On the Emerging Notion of Evolutionary Multitasking: A Computational Analog of Cognitive Multitasking

Abhishek Gupta, Bingshui Da, Yuan Yuan and Yew-Soon Ong

Abstract Over the past decades, Evolutionary Computation (EC) has surfaced as a popular paradigm in the domain of computational intelligence for global optimization of complex multimodal functions. The distinctive feature of an Evolutionary Algorithm (EA) is the emergence of powerful implicit parallelism as an offshoot of the simple rules of population-based search. However, despite the known advantages of implicit parallelism, it is interesting to note that EAs have almost exclusively been developed to solve only a single optimization problem at a time; seldom has any effort been made to multitask, i.e., to tackle multiple self-contained optimization problems concurrently using the same population of evolving individuals. To this end, inspired by the remarkable ability of the human brain to perform multiple tasks with apparent simultaneity, we present *evolutionary multitasking* as an intriguing direction for EC research. In particular, the paradigm opens doors to the possibility of autonomously exploiting the underlying complementarities between separate (but possibly similar) optimization exercises through the process of *implicit genetic transfer*, thereby enhancing productivity in decision making processes via accelerated convergence characteristics. Along with the design of an appropriately unified solution representation scheme, we present the outline of a recently proposed algorithmic framework for effective multitasking. Thereafter, the efficacy of the approach is substantiated through a series of practical examples in continuous and discrete optimization that highlight the real-world utility of the paradigm.

A. Gupta (✉) · B. Da · Y. Yuan · Y.-S. Ong
School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore
e-mail: abhishekg@ntu.edu.sg

B. Da
e-mail: DA0002UI@e.ntu.edu.sg

Y. Yuan
e-mail: yuanyuan@ntu.edu.sg

Y.-S. Ong
e-mail: asysong@ntu.edu.sg

S. Bechikh et al. (eds.), *Recent Advances in Evolutionary Multi-objective Optimization*, Adaptation, Learning, and Optimization 20,
DOI 10.1007/978-3-319-42978-6_5

Keywords Evolutionary multitasking · Multi-objective optimization · Memetic computation

1 Introduction

One of the most astonishing aspects of human cognition is its ability to manage and execute multiple tasks with what appears to be apparent simultaneity. It is recognized that in this fast-paced, technologically driven world that we live in, the explosion in volume and variety of incoming information streams presents unprecedented opportunity, tendency, and (even) the need to effectively multitask. Merely a fleeting glance at the world around us reveals the ubiquity of supposed cognitive multitasking. From relatively straightforward examples, such as phoning while walking, to more complex ones, such as media multitasking, the human brain has shown notable adaptability to multitask settings. In fact, it is generally acknowledged that multitasking is perhaps the only way to fit in all our priorities into increasingly busy schedules, albeit at the (often tolerable) cost of a marginal drop in the quality of output achieved. Thus, it is not unnatural to expect the pursuit of intelligent systems and algorithms capable of effective multitasking to gain popularity among scientists and engineers who are constantly aiming for enhanced productivity in a world that routinely presents a multiplicity of complex challenges.

It is noted that a major criticism leveled against cognitive multitasking originates from an observed *switching cost* during which the brain attempts to overcome the interference between tasks and adjusts to the new task [1]. Thus, while constantly switching between competing tasks, an individual may often experience slower response times, degraded performance, and/or increased error rates [2]. In this regard, while developing computational analogues of multitasking, it is observed that modern-day computers are in the most part free from any significant switching cost while handling multiple tasks at once. This observation forms grounds for our contention that an artificial (computational) multitasking engine may be capable of retaining many of the advantages of cognitive multitasking, while effectively overcoming its potential perils.

In the field of computational intelligence, Evolutionary Algorithms (EAs) constitute a family of stochastic optimizers that are inspired by Darwinian principles of natural selection [3–5]. The increasing popularity of EAs as a mainstay of optimization in science, operations research, and engineering is largely due to the emergent properties of implicit parallelism of population-based search [6], which circumvents the need for derivative-based techniques that impose continuity and differentiability requirements on objective function landscapes. In fact, it is largely due to the efficient exploitation of implicit parallelism that Multi-objective Evolutionary Algorithms (MOEAs) have rapidly gained in popularity in recent decades, enabling synchronous convergence to a diverse set of near optimal trade-off points [7–9]. Encouraged by this observation, a central goal of the present proposition is to further leverage upon the known power of implicit parallelism, thereby establishing a new niche for EAs that undeniably sets them apart from existing mathematical optimization procedures. In particular, we investigate the potential utility of EAs towards *multitask optimization*, i.e., the solution of multiple self-contained (but possibly similar) optimization tasks at the same time using a single population of evolving individuals. While the proposition bears resembling conceptual motivation to the field of multitask learning [10, 11], it operates from the standpoint of nature-inspired computing, facilitating implicit information exchange across different numerical optimization tasks. To elaborate, we contend that useful inductive biases or some form of knowledge overlap may exist in the evolutionary search of one or more optimization tasks that lie outside the self-contained scope of a particular problem of interest. Neglecting this information, as is typically the case in tabula rasa optimization, may be deemed highly counterproductive, especially given the increasing complexity of real-world problems. In such scenarios, *evolutionary multitasking* provides the scope for autonomously exploiting the complementarities in an implicit manner (through the process of *genetic transfer*), and consequently accelerating convergence characteristics by circumventing several (often impeding) function evaluations [12–14].

For a more detailed illustration of the various notions discussed heretofore, the remainder of this chapter is organized as follows. In Sect. 2, we introduce the preliminaries of multitask optimization. Following [12], we hereafter label the paradigm as *multifactorial optimization* (MFO) in order to emphasize that each task presents an additional *factor* influencing the evolution of a single population. Further, we highlight the key conceptual distinction between multitasking and multi-objective optimization in order to address several queries arising in this regard. In Sect. 3, we present the *Multifactorial Evolutionary Algorithm* (MFEA) from [12], an approach that draws inspiration from bio-cultural models of multifactorial inheritance [15–18]. The means by which the MFEA facilitates knowledge transfer across tasks is also briefly discussed therein. Thereafter, Sect. 4 contains recent case studies for a variety of practical applications of multitasking, including examples in continuous and discrete optimization. In essence, it is reasoned that there exist numerous promising opportunities for MFO in real-world problems, which encourages future research efforts in this direction. Finally, Sect. 5 summarizes the chapter, highlighting important research questions brought to the table by the promising future prospects of multitask optimization.

