parents possess different skill factors, a multicultural environment is created for offspring to be reared in. This condition leads to the possibility of implicit genetic transfer (refer to main text for details).

In [21], the classical Simulated Binary Crossover (SBX) operator was employed in the chromosomal mating step. A

salient property of the SBX operator, which happens to be of interest in the process of multitasking, is that it creates offspring that are located in close proximity of the parents with high probability [38], [39]. With this background, consider the situation in Fig. 3 where two parents p_1 and p_2 perform crossover in a hypothetical 2-D search space. Note that p_1 has skill factor T_1 while p_2 has skill factor T_2, with $T_1 \neq T_2$. As is generally the case, offspring c_1 and c_2 are created close to the parents. To elaborate, c_1 inherits much of its genetic material from p_1, while c_2 is found to be genetically closer to p_2. Given such a setup, if c_1 randomly imitates the skill factor of p_2 (i.e., c_1 is evaluated for T_2) and/or if c_2 randomly imitates the skill factor of p_1 (i.e., c_2 is evaluated for T_1), then implicit genetic transfer is said to have occurred between the two tasks. Now, if the genetic material associated with T_1 (being carried by c_1) happens to be useful for T_2 as well, or vice versa, then the transfer is beneficial. Else, there exists a possibility that the transfer turns out to be impeding (or negative) [40]-[42]. However, the remarkable feature of evolution is that when negative transfer occurs, the inferior genes get automatically depleted and eventually removed from the population (over the course of a few generations) by the natural process of survival of the fittest.

With regard to the random mating probability, it plays an additional role of reflecting the intuition of the designer or decision maker towards the possibility of fruitful genetic transfer between tasks. If the tasks are known to be closely related, then *rmp* should be set close to 1 in order to facilitate unhindered exchange of genetic material. In contrast, when the relationship between tasks is uncertain, a lower value of *rmp* may be used to imply only occasional genetic exchange. The latter strategy acts as a means of lowering the chance of excessive negative transfer that could encumber the search.

IV. The Boon of Multitasking

In this section, we showcase the utility of the proposed evolutionary multitasking paradigm by presenting experimental studies for some synthetic benchmark functions that are expected to pose challenges for many existing optimization algorithms. The performance of the MO-MFEA (which executes multitasking) shall be compared against its base algorithm, i.e., the NSGA-II (which executes a single task at a time). Recall that in the special case of a 1-factorial environment, MO-MFEA is in fact equivalent in form to NSGA-II. Since the purpose of this comparison is to demonstrate the performance improvements achievable via multitasking alone, we employ *identical* solution encoding, genetic operators, and parameter settings in the MO-MFEA and in NSGA-II.

A. Experimental Setup for Benchmark Functions

Both algorithms are initiated with a total population of N = 100 individuals, and are executed for a maximum of 25000 solution evaluations on each run. Note that in the present implementation of the MO-MFEA, the available budget of solution evaluations gets (roughly) evenly split among the various tasks in the multitasking environment. Thus, for the 2-factorial benchmark experiments, there occur approximately 12500 evaluations per task in the MO-MFEA.

With regard to genetic operators, we employ SBX crossover and polynomial mutation [43]. Further, a crossover probability of $p_c = 1$ is considered with no uniform crossover-like variable swap between offspring. The latter condition helps reduce chromosome disruption [44] and allows the effects of inter-task implicit genetic transfer to be clearly seen. The mutation probability is set to $p_m = 1/D$. Note that, for NSGA-II, $D = D_j$ while solving T_j. On the other hand, for the MO-MFEA, $D = D_{multitask}$. As we evaluate several difficult (Multimodal + Rotated) benchmark functions in our experiments (which pose a stiff challenge in terms of convergence to the true PF), we emphasize search space exploration by using relatively low distribution indices for the SBX and polynomial mutation operations. Accordingly, we fix the crossover index and the mutation index as $\eta_c = 10$ and $\eta_m = 10$, respectively. Lastly, regarding the random mating probability of the MO-MFEA, we set it to 1 in order to facilitate unhindered exchange of genes between tasks.

For the purpose of effective performance comparison between the MO-MFEA and NSGA-II, we make use of the *Inverted Generational Distance* (IGD) metric [45]. For benchmark functions where the true PF is known beforehand, the IGD simply measures the Euclidean distance between the true PF and the approximate PF obtained by the algorithms. The metric has been especially defined to provide a reasonable representation of the convergence and diversity of the approximate PF.

B. Specification of Benchmark MOOPs

We consider a set of highly multimodal MOOPs. The challenge with multimodality lies in convergence to the true PF, given how easy it is to get trapped at a local PF. However, during evolutionary multitasking, the MO-MFEA is expected to autonomously exploit the genetic complementarities between tasks whenever available, thereby bypassing obstacles more easily to converge faster.

The task specifications presented hereafter are essentially variants of the ZDT4 benchmark function introduced by Zitzler *et al*. in [46]. Similar variants have also been used by Deb and Goel in [47] while verifying the effects of controlled elitism in the NSGA-II. The basic structure of the ZDT4 function is as follows,

$$minimize\ F(\boldsymbol{x}) = (f_1(x_1), f_2(\boldsymbol{x})),$$

$$where, f_1 = x_1\ and\ f_2 = g(\boldsymbol{x})\left[1 - \sqrt{x_1/g(\boldsymbol{x})}\right]. \quad (2)$$

In the above, by simply using different multimodal functions as $g(\boldsymbol{x})$, a variety of difficult multi-objective test problems may be constructed. Four such MOOPs are described next.

1) ZDT4-R

This problem is similar to the original ZDT4 where $g(\boldsymbol{x})$ takes the form of the Rastrigin function [48],

$$g(\boldsymbol{x}) = 1 + 10(D-1) + \sum_{i=1}^{D-1} z_i^2 - 10\cos(4\pi z_i),$$

$$where, \boldsymbol{z} = M \cdot (x_2, x_3, \ldots, x_D)^T, \quad (3)$$

M being a randomly generated (D-1) × (D-1) rotation matrix.

