# Evolutionary Multitasking: A Computer Science View of Cognitive Multitasking

Yew-Soon Ong, Abhishek Gupta

School of Computer Engineering, Nanyang Technological University, Singapore 639798.

Corresponding author contact details,

Email: asysong@ntu.edu.sg

Tel: +65-6790-6448; Fax: +65-6792-6559

## Abstract

The human mind possesses the most remarkable ability to perform multiple tasks with apparent simultaneity. In fact, with the present-day explosion in the variety and volume of incoming information streams that must be absorbed and appropriately processed, the opportunity, tendency, and (even) the need to multitask is unprecedented. Thus, it comes as little surprise that the pursuit of intelligent systems and algorithms that are capable of efficient multitasking is rapidly gaining importance among contemporary scientists who are faced with the increasing complexity of real-world problems. To this end, the present paper is dedicated to a detailed exposition on a so-far underexplored characteristic of population-based search algorithms, i.e., their inherent ability (much like the human mind) to handle multiple optimization tasks at once. We present a simple evolutionary methodology capable of *cross-domain multitask optimization* in a *unified genotype space*, and show that there exist many potential benefits of its application in practical domains. Most notably, it is revealed that multitasking enables one to automatically leverage upon the underlying commonalities between distinct optimization tasks, thereby providing the scope for considerably improved performance in real-world problem solving.

***Keywords*:** multitask optimization, evolutionary multitasking, evolutionary algorithm, cross-domain optimization, memetic computation

# 1. Background and Motivation

The ability of the human mind to manage and execute multiple tasks in what seems like apparent simultaneity is perhaps once of its most remarkable features. A quick glance at the world around us reveals the ubiquity of supposed cognitive multitasking. Some of the most natural examples include talking (on a mobile phone or otherwise) while walking or driving, the simple act of cooking where a number of resources are to be managed at once in order to complete a set of orders in time, media multitasking (as illustrated in Fig. 1), simultaneously scanning through endless streams of information received across various social networking platforms (Facebook, Twitter, etc.) while reading and sharing the posts we find most interesting. Although interleaving multiple tasks may entail a considerable *switching cost* during which the brain must readjust [1], it is conceded by psychologists that bundling together related tasks (particularly when they possess some form of re-usable knowledge overlap) may allow one to move more fluidly between them, thereby potentially improving productivity via multitasking [1-3]. Most importantly, it is recognized that in this fast-paced, technologically-driven world that we live in, multitasking is perhaps the best way to fit in all of our priorities, albeit at the (often tolerable) cost of a marginal reduction in the quality of output achieved.

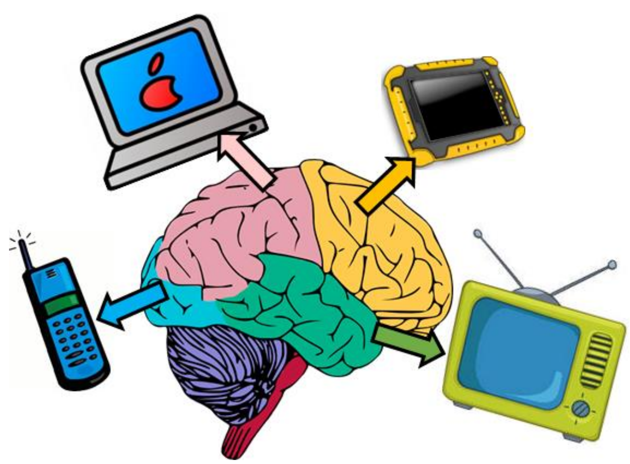

Fig. 1. Media multitasking is perhaps the most common type of present-day cognitive multitasking wherein the brain is exposed to various forms of media or communication devices at the same time.

A major criticism levelled against cognitive multitasking originates from the observed switching cost (as mentioned earlier) during which the brain attempts to overcome the resulting interference between tasks and adjusts to the new task [1]. Thus, while constantly switching between competing tasks, an individual may experience undesirable distractions that lead to slower response time, degraded performance, and/or increased error rates [4]. However, while developing computational analogues of the multitasking phenomenon, it is noted that modern-day computers are largely 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 or eliminating most of its potential perils.

In the rapidly advancing research on machine learning techniques, the notion of multitask learning has been prevalent for at least the past two decades [5]. Essentially, the idea is to leverage upon the relevant information available in related tasks by performing learning using a shared representation. As a result, the external information source acts as a form of inductive bias which often helps the main tasks in being learned better. Put simply, if different tasks can by some means share what they learn (even during the process of learning itself), it might be simpler for a learner to tackle them together, rather than to tackle each one in isolation. The practical motivation behind the paradigm lies mainly in the claim that a potentially rich source of information may be contained in the training signals of other learning tasks drawn from the same domain. Thus, discounting such information that lies outside the self-contained scope of a particular problem, as is typically the case in tabula rasa learning, may be deemed highly counterproductive [5], especially under the limited availability of training data.

Keeping the aforementioned arguments in mind, we shall hereafter call the attention of the reader to another sub-discipline of computer science that has seen notable advances in recent years, namely, complex real-world problem solving via *optimization*. In particular, the main aim of this paper is to reveal the multitasking potential of Evolutionary Algorithms (EAs), which form a popular class of population-based search metaheuristics in the domain of computational intelligence [6-8]. While the proposition bears similar motivations to the field of multitask learning, it operates from the standpoint of nature-inspired computing, facilitating implicit information exchange across different optimization tasks. Over the years, EAs have been used with much success in solving a variety of problems in science, operations research, and engineering, gradually establishing their current status as a mainstay of optimization. The fundamental mechanics of an EA involves simulating Darwinian principles of *natural selection* or *survival of the fittest* [8], according to which a population of candidate solutions are steered through complex search spaces. The navigation typically culminates in convergence of the population to a near-optimal region of the search space, which marks the satisfactory completion of a specific underlying optimization task.

The distinctive feature of an EA is the emergence of powerful implicit parallelism which materializes from the simple rules of its population-based search strategy [7, 9-11]. However, despite the known power of implicit parallelism, it is interesting to note that EAs have almost exclusively been used to solve only a single optimization problem at a time. Regardless of the fact that the singular problem may involve mono, multi, or even many objective functions [12-16], seldom has an effort been made to multitask, i.e., to solve multiple independent optimization problems simultaneously using a single population of evolving individuals; here the notion of *independence* suggests that each ongoing task is self-contained (i.e., each task is traditionally viewed in a stand-alone manner) and is free to comprise either mono, multi, or many objectives. In order to clarify the distinction between the conceived *multitask optimization paradigm*, and the richly studied topic of multi/many-objective optimization, the reader is referred to the illustration in Fig. 2. Accordingly, this paper shall present multitask optimization, and particularly *evolutionary multitasking*, as a novel paradigm in the realm of evolutionary computation. Effective evolutionary multitasking is expected to provide the scope for efficient utilization of fruitful knowledge contained in other (possibly related) tasks (as has been discussed previously in the context of machine learning), thereby bolstering the utility of optimization tools in complex real-world problem solving exercises.

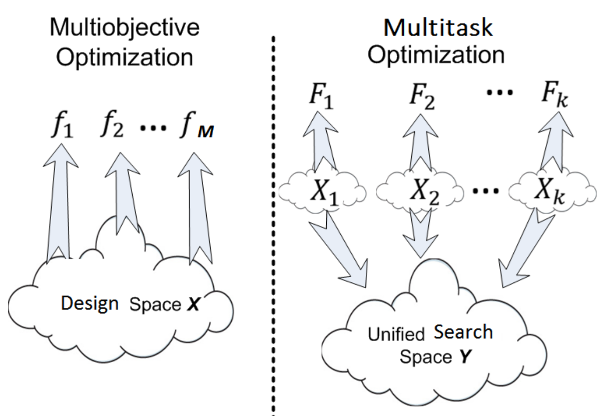


Fig. 2. Distinguishing between multi-objective and multitask optimization. While multi-objective optimization typically has a single search space encompassing all objectives, in contrast, multitask optimization involves different distinctive search spaces corresponding to the different self-contained optimization tasks. Thus, an additional unification step is needed for effective multitask optimization. Interestingly, note that each task in a multitasking environment may itself comprise multiple objectives, which highlights the greater generality of the paradigm.

For a detailed exposition of the notions discussed above, the remainder of this paper has been organized as follows. In Section 2, we discuss a simple evolutionary methodology, recently proposed in [17], for efficient multitask optimization. One of the major points of emphasis in the algorithm is the use of a *unified genotype space* (i.e., a *unified search space* or a *unified solution representation scheme*) encompassing all tasks in a multitasking environment. The unification forms the basis of implicit knowledge transfer across optimization tasks in the form of encoded *genetic material*. Interestingly, it shall be shown that unified representation schemes can be defined to encompass a variety of tasks belonging to different domains (i.e., in continuous as well as discrete optimization), thereby laying the foundation for *cross-domain multitask optimization* frameworks.

Section 3 demonstrates that evolutionary multitasking indeed works. The performance of the proposed multitasking engine is compared against a more traditional methodology on some insightful benchmark examples, as well as for a more realistic optimization setting involving the path-planning of multiple Unmanned Aerial Vehicles (multi-UAVs). In Section 4, we discuss *how* evolutionary multitasking works. In particular, we show that the crossover (or recombination) operator when applied in the unified genotype space, in conjunction with the effects of evolutionary selection pressure, form the key ingredients driving implicit knowledge transfer across constitutive optimization tasks in multitasking environments. Next, in Section 5, it is reasoned that there indeed are numerous promising opportunities for evolutionary multitasking, and multitask optimization in general, in real domains. The validity of our claim is demonstrated through a variety of illustrations, including, (a) a sample study in last mile logistics, (b) an exercise in complex engineering design from the composites manufacturing industry, (c) potential application in complex multi-echelon supply chain networks, and finally (d) an example from a conceived cloud-based on-demand optimization software.