2 Preliminaries

Consider a hypothetical situation wherein K self-contained optimization tasks are to be performed concurrently. Without loss of generality, all tasks are assumed to be minimization problems. The j-th task, denoted T_j, is considered to have a search space X_j on which the objective function is defined as $F_j : \boldsymbol{X}_j \rightarrow \mathbb{R}$. In addition, each task may be constrained by several equality and/or inequality conditions that must be satisfied for a solution to be considered feasible. In such a setting, we define MFO as an evolutionary multitasking paradigm that aims to simultaneously navigate the design space of all tasks, constantly building on the implicit parallelism of population-based search so as to rapidly deduce $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{K-1}, \boldsymbol{x}_K\} =$

$argmin\{F_1(\boldsymbol{x}), F_2(\boldsymbol{x}), \ldots, F_{K-1}(\boldsymbol{x}), F_K(\boldsymbol{x})\}$, where $\boldsymbol{x}_j$ is a feasible solution in $\boldsymbol{X}_j$. As suggested by the nomenclature, herein each F_j is treated as an additional factor influencing the evolution of a single population of individuals. For this reason, the composite problem may also be referred to as a K-factorial problem.

While designing evolutionary solvers for MFO, it is necessary to formulate a general technique for comparing population members in a multitasking environment. To this end, we first define a set of properties for every individual p_i, where $i \in \{1, 2, |P|\}$, in a population P. Note that the individuals are encoded in a unified search space $\boldsymbol{Y}$ encompassing $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K$, and can be decoded into a task-specific solution representation with respect to each of the K optimization tasks. The decoded form of p_i can thus be written as $\{\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots, \boldsymbol{x}_{iK}\}$, where $\boldsymbol{x}_{i1} \in \boldsymbol{X}_1$, $\boldsymbol{x}_{i2} \in \boldsymbol{X}_2$, ..., and $\boldsymbol{x}_{iK} \in \boldsymbol{X}_K$.

- *Definition 1*(*Factorial Cost*): For a given task T_j, the *factorial cost* Ψ_{ij} of individual p_i is given by $\Psi_{ij} = \lambda \cdot \delta_{ij} + F_{ij}$; where λ is a large penalizing multiplier, F_{ij} and δ_{ij} are the objective value and the total constraint violation, respectively, of p_i with respect to T_j. Accordingly, if p_i is feasible with respect to T_j (zero constraint violation), we have $\Psi_{ij} = F_{ij}$.
- *Definition 2*(*Factorial Rank*): The *factorial rank* r_{ij} of p_i on task T_j is simply the index of p_i in the list of population members sorted in ascending order with respect to factorial cost Ψ_{ij}.

Note that, while assigning factorial ranks, whenever $\Psi_{1j} = \Psi_{2j}$ for a pair of individuals p_1 and p_2, the parity is resolved by random tie-breaking.

- *Definition 3*(*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 K tasks then $\tau_i = argmin_j\{r_{ij}\}$, where $j \in \{1, 2, \ldots, K\}$.
- *Definition 4*(*Scalar Fitness*): The *scalar fitness* of p_i in a multitasking environment is given by $\varphi_i = 1/r_{iT}$, where $T = \tau_i$. Notice that $max\{\varphi_i\} = 1$.

Once the fitness of every individual has been scalarized according to Definition 4, performance comparison can then be carried out in a straightforward manner. For example, individual p_1 will be considered to dominate individual p_2 in multifactorial sense simply if $\varphi_1 > \varphi_2$.

It is important to note that the procedure described heretofore for comparing individuals is not absolute. As the factorial rank of an individual, and implicitly its scalar fitness, depends on the performance of every other individual in the population, the comparison is in fact population dependent. Nevertheless, the procedure guarantees that if an individual p^* uniquely attains the global optimum of any task then $\varphi^* = 1$, which implies that $\varphi^* \geq \varphi_i$ for all $i \in \{1, 2, \ldots, |P|\}$. Therefore, it can be said that the proposed technique is indeed consistent with the ensuing definition of multifactorial optimality.

- *Definition 5*(*Multifactorial Optimality*): An individual p^* is considered to be optimum in multifactorial sense if there exists at least one task in the K-factorial environment which it globally optimizes.

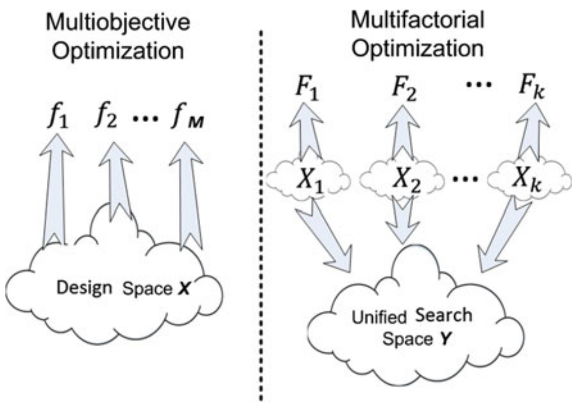

Fig. 1 Multi-objective optimization typically comprises a single design space encompassing all objective functions. On the other hand, multitask optimization unifies (into $\boldsymbol{Y}$) multiple heterogeneous design spaces belonging to distinct tasks [13]

2.1 Multitask Versus Multi-objective Optimization

Since multitask and multi-objective optimization are both concerned with processing a set of objective functions, a conceptual overlap may be seen to exist between them. However, it must be observed that there exists a vital difference between the fundamental principles of the two paradigms. While MFO aims to *leverage upon the implicit parallelism of population-based search to exploit the underlying commonalities and/or complementarities between multiple separate (but possibly similar) optimization tasks*, the formulation of a multi-objective optimization problem and its associated solution algorithms (such as any MOEA) attempt to effectively resolve conflicts among competing objectives of *the same task*. An illustration summarizing the statement is depicted in Fig. 1. The key ingredient distinguishing the two paradigms is the simultaneous existence of multiple heterogeneous design spaces in the case of multitasking, each corresponding to a distinct task. On the other hand, for the case of multi-objective optimization, there typically exists a single design space for a given task of interest, with all objective functions depending on variables contained within that space. Furthermore, note that a multitasking environment could potentially include a multi-objective optimization task as one among many other concurrent tasks, which highlights the greater generality of the proposed paradigm.