2) ZDT4-G

Here, $g(\boldsymbol{x})$ takes the form of the Griewank function [49],

$$g(\boldsymbol{x}) = 2 + \sum_{i=1}^{D-1} z_i^2/4000 - \prod_{i=1}^{D-1} \cos(z_i/\sqrt{i}),$$

$$where, \boldsymbol{z} = M \cdot (x_2, x_3, \dots, x_D)^T, \quad (4)$$

3) ZDT4-A

Here, $g(\boldsymbol{x})$ takes the form of the Ackley function [50],

$$g(\boldsymbol{x}) = 21 + \exp(1) - 20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{D-1} z_i^2}{D-1}}\right) - \exp\left(\frac{\sum_{i=1}^{D-1}\cos(2\pi z_i)}{D-1}\right),$$

$$where, \boldsymbol{z} = M \cdot (x_2, x_3, \dots, x_D)^T, \quad (5)$$

4) ZDT4-RC

This is a constrained version of ZDT4-R. As prescribed in [51], the constraints are imposed as follows,

$$\cos(\theta)\,(f_2 - e) - \sin(\theta)\,f_1 \geq a|\sin(b\pi(\sin(\theta)\,(f_2 - e) + \cos(\theta)f_1)^c)|^d,$$

$$where\ \theta = -0.05\pi, a = 40, b = 5, c = 1, d = 6, e = 0. \quad (6)$$

Details of the design space corresponding to the four MOOPs are reported in Table I. In all cases, the dimensionality of the design space is assumed to be 10, i.e., D = 10 in Eq. (3) to Eq. (5).

TABLE I

SUMMARY OF THE BENCHMARK TEST FUNCTIONS

Task Label	Extent of Design Space	Properties
ZDT4-R	$x_1 \in [0,1]$; $x_i \in [-5, 5]$ for $i = 2, \dots, D$	Multimodal + Rotated
ZDT4-G	$x_1 \in [0,1]$; $x_i \in [-512, 512]$ for $i = 2, \dots, D$	Multimodal + Rotated
ZDT4-A	$x_1 \in [0,1]$; $x_i \in [-32, 32]$ for $i = 2, \dots, D$	Multimodal + Rotated
ZDT4-RC	$x_1 \in [0,1]$; $x_i \in [-5, 5]$ for $i = 2, \dots, D$	Multimodal + Rotated + Constrained

C. Numerical Results and Discussions

We construct a pair of 2-factorial multitasking instances by combining the multimodal MOOPs described earlier. In the first instance, we integrate ZDT4-R and ZDT4-G into a single multitasking environment. The combined problem is referred to as (ZDT4-R, ZDT4-G). Similarly, in the second instance, we combine ZDT4-RC and ZDT4-A, with the combined problem denoted as (ZDT4-RC, ZDT4-A). Observe that each multitasking instance comprises tasks with heterogeneous design spaces, thereby highlighting the utility of the unified search space. Moreover, the true PFs of the tasks are known to intersect in the unified search space, thereby reflecting the prevalence of transferrable knowledge between distinct tasks in the real-world. Finally, note that all performance results reported in the subsection are averaged across 31 independent runs of the optimizers. The performance of evolutionary multitasking via the MO-MFEA is compared against the performance of NSGA-II which operates only on a single MOOP at a time.

1) Instance 1: (ZDT4-R, ZDT4-G)

On analyzing the landscapes of ZDT4-R and ZDT4-G, it is noted that while the former comprises 21^9 local PFs, the latter has 163^9 local PFs (of which only one is the global optimum). Interestingly, we find that the basin of attraction corresponding to each local PF of ZDT4-G is narrow, at least when measured in the unified search space. This implies that it is relatively straightforward for a population of searching individuals to quickly jump across local optimums, without getting trapped in one. In contrast, the ZDT4-R function possesses wide basins of attraction corresponding to each local optimum, making it much harder for individuals to escape. Thus, while combining ZDT4-R and ZDT4-G in a single multitasking environment, it is expected that ZDT4-G will experience faster convergence, thereby making refined genetic material continuously available for transfer to ZDT4-R.

Our expectations are closely borne out by the numerical results depicted in Fig. 4. The figure shows that while solving ZDT4-R separately using NSGA-II, the IGD progresses slowly, which highlights the tendency of the population to get trapped in a local PF. Notably, when ZDT4-R and ZDT4-G are integrated, convergence characteristics are significantly improved due to the exploitation of shared genetic building blocks during evolutionary multitasking. This fact is aptly demonstrated by the accelerated convergence of the curve MO-MFEA(ZDT4-R**, ZDT4-G) in Fig. 4.

It is also noteworthy that the convergence characteristic of the less challenging ZDT4-G function is not impeded by multitasking. In fact, the rapid convergence of the curve MO-MFEA(ZDT4-R, ZDT4-G**), as shown in Fig. 4, implies that (at least in this case) evolutionary multitasking successfully aids both functions. In other words, the continuous exchange of genetic material between tasks allows the exploitation of both landscapes simultaneously, thereby enabling the population to effectively avoid obstacles to converge faster.

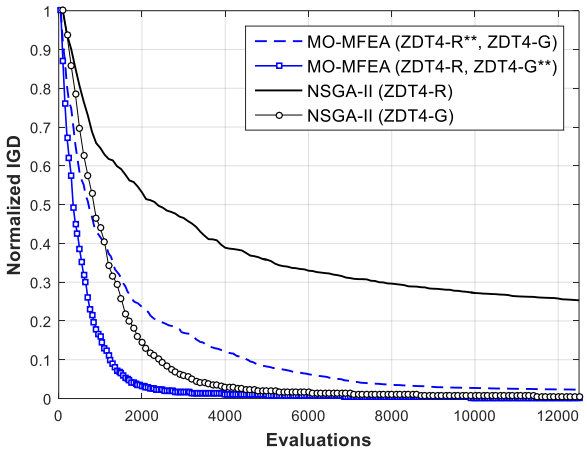

Fig. 4. Comparing averaged convergence characteristics of the normalized IGD metric for instance (ZDT4-R, ZDT4-G). Note that the curve corresponding to a particular task in MO-MFO is denoted by appending asterisks (**) to the task label.