Finally, in Section 6, we briefly discuss the plethora of research opportunities opened up by the promising future prospects of multitask optimization, including the scope for algorithmic advancements, improved theoretical understanding, and a variety of applications to complex real-world problems of today.

## 2. An Evolutionary Methodology for Multitask Optimization

During the progress of a typical EA, a population of individuals navigate through a predefined search space, visiting a number of locations therein, in pursuit of at least a near-optimal solution; where optimality is defined according to the maximization or minimization of a specified objective function. Interestingly, by assuming the search space to be a unification or amalgamation of multiple heterogeneous search spaces corresponding to different optimization tasks, it is intuitively possible that a visited location that is inferior for one optimization task may in fact be of high quality with respect to another. Or otherwise, it may so happen that a region of the unified search space that is good for one optimization task, is also good for another task at the same time.

The simple scenarios mentioned above constitute the basic insights that have led to the recent development of an evolutionary algorithm capable of efficient multitask optimization; one that can harnesses latent relationships between distinct optimization tasks to enhance the effectiveness of the search process. To elaborate, the *Multifactorial Evolutionary Algorithm* (MFEA) proposed in [17] was shown to be inspired by the bio-cultural models of *multifactorial inheritance* [18, 19], which essentially describe how complex developmental traits among offspring emerge from the interactions of various genetic and cultural factors. The central ingredient of the MFEA is that it makes use of a unified search space encompassing all tasks in a multitasking environment, with each task contributing a unique *cultural factor* (in a metaphorical sense) imparting a distinctive influence on the evolution of individuals in the population. The algorithm can in fact be classified under the umbrella of *memetic computation* [20, 21] as it considers the transmission of biological as well as cultural building blocks (i.e., genes and memes [22]) from parents to their offspring. The principal means of incorporating cultural effects is by the creation of a computational analogue of the natural phenomenon of *vertical cultural transmission*, which suggests that the phenotype of an offspring is directly affected by the phenotype of its parents [23, 24]. Another feature that is considered to take importance while multitasking across significantly disparate problems is that of *assortative mating* [17, 18] which advocates that individuals prefer to mate with those possessing the cultural background. The algorithmic manifestations of the aforesaid ideas are discussed next.

The basic workflow of the MFEA for tackling multiple mono-objective optimization problems simultaneously is summarized in Algorithm 1. Therein, $P_t$ denotes the current population of the algorithm at generation $t$, whereas $C_t$ denotes the offspring population derived from $P_t$ via genetic variation operators (such as crossover and mutation). Notice that the majority of the algorithm resembles a standard evolutionary procedure, with the inclusion of a new entity, namely, the *skill factor* ($\tau$).

The skill factor, which is viewed as a computational representation of an offspring's cultural background, represents the one task, amongst all other tasks in a multitasking environment, to which the individual is associated [17]. The assignment of a particular skill factor ($\tau_i$) to an offspring ($c_i$) is in fact based on the concept of vertical cultural transmission. As is well known, *imitation* is one of the most prevalent forms of cultural transmission [20]. By following the same principle, an offspring in the MFEA is designed to randomly imitate the skill factor of any one of the parent individual(s) from which it is created. Observe that in addition to inducing a form of cultural bias into the evolutionary search process, the inclusion of the so-called skill factor serves as a means to reduce the overall computational cost of the MFEA, as an offspring is evaluated only with respect to the skill factor that it imitates. The possibility of implicit knowledge transfer between tasks (the mechanics of which shall be described in Section 4) emerges autonomously when parents belonging to different cultural backgrounds (i.e., possessing different skill factors) undergo recombination, thereby creating a multicultural environment for their offspring to be reared in. However, note that when constitutive tasks in multitasking environments are known to have little in common, the principal of assortative mating is allowed to kick in, which prevents excessive mixing of genetic material by controlling the amount of cross-cultural breeding in the MFEA [17].

**Algorithm 1:** Pseudocode of the MFEA

**while** (stopping conditions are not satisfied) **do**
  $C_t$ = Crossover + Mutate ($P_t$)
  **for every** $c_i$ in $C_t$ **do**
    Assign skill factor $\tau_i$ via imitation-based vertical cultural transmission
    Evaluate $c_i$ for task $\tau_i$ only
  **end**
  $R_t = C_t \cup P_t$
  Select the fittest members from $R_t$ to form $P_{t+1}$.
  Set $t = t + 1$
**end while**

Returning to Algorithm 1, note that a rank-based fitness calculation scheme, as has been described in [17, 25] for inferring the overall fitness of an individual in a multitasking environment, is employed in the generational selection step of the MFEA. Further, bear in mind that the skill factors of all individuals in the initial population, as they clearly cannot be assigned via the inductive process of imitation, can either be assigned randomly (while ensuring uniform representation for all constitutive optimization tasks) or by the exhaustive evaluation procedure considered in [17].

## 2.1. The Unified Search Space and Cross-Domain Decoding Exemplars

For implicit knowledge transfer to take place efficiently during multitasking, it is pivotal to describe a unified genotype space encompassing all the constitutive optimization tasks. In fact, 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 related problem solving exercises.

A unification implies that the genetic building blocks (or schemata [7]) 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 can 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 will be crucial in order to ensure the exploitation of transferrable knowledge across tasks. On the other hand, in many naive cases, as were described in [17], simply extracting the *first* $D_j$ variables from the chromosome is also a viable alternative.

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* [17] is adopted. According to the random-key scheme, each variable of a chromosome is encoded by a *continuous value* in the range [0, 1]. The salient feature of this representation is that it elegantly encompasses a wide variety of problems in continuous as well as discrete optimization. Illustrative decoding examples for continuous optimization and popular instantiations of combinatorial optimization shall be discussed hereafter.

As a side note, we acknowledge that while the prescribed random-key representation scheme indeed provides notable flexibility to the MFEA in terms of its cross-domain multitasking capability, it is certainly not a strict pre-requisite for successful multitask optimization. In fact, if multitasking is to be performed across problems belonging to similar domains (particularly those involving discrete variables), superior performance with improved knowledge transfer is likely to be achieved if a domain-specific representation scheme is utilized instead.

### *2.1.1. 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 search space of the appropriate optimization task. For instance, consider a continuous optimization task in which the $i^{th}$ variable ($x_i$) is bounded in the range $[L_i, U_i]$. If the $i^{th}$ random-key of a chromosome $\boldsymbol{y}$ takes a value $y_i \in [0, 1]$, then the decoding procedure may simply be given by,

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

### *2.1.2. Decoding for Sequencing Problems*

In the domain of combinatorial optimization, sequencing problems (also referred to as permutation-based optimization problems) include classical examples such as the Travelling Salesman Problem (TSP), the Job-Shop Scheduling Problem (JSP), the Quadratic Assignment Problem (QAP), the Vehicle Routing Problem (VRP), etc. Their salient feature 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 first investigated in [26]. 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. Consequently, the methodology detailed below has found notable interest over the past two decades in the field of operations research [27].

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 $\boldsymbol{y}$ = (0.7, 0.1, 0.3, 0.9, 0.05), such that the first entity is labeled as 0.7, the second entity is labeled as 0.1, the third is labeled as 0.3, and so on. Following the technique suggested in [26], the order of entities encoded by the chromosome $\boldsymbol{y}$ is given by the sequence $\boldsymbol{s}$ = (5, 2, 3, 1, 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.

### *2.1.3. Decoding for Binary Variables*

Some of the most commonly encountered examples of optimization problems involving binary variables include the 0/1 Knapsack Problem (KP), the 0/1 Multiple Knapsack Problem (MKP), the set covering problem, etc. There can be several conceivable ways of deducing binary variables from random-keys. One simplistic approach is to set $x_i = 1$ if $y_i \geq 0.5$, and $x_i = 0$ otherwise. However, we find that this technique generally leads to poor performance of the MFEA on KP and MKP instances of any substantial size. Thus, in [17] a clustering-based method was proposed for optimization problems of this kind. Details of the methodology are not reproduced herein for the sake of brevity.

### *2.1.4. A Summary of Cross-Domain Decoding*

The unification scheme coupled with the decoding process is perhaps the most important ingredient of effective multitask optimization. Thus, in order to clearly demonstrate the manner in which a single random-key chromosome may be decoded into different task-specific solutions, we refer to the sample illustration presented in Fig. 3. Therein, a 12-D KP is to be solved in conjunction with a 6-D TSP. It is shown that by extracting the required number of variables from the unified 12-D random-key chromosome ($\boldsymbol{y}$), and subsequently applying a decoding step, task-specific solutions can immediately be obtained.

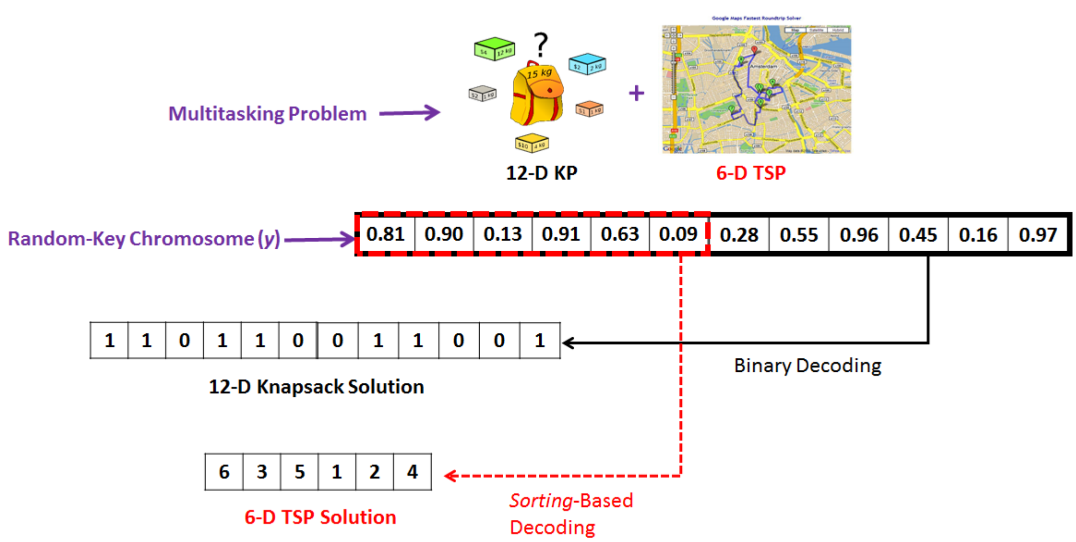


Fig. 3. An illustrative example of the manner in which a single random-key chromosome may be decomposed into different task-specific solutions.