3 Multifactorial Evolution: A Framework for Effective Multitasking

In this section we describe the Multifactorial Evolutionary Algorithm (MFEA), an effective multitasking framework that draws upon the bio-cultural models of multifactorial inheritance [15, 16]. As the workings of the approach are based on the transmission of biological as well as cultural building blocks from parents to their

Algorithm 1 Pseudocode of the MFEA

Randomly generate n individuals in $\boldsymbol{Y}$ to form initial population P_0
for every p_j in P_0 **do**
  Assign skill factor $\tau_j = \mod(j, K) + 1$, for the case of K tasks
  Evaluate p_j for task τ_j only
end for
Compute scalar fitness φ_j for every p_j
Set $t = 0$
while stopping conditions are not satisfied **do**
  C_t = Crossover + Mutate(P_t)
  for every c_j in C_t **do**
    Determine skill factor $\tau_j \rightarrow$ Refer Algorithm 2
    Evaluate c_j for task τ_j only
  end for
  $R_t = C_t \cup P_t$
  Update scalar fitness of all individuals in R_t
  Select N fittest members from R_t to form P_{t+1}
  Set $t = t + 1$
end while

offspring, the MFEA is regarded as belonging to the realm of *memetic computation* [19, 20]—a field that has recently emerged as a successful computational paradigm synthesizing Darwinian principles of natural selection with the notion of memes, as put forth by Richard Dawkins, as the basic unit of cultural evolution [21]. An overview of the procedure is provided next.

As shown in Algorithm 1, the MFEA starts by randomly creating a population of n individuals in the unified search space $\boldsymbol{Y}$. Moreover, each individual in the initial population is pre-assigned a specific skill factor (see Definition 3) in a manner that guarantees every task to have uniform number of representatives. We would like to emphasize that the skill factor of an individual (i.e., the task with which the individual is associated) is viewed as a computational representation of its pre-assigned cultural trait. The significance of this step is to ensure that an individual is only evaluated with respect to a single task (i.e., only its skill factor) amongst all other tasks in the multitasking environment. Doing so is considered practical since evaluating every individual exhaustively for every task will generally be computationally demanding, especially when K (the number of tasks in the multitasking environment) becomes large. The remainder of the MFEA proceeds similarly to any standard evolutionary procedure. In fact, it must be mentioned here that the underlying genetic mechanisms may be borrowed from any of the plethora of population-based algorithms available in the literature, keeping in mind the properties and requirements of the multitasking problem at hand. The only significant deviation from a traditional approach occurs in terms of offspring evaluation which accounts for cultural traits via individual skill factors.

3.1 Offspring Evaluation in the MFEA

Following the memetic phenomenon of *vertical cultural transmission* [17–19], offspring in the MFEA experience strong cultural influences from their parents, in addition to inheriting their genes. In gene-culture co-evolutionary theory, vertical cultural transmission is viewed as a mode of inheritance that operates in tandem with genetics, and leads to the phenotype of an offspring being directly influenced by the phenotype of its parents. The algorithmic realization of the aforementioned notion is achieved in the MFEA via a *selective imitation strategy*. In particular, selective imitation is used to mimic the commonly observed phenomenon that offspring tend to imitate the cultural traits (i.e., skill factors) of their parents. Accordingly, in the MFEA, an offspring is only decoded (from the unified genotype space $\boldsymbol{Y}$ to a task-specific phenotype space) and evaluated with respect to a single task with which at least one of its parents is associated. As has been mentioned earlier, selective evaluation plays a role in managing the computation expense of the MFEA. A summary of the steps involved is provided in Algorithm 2.

Algorithm 2 Vertical cultural transmission via selective imitation

Consider offspring $c \in C_t$ where $c = \text{Crossover} + \text{Mutate}(p_1, p_2)$
1: Generate a random number $rand$ between 0 and 1
2: **if** $rand \leq 0.5$ **then**
    c imitates skill factor of p_1
3: **else**
    c imitates skill factor of p_2
4: **end if**

3.2 Search Space Unification and Cross-Domain Decoding Exemplars

The core motivation behind the evolutionary multitasking paradigm is the autonomous exploitation of known or latent commonalities and/or complementarities between distinct (but possibly similar) optimization tasks for achieving faster and better convergence characteristics. One of the possible means of harnessing the available synergy, at least from an evolutionary perspective, is through implicit genetic transfer during crossover operations. However, for the relevant knowledge to be transferred across appropriately, i.e., to ensure effective multitasking, it is pivotal to first describe a genotypic unification scheme that suits the requirements of the multitasking problem at hand. In particular, the unification serves as a higher-level abstraction that constitutes a *meme space*, wherein building blocks of encoded knowledge are processed and shared across different optimization tasks. This perspective is much in alignment with the workings of the human brain, where knowledge pertaining to different tasks are abstracted, stored, and re-used for relevant problem solving exercises whenever needed.

Unification implies that genetic building blocks [22] corresponding to different tasks are contained within a single pool of genetic material, thereby facilitating the MFEA to process them in parallel. To this end, assuming the search space dimensionality of the j-th optimization task (in isolation) to be D_j, a unified search space $\boldsymbol{Y}$ comprising K (traditionally distinct) tasks may be defined such that $D_{multitask} = max_j\{D_j\}$, where $j \in \{1, 2, \ldots, K\}$. In other words, while handling K optimization tasks simultaneously, the chromosome $\boldsymbol{y} \in \boldsymbol{Y}$ of an individual in the MFEA is represented by a vector of $D_{multitask}$ variables. While addressing the j-th task, we simply extract D_j variables from the chromosome and decode them into a meaningful solution representation for the underlying optimization task. In most cases, an appropriate selection of D_j task-specific variables from the list of $D_{multitask}$ variables is crucial for the success of multitasking. For instance, if two distinct variables belonging to two different tasks have similar phenotypic meaning, then they should intuitively be associated to the same variable in the unified search space $\boldsymbol{Y}$. On the other hand, in many naive cases where no a priori understanding about the phenotype space is available, simply extracting the *first* D_j variables from the chromosome can oftentimes be a viable alternative [12].

In what follows, we demonstrate how chromosomes in a unified genotype space can be decoded into meaningful task-specific solution representations when a *random-key unification scheme* [23] is adopted. According to the random-key scheme, each variable of a chromosome is simply encoded by a *continuous value* in the range [0, 1]. The salient feature of this representation is that it elegantly accommodates a wide variety of problems in continuous as well as discrete optimization, thereby laying the foundation for a cross-domain multitasking platform. Some decoding examples for continuous and popular instantiations of combinatorial optimization shall be discussed hereafter. At this juncture, it must however be emphasized that the concept of multitasking is not necessarily tied to cross-domain optimization. In fact, domain-specific schemes can indeed be used (often with greater success) when all constitutive tasks belong to similar domains.