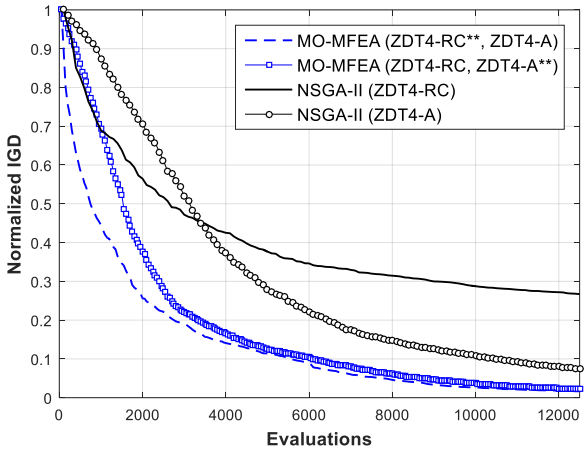

Fig. 5. Comparing averaged convergence characteristics of the normalized IGD metric for instance (ZDT4-RC, ZDT4-A). Note that the curve corresponding to a particular task in MO-MFO is denoted by appending asterisks (**) to the task label.

2) Instance 2: (ZDT4-RC, ZDT4-A)

In the second multitasking instance, both constitutive optimization tasks are considered to be significantly challenging. ZDT4-RC is a constrained version of ZDT4-R, thereby adding to the complexity of the MOOP. Further, ZDT4-A introduces its own challenges. In particular, when the population is far from the true PF, the local optimum structure of the ZDT4-A function poses difficulties in accurately identifying the progress made by population members. This is because distantly located local PFs are found to have approximately the same objective function values.

Based on the above-stated observations, the second instance provides an ideal setting to showcase the manner in which evolutionary multitasking facilitates a fruitful interplay between the objective function landscapes of constitutive tasks. Specifically, in (ZDT4-RC, ZDT4-A), the landscape of ZDT4-RC drives evolution during the initial stages (when the population is far from the true PF) while ZDT4-A takes over during the latter stages. As depicted in Fig. 5, the implicit genetic transfer lends a strong impetus to the evolutionary search process of both tasks, which explains the accelerated convergence characteristics achieved by the MO-MFEA.

V. A Real-World Case Study from the Composites Manufacturing Industry

Among humans, the inherent ability to multitask is brought to the fore on a daily basis while simultaneously dealing with a variety of responsibilities or tasks. Whenever a family of tasks possesses some underlying commonalities or complementarities, the process of multitasking provides the means for spontaneous exchange of information among them, which can often assist in improved task execution. In [21], this anthropic phenomenon was executed computationally in the form of evolutionary multitasking. For demonstrating the utility of this novel paradigm in real-world problem solving, we present an illustrative case study from the field of complex engineering design where the practice of manual adaption and re-use of relevant knowledge is routine.

A. A Brief Introduction to Composites Manufacturing

We consider two distinct composites manufacturing techniques that belong to the same family of rigid-tool Liquid Composite Molding (LCM) processes [52]. To elaborate, (a) Resin Transfer Molding (RTM) and (b) Injection/Compression Liquid Composite Molding (I/C-LCM) are popular techniques for high volume production of Fiber-Reinforced Polymer (FRP) composite parts, and are distinguished by the use of highly stiff (rigid) molds. As the mold undergoes negligible deflection in response to the large internal forces originating from the combined effect of high injection pressure of liquid resin and compaction of the (solid) fibrous reinforcement, rigid-tool LCM processes find numerous applications in areas requiring high geometrical precision, such as the automobile and aerospace industries. However, in order to ensure feasibility of the manufacturing cycle, sophisticated peripheral equipment is often needed to equilibrate the large internal forces. Accordingly, two crucial objectives emerge in the optimal design of LCM processes: (a) maximization of throughput, and (b) minimization of estimated capital layout and running costs of the peripheral equipment. While the former objective is directly related to the manufacturing time per part, the second objective is estimated to be proportional to the magnitude of internal force [53], [54].

Next, we briefly describe the two composites manufacturing processes under consideration in this study.

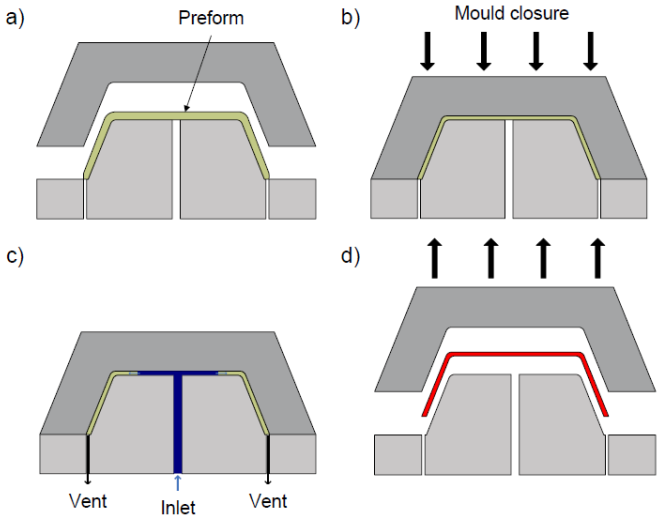

Fig. 6. Workflow of RTM [52]: (a) placement of preform, (b) complete mold closure to desired part thickness, (c) high pressure liquid resin injection, (d) extraction of final part.

1) Resin transfer molding

The setup of the RTM process typically consists of a metallic mold machined according to the geometry of the FRP part to be manufactured. The first step is to place a preform of the fibrous reinforcement inside the mold cavity (as shown in step [a] of Fig. 6). The mold is then completely closed, fully compressing the preform to the final thickness of the part (step [b] in Fig. 6). Before liquid resin injection, the mold is heated to a desired (optimum) operation temperature. Then, a thermosetting resin is injected into the closed mold at high pressure until the resin reaches the vents (refer to step [c] in

Fig. 6). To conclude, the filled mold is allowed to rest until the liquid resin sufficiently solidifies, followed by extraction of the final part (step [d] in Fig. 6). The optimization of the RTM cycle involves four design variables, namely, (a) speed of mold closure ($V_{closure}$), (b) resin injection pressure (P_{inj}), (c) preheated mold temperature (T_{mould}), and (d) preheated resin temperature (T_{resin}). Thus, a design vector for the RTM cycle can be summarized as ($V_{closure}$, P_{inj}, T_{mould}, T_{resin}).