# 3. Multitask Optimization at Work

Prior to discussing about the fundamental mechanics of implicit knowledge transfer in evolutionary multitasking, we present three computational studies that demonstrate multitasking at work. This section serves the purpose of illustrating that the potential implications of effective multitask optimization are truly noteworthy from a practical standpoint.

In the first numerical study, we consider the case of solving a pair of continuous benchmark functions simultaneously in the MFEA. Thereafter, in the second example, we tackle different multi-UAV path-planning problems at once, thereby showcasing the utility of multitasking in realistic domains. Finally, in the third case, we consider multitasking across distinct combinatorial optimization problems, thereby highlighting the diverse capabilities of the proposed framework. The experiments will hopefully provide some intuition towards the kind of problems where evolutionary multitasking is applicable. After all, developing the correct *intuition*, on the part of the user, is contended to be a major driving factor for future applications of multitask optimization.

## 3.1. Numerical Experiments with Continuous Benchmark Functions

We consider three benchmark functions that are commonly encountered in the literature on real parameter optimization. The three functions, namely, (a) sphere function [28], (b) Ackley function [28, 29], and (c) Rastrigin function [30], are combined pairwise to form three different multitasking instances. Table 1 summarizes the properties of the three functions as considered in this experimental study. Firstly, note that the extent of the search space corresponding to each function is distinct, which emphasizes the utility of the unified solution representation scheme. Secondly, notice that the global optimums of all functions intersect not only in the phenotype space, but also in the unified genotype space described by the random-key representation scheme, thereby ensuring the availability of transferrable knowledge across different optimization tasks.

Table 1. Properties of the benchmark functions as used in the first experimental study

| **Function Name** | $D_j$ | **Extent** | **Global Optimum** | **Properties** |
|---|---|---|---|---|
| Sphere | 30 | $x_i \in [-100,100]\ \forall i$ | [0, 0, …, 0] | Unimodal |
| Ackley | 30 | $x_i \in [-32, 32]\ \forall i$ | [0, 0, …, 0] | Multimodal + Rotated |
| Rastrigin | 30 | $x_i \in [-5, 5]\ \forall i$ | [0, 0, …, 0] | Multimodal + Rotated |

In the numerical experiments, the MFEA is employed as an instantiation of an evolutionary multitasking engine. In order to highlight performance variations due to multitasking, the output of the MFEA shall be presented alongside an elitist Single-Objective Evolutionary Algorithm (SOEA) with essentially the same algorithmic specifications and parameter settings as the MFEA. The use of identical experimental settings ensures that the observed changes in performance during multitasking can be entirely attributed to the presence of inductive bias from other optimization tasks in the multitasking environment. Note that, here the mutation operator is suppressed in both optimizers. Doing so allows us to focus our attention on the results achieved via unadulterated transmission of genetic material during crossover. In particular, removing mutation makes it easier to decipher the effects of implicit knowledge transfer during multitasking. The random exploration of the search space that transpires due to mutation is often found to obscure the underlying phenomena of interest, and is therefore removed herein. We incorporate the Simulated Binary Crossover (SBX) operator [31] into both EAs, and set its distribution index for the benchmark problems to $\eta_c = 4$. Further, note that we do not perform any additional uniform crossover-like variable swap between offspring. The latter condition supports minimum schema disruption as uniform crossover (albeit good for exploration) is known to be highly disruptive, and ill-suited for preserving genetic linkages between the variables of optimization tasks [32]. In order to compensate for the suppressed exploration, a sufficiently large population of 100 individuals is deployed with the added feature that every individual undergoes local solution refinements in the spirit of Lamarckian learning [30]. The BFGS quasi-Newton method is utilized for this purpose.

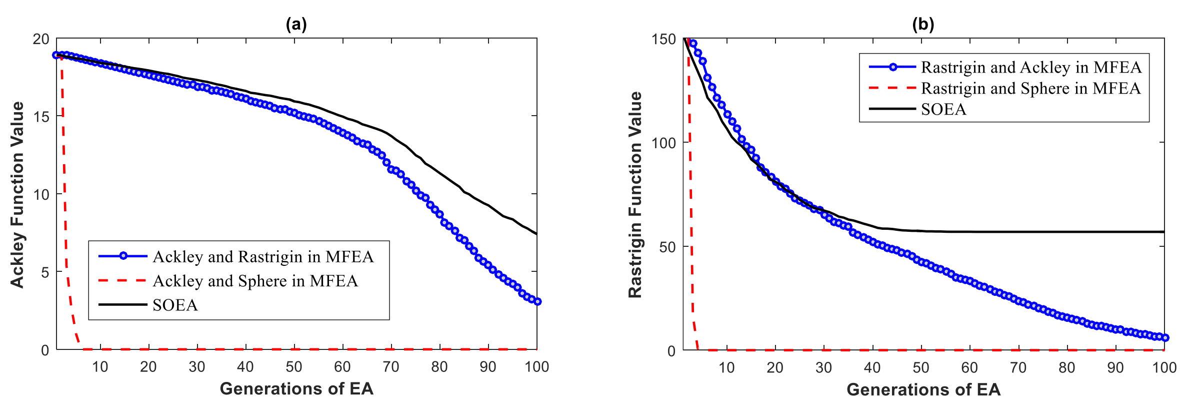


Fig. 4. Averaged convergence trends for numerical experiments with synthetic benchmark functions: (a) performance on the Ackley function, and (b) performance on the Rastrigin function.

We first consider the convergence trends in Figs. 4(a) and 4(b) for the more intuitive case where the multimodal Ackley and Rastrigin functions are paired with the unimodal sphere function, respectively. The setup of this experiment serves as a clear representation of the potential benefits of evolutionary multitasking. We find that the rapid optimization achieved for the simple unimodal function provides a strong inductive bias to the search process of the considerably more challenging multimodal functions. Due to the intersecting global optimums, the inductive bias is sure to be helpful, thereby leading to predominantly *positive* transfer of knowledge [33-35] which greatly accelerates the convergence process for the otherwise complex optimization tasks.

For a more noteworthy but less exaggerated outcome of multitask optimization, we refer to the averaged convergence trends in Figs. 4(a) and 4(b) that correspond to the case where the Ackley function and the Rastrigin function are paired together in a multitasking environment. The improvement in convergence characteristics achieved for both functions during multitasking is substantial in comparison to the SOEA. Remarkably, it is not merely the case that one function benefits from the other. Instead, the two functions are found to be strongly mutually complementing. Therefore, accounting for the fact that Ackley and Rastrigin functions are multimodal, it is hypothesized that the positive transfer of knowledge may depend on more than the mere proximity (or apparent intersection) of the global optimums of different optimization tasks. In fact, it is contended that the *objective function landscapes of constitutive tasks may somehow complement each other throughout the unified search space*. As a result, the evolving population in the MFEA can exploit the landscape of both functions simultaneously, efficiently overcoming obstacles to converge faster. For the case of the more traditional SOEA, the lack of any useful inductive bias from a different task causes the population to often stagnate at a local optimum (as is revealed in Fig. 4[b]).

## 3.2. Multitask Optimization in Multi-UAV Path-Planning

Multi-UAV path-planning is a complex optimization exercise that must satisfy a variety of conditions, including maneuvering constraints, avoidance of no-fly zones (or geo-fences), collision prevention, etc., while pursuing smooth navigation trajectories that improve one or more underlying objective functions [36]. When the constraints are highly stringent, it is often extremely challenging to design intelligent path-planning algorithms that can consistently and efficiently converge to the desired trajectories. Furthermore, when there is a lack of human supervision, the problem becomes even harder to resolve. To this end, we present a case study that showcases the utility of multitask optimization in such realistic domains. In particular, we find that the knowledge contained in related path-planning exercises may be autonomously harnessed by an evolutionary multitasking engine to accelerate the discovery of high quality solutions.

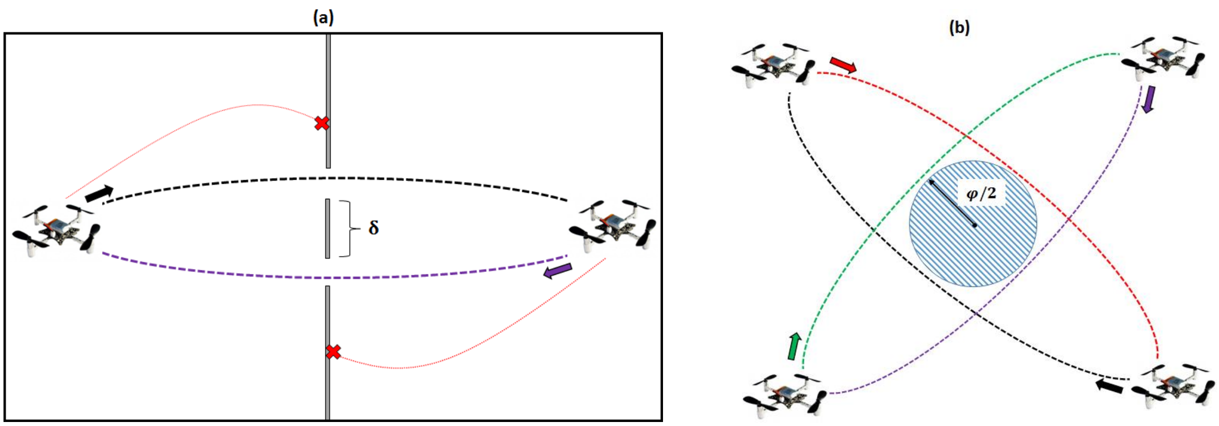


Fig. 5. Schematic of optimum paths in multi-UAV path-planning missions. (a) *Mission 1*: 2 UAVs must fly through a barrier with two narrow openings as they traverse the shortest path to their respective destinations, and (b) *Mission 2*: 4 UAVs must fly around a centrally located geo-fence of circular planform without colliding into one another as they travel to their respective destinations along the shortest path.