3.2.1 Decoding for Continuous Optimization Problems

In the case of continuous optimization, decoding can be achieved in a straightforward manner by linearly mapping each random-key from the genotype space to the box-constrained phenotype space of the relevant optimization task [12].

3.2.2 Decoding for Discrete Sequencing Problems

In the domain of combinatorial optimization, sequencing problems include a variety of classical examples such as the Travelling Salesman (TSP), Job-Shop Scheduling (JSP), Quadratic Assignment (QAP), Vehicle Routing (VRP), etc. The common feature of these problems is that they involve the *ordering* of a finite set of distinct entities in a manner that optimizes a given objective function. The applicability of the real parameter random-key chromosome representation scheme to discrete problems of this kind was perhaps first investigated in [23]. In particular, it was observed that under any real-coded variation operation, the decoding procedure ensures feasibility of the generated offspring. This outcome is in contrast to domain-specific representations of sequencing problems wherein specially designed variation operators are needed to ensure offspring feasibility. As a consequence, the random-key representation has found notable interest over the past two decades in the field of operations research [24–26].

For an illustration of the decoding scheme, consider a case where 5 distinct entities are to be ordered optimally. To this end, a sample random-key chromosome in the MFEA may look like $\boldsymbol{y} = (0.1, 0.7, 0.2, 0.9, 0.04)$, such that the first entity is labeled as 0.1, the second entity is labeled as 0.7, the third is labeled as 0.2, and so on. Following the technique suggested in [23], the order of entities encoded by the chromosome $\boldsymbol{y}$ is given by the sequence $\boldsymbol{s} = (5, 1, 3, 2, 4)$. In other words, the sequence can be deduced simply by *sorting* the random-key labels in ascending order. Each entity is assigned an index in $\boldsymbol{s}$ that corresponds to the position of its label in the sorted list.

3.3 Implicit Knowledge Transfer in the MFEA

For any proposed unification scheme to be useful for multitasking, a matter of critical importance is the means of knowledge transfer in the unified space. In this regard, it has been stated that knowledge transfer across two or more optimization tasks, being simultaneously solved in the MFEA, occurs in the form of implicit genetic exchange between cross-cultural parents undergoing crossover [13]. While there are a plethora of such operators available in the literature, many of which exploit unique features of the underlying optimization tasks, herein we focus on the mechanics of the well-established simulated binary crossover (SBX) operator [27] from the standpoint of multitasking.

A salient feature of the SBX operator is that it emphasizes (with high probability) on creating offspring that are located close to their parents [28]. In other words, in a continuous search space, it is often the case that a generated offspring possesses genetic material that is in close proximity to at least one of its parents. With this background, consider the situation in Fig. 2 where two parents p_1 and p_2, with different cultural traits or skill factors (recall Definition 3), undergo crossover in a hypothetical 2-D unified search space. In particular, p_1 is assigned skill factor τ_1 while p_2 is assigned skill factor τ_2, with $\tau_1 \neq \tau_2$. Further, a pair of offspring, namely c_1 and c_2, is generated in the neighborhood of the parents by the SBX operator. Notice that c_1 is found to inherit much of its genetic material from p_1, while c_2 is found to inherit much of its genetic material from p_2. In such a scenario, if c_1 imitates the skill factor of p_2 (i.e., if c_1 is evaluated for τ_2) and/or if c_2 imitates the skill factor of p_1 (i.e., if c_2

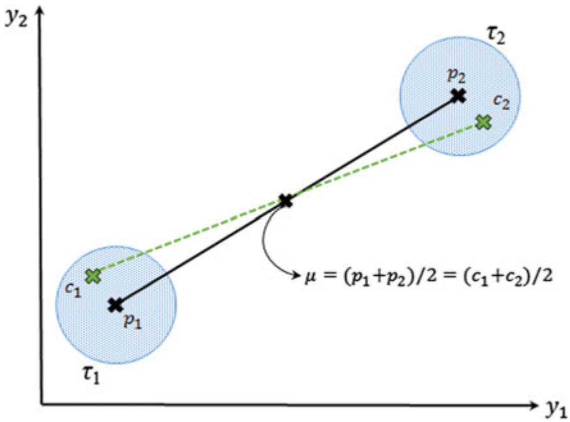

Fig. 2 Parent candidates p_1 and p_2 undergo standard SBX crossover to produce offspring c_1 and c_2 that are located close to their parents with high probability. Parent p_1 possesses skill factor τ_1 and p_2 possesses skill factor τ_2 with $\tau_1 \neq \tau_2$, thereby creating a multicultural environment for offspring to be reared in. Now, if c_1 imitates p_1 and/or if c_2 imitates p_2, then implicit genetic transfer is said to occur between the two tasks [13]

is evaluated for τ_1), then implicit transfer of knowledge occurs between the two tasks. At this juncture, if the genetic material corresponding to τ_1 (carried by c_1) is found to be useful for τ_2, or vice versa, then the transfer is deemed beneficial. Thereafter, the evolutionary selection pressure takes over to ensure that the positively transferred knowledge survives through generations. On the other hand, if the transfer turns out to be unproductive, the fundamental property of evolution is to eliminate the weak (negatively transferred [29–31]) genes by the natural process of survival of the fittest.

3.4 A Summary of the Salient Features of Evolutionary Multitasking

Standard EAs typically generate a large population of candidate solutions, all of which are unlikely to be competent for the task at hand. In contrast, in a multitasking environment, wherein all constitutive tasks are assimilated into a unified search space, it is intuitively more probable that a randomly generated or genetically modified individual is competent for at least one task. The mechanisms of the MFEA leverage upon this observation by effectively coordinating the search via the metaphorical interactions of genetic and cultural factors, thereby facilitating enhanced productivity in decision making processes in real-world settings.

Interestingly, during the combined optimization process it may so happen that the refined genetic material created within individuals of a particular skill factor (i.e., of a particular cultural trait) may also be useful for another group of individuals with a different skill factor. Thus, in such situations, the scope for implicit genetic transfer across tasks can potentially lead to accelerated convergence characteristics and/or the discovery of hard to find global optima. For the MFEA in particular, the transfer of genetic material occurs whenever cross-cultural parents with different skill factors undergo chromosomal crossover, as described in the previous subsection.