2) Injection/compression liquid composite molding

While mold filling in RTM is generally viewed as a single phase process, the same occurs in a two-phase manner in I/C-LCM. As illustrated in Fig. 7, during I/C-LCM, the mold is only *partially* closed prior to resin injection (as shown in step [b] in Fig. 7). After the required volume of liquid resin has been injected into the (partially) open mold (step [c] in Fig. 7), the mold is fully closed to the desired part thickness using a velocity-controlled mechanism (refer to step [d] in Fig. 7). Due to the inclusion of the *in situ* mold closure phase (step [d] in Fig. 7), the I/C-LCM cycle introduces two additional design variables, namely, (a) mold cavity thickness during resin injection (H_{inj}), and (b) velocity of final mold closure ($V_{closure}^{final}$). Thus, a design vector for the I/C-LCM cycle can be summarized as ($V_{closure}^{initial}$, P_{inj}, T_{mould}, T_{resin}, H_{inj},$V_{closure}^{final}$).

B. The Multi-Objective Optimization Tasks

The I/C-LCM cycle is often considered less competitive than RTM due to the practical difficulty of *in situ* mold compression (step [d] in Fig. 7) [55]. The difficulty arises particularly in the manufacture of FRP parts of large size and/or complex curvature. Furthermore, while the RTM cycle only requires simple perimeter clamps as peripheral equipment, I/C-LCM demands the equivalent of a hydraulic press. Nevertheless, a significant advantage of I/C-LCM can be that it allows notably faster manufacturing times as compared to RTM [52]. Thus, while determining a preferred manufacturing technique for a particular FRP composite part, the manufacturer is expected to carefully explore both processes in terms of practicality, setup and running costs, as well as the production rate. To this end, the optimization task corresponding to each manufacturing technique may be formulated as [54],

$$minimize\ (Mold\ Filling\ Time, Peak\ Internal\ Force),$$

$$subject\ to: F_{fluid} + F_{fibre} \leq F_{capacity}. \qquad (7)$$

Here, F_{fluid} represents the internal force originating from resin pressure, F_{fibre} is the response of the compressed fibrous reinforcement, and $F_{capacity}$ is the maximum allowable internal force as dictated by the availability of peripheral equipment. The values of F_{fluid}, F_{fibre}, *mold filling time*, and *peak internal force* (for a given combination of input design variables) are obtained via a process simulation software [56], [57] which evaluates a set of partial differential equations (see Appendix) that govern the complex non-isothermal and chemically reactive resin flow through porous media. As is well known, such simulations are generally computationally time consuming, often taking several minutes for a single evaluation of sufficiently high fidelity. This feature presents a considerable roadblock to efficient optimization.

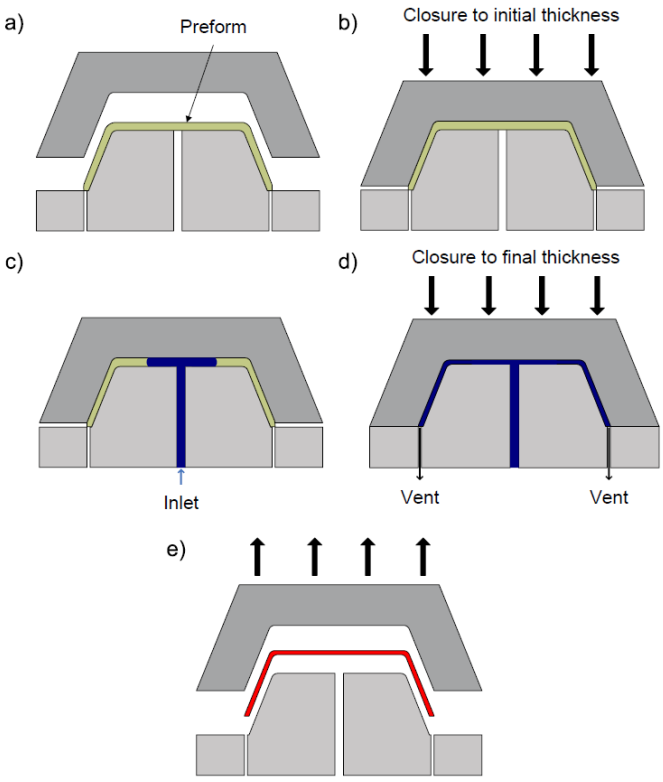

Fig. 7. Workflow of I/C-LCM [52]: (a) placement of preform, (b) partial mold closure, (c) high pressure liquid resin injection, (d) *in situ* mold closure to desired part thickness, (e) extraction of final part.

1) The potential utility of evolutionary multitasking in engineering design

The described composites manufacturing problem provides an ideal setting for us to demonstrate the real-world utility of exploiting knowledge overlaps in evolutionary multitasking. Since RTM and I/C-LCM belong to the same family of rigid-tool LCM processes and have several recurring design variables, one intuitively expects there to exist some underlying synergy between the two techniques, especially when dealing with the manufacture of the same FRP composite part. In particular, referring to Fig. 8, the common knowledge is expected to be (primarily) contained in the intersecting region of the phenotype space, i.e., in $\boldsymbol{x}_{overlap}$ = (P_{inj}, T_{mould}, T_{resin}). Note that despite the phenotypic overlap, the variables P_{inj}, T_{mould}, and T_{resin} need not assume identical numeric values with respect to both tasks.

Situations such as the above, where distinct optimization tasks have several overlapping design variables, frequently occur in the conceptualization phase of most product/process development cycles [58]. Taking the current example of manufacturing process design, the fabrication of a product typically follows careful selection of a process that minimizes capital layout on equipment and running costs while maximizing throughput [59]. To this end, the set of all candidate processes must be thoroughly explored before making a (delayed) decision about the most suitable one [60]. Since different processes often exhibit several recurrent design variables, the designer instinctively anticipates there to exist some useful or adaptable knowledge that may be common across various design exploration exercises. Thus, in such

situations, instead of having to investigate each candidate manufacturing process serially with manual knowledge adaptation, evolutionary multitasking provides the scope for knowledge refinement and exchange to occur spontaneously in the form of implicit genetic transfer. The resultant stimulus provided to the evolutionary search can significantly shorten the often exorbitantly time consuming design stage.

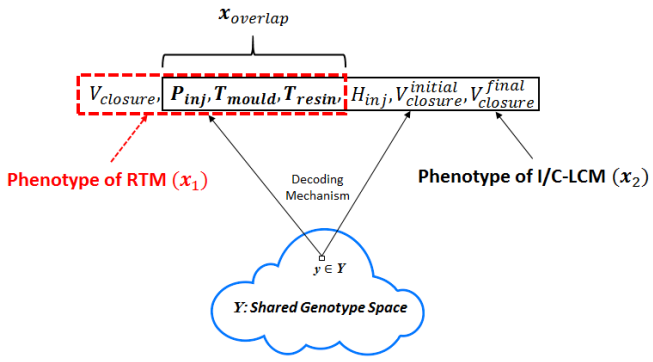

Fig. 8. A summary of the 2-task environment in composites manufacturing. Here, $\boldsymbol{x}_1$ represents a solution in the phenotype space of the RTM cycle, and $\boldsymbol{x}_2$ represents a solution in the phenotype space of the I/C-LCM cycle. The interesting aspect of this multitasking instance is that there exists an overlap in the phenotype space, giving rise to synergies that can be exploited during multitasking.