The schematic of two distinct path-planning missions are depicted in Fig. 5. The first mission portrayed in Fig. 5(a) involves a pair of UAVs that are to fly through two narrow openings in a barrier as they traverse the shortest possible path to their respective destinations. In the second mission, depicted in Fig. 5(b), there are four UAVs that must fly around a centrally located geo-fence of circular planform, without colliding into one another, as they travel to their respective destinations along the shortest possible path. Note that in both cases the maintenance of a smooth trajectory is vital as a UAV cannot suddenly change directions. Clearly, these missions pose a stiff challenge to the path-planning algorithm due to the prevalence of stringent constraints. We show hereafter that by using the proposed MFEA to tackle both missions in concurrence, any available knowledge overlap between the tasks can get automatically harnessed, thereby enabling efficient convergence to improved solutions.

The likelihood of helpful inductive bias (or complementarity) between the two path-planning missions described above emerges from the condition that $|\delta|$ in Fig. 5(a) is similar to $|\varphi|$ in Fig. 5(b), i.e., $|\delta| \approx |\varphi|$. In other words, a similar magnitude of stray from an ideal (straight line) path is required in both path-planning missions, thereby giving rise to a useful nugget of knowledge that is common to the two tasks. The fact that the MFEA can successfully exploit the knowledge to accelerate convergence characteristics is highlighted in Fig. 6. Interestingly, notice that *Mission 2* comprising 4 UAVs (depicted in Fig. 6[b]) is substantially benefitted by multitasking, while *Mission 1* comprising 2 UAVs (depicted in Fig. 6[a]) is affected to a much lesser extent. This observation leads to the contention that evolutionary multitasking may serve as a promising means of improving performance on complex optimization tasks by utilizing information from comparatively simpler tasks. It must however be stated that the success of multitasking in the current example is strongly influenced by the setup of the unified random-key representation scheme and the subsequent task-specific decoding mechanism, so as to facilitate the transfer of knowledge across tasks in the most effective manner possible.

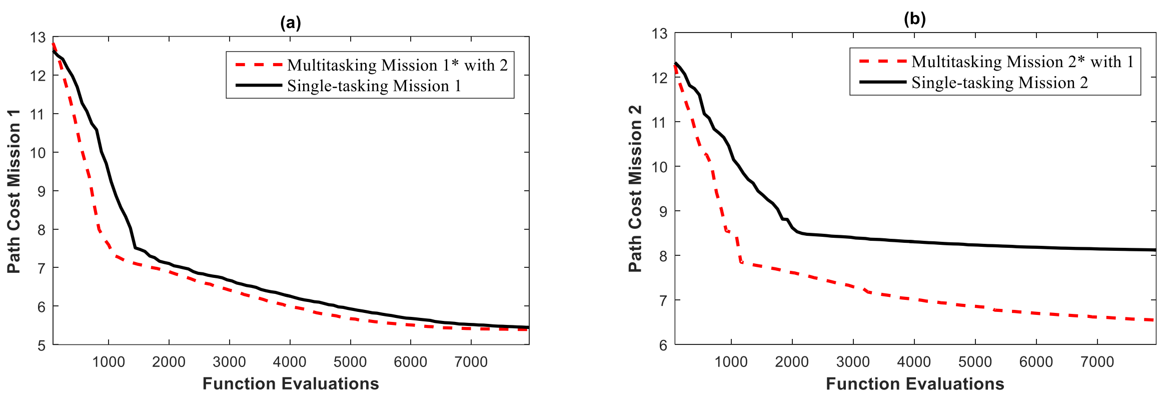


Fig. 6. Averaged convergence trends achieved by single-tasking and via multitasking for (a) *Mission 1* and (b) *Mission 2*.

### 3.3. Multitasking Across Combinatorial Optimization Problems

In both computational studies presented so far, it was possible to *a priori* recognize a nugget of knowledge (common to the constitutive tasks) that was guaranteed to be useful during the process of multitasking. For instance, in the study with continuous benchmark functions, it was known beforehand that the optimums of the respective objective functions were located in close proximity of each other. Similarly, in the multi-UAV path-planning example, a nugget of transferrable knowledge was explicitly identified. However, in many real-world applications, it may be extremely difficult to make any a priori judgement about the complementarity between different optimization tasks.

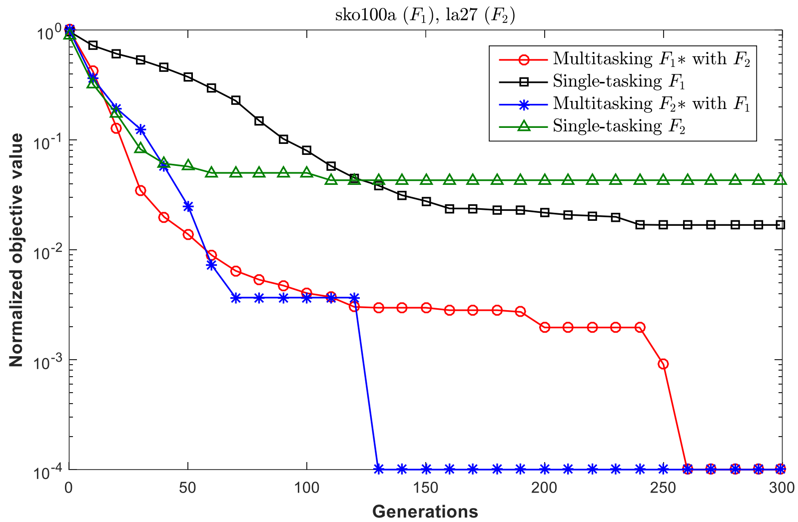


Fig. 7. Averaged convergence trends achieved while multitasking across combinatorial optimization problems: QAP (Sko100a) and JSP (la27).

In this example, we demonstrate that prior familiarity about the relationship between tasks is not a pre-requisite for successful evolutionary multitasking. While it is acknowledged that the availability of background information may be incorporated to significantly improve the performance of a multitasking engine, interestingly, it has been found that the proposed MFEA can autonomously exploit *latent* complementarities even between seemingly disparate optimization tasks, thereby successfully accelerating convergence characteristics. As an illustration of this phenomenon, we present an example of multitasking across a pair of combinatorial optimization problems. As is well known, combinatorial problems possess complex objective function landscapes that are difficult to analyze. Thus, in many cases, it is very challenging, if not impossible, to infer the availability of transferrable knowledge across tasks in any practicable amount of time. Nevertheless, it can be concluded from the convergence trends in Fig. 7 that even in such cases of essentially *blind* multitasking, performance enhancement is possible via the MFEA.

The depicted instance comprises a QAP (Sko100a) and a JSP (la27) solved together. While the single-tasking approach is found to get trapped in local optimum regions of the search space, the diversified search facilitated by multitasking, as a result of the constant transfer of genetic material from one task to the other, substantially improves performance characteristics. It is contended that while no decipherable complementarity may exist between tasks when viewed in the task-specific or phenotype spaces, some hidden form of complementarity may emerge in the unified genotype space. Thus, the scope of evolutionary multitasking encompasses all such problems, even when no direct relationship between optimization tasks can be inferred.

# 4. How Does Evolutionary Multitasking Work?

In this section, we elaborate upon the fundamental mechanisms driving effective evolutionary multitasking. In particular, we discuss the workings of the crossover operator and the effects of evolutionary selection pressure in multitask settings, which serve as the core ingredients facilitating implicit knowledge transfer across tasks.

## 4.1. The Mechanics of Implicit Knowledge Transfer

Notice that 'knowledge', in the MFEA, exists in the form of encoded genetic material. Thus, the transfer of knowledge between tasks is in fact achieved in the form of *implicit genetic transfer*, which naturally occurs during crossover operations. Specifically, when parents possessing different skill factors (i.e., belonging to different cultural backgrounds) undergo recombination in the unified search space, the exchange of genetic material automatically takes place. Some sample modes by which genetic transfer may occur are shown herein.

### *4.1.1. Transfer in Continuous Search Spaces*

We first discuss the case of the random-key unification scheme in which all constitutive optimization tasks in a multitasking environment are assimilated into a continuous search space. For the experimental study in Section 3.1, we had made use of the SBX operator to navigate the continuous space. A salient feature of the SBX operator, which happens to be of particular interest from the standpoint of multitasking, is that it creates offspring that are located close to the parents with very high probability [31, 37]. With this background, consider the situation in Fig. 8 where two parents $p_1$ and $p_2$, with different skill factors, undergo crossover in a hypothetical 2-D unified search space. To be precise, $p_1$ has skill factor $\tau_1$ and $p_2$ has skill factor $\tau_2$, with $\tau_1 \neq \tau_2$. A pair of offspring, $c_1$ and $c_2$, are created by the SBX operator in close proximity of the parents. In other words, $c_1$ 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 $\tau_2$) and/or if $c_2$ imitates the skill factor of $p_1$ (i.e., if $c_2$ is evaluated for $\tau_1$), then implicit transfer of knowledge is said to have occurred between the two tasks.

At this juncture, if the genetic material corresponding to $\tau_1$ (carried by $c_1$) is found to be useful for $\tau_2$, 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 unhelpful or *negative* [33], the nice property of evolution is that the negatively transferred genes get automatically eliminated from the population (over the course of a few generations) by the process of natural selection or survival of the fittest.