Practical scenarios amenable to multitasking are likely to occur in a variety of domains, including engineering, business, operations, etc., wherein optimization tasks with essentially identical underlying characteristics recur in large numbers. As per traditional practices, the knowledge contained in these related tasks is generally ignored by taking a tabula rasa approach to optimization. To this end, evolutionary multitasking provides a novel means of harnessing the so-far untapped source of knowledge, thereby opening doors to a plethora of real-world opportunities, some of which shall be showcased next.

4 Scope for Multitasking in the Real-World

Humans demonstrate cognitive multitasking capabilities on a daily basis. In [12], this anthropic phenomenon was realized computationally in the form of evolutionary multitasking for optimization. In order to emphasize the considerable real-world scope of multitasking, we present some guiding thoughts to aid effective utilization of the concept. It is contended that insights for a variety of practical applications can naturally be inferred from our discussions.

Without loss of generality, consider a hypothetical 2-factorial scenario where the first task is labeled τ_1 and the second task is labeled as τ_2. The setup of the multitasking environment is depicted in Fig. 3. Therein, notice the presence of a unified genotype space $\boldsymbol{Y}$ that encodes solutions to each of the constitutive tasks. In particular, $\boldsymbol{x}_1$ represents a solution in the phenotype space of τ_1 while $\boldsymbol{x}_2$ represents a solution in the phenotype space of τ_2. With this background, we categorize multitasking problem instances based on the amount of overlap in the phenotype space. We quantify the *overlap* (χ) as the number of variables in a task-specific solution space that have similar phenotypic meaning with respect to the other task, i.e., $\chi = |\boldsymbol{x}_{overlap}|$, leading to three broad categories, namely, complete, partial, and no overlap.

4.1 Complete Overlap in Phenotype Space

The first scenario we consider is perhaps the most intuitively pleasing application domain for evolutionary multitasking. In particular, we assume $\boldsymbol{x}_1 \backslash \boldsymbol{x}_{overlap} = \boldsymbol{x}_2 \backslash \boldsymbol{x}_{overlap} = \varnothing$ in Fig. 3. Accordingly, the only feature distinguishing the tasks is the set of task-specific *auxiliary variables* which are not explicitly part of the search space but describe the background in which the optimization tasks play out. A variety of possible real-world manifestations of this category in fields such as complex

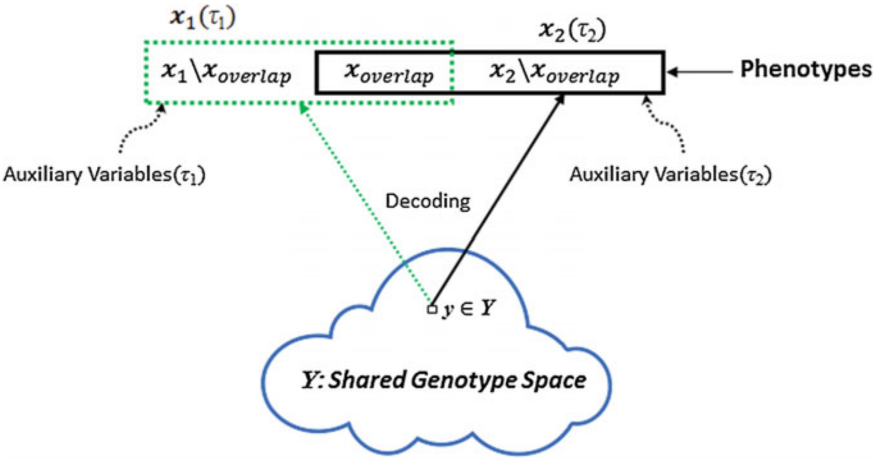

Fig. 3 Setup of a 2-factorial environment. The overlap in phenotype space represents the variables that have similar phenotypic interpretation with respect to either task. Note that although the overlapping variables need not bear identical numeric values for both tasks, they often provide the scope for useful genetic transfer due to similarities in their underlying behavior

engineering design and operations research have been discussed in [13]. In the present chapter, we delve into recent advancements in other areas of interest that have not been reviewed in previous papers.

A promising approach for improving optimization performance is the creation of *artificial helper (or catalyst) tasks that can aid the search process for a target optimization task of interest*, i.e., when both are combined in a single multitasking environment. While this possibility has been exploited in the field of machine learning [32], little has been done in the context of optimization. The lack of related approaches in optimization is particularly surprising given the availability of population-based methods that are endowed with the power of implicit parallelism. In light of this fact, preliminary investigations show that combining a target single-objective optimization task together with an artificially created multi-objective reformulation of the same task can improve convergence characteristics [33]. A representative example is depicted in Fig. 4 for a TSP instance where the target task and the helper task have completely overlapping phenotype spaces. In essence, the multi-objective reformulation, which has often been found to remove local optima [34], aids performance by leveraging on the scope for implicit genetic transfer.

In addition to the above, a recent study in bi-level optimization has shown the potential utility of evolutionary multitasking therein [14]. It was found that the notion of multitasking naturally emerges in the realm of evolutionary bi-level optimization where several lower level optimization tasks are to be solved with respect to different upper level population members. In particular, lower level tasks corresponding to neighboring upper level individuals, such as those belonging to the same cluster (as shown in Fig. 5), are likely to possess useful underlying commonalities that can

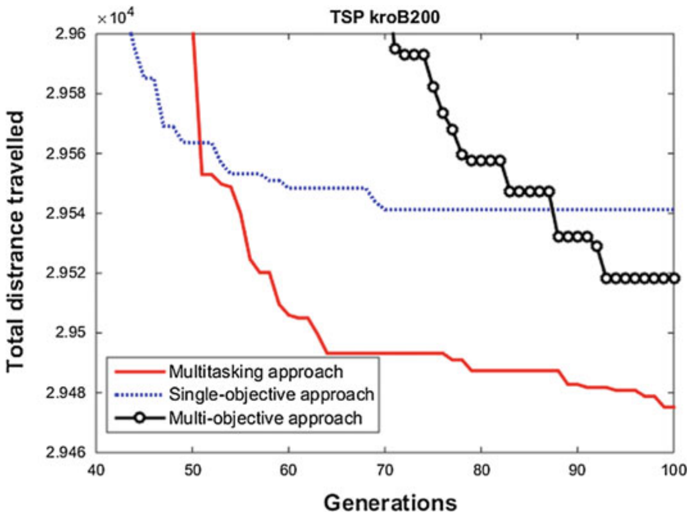

Fig. 4 Convergence trends for single-objective, multi-objective, and multitasking approaches for TSP kroB200. Multitasking harnesses the unique advantages of the single-objective and multi-objective formulations to accelerate convergence. Here, the artificially formulated multi-objective task acts as a catalyst during multitasking