Through a comparative study between MO-MFEA and NSGA-II for the described composites manufacturing problem, we present a substantiation of our claims.

C. Computational Results

Herein, we carry out the simulation-based exploration (via multi-objective optimization) of two candidate manufacturing processes for a FRP composite disc (or plate). The plate diameter is 1 m, with a central injection hole of 2 cm (as shown in Fig. 9). The desired part thickness is 0.75 cm. A glass-fiber reinforced epoxy resin matrix is considered, with fiber volume fraction of the finished part to be 50%. Further, it is prescribed that the availability of peripheral equipment restricts $F_{capacity}$ to 30 tons (3E+05 N). For full details of material properties required for numerical simulations the reader is referred to [59]. We do not reproduce the data in this paper for the sake of brevity.

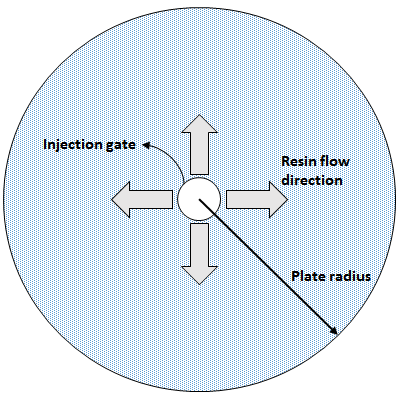

Fig. 9. Plan of a FRP composite disc with a circular injection gate.

TABLE II

DESIGN VARIABLES AND EXTENT OF THE DESIGN SPACE FOR THE RTM AND I/C-LCM MANUFACTURING CYCLES

Design Variable	Lower Bound	Upper Bound
$V_{closure}^{initial/final}$	1 mm/min	10 mm/min
P_{inj}	1 MPa	10 MPa
T_{mould}	293 K	348 K
T_{resin}	293 K	348 K
H_{inj}	0.8 cm	1 cm

The extent of the design space for all design variables are provided in Table II (the table includes variables corresponding to the RTM cycle and the I/C-LCM cycle. In order to account for the computational expense of the numerical simulations, we consider a small population of N = 50 individuals (in the MO-MFEA and the NSGA-II) which are evolved for a limited budget of 5000 solution evaluations. Recall that the MO-MFEA causes the evaluation budget to be (roughly) equally shared among constitutive optimization tasks, thereby limiting the available evaluations to approximately 2500 per task in this 2-factorial environment. In order to compare the convergence characteristics of the MO-MFEA and NSGA-II, we make use of the normalized *Hypervolume* (HV) metric [45], as described in Fig. 10. The reference point is set to (220 sec, 33 tons) and the prior estimate of the ideal point is (18 sec, 13.5 tons). The IGD metric is no longer preferred in this case as the true PF is not known beforehand for real-world problems. Note that, in contrast to IGD, HV is expected to increase gradually for an evolving population, with higher HV values indicating superior convergence and diversity of the population. The values of the normalized HV metric reported hereafter are averages across 3 independent runs of the optimizers.

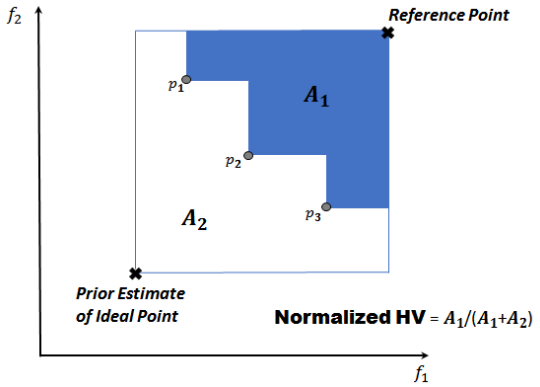

Fig. 10. The normalized Hypervolume metric.

Figs. 11 and 12 represent the evolution of the HV for the case of RTM and I/C-LCM, respectively. As is clear from both figures, when the two manufacturing processes are explored simultaneously in a single multitasking environment

(as is the case in the MO-MFEA), the overall convergence characteristics can be significantly improved as opposed to tackling a single optimization task at a time with NSGA-II. Since the MO-MFEA has several identical features as the NSGA-II, the improved performance during multitasking can be credited majorly to the effective utilization of the underlying synergies between the two tasks. The resultant impetus to the evolutionary search enables higher HV values to be achieved within significantly fewer number of solution evaluations, as is revealed in Figs. 11 and 12. Thus, in typical engineering design exercises involving time consuming computational simulations, the potential for multitasking can greatly accelerate the design stage.

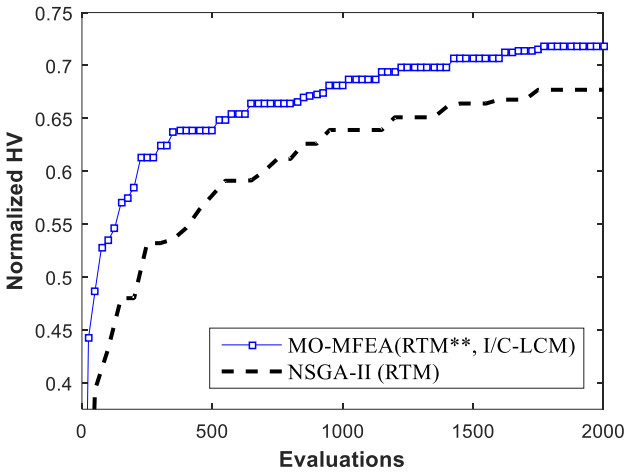

Fig. 11. Comparing the evolution of the HV metric for the case of RTM. Note that the curve corresponding to a particular task in MO-MFO is denoted by appending asterisks (**) to the task label.

