Swarm and Evolutionary Computation 48 (2019) 220–250

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Bio-inspired computation: Where we stand and what's next

Javier Del Ser [a,b,c,*], Eneko Osaba [b], Daniel Molina [d], Xin-She Yang [e], Sancho Salcedo-Sanz [f], David Camacho [g], Swagatam Das [h], Ponnuthurai N. Suganthan [j], Carlos A. Coello Coello [i], Francisco Herrera [d]

[a] University of the Basque Country (UPV/EHU), 48013, Bilbao, Spain
[b] TECNALIA, 48160, Derio, Spain
[c] Basque Center for Applied Mathematics (BCAM), 48009, Bilbao, Spain
[d] DaSCI Andalusian Institute of Data Science and Computational Intelligence, University of Granada, 18071, Granada, Spain
[e] Middlesex University London, The Burroughs, London, NW4 4BT, United Kingdom
[f] University of Alcala, 28871, Alcala de Henares, Spain
[g] Universidad Autonoma de Madrid, 28049, Madrid, Spain
[h] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, 700108, India
[i] CINVESTAV-IPN, 07360, Mexico, DF, Mexico
[j] Nanyang Technological University, 639798, Singapore

ARTICLE INFO

Keywords:
Bio-inspired computation
Evolutionary computation
Swarm intelligence
Nature-inspired computation
Dynamic optimization
Multi-objective optimization
Many-objective optimization
Multi-modal optimization
Large-scale global optimization
Topologies
Ensembles
Hyper-heuristics
Surrogate model assisted optimization
Computationally expensive optimization
Distributed evolutionary computation
Memetic algorithms
Parameter tuning
Parameter adaptation
Benchmarks

ABSTRACT

In recent years, the research community has witnessed an explosion of literature dealing with the mimicking of behavioral patterns and social phenomena observed in nature towards efficiently solving complex computational tasks. This trend has been especially dramatic in what relates to optimization problems, mainly due to the unprecedented complexity of problem instances, arising from a diverse spectrum of domains such as transportation, logistics, energy, climate, social networks, health and industry 4.0, among many others. Notwithstanding this upsurge of activity, research in this vibrant topic should be steered towards certain areas that, despite their eventual value and impact on the field of bio-inspired computation, still remain insufficiently explored to date. The main purpose of this paper is to outline the state of the art and to identify open challenges concerning the most relevant areas within bio-inspired optimization. An analysis and discussion are also carried out over the general trajectory followed in recent years by the community working in this field, thereby highlighting the need for reaching a consensus and joining forces towards achieving valuable insights into the understanding of this family of optimization techniques.

1. Introduction

Over millions of years, Nature has evolved to give rise to intelligent behavioral characteristics and biological phenomena, where adaptability, self-learning, robustness, and efficiency enable biological agents (such as insects and birds) to undertake complex tasks. While cases exemplifying these capabilities are truly multi-fold, the most illustrative ones revolve around the social behavior of animals such as ant colonies, beehives and bird flocks, where concepts such as stigmergy and the collective swarming movement of organisms often lead to the so-called Swarm Intelligence (SI), where improved exploration mechanisms over complex search spaces can be achieved by agents obeying local rules without any central control. The overall functionalities of the swarm are much richer than the simple sum of individual actions. Similarly, other renowned examples arise from the genetic inheritance process, the immune system of the human body or the neural activity

* Corresponding author. University of the Basque Country (UPV/EHU), Alameda Urquijo S/N, 48013, Bilbao, Bizkaia, Spain.
E-mail address: javier.delser@tecnalia.com (J. Del Ser).

https://doi.org/10.1016/j.swevo.2019.04.008
Received 10 December 2018; Received in revised form 11 April 2019; Accepted 15 April 2019
Available online 29 April 2019

of the brain. We refer to Ref. [1] for a comprehensive material summarizing these inspirational sources found in Nature.

Inspired by different behaviors observed in biological systems, many researchers in the research community investigating on computational paradigms have emulated intelligent bio-inspired processes in the form of computational algorithms, in an attempt to mimic the inherent advantages of such biological systems to address complex modeling, simulation, and optimization problems. In this regard, special attention has been paid to optimization problems, whose complexity has unleashed a rich substratum where to grow many bio-inspired population-based heuristic approaches, each differently balancing between computational efficiency and optimality of solutions. While the first contributions in this area are largely based on observation and emulation of Darwinian evolutionary principles, nowadays the number of bio-inspired solvers in the literature has increased dramatically, with very diverse inspirational rationale underneath their algorithmic design. This spotted flourishing of novel bio-inspiration optimization methods becomes even more intense when shifting the focus on other aspects related to optimization, such as multi-objective criteria, evolving (dynamic) optimization problems or distributed computing schemes, to mention a few.