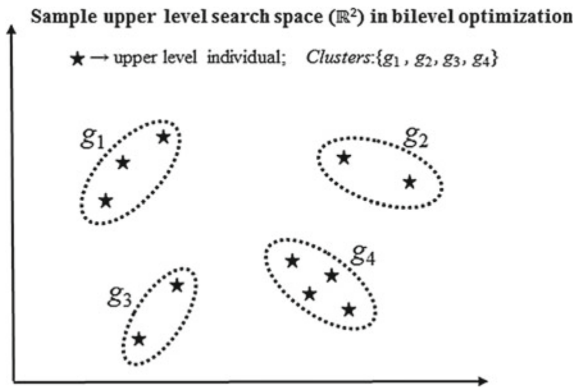

Fig. 5 In evolutionary bilevel optimization, lower level tasks corresponding to closely located upper level individuals (such as those belonging to the same cluster) are likely to possess commonalities that are exploitable by multitasking

be exploited via multitasking. The efficacy of the proposition was demonstrated by a proof-of-concept case study from the composites manufacturing industry which led to a computational cost saving of nearly 65 % for an expensive simulation-based optimization exercise [14]. A representative plot comparing the convergence trends achieved in practical bi-level optimization with and without evolutionary multitasking is provided in Fig. 6.

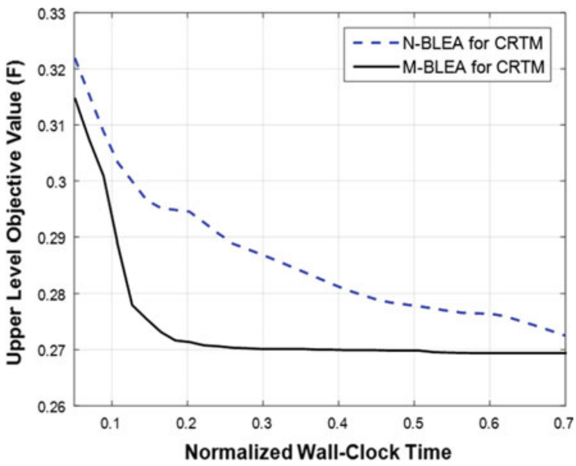

Fig. 6 Comparing averaged convergence trends of a standard Nested-Bilevel Evolutionary Algorithm (N-BLEA) with a Multitasking-Bilevel Evolutionary Algorithm (M-BLEA) for a Compression Resin Transfer Molding (CRTM) based composites manufacturing cycle [14]

4.2 Partial Overlap in Phenotype Space

Next, we consider the case where the phenotype spaces of constitutive tasks are only partially overlapping. For the 2-factorial setup in Fig. 3, this implies that $\boldsymbol{x}_1 \backslash \boldsymbol{x}_{overlap} \neq \emptyset$ and/or $\boldsymbol{x}_2 \backslash \boldsymbol{x}_{overlap} \neq \emptyset$ and $\chi \geq 1$. Thus, the transferrable knowledge between tasks is largely contained in the overlapping region, i.e., in $\boldsymbol{x}_{overlap}$. Real-world instantiations of such situations appear aplenty in the *conceptualization phase* of engineering design exercises. The process of conceptualization, as depicted in Fig. 7, is a human creativity driven preliminary design stage dealing with the formulation of an idea or concept which determines the scope of a project in terms of

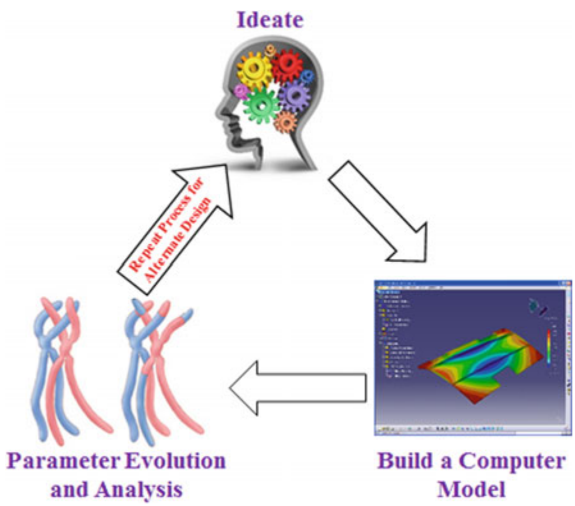

Fig. 7 Workflow of the conceptualization phase in engineering design [13]. Immense scope for multitasking exists due to the emergence of multiple alternative concepts to be analyzed. The concepts are likely to share some underlying commonalities as they all cater to the same product or process. This knowledge may be harnessed during multitasking to accelerate the design process

desired design features and requirements [35–37]. Typically, numerous alternative approaches will be proposed and analyzed before agreeing upon the single most suitable one. In these situations, the scope for evolving similar concepts via multitasking is quite intuitive, especially because several overlapping (i.e., recurring) design variables appear in different conceptual designs. Therefore, useful transferrable knowledge is instinctively known to exist among the tasks as they pertain to the same underlying product or process [13].

4.3 No Overlap in Phenotype Space (Blind Multitasking)

In both categories discussed so far, it is generally possible to make an a priori inference about the existence of transferrable knowledge that can be exploited by the process of multitasking. However, in many other real-world applications, it may be extremely difficult, if not impossible, to make such prior judgment about the complementarity between different optimization tasks. Multitasking instances belonging to the third category of no overlap in phenotype space, i.e., $\boldsymbol{x}_{overlap} = \varnothing$, are examples of such blind multitasking. However, even in these cases, it is noted that some latent complementarity between tasks may continue to exist in the unified genotype space. Thus, it often makes sense to allow evolution to take over and autonomously harness the complementarities whenever available, without the need to explicitly identify and inject domain knowledge into the algorithm. Needless to say, the execution of blind multitasking in the proposed naïve manner raises the fear of predominantly negative transfer. Whether the potential for enhanced productivity is sufficient to subdue such fears remains to be seen in the future. In the long run however, an *ideal* evolutionary multitasking engine is envisaged to be a complex adaptive system that is capable of inferring and appropriately responding to inter-task relationships on the fly, with its overall performance being at least comparable to the single-task solvers of the present day.