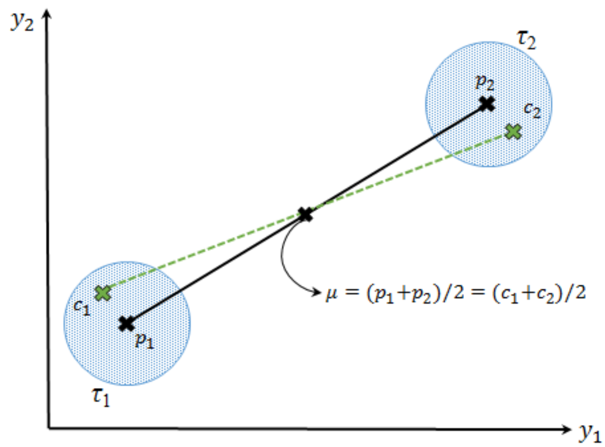


Fig. 8. Parent candidates $p_1$ and $p_2$ undergo standard SBX crossover to produce offspring $c_1$ and $c_2$ that are located close to their parents with high probability. Parent $p_1$ possesses skill factor $\tau_1$ and $p_2$ possesses skill factor $\tau_2$ with $\tau_1 \neq \tau_2$, thereby creating a multicultural environment for offspring to be reared in. If $c_1$ is evaluated for $\tau_2$ and/or if $c_2$ is evaluated for $\tau_1$, then implicit genetic transfer is said to occur between the two tasks.

### *4.1.2. Transfer in Discrete Search Spaces*

Next, we present the case of combining discrete optimization problems into a unified search space which also happens to be discrete in nature. For simplicity of exposition, we assume that the search space is comprised of a string of binary variables. Moreover, the one-point crossover operator [38] is chosen for our discussion.

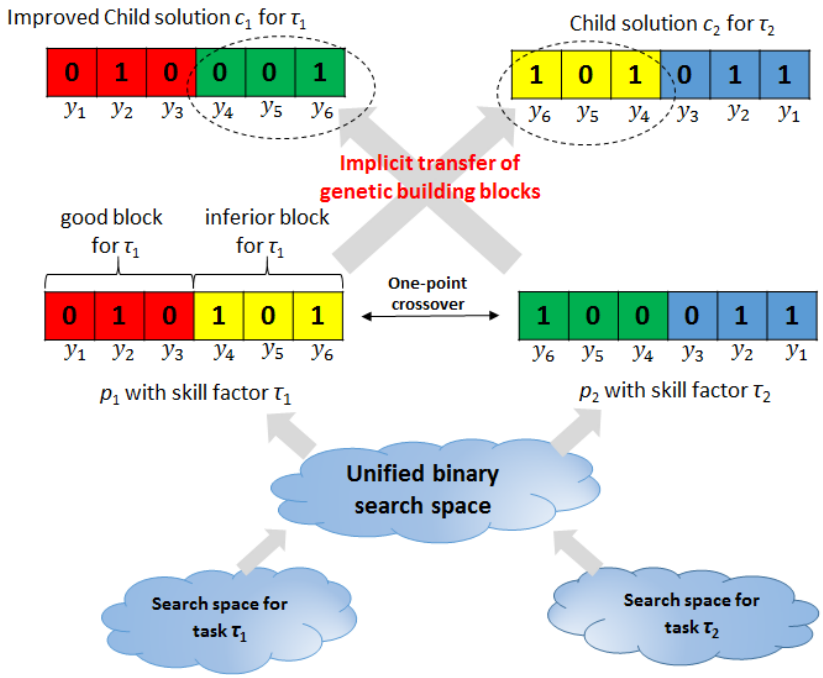


Fig. 9. During one-point crossover of parents $p_1$ and $p_2$ possessing different skill factors, genetic building blocks (or schemata) get automatically transferred from one task to the other. Since the schema from task $\tau_2$ also happens to be good for task $\tau_1$, the genetic transfer is positive.

From the graphic in Fig. 9, it can be seen that implicit genetic transfer occurs in a simplistic manner in discrete search spaces. In fact, the mechanism is very much similar to that of standard single-task optimization, where good genetic building blocks from different individuals combine to form better offspring. The only difference being that in the case of multitasking building blocks are assembled and exchanged across different tasks at once, thereby leading to the possibility of fruitful knowledge sharing and accelerated convergence characteristics whenever there exists some form of underlying commonality between tasks.

The feature described above shows the intrinsic ability of EAs to facilitate autonomous knowledge transfer while effortlessly accommodating multitask optimization. Therefore, it is indeed surprising that the utilization of this elegant attribute of EAs has seldom been investigated in the past. In fact, to the best of our knowledge, the notion of multitasking and its potential practical significance has largely eluded researchers in mathematical optimization. Accordingly, the present paper concerns itself with revealing the true power of EAs, as naturally emerges from the implicit parallelism of population-based search. The numerous implications for real-world problem solving shall be discussed in the next section.

# 5. Multitasking in the Real World

On a daily basis, we humans draw upon our cognitive ability to multitask in order to satisfactorily meet all our responsibilities. On occasions when different tasks possess underlying commonalities, the process of multitasking provides a platform for spontaneous exchange of information that can often assist in improved task execution. In [17], this anthropic phenomenon was realized computationally in the form of evolutionary multitasking in optimization. In particular, the booming field of cloud computing was identified as an arena naturally faced with multiple jobs from multiple users at the same time. Thus, in an envisioned cloud-based on-demand optimization tool, it is conceivable to devise techniques by which the underlying synergies between distinct optimization tasks are automatically harnessed, thereby providing customers with faster and better solutions.

In order to emphasize the considerable generality of the scope of multitask optimization, we present some guiding thoughts to aid effective utilization of the concept for real-world problem solving. Based on our discussions, numerous practical applications of the paradigm naturally emerge, a selection of which are discussed in this paper. For instance, significant relevance of the proposed ideas is found in the field of complex engineering design where the practice of adapting and re-using relevant knowledge is commonplace.

## 5.1. The Scope for Effective Multitask Optimization

Consider a hypothetical 2-task scenario where the first task is labeled $\tau_1$ and the second task is labeled as $\tau_2$. The setup of the multitasking environment is depicted in Fig. 10. Adhering to the need for unification, the genotype space is assumed to encode solutions in the phenotype space of either task. For instance, in Fig. 10, $\boldsymbol{x}_1$ represents a solution in the phenotype space of $\tau_1$ while $\boldsymbol{x}_2$ represents a solution in the phenotype space of $\tau_2$.

With this background, we classify potentially fruitful multitasking exercises into three broad categories based upon the amount of overlap in the phenotype space of the constitutive optimization tasks. We quantify *overlap* ($\chi$) as the number of variables in a task-specific solution space that happen to have similar phenotypic meaning with respect to the other task, i.e., $\chi = |\boldsymbol{x}_{overlap}|$ in Fig. 10. As has been alluded to earlier, it is not hard to conceive numerous real-world implications of multitasking in each of the three categories. In the discussions that follow, we shall present a selection of noteworthy applications of the paradigm in diverse domains.

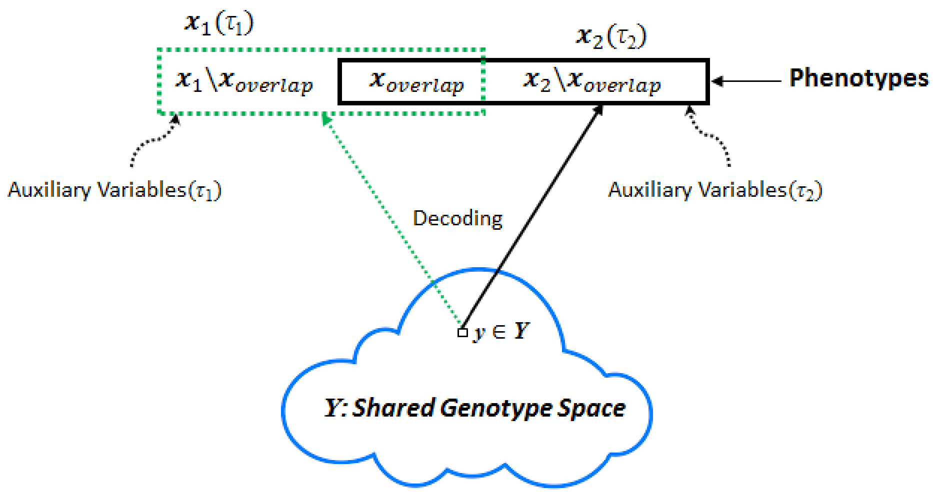


Fig. 10. Setup of a 2-task environment with a unified genotype space. The overlap in phenotype space represents the variables that share similar phenotypic meaning with respect to either task (although they may not necessarily assume identical numeric values for both tasks).

### *5.1.1. Complete Overlap in Phenotype Space*

We begin by considering the extreme case of tasks that are completely overlapping in phenotype space, i.e., $\boldsymbol{x}_1 \backslash \boldsymbol{x}_{overlap} = \boldsymbol{x}_2 \backslash \boldsymbol{x}_{overlap} = \emptyset$ in Fig. 10. Thus, the only feature distinguishing the tasks is the set of task-specific *auxiliary variables* which essentially describe the background in which the optimization tasks play out. In Fig. 10, the auxiliary variables are presented alongside the phenotype space, and are not explicitly a part of the search space. In problems of complex engineering design, operations research, etc., auxiliary variables can be manifested in various forms. For example, in the wing-shape optimization problem [39], the operating conditions of the aircraft (i.e., the Mach and Reynolds numbers) form a possible set of auxiliary variables that can vary between different design exercises. Intuitively, the wing-shapes obtained under different operating conditions are likely to possess some design commonalities, mainly because they all share the same underlying physical phenomena.