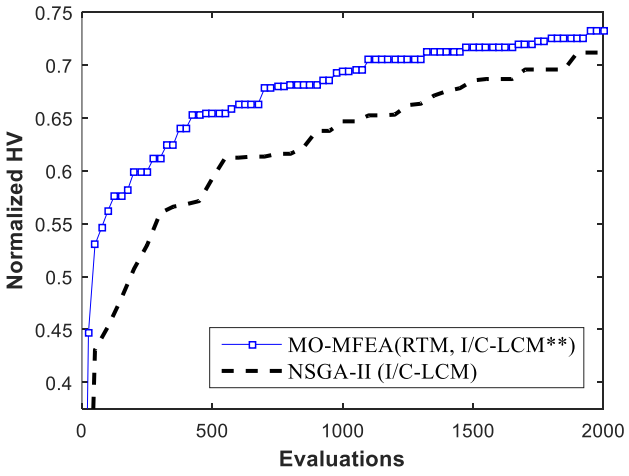

Fig. 12. Comparing the evolution of the HV metric for the case of I/C-LCM. Note that the curve corresponding to a particular task in MO-MFO is denoted by appending asterisks (**) to the task label.

Finally, to conclude the real-world case study, we present the approximate PFs of the RTM and I/C-LCM cycle, as are obtained by the MO-MFEA, in Fig. 13. We find that the I/C-LCM cycle does indeed provide the scope for considerably faster mold filling (thereby reducing the overall manufacturing time). On the other hand, the internal force that may be achieved during RTM is lower than what is achievable during I/C-LCM. This indicates that the capital layout on equipment and running cost of the RTM cycle may be lower. Armed with such information, the designer/manufacturer can make an informed *a posteriori* decision regarding the most suitable manufacturing process; that which presents an ideal balance between practicality, maximizing throughput, and minimizing cost of layout and running peripheral equipment.

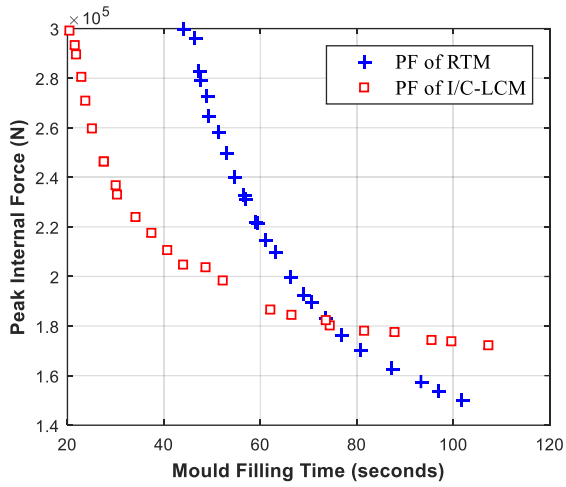

Fig. 13. Approximate PFs of the RTM and I/C-LCM cycles.

VI. Conclusions and Directions for Future Research

In this paper, we have presented an amalgamation of multi-objective optimization with the notion of evolutionary multitasking. We label the resultant paradigm as *Multi-Objective Multifactorial Optimization* (MO-MFO), where each MOOP in a multitasking environment acts as an additional *factor* influencing the evolution of a single population of individuals. The mechanics of evolution are carried out via a new multitasking engine labelled as the *Multi-Objective Multifactorial Evolutionary Algorithm* (MO-MFEA). It is contended that the process of multitasking in optimization opens doors to the so far underexplored possibility of harnessing the underlying commonalities between different optimization tasks. In fact, evolutionary multitasking is deemed to be particularly well suited for automated transfer, adaptation, and refinement of knowledge (in the form of encoded genetic material), without the need for any external human intervention.

The claims stated above have been substantiated through computational studies on some benchmark MOOPs as well as on a real-world manufacturing process design problem from the composites industry. It is found that the phenomenon of implicit genetic transfer in multitasking can exploit the presence of transferrable knowledge between optimization tasks, thereby facilitating improved performance characteristics for multiple tasks at the same time. In the field of complex engineering design, which typically involves computationally expensive simulation-based optimization, the

impetus provided to evolutionary search during multitasking can considerably shorten the exorbitantly time consuming design process.

Although the results presented in the paper are encouraging and demonstrate the potential implications of evolutionary multitasking towards real-world problem solving, we acknowledge that the concept gives rise to several research questions that must be thoroughly studied in the future. For instance, it must be noted that evolutionary multitasking cannot always be expected to provide performance improvements. As has been discussed in the paper, while some genetic transfer may be useful, others may impede the search. The chance of predominantly negative transfer may be high if the unification scheme, decoding mechanism, and genetic operators are not appropriately designed to comply with the features of the underlying optimization tasks. While even the simple schemes used in this study have shown promising performance enhancements, there is indeed the scope for development of more advanced techniques that are better suited for multitasking under different circumstances. In particular, the design of unified representations and decoding mechanisms which incorporate domain knowledge and/or account for the associations between variables of different tasks are considered invaluable for consistently effective evolutionary multitasking.

Finally, notice that in the real-world application presented in this paper, an explicit overlap is known to exist between the phenotype spaces of the optimization tasks, which can be exploited during multitasking. However, our methods can in principle be utilized for a variety of other real-world instances where the amount of overlap may be lower or less apparent. Admittedly, this may at times reduce the success rate of multitasking. However, according to the study carried out in [21], in many cases the multifactorial evolutionary algorithm was found to successfully harness hidden complementarities even among cross-domain optimization tasks (i.e., including continuous and combinatorial problems). From the standpoint of conceived cloud-based on-demand machine learning and optimization services [61]-[63], the implication of such a feature calls for further research attention.

APPENDIX

The filling of the mold with a liquid resin in a generic composites manufacturing process is governed by the following PDEs:

$$\nabla . \left(h \frac{\boldsymbol{K}}{\mu} \nabla p \right) = \frac{\partial h}{\partial t}, \quad (8)$$

$$\rho\, C_p \frac{\partial T}{\partial t} + \rho_r C_{pr} (\boldsymbol{u} . \nabla T) = \nabla . (k \nabla T) + (1 - V_f) \cdot \dot{H}, \quad (9)$$

$$\varphi \frac{\partial \alpha}{\partial t} + \boldsymbol{u} . \nabla \alpha = (1 - V_f) \cdot R_\alpha . \quad (10)$$

Here, Eq. 8 represents Darcy's Law that governs fluid flow in porous media, h is the thickness of the mold cavity, p is the local resin pressure, $\boldsymbol{K}$ is the reinforcement permeability tensor, t is the time, and $\partial h/\partial t$ represents the speed of mold closure. Note that while $\partial h/\partial t$ is zero throughout the RTM cycle, it is strictly negative for the I/C-LCM cycle during *in situ* mold compression. The viscosity μ of the resin is a function of the local temperature T and the degree of resin cure α. The relation may be captured by the following widely used rheological model,

$$\mu = A_\mu e^{E_\mu / R\,T} \left(\frac{\alpha_g}{\alpha_g - \alpha} \right)^{a + b\alpha}, \quad (11)$$

where α_g is the degree of cure at which resin gel conversion occurs, R is the universal gas constant, E_μ is the activation energy, and A_μ, a and b are other experimentally determined constants.