For the purpose of demonstration, we present a multitasking instance where performance enhancements are achieved despite the lack of any apparent overlap in the phenotype spaces of constitutive tasks. The example combines a pair of combinatorial optimization problems. As is well known, combinatorial problems possess complex objective function landscapes that are generally difficult to analyze. Thus, in most cases it is extremely challenging to make any prior inference about the availability of transferrable knowledge across tasks. Nevertheless, it can be concluded from the convergence trends in Fig. 8 that even in such cases of blind multitasking performance enhancement is achievable via the MFEA.

The 2-factorial problem depicted in Fig. 8 comprises a TSP (kroA200) and JSP (la39). For both tasks, the single-tasking approach is found to consistently get trapped in a local optimum. On the other hand, the diversified search facilitated by multitasking substantially improves performance characteristics, primarily as a result of the constant transfer of genetic material from one task to the other. It is therefore contended that while no decipherable complementarity exists between the tasks when

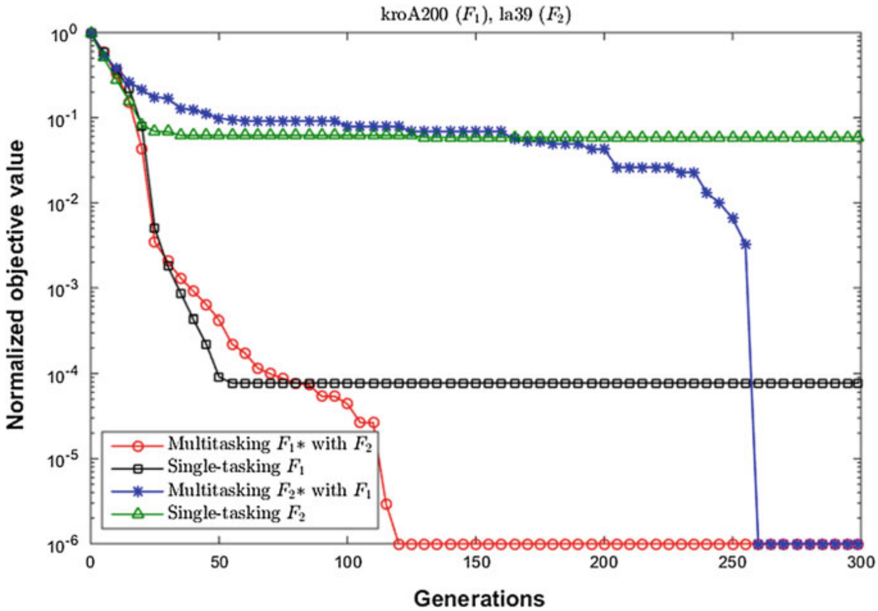

Fig. 8 Averaged convergence trends achieved while single-tasking and while multitasking across combinatorial optimization problems occurring in complex supply chain networks: TSP (kroA200) and JSP (la39) [13]

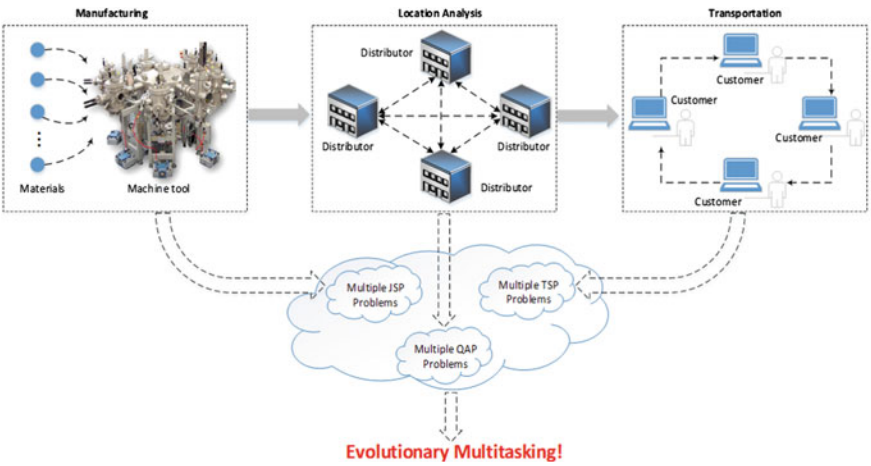

Fig. 9 Complex multi-echelon supply chain networks provide promising future prospects for the application of evolutionary multitasking [13]

viewed in the phenotype space, some latent complementarity may emerge in the unified genotype space. A real-world setting where the need to multitask across such seemingly disparate problems may arise is that of complex multi-echelon supply chain networks. The increase in productivity can help ease bottlenecks in decision making across multiple silos at once. Accordingly, as illustrated in Fig. 9, the domain of supply chain management can be a notable future beneficiary of evolutionary multitasking. For instance, while a TSP may represent a transportation (or logistics) silo of a supply chain, the JSP may represent a manufacturing silo, together forming key ingredients of the overall network.

5 Conclusions and Directions for Future Research

Evolutionary multitasking is a novel optimization paradigm that, albeit in its in-fancy, is showcasing significant promise with regard to unleashing the true power of implicit parallelism of population-based search [38]. To highlight the fact that each task in a multitasking environment presents an additional factor influencing the evolution of single population of individuals, the paradigm has also been formally labeled as Multifactorial Optimization (MFO). Sharing similar motivations as the field of multitask learning, MFO provides the scope for exploiting the underlying commonalities and/or complementarities between different (but possibly similar) optimization tasks, thereby achieving accelerated convergence characteristics in comparison to standard single-task optimizers. Furthermore, the quality of results obtained in a variety of domains of practical interest strongly encourages more comprehensive research pursuits in the future. It is envisaged that with increasing contributions from the community of EC researchers, as well as from the computer science and engineering communities at large, the notion of multitasking has the potential to change the current landscape of optimization techniques by seamlessly incorporating the scope of autonomous knowledge adaptation from various sources. In particular, it is contended that an artificial (computational) multitasking engine may be capable of retaining many of the advantages of cognitive multitasking, while effectively overcoming its potential perils.

In summary, it is recognized that so far we have merely scratched the surface of a potentially rich research topic. Rigorous examination of several practical and theoretical aspects of the paradigm is needed in the future. To begin with, a fundamental question that may arise in the mind of a practitioner is whether multitasking will always improve performance. In this regard, it must be noted that evolutionary multitasking acts as a means of harnessing the inductive bias provided by other optimization tasks in the same multitasking environment. Thus, while some inductive biases are helpful, some other inductive biases may hurt [10]. In fact, in the current simplistic description of the MFEA, we have indeed encountered some counter examples where the observed performance deteriorates during multitasking. However, in the long run, an ideal evolutionary multitasking engine is conceived to be an adaptive system that will be capable of estimating and autonomously responding to the level of complementarity between tasks on the fly. Thus, with the aim of enhancing productivity in complex decision making environments, it is the design of such intelligent algorithms that shall form the crux of our future research endeavours.