For the purpose of demonstrating the efficacy of multitask optimization under complete phenotypical overlap, we present a case study in last mile logistics. Here, the phenotype space comprises a set of geographically distributed nodes (customers) that must be assigned to a fleet of delivery vehicles originating from a depot. Moreover, the customers must be ordered so as to specify the chronology in which they are to be serviced [40]. In real-world settings, the auxiliary variables may occur as additional constraints for the VRP, such as, (a) customer time windows [41], (b) variable pickup and/or delivery demands, (c) variable traffic conditions, etc. For setting up the multitasking environment, we consider two separate VRP instances (VRP-1 and VRP-2) which have similar geographical distribution of customers (i.e., there is complete overlap in phenotype space) but different time window constraints [41]. To highlight the presence of transferrable knowledge between two such tasks, we consider the VRP instances to be presented to the MFEA asynchronously. Specifically, the VRP-2 is introduced into the multitasking engine at the intermediate stages of solving VRP-1. As shown in Fig. 11, the significant impetus received by VRP-2 during multitasking, as opposed to standard single-tasking, serves as strong empirical evidence of the helpful inductive bias provided by VRP-1.

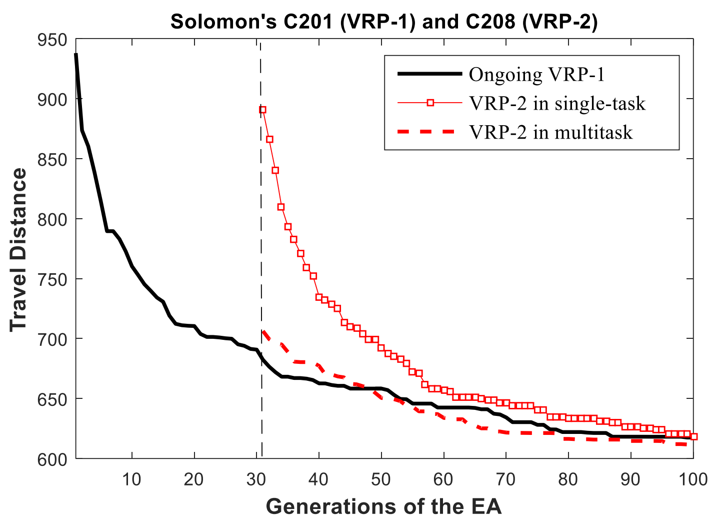


Fig. 11. Averaged convergence trends achieved while solving two separate VRP instances with complete overlap in the phenotype space.

### *5.1.2. Partial Overlap in Phenotype Space*

As a more general setting for multitasking, we consider the case where the phenotype spaces of constitutive tasks are only partially overlapping. For the 2-task setup in Fig. 10, 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 the two 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 illustrated in Fig. 12, 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 desired design features and requirements [42-44]. Typically, numerous alternative approaches or conceptual designs will be proposed and analyzed before agreeing upon the single most suitable one. In these situations, there naturally emerges the scope for evolving similar concepts (i.e., bearing several overlapping design variables) concurrently using an evolutionary multitasking engine, particularly because useful transferrable knowledge is instinctively known to exist among tasks pertaining to the same underlying product or process.

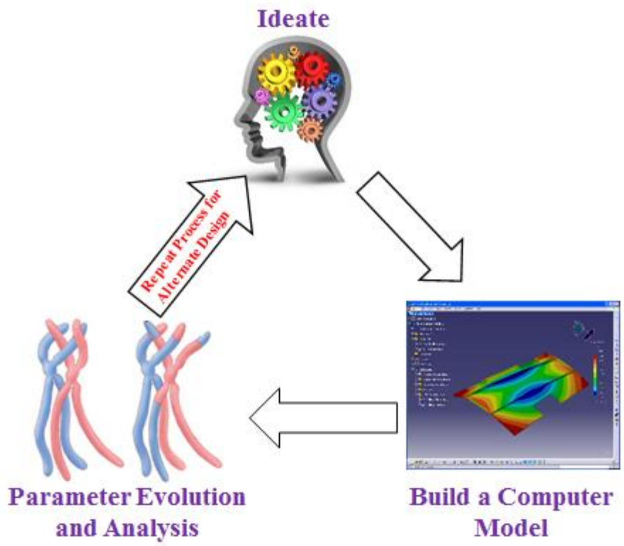


Fig. 12. Typical workflow of the conceptualization phase of a modern-day design exercise. Immense scope for multitasking emerges due to the presence of several conceptual designs to be analyzed. These conceptual designs are likely to share some underlying commonalities that may be harnessed during multitasking to accelerate the design process.

As an illustration of evolutionary multitasking across optimization tasks with partial overlap, we consider a case study from the composites manufacturing industry. In particular, the purpose is to investigate the performance of two candidate manufacturing processes (for the manufacture of a glass-fiber reinforced epoxy composite disc [45]) that belong to the same family of Liquid Composite Molding (LCM) methods. The schematic of a typical LCM process is depicted in Fig. 13. The candidate processes are (a) Resin Transfer Molding (RTM) [45, 46], and (b) Injection/Compression Liquid Composite Molding (I/C-LCM) [45-47].

Traditionally, I/C-LCM has been less popular due to the complexity of its *in situ* mold compression step [48]. On the other hand, a notable advantage of I/C-LCM, as compared to RTM, is that it usually enables significantly faster manufacturing cycle time [49]. Thus, while deciding on a suitable manufacturing technique for a particular fiber reinforced polymer composite part, the manufacturer must thoroughly explore both processes in terms of practicality, setup and running cost, as well as throughput, which typically culminates in the formulation of a multi-objective optimization problem.

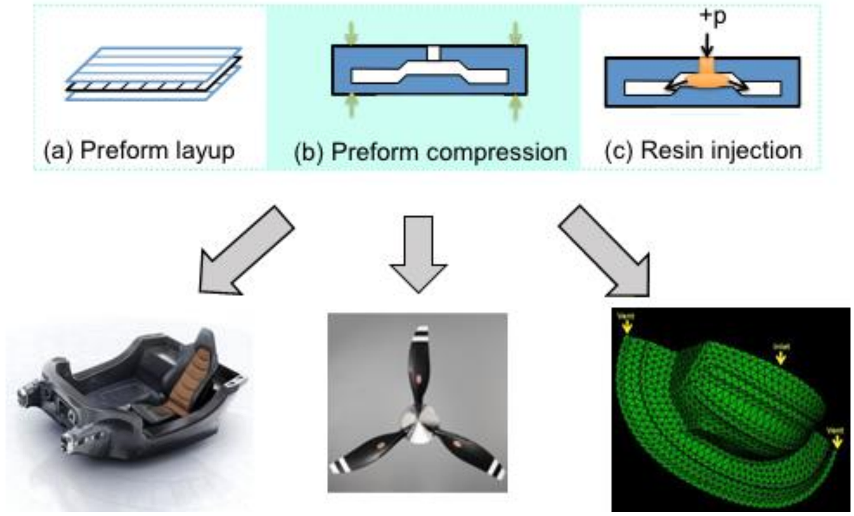


Fig. 13. Typical workflow of a generic Liquid Composite Molding (LCM) process for composite part manufacturing. RTM and I/C-LCM are members of the LCM family of manufacturing methods that have found popularity in the automotive and aerospace industries, as well as in a variety of other applications.

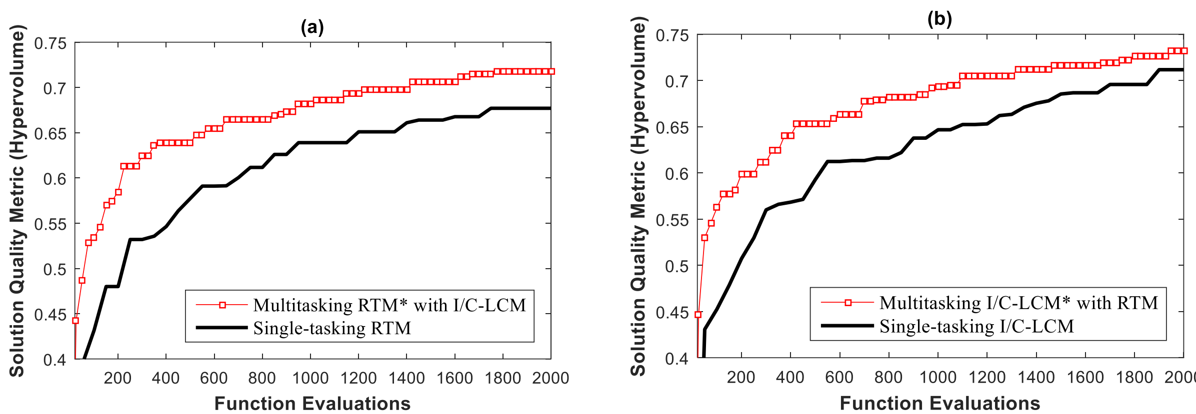


Fig. 14. Comparing the evolution of the hypervolume [50] metric achieved during multitasking and during standard single-tasking for the case of (a) RTM, and (b) I/C-LCM.

While the RTM and I/C-LCM cycles possess some distinct variables, they also possess a set of baseline variables that appear in the phenotype spaces of both processes [46]. This overlap stems from the fact that both procedures belong to the same family of LCM methods. As a result, there is bound to exist some transferrable knowledge between the two associated multi-objective optimization tasks. As shown in Fig. 14(a) (pertaining to RTM) and Fig. 14(b) (pertaining to I/C-LCM), by engaging the MFEA to tackle the tasks in conjunction, it becomes possible to autonomously harness their commonalities, thereby accelerating convergence to higher quality solutions.

Note that the solution quality is measured as a combination of (a) the estimated equipment cost and (b) the throughput [51]. As a matter of particular interest, it must be noted that the results presented in Fig. 14 are for simulation-based optimization exercises. Since the numerical (finite element) simulation of a composites manufacturing process is computationally demanding (requiring several minutes of runtime per objective function evaluation), the accelerated convergence may significantly ease the bottleneck caused by an exorbitantly time consuming design stage.

### *5.1.3. No Overlap in Phenotype Space*

In the two prior cases, the measurable overlap in the phenotype space implies that one can often intuitively predict the likelihood of transferrable knowledge between optimization tasks. However, for multitasking instances belonging to the third category of *no overlap in phenotype space*, i.e., $\boldsymbol{x}_{overlap} = \emptyset$, it is generally impossible to make any a priori judgement. However, even in such cases of essentially blind multitasking, it is noted that some latent complementarity between tasks may continue to exist in the unified genotype space. Thus, it continues to make sense to allow evolutionary multitasking 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 raises the fear of predominant negative transfer. Whether the potential for performance gains is adequate to quell such fears, remains to be fully established.