Eq. 9 is a lumped energy equation which governs the temperature distribution within the mold. The material properties ρ, C_p, and k represent the average density, specific heat capacity, and thermal conductivity of the resin-fiber system, respectively. Further, $\boldsymbol{u}$ is the volume averaged resin flow velocity, V_f is the fiber volume fraction, and $\dot{H}$ is a source term representing the thermal energy generated by the resin during its exothermic polymerization reaction.

Finally, Eq. 10 models how the degree of resin conversion varies in the part during filling. Therein, R_α $(= d\alpha/dt)$ represents the rate of resin polymerization. Kamal and Sourour [64] proposed the following general model which is widely used to describe the polymerization reaction,

$$\frac{d\alpha}{dt} = \left(A_1 . \, e^{(-E_1/R\,T)} + A_2 . \, e^{(-E_2/R\,T)} . \alpha^{m_1} \right) . (1 - \alpha)^{m_2}, \quad (12)$$

where A_1, A_2, E_1, E_2, m_1, and m_2 are experimentally determined constants.

For complete details on the material properties and values of empirical constants used in the composites manufacturing case study, the reader is referred to [59].

REFERENCES

[1] M. Ehrgott. *Multicriteria Optimization*. Springer, 2005.

[2] V. A. Shim, K. C. Tan, and H. J. Tang, "Adaptive memetic computing for evolutionary multi-objective optimization", *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 610-621, 2014.

[3] L. Ke, Q. Zhang, and R. Battiti, "Hybridization of Decomposition and Local Search for Multiobjective Optimization", *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1808 - 1820, 2014.

[4] E. G. Talbi, S. Mostaghim, T. Okabe, H. Ishibuchi, G Rudolph, and C. A. Coello Coello, "Parallel approaches for multi-objective optimization," in *Muliobjective Optimization*, Springer Berlin Heidelberg, 2008, pp. 349-372.

[5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

[6] W. F. Leong and G. G. Yen, "PSO-based multi-objective optimization with dynamic population size and adaptive local archives," *IEEE Trans. SMC Part B: Cybernetics*, vol. 38, no. 5, pp. 1270-1293, 2008.

[7] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. SMC Part C*, vol. 28, no. 3, pp. 392-403, 1998.

[8] K. C. Tan, C. K Goh, Y. J. Yang, and T. H. Lee, "Evolving better population distribution and exploration in evolutionary multi-objective optimization", *European Journal of Operational Research*, vol. 171, no. 2, pp. 463-495, 2006.

[9] K.C. Tan, T. H. Lee, D. Khoo, and E. F. Khor, "A multi-objective evolutionary algorithm toolbox for computer-aided multi-objective optimization", *IEEE Transactions on Systems, Man and Cybernetics: Part B (Cybernetics)*, vol. 31, no. 4, pp. 537-556, 2001.

[10] S. W. Jiang, J. Zhang, Y. S. Ong, A. N. S. Zhang, and P. S. Tan, "A Simple and Fast Hypervolume Indicator-based Multiobjective Evolutionary Algorithm", *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2202-2213, 2014.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-2," *IEEE Trans. Evo. Comp*., vol. 6, no. 2, pp. 182-197, 2002.

[12] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, Part I: Solving problems with box constraints" *IEEE Trans. on Evo. Comp*., vol. 18, no. 4, pp. 577-601, 2013.

[13] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach" *IEEE Trans. on Evo. Comp*., vol. 18, no. 4, pp. 602-622, 2013.

[14] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *in Proc. EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., Athens, Greece, Sept. 2001.

[15] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition", *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

[16] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-Based Selection for Decomposition Multiobjective Optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2076-2088, 2015.

[17] S. Jiang and S. Yang, "An Improved Multiobjective Optimization Evolutionary Algorithm Based on Decomposition for Complex Pareto Fronts," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421-437, 2016.

[18] J. Bader, E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45-76, 2011.

[19] T. Back, U. Hammel, and H. P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. on Evo. Comp.*, vol. 1, no. 1, pp. 3-17, 1997.

[20] M. Srinivas and L.M. Patnaik, "Genetic Algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17-26, June 1994.

[21] A. Gupta, Y. S. Ong, and L. Feng, "Multifactorial Evolution: Toward Evolutionary Multitasking," *IEEE Trans. on Evo. Comp*., Accepted, 2015.

[22] J. Rice, C. R. Cloninger, and T. Reich, "Multifactorial inheritance with cultural transmission and assortative mating. I. Description and basic properties of the unitary models," *Am. J. Hum. Genet*., vol. 30, pp. 618-643, 1978.

[23] C. R. Cloninger, J. Rice, and T. Reich, "Multifactorial inheritance with cultural transmission and assortative mating. II. A general model of combined polygenic and cultural inheritance," *Am. J. Hum. Genet*., vol. 31, pp. 176-198, 1979.

[24] L. L. Cavalli-Sforza and M. W. Feldman, "Cultural vs Biological inheritance: Phenotypic transmission from parents to children (A theory of the effect of parental phenotypes on children's phenotypes)," *Am. J. Hum. Genet*., vol. 25, 618-637, 1973.

[25] R. Dawkins. *The Selfish Gene*. Oxford University Press, 1976.

[26] X. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Trans. Evo. Comp*., vol. 15, no. 5, pp. 591-606, Oct. 2011.

[27] Y. S. Ong, M. H. Lim, and X. S. Chen, "Research Frontier: Memetic Computation – Past, Present & Future," *IEEE Comp. Intel. Mag*., vol. 5, no. 2, pp. 24-36, 2010.