References

1. Dzubak, C.M., et al.: Multitasking: the good, the bad, and the unknown. J. Assoc. Tutoring Prof. **1**(2), 1–12 (2008)
2. Just, M.A., Buchweitz, A.: What brain imaging reveals about the nature of multitasking (2011)
3. Bäck, T., Hammel, U., Schwefel, H.-P.: Evolutionary computation: comments on the history and current state. IEEE Trans. Evol. Comput. **1**(1), 3–17 (1997)
4. Goldberg, D.E., et al.: Genetic Algorithms in Search Optimization and Machine Learning, vol. 412. Addison-Wesley, Reading (1989)
5. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. Computer **27**(6), 17–26 (1994)
6. Bertoni, A., Dorigo, M.: Implicit parallelism in genetic algorithms. Artif. Intell. **61**(2), 307–314 (1993)
7. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms, vol. 16. Wiley, New York (2001)
8. Branke, J.: MCDA and multiobjective evolutionary algorithms. Multiple Criteria Decision Analysis, pp. 977–1008. Springer, New York (2016)
9. Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: a survey. ACM Comput. Surv. (CSUR) **48**(1), 13 (2015)
10. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
11. Ben-David, S., Schuller, R.: Exploiting task relatedness for multiple task learning. Learning Theory and Kernel Machines, pp. 567–580. Springer, New York (2003)
12. Gupta, A., Ong, Y.-S., Feng, L.: Multifactorial evolution: towards evolutionary multitasking. Accepted IEEE Trans. Evol. Comput. **10**, 1109 (2015)
13. Ong, Y.-S., Gupta, A.: Evolutionary multitasking: a computer science view of cognitive multitasking. Cogn. Comput. **8**(2), 125–142 (2016)
14. Gupta, A., Mańdziuk, J., Ong, Y.-S.: Evolutionary multitasking in bi-level optimization. Complex Intell. Syst. **1**(1–4), 83–95 (2015)
15. Rice, J., Cloninger, C., Reich, T.: Multifactorial inheritance with cultural transmission and assortative mating. i. description and basic properties of the unitary models. Am. J. Hum. Genet. **30**(6), 618 (1978)
16. Cloninger, C.R., Rice, J., Reich, T.: Multifactorial inheritance with cultural transmission and assortative mating. ii. a general model of combined polygenic and cultural inheritance. Am. J. Hum. Genet. **31**(2), 176 (1979)
17. Cavalli-Sforza, L.L., Feldman, M.W.: Cultural versus biological inheritance: phenotypic transmission from parents to children (a theory of the effect of parental phenotypes on children's phenotypes). Am. J. Hum. Genet. **25**(6), 618 (1973)
18. Feldman, M.W., Laland, K.N.: Gene-culture coevolutionary theory. Trends Ecol. Evol. **11**(11), 453–457 (1996)
19. Chen, X., Ong, Y.-S., Lim, M.-H., Tan, K.C.: A multi-facet survey on memetic computation. IEEE Trans. Evol. Comput. **15**(5), 591–607 (2011)
20. Ong, Y.-S., Lim, M.H., Chen, X.: Research frontier-memetic computation past, present and future. IEEE Comput. Intell. Mag. **5**(2), 24 (2010)
21. Dawkin, R.: The Selfish Gene, vol. 1, p. 976. Oxford University Press, Oxford (1976)
22. Iqbal, M., Browne, W.N., Zhang, M.: Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems. IEEE Trans. Evol. Comput. **18**(4), 465–480 (2014)
23. Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA J. Comput. **6**(2), 154–160 (1994)

24. Gonçalves, J.F., Resende, M.G.: Biased random-key genetic algorithms for combinatorial optimization. J. Heuristics **17**(5), 487–525 (2011)
25. Snyder, L.V., Daskin, M.S.: A random-key genetic algorithm for the generalized traveling salesman problem. Eur. J. Oper. Res. **174**(1), 38–53 (2006)
26. Gonçalves, J.F., Resende, M.G.: A parallel multi-population biased random-key genetic algorithm for a container loading problem. Comput. Oper. Res. **39**(2), 179–190 (2012)
27. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. **9**(3), 1–15 (1994)
28. Deb, K., Sindhya, K., Okabe, T.: Self-adaptive simulated binary crossover for real-parameter optimization. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 1187–1194. ACM (2007)
29. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
30. Feng, L., Ong, Y.-S., Lim, M.-H., Tsang, I.W.: Memetic search with interdomain learning: a realization between CVRP and CARP. IEEE Trans. Evol. Comput. **19**(5), 644–658 (2015)
31. Feng, L., Ong, Y.-S., Tan, A.-H., Tsang, I.W.: Memes as building blocks: a case study on evolutionary optimization+transfer learning for routing problems. Memet. Comput. **7**(3), 159–180 (2015)
32. Krawiec, K., Wieloch, B.: Automatic generation and exploitation of related problems in genetic programming. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2010)
33. Da, B., Gupta, A., Ong, Y.-S., Feng, L.: Evolutionary multitasking across single and multi-objective formulations for improved problem solving. In: 2010 IEEE Congress on Evolutionary Computation (CEC). IEEE (2016)
34. Knowles, J.D., Watson, R.A., Corne, D.W.: Reducing local optima in single-objective problems by multi-objectivization. Evolutionary Multi-criterion Optimization, pp. 269–283. Springer, New York (2001)
35. Avigad, G., Moshaiov, A.: Set-based concept selection in multi-objective problems: optimality versus variability approach. J. Eng. Des. **20**(3), 217–242 (2009)
36. Avigad, G., Moshaiov, A., Brauner, N.: MOEA-based approach to delayed decisions for robust conceptual design. Applications of Evolutionary Computing, pp. 584–589. Springer, New York (2005)
37. Avigad, G., Moshaiov, A.: Interactive evolutionary multiobjective search and optimization of set-based concepts. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **39**(4), 1013–1027 (2009)
38. Gupta, A., Ong, Y.-S., Feng, L., Tan, K.C.: Multiobjective multifactorial optimization in evolutionary multitasking (2016)