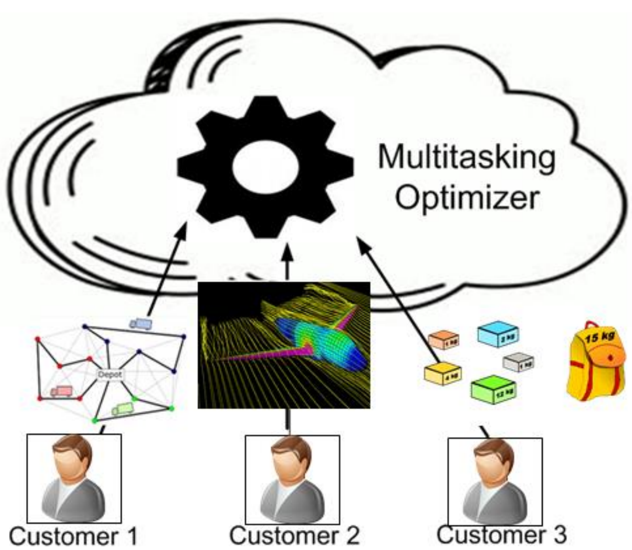


Fig. 15. A cloud computing environment comprising an on-demand “optimization” service can give rise to a scenario wherein multiple optimization tasks, each with distinct properties, are received from multiple customers at the same time. A multitasking engine, with the ability to autonomously harness the commonalities between tasks, is potentially of much value in such settings.

Being the most general form of multitask optimization, instances with no phenotypical overlap finds manifold real-world applications. Among these, some of the most noteworthy applications are in complex multi-echelon supply chain networks, and in the novel conception of cloud-based on-demand optimization services [17]. As illustrated in Figs. 15 and 16, both domains provide immense scope for evolutionary multitasking. While supply chain networks comprise a number of distinct silos, a cloud-based optimization service provides a platform for different users to solve their individual problems of interest. In such settings, the ability to autonomously leverage upon the hidden commonalities between different optimization tasks, is contended to be invaluable. A realization is depicted in Fig. 17 where results are presented for a JSP solved in conjunction with a TSP, representing the manufacturing and transportation (or logistics) silos, respectively, of a supply chain network. As has been seen previously, the diversified search facilitated by multitasking substantially improves performance characteristics, while standard single-tasking is found to get trapped in local optimum regions of the search space.

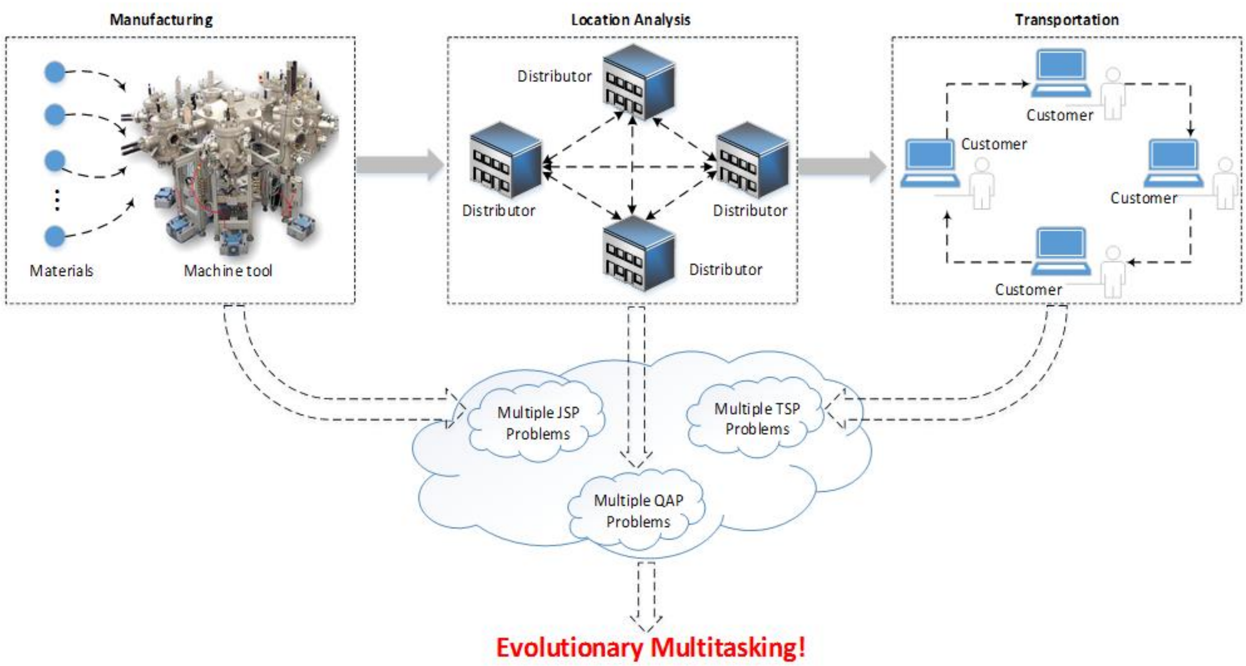


Fig. 16. Complex multi-echelon supply chain networks provide a setting where evolutionary multitasking may be applied to simultaneously tackle multiple optimization tasks with distinctive properties.

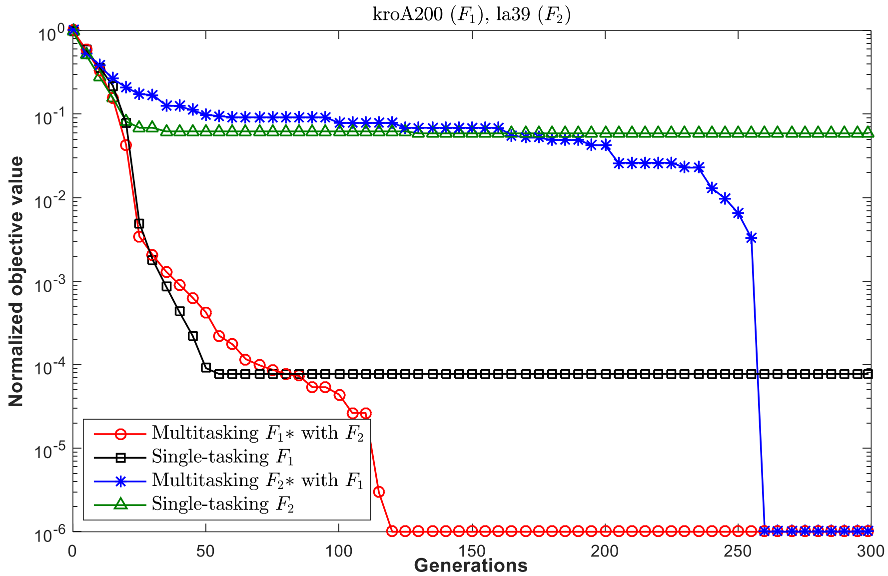


Fig. 17. Averaged convergence trends achieved while single-tasking and while multitasking across a pair of optimization problems commonly arising in complex supply chain networks: TSP (kroA200) and JSP (la39).

# 6. Discussion and Future Directions

In this paper, we have described the novel concept of multitasking in optimization, and have demonstrated its implications on some benchmark functions and on other realistic optimization examples. The quality of results achievable, strongly encourages more comprehensive research pursuits in the future. It is envisaged that with increasing contributions from the community of evolutionary computation 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 or eliminating most of its potential perils. To conclude the paper, we discuss some of the most promising future directions and research questions that emerge from the paradigm.

## 6.1. Investigating Novel Operators for Effective Evolutionary Multitasking

As has been gathered from existing multitask problem solving exercises, the description of the unified genotype space, the process of decoding, and the mechanism of implicit genetic transfer (generally via crossover), together form perhaps the most critical ingredients of successful evolutionary multitasking. If the adopted methodology does not appropriately suit the behavior of the underlying optimization tasks, the process of multitasking can often be counterproductive. To this end, it must be noted that only the SBX crossover operator has so far been tested in conjunction with the continuous random-key representation scheme. There exist a plethora of other real-coded crossover operators that may prove better suited for multitasking under different circumstances. Similarly, when discrete unification schemes are adopted, there is much scope for exploring the implications of a variety of other genetic operators available in the literature.

### 6.2. Scope for Theoretical Advancements

A natural question that may arise in the mind of a practitioner is whether multitasking *always* improves performance. From our current experience with the MFEA, we find that multitask optimization may not always be beneficial due to the possibility of negative knowledge transfer. Note that evolutionary multitasking acts as a means of harnessing the inductive bias provided by other optimization tasks in the same multitasking environment. While some inductive biases are helpful, some other inductive biases may hurt [5]. Thus, while there are indeed several examples, even in blind multitasking, where the diversified search significantly improves performance characteristics, there exist some counter examples where the observed performance deteriorates during multitasking. This is especially true in the case of combinatorial optimization where it is extremely challenging to infer any relationship/relatedness between tasks.

In light of the above, an important research topic is the formulation of approximate online models that can make use of the data generated during the optimization process to somehow quantify the relatedness between tasks. Efforts into offline measurement of relatedness, via correlation quantification between objective function landscapes of benchmark optimization tasks, are currently ongoing. As is outlined next, such models will most likely play a central role in the design of evolutionary multitasking engines of the future.

### 6.3. Scope for Algorithmic Advancements: Many-tasking, Adaptation

In addition to the core requirement of having a suitable unification and transfer mechanism, the peripheral workings of the evolutionary multitasking engine will certainly play a crucial role in the future progress of multitask optimization. In this regard, some of the most noteworthy research prospects are in (a) the effective handling of *many* tasks at a time, and (b) developing adaptive EAs that are capable of modelling (online) the complementarity between optimization tasks and accordingly adapting their core mechanisms (such as genetic operators, population size and distribution, choice of local search steps, etc.). In fact, in the long run, an *ideal* evolutionary multitasking engine is envisioned to be a complex adaptive system with its performance being at least comparable to that of standard single-tasking evolutionary optimizers of the present day.