[28] Y. S. Ong and A. J. Keane, "Meta-Lamarckian learning in memetic algorithms," *IEEE Trans. Evo. Comp*., vol. 8, no. 2, pp. 99-110, 2004.

[29] S. Bechikh, A. Chaabani, and L. B. Said, "An efficient chemical reaction optimization algorithm for multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2051-2064, 2015.

[30] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modelling," *IEEE Trans. Evo. Comp.*, vol. 19, no. 6, pp. 838-856, 2015.

[31] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comp. Methods in Appl. Mech. and Engg*., vol. 186, pp. 311-338, 2000.

[32] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: Analysing the state-of-the-art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125-147, 2000.

[33] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1-16, 1995.

[34] R. Mills, T. Jansen, and R. A. Watson, "Transforming evolutionary search into higher-order evolutionary search by capturing problem structure," *IEEE Trans. Evo. Comp*., vol. 18, no. 5, pp. 628-642, 2014.

[35] M. Iqbal, W. N. Browne, and M. Zhang, "Reusing building blocks of extracted knowledge to solve complex, large-scale Boolean problems," *IEEE Trans. Evo. Comp*., vol. 18, no. 4, pp. 465-480, 2013.

[36] J. Goncalves and M. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *J. Heuristics*, vol. 17, no. 5, pp. 487–525, 2011.

[37] J. C. Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA J. on Computing*, 6:154–160, 1994.

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

[39] K. Deb, S. Karthik, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," *GECCO'07*, pp. 1187-1194.

[40] S. J. Pan, Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge & Data Engg*., vol. 22, no. 10, pp. 1345-1359, 2009.

[41] L. Feng, Y. S. Ong, M. H. Lim, and I. W. Tsang, "Memetic Search with Inter-Domain Learning: A Realization between CVRP and CARP," *IEEE Trans. Evo. Comp.*, vol. 19, no. 5, pp. 644-658, 2014.

[42] L. Feng, Y. S. Ong, A. H. Tan, and I. W. Tsang, "Memes as building blocks: a case study on evolutionary optimization + transfer learning for routing problems," *Memetic Computing*, vol. 7, no. 3, pp. 159-180, 2015.

[43] K. Deb and D. Deb, *Analyzing mutation schemes for real-parameter genetic algorithms*, KanGAL Report No. 2012016.

[44] E. A. Williams and W. A. Crossley, "Empirically-derived population size and mutation rate guidelines for a genetic algorithm with uniform crossover," *Soft computing in engineering design and manufacturing*, pp. 163-172, 1998.

[45] S. Jiang, Y. S. Ong, J. Zhang, and L. Feng, "Consistencies or Contradictions of Performance Metrics in Multiobjective Optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2391 - 2404, 2014.

[46] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multi-objective evolutionary algorithms: Empirical results," *Evo. Comp*., vol. 8, no. 2, pp. 173-195, 2000.

[47] K. Deb and T. Goel, "Controlled elitist non-dominated sorting genetic algorithms for better convergence," *Evolutionary Multi-Criterion Optimization*, *LNCS*, vol. 1993, pp. 67-81, 2001.

[48] L. A. Rastrigin, *Systems of extreme control*. 1974.

[49] A. O. Griewank, "Generalized decent for global optimization," *J. Opt. Th. Appl.*, vol. 34, no. 1, pp. 11-39, 1981.

[50] D. H. Ackley, *A connectionist machine for genetic hillclimbing*. Boston: Kluwer Academic Publishers, 1987.

[51] K. Deb, A. P. Mathur, T. Meyarivan, *Constrained test problems for multi-objective evolutionary optimization*, Tech. Report No. 200002, Kanpur Genetic Algorithms Lab, IIT Kanpur, India.

[52] W. A. Walbran. *Experimental validation of local and global force simulations for rigid tool liquid composite moulding processes*. PhD Thesis – University of Auckland, 2011.

[53] A. Gupta. *Numerical modelling and optimization of non-isothermal, rigid tool liquid composite moulding processes*. PhD Thesis – University of Auckland, 2013.

[54] S. Hsu, M. Ehrgott, and P. Kelly, "Optimization of mould filling parameters of the compression resin transfer moulding process," *Proceedings of the 45th Annual Conference of the ORSNZ*, 2010.

[55] S. G. Advani and K. T. Hsiao (Eds.). *Manufacturing techniques for polymer matrix composites (PMCs)*. Elsevier, 2012.

[56] P. A. Kelly and S. Bickerton, "A comprehensive filling and tooling force analysis for rigid mould LCM processes," *Composites Part A*, vol. 40, no. 11, pp. 1685-1697, 2009.

[57] A. Gupta and P. A. Kelly, "Optimal Galerkin finite element methods for non-isothermal liquid composite moulding process simulations," *International Journal of Heat and Mass Transfer*, vol. 64, pp. 609-622, 2013.

[58] G. Avigad and A. Moshiaov, "Interactive evolutionary multiobjective search and optimization of set-based concepts," *IEEE Trans. SMC Part B (Cybernetics)*, vol. 39, no. 4, pp. 1013-1027, 2009.

[59] A. Gupta, P. A. Kelly, S. Bickerton, and W. A. Walbran, "Simulating the effect of temperature elevation on clamping force requirements during rigid-tool liquid composite moulding processes," *Composites Part A*, vol. 43, no. 12, pp. 2221-2229, 2012.

[60] G. Avigad and A. Moshaiov, “Set-based concept selection in multi-objective problems with delayed decisions,” *Journal of Engineering Design*, vol. 21, no. 6, pp. 619-646, 2010.
[61] Y. S. Ong and A. Gupta. “Evolutionary Multitasking: A Computer Science View of Cognitive Multitasking,” *Cognitive Computation*, pp. 1-18, 2016.
[62] I. Arnaldo, K. Veeramachaneni, A. Song, and U. -M. O'Reilly, “Bring Your Own Learner: A Cloud-Based, Data-Parallel Commons for Machine Learning,” *IEEE Computational Intelligence Magazine*, vol. 10, no. 1, pp. 20-32, 2015.
[63] M. Guzek, P. Bouvry, E.-G. Talbi, “A Survey of Evolutionary Computation for Resource Management of Processing in Cloud Computing,” *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 53-67, 2015.
[64] M. Kamal and S. Sourour, “Kinetics and thermal characterization of thermoset resin,” *Polymer Engineering and Science*, vol. 13, no. 1, pp. 59-64, 1973.