### 6.4. Further Applications in Complex Real-World Problem Solving

Finally, we believe that the notion of multitask optimization provides a fresh perspective with regard to the harnessing of available information for improved problem solving. The knowledge contained in related optimization exercises, which has traditionally been disregarded in standard single-task optimization (primarily because there was no known way of inferring and exploiting such knowledge in a simplistic manner), can now be leveraged upon due to the new opportunities provided by multitasking. Several practical problems in science, engineering, operations research, etc., some examples of which have been presented in this paper, will benefit immensely from the proposed ideas. In fact, with the growing complexity, volume, and speed of real-world challenges in the technologically-driven world of today, multitasking appears to be an indispensable tool for the future.

## Acknowledgement

The authors would like to thank Dr. Yuan Yuan, Research Fellow in the School of Computer Engineering, NTU, Singapore, for his assistance in producing some of the figures presented in the paper.

## References

1. Dzubak CM. Multitasking: The good, the bad, and the unknown. The Journal of the Association for the Tutoring Profession. 2008 Sep;1(2):1-12.
2. Adler RF, Benbunan-Fich R. Juggling on a high wire: Multitasking effects on performance. International Journal of Human-Computer Studies. 2012 Feb 29;70(2):156-68.
3. Salvucci DD, Taatgen NA. Threaded cognition: an integrated theory of concurrent multitasking. Psychological review. 2008 Jan;115(1):101.
4. Just MA, Buchweitz A. What brain imaging reveals about the nature of multitasking. 2011.

5. Caruana R. Multitask learning. Machine learning. 1997 Jul 1;28(1):41-75.
6. Bäck T, Hammel U, Schwefel HP. Evolutionary computation: Comments on the history and current state. Evolutionary computation, IEEE Transactions on. 1997 Apr;1(1):3-17.
7. Golberg DE. Genetic algorithms in search, optimization, and machine learning. Addion wesley. 1989;1989.
8. Srinivas M, Patnaik LM. Genetic algorithms: A survey. Computer. 1994 Jun;27(6):17-26.
9. Grefenstette JJ, Baker JE. How genetic algorithms work: A critical look at implicit parallelism. In Proceedings of the third international conference on Genetic algorithms 1989 Dec 1 (pp. 20-27). Morgan Kaufmann Publishers Inc.
10. Bertoni A, Dorigo M. Implicit parallelism in genetic algorithms. Artificial Intelligence. 1993 Jun 30;61(2):307-14.
11. Wright AH, Vose MD, Rowe JE. Implicit parallelism. InGenetic and Evolutionary Computation—GECCO 2003 2003 Jan 1 (pp. 1505-1517). Springer Berlin Heidelberg.
12. Seada H, Deb K. A unified evolutionary optimization procedure for single, multiple, and many objectives. doi: 10.1109/TEVC.2015.2459718
13. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on. 2002 Apr;6(2):182-97.
14. Deb K. Multi-objective optimization using evolutionary algorithms. John Wiley & Sons; 2001 Jul 5.
15. Ishibuchi H, Tsukamoto N, Nojima Y. Evolutionary many-objective optimization: A short review. In IEEE congress on evolutionary computation 2008 Jun 1 (pp. 2419-2426).
16. Asafuddoula M, Ray T, Sarker R. A Decomposition Based Evolutionary Algorithm for Many Objective Optimization. Evolutionary Computation, IEEE Transactions on. 2014; 19(3):445-460.
17. Gupta A, Ong YS, Feng L. Multifactorial evolution: Toward evolutionary multitasking. Accepted IEEE Trans. Evol. Comput. doi: 10.1109/TEVC.2015.2458037
18. Rice JO, Cloninger CR, Reich TH. Multifactorial inheritance with cultural transmission and assortative mating. I. Description and basic properties of the unitary models. American journal of human genetics. 1978 Nov;30(6):618.

19. Cloninger CR, Rice J, Reich T. Multifactorial inheritance with cultural transmission and assortative mating. II. a general model of combined polygenic and cultural inheritance. American journal of human genetics. 1979 Mar;31(2):176.

20. Chen X, Ong YS, Lim MH, Tan KC. A multi-facet survey on memetic computation. IEEE Transactions on Evolutionary Computation. 2011 Oct 1;15(5):591-607.

21. Ong YS, Lim MH, Chen X. Research Frontier-Memetic Computation—Past, Present & Future. IEEE Computational Intelligence Magazine. 2010 May 5;5(2):24.

22. Dawkins R. The selfish gene. Oxford university press; 2006 Mar 16.

23. Cavalli-Sforza LL, Feldman MW. Cultural versus biological inheritance: phenotypic transmission from parents to children.(A theory of the effect of parental phenotypes on children's phenotypes). American journal of human genetics. 1973 Nov;25(6):618.

24. Feldman MW, Laland KN. Gene-culture coevolutionary theory. Trends in Ecology & Evolution. 1996 Nov 30;11(11):453-7.

25. Krawiec K, Wieloch B. Automatic generation and exploitation of related problems in genetic programming. In IEEE Congress on Evolutionary Computation (CEC), 2010 Jul 18 (pp. 1-8).

26. Bean JC. Genetic algorithms and random keys for sequencing and optimization. ORSA journal on computing. 1994 May;6(2):154-60.

27. Gonçalves JF, Resende MG. Biased random-key genetic algorithms for combinatorial optimization. Journal of Heuristics. 2011 Oct 1;17(5):487-525.

28. Chauhan P, Deep K, Pant M. Novel inertia weight strategies for particle swarm optimization. Memetic Computing. 2013 Sep 1;5(3):229-51.

29. Ong YS, Zhou Z, Lim D. Curse and blessing of uncertainty in evolutionary algorithm using approximation. In Congress on Evolutionary Computation, 2006 (pp. 2928-2935).

30. Ong YS, Keane AJ. Meta-Lamarckian learning in memetic algorithms. Evolutionary Computation, IEEE Transactions on. 2004 Apr;8(2):99-110.

31. Deb K, Agrawal RB. Simulated binary crossover for continuous search space. Complex Systems. 1994 Nov;9(3):1-15.

32. Watson RA, Jansen T. A building-block royal road where crossover is provably essential. In Proceedings of the 9th annual conference on Genetic and evolutionary computation 2007 Jul 7 (pp. 1452-1459). ACM.
33. Pan SJ, Yang Q. A survey on transfer learning. Knowledge and Data Engineering, IEEE Transactions on. 2010 Oct;22(10):1345-59.
34. Feng L, Ong YS, Lim M, Tsang I. Memetic Search with Inter-Domain Learning: A Realization between CVRP and CARP. Evolutionary Computation, IEEE Transactions on. 2014; 19(5):644-658.
35. Feng L, Ong YS, Tan AH, Tsang IW. Memes as building blocks: a case study on evolutionary optimization+ transfer learning for routing problems. Memetic Computing. 2015 Sep 1;7(3):159-80.
36. Besada-Portas E, de la Torre L, de la Cruz JM, de Andrés-Toro B. Evolutionary trajectory planner for multiple UAVs in realistic scenarios. Robotics, IEEE Transactions on. 2010 Aug;26(4):619-34.
37. Deb K, Sindhya K, Okabe T. Self-adaptive simulated binary crossover for real-parameter optimization. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007), UCL London 2007 (pp. 1187-1194). Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007), UCL London.
38. Poli R, Langdon WB. Schema theory for genetic programming with one-point crossover and point mutation. Evolutionary Computation. 1998;6(3):231-52.
39. Ong YS, Nair PB, Keane AJ. Evolutionary optimization of computationally expensive problems via surrogate modeling. AIAA journal. 2003 Apr;41(4):687-96.
40. Dantzig GB, Ramser JH. The truck dispatching problem. Management science. 1959 Oct;6(1):80-91.
41. Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research. 1987 Apr;35(2):254-65.
42. Avigad G, Moshaiov A. Set-based concept selection in multi-objective problems: optimality versus variability approach. Journal of Engineering Design. 2009 Jun 1;20(3):217-42.
43. Avigad G, Moshaiov A, Brauner N. MOEA-based approach to delayed decisions for robust conceptual design. In Applications of evolutionary computing 2005 Jan 1 (pp. 584-589). Springer Berlin Heidelberg.
44. Avigad G, Moshaiov A. Interactive evolutionary multiobjective search and optimization of set-based concepts. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on. 2009 Aug;39(4):1013-27.

45. Gupta A. Numerical modelling and optimization of non-isothermal, rigid tool liquid composite moulding processes (Doctoral dissertation, ResearchSpace@ Auckland). 2013.

46. Gupta A, Kelly PA, Bickerton S, Walbran WA. Simulating the effect of temperature elevation on clamping force requirements during rigid-tool Liquid Composite Moulding processes. Composites Part A: Applied Science and Manufacturing. 2012 Dec 31;43(12):2221-9.

47. Gupta A, Kelly PA, Ehrgott M, Bickerton S. A surrogate model based evolutionary game-theoretic approach for optimizing non-isothermal compression RTM processes. Composites Science and Technology. 2013 Jul 29;84:92-100.

48. Advani SG, Hsiao KT, editors. Manufacturing techniques for polymer matrix composites (PMCs). Elsevier; 2012 Jul 18.

49. Walbran WA. Experimental validation of local and global force simulations for rigid tool liquid composite moulding processes (Doctoral dissertation, ResearchSpace@ Auckland). 2011.

50. Jiang S, Ong YS, Zhang J, Feng L. Consistencies and contradictions of performance metrics in multiobjective optimization. Cybernetics, IEEE Transactions on. 2014 Dec;44(12):2391-404.

51. Hsu S, Ehrgott M, Kelly P. Optimisation of mould filling parameters of the compression resin transfer moulding process. In 45th Annual conference of the Operations Research Society of New Zealand (ORSNZ) 2010 Nov.