However, quantity and diversity do not necessarily reflect scientific value when it comes to science. The development of the field has lately undergone a *gold rush* for bio-inspired streamlines that stimulate new algorithmic strands, around which some controversy has sprung regarding their relevance and novelty [2]. Debates around this topic are counterproductive, for which they waste efforts towards research directions with scarce – or even null – added scientific value. The same may occur in other research subareas as the ones exemplified above, where most algorithmic contributions build upon empirical performance observations rather than upon a deep, thoughtful and rigorous analysis of their design and internal operation. Futile debates should set aside to allow the entire community to start over with a clean common ground on the key research directions to be pursued in the future. We must ally to focus our efforts on important unresolved questions that can potentially produce greater insights into bio-inspired optimization techniques, ultimately leading to valuable advances and improved methods. Without a consensus, research niches of acknowledged relevance in the field will remain largely unexplored and unfairly dominated by controversial discussions, subtle and incremental algorithmic proposals, and a worrying lack of fresh breezes and fertile prospects.

This work responds to this need for a common meeting point by suggesting the audience to pause and reflect on which research directions should be pursued in the future in regards to bio-inspired optimization and related areas. For this purpose we pay special attention to numerical optimization, which was underneath the advent of the first bio-inspired solvers that were later adapted to combinatorial optimization. In this manuscript we provide an informed insight of the status of this field from both descriptive (*where we stand*) and prescriptive (*what's next*) points of view. This manuscript suggests and highlights several key research challenges that should captivate newcomers and experienced researchers for years to come, with scientific soundness at the core of their *raison d'être*. We hope that our envisioned future for bio-inspired computation acts as a suggestive guiding light for the community, bringing together different views that have remained so far quite different from each other to date, and potentially unifying them into a comprehensive multi-disciplinary view of the field.

The remainder of the paper is structured as follows: first, Section 2 provides a brief albeit informative overview of the history of bio-inspired computation. Section 3 and subsections therein undertake a comprehensive analysis of several areas of the field, stressing on their current status, trends, and open challenges. Section 4 elaborates on the general issues and research niches of bio-inspired computation, and finally Section 5 concludes this paper.

2. Recent history of bio-inspired computation

Bio-inspired computation has emerged as one of the most studied branches of Artificial Intelligence during the last decades. Hundreds of novel approaches have been reported along the years, showcasing the adoptability of different bio-inspired behaviors and characteristics to yield a near-optimal performance over a wide range of complex academic and real-world problems. This growing attention has led to a continuous increase in the number of publications related to the field, mainly focusing on the analysis, adaptation and/or improvement of different heuristic solvers.

Over the past, a diversity of optimization problems has been tackled using different bio-inspired techniques. According to the practical concerns of the time, the first ones addressed with this algorithmic portfolio were continuous and combinatorial optimization problems, which hinged on seminal formulations such as the Traveling Salesman Problem (TSP) or the Knapsack Problem (KP). In the late 60s and 70s, Fogel, Rechenberg and Schwefel reported their first pioneering studies related to Evolutionary Programming (EP, designed to optimize the behavioral linkage between solutions to a problem and their offspring) and Evolution Strategies (ES, conceived for numerical optimization since its very inception), laying the first compounding bricks of an incipient community that revolved around these concepts [3–5]. Another groundbreaking contribution was made by Holland with the publication of his seminal book in 1975 [1] on Genetic Algorithms (GA). This optimization method impacted deeply on the community at that time, unleashing a flurry of activity and a vast area of research that gave birth to what would be subsequently coined as bio-inspired computation. These three streams developed in isolation until the early 90s, when they came to be the cornerstone of a unified algorithmic branch denoted as Evolutionary Computation (EC, [6]). A special mention should be also given to Differential Evolution (DE) by Storn and Price [7], which embodied another breakthrough achievement within the EC community. Several other milestones in the EC realm were reported to the community in years thereafter, with emphasis on improved versions of nominal EC solvers (e.g. CMA-ES [8], IPOP-CMA-ES [9] or the more recent SHADE approach [10]).

Simultaneously to the forge of the EC field, the availability of more computational resources made it possible to cope with more complex and diverse optimization problems, especially those for which an analytical formulation cannot be stated mathematically (as occurs in e.g. simulation-based or black-box optimization). New problem flavors such as multi-modal and multi-objective optimization paved their way within a research community eager to delve into the benefits of bio-inspired computation. As for multi-modal optimization (i.e., problems where most of its multiple optimal solutions must be found), several studies have evinced that bio-inspired computation approaches can perform extremely well in practice [11–13], as opposed to classical optimization methods. In regards to problems comprising several conflicting objectives, the rise and development of this subarea has gone side by side with multi-objective optimization methods relying on bio-inspired processes (e.g. NSGA-II [14], MOEA/D [15], SMPSO [16] and others proposed along the years [17,18]). Recently, this stream has evolved to what is now referred to as *many-objective optimization*, an emerging topic where bio-inspired computation still prevails [19]. Another class of optimization problems gaining momentum today is Large-Scale Global Optimization (LSGO), which aims at dealing efficiently with the ever-growing complexity of real-world problems in any of its design dimensions (number of objectives, variables and/or constraints). LSGO techniques are designed to capitalize on the higher availability of computational resources and the emergence of new computing paradigms for massively parallel processing, which unleashes a very suitable scenario for the design and development of parallel or multi-population bio-inspired approaches. Some comprehensive compendiums on this noted suitability can be found in Ref. [20]. In addition, another class of optimization problems with the assumption that fitness function(s) and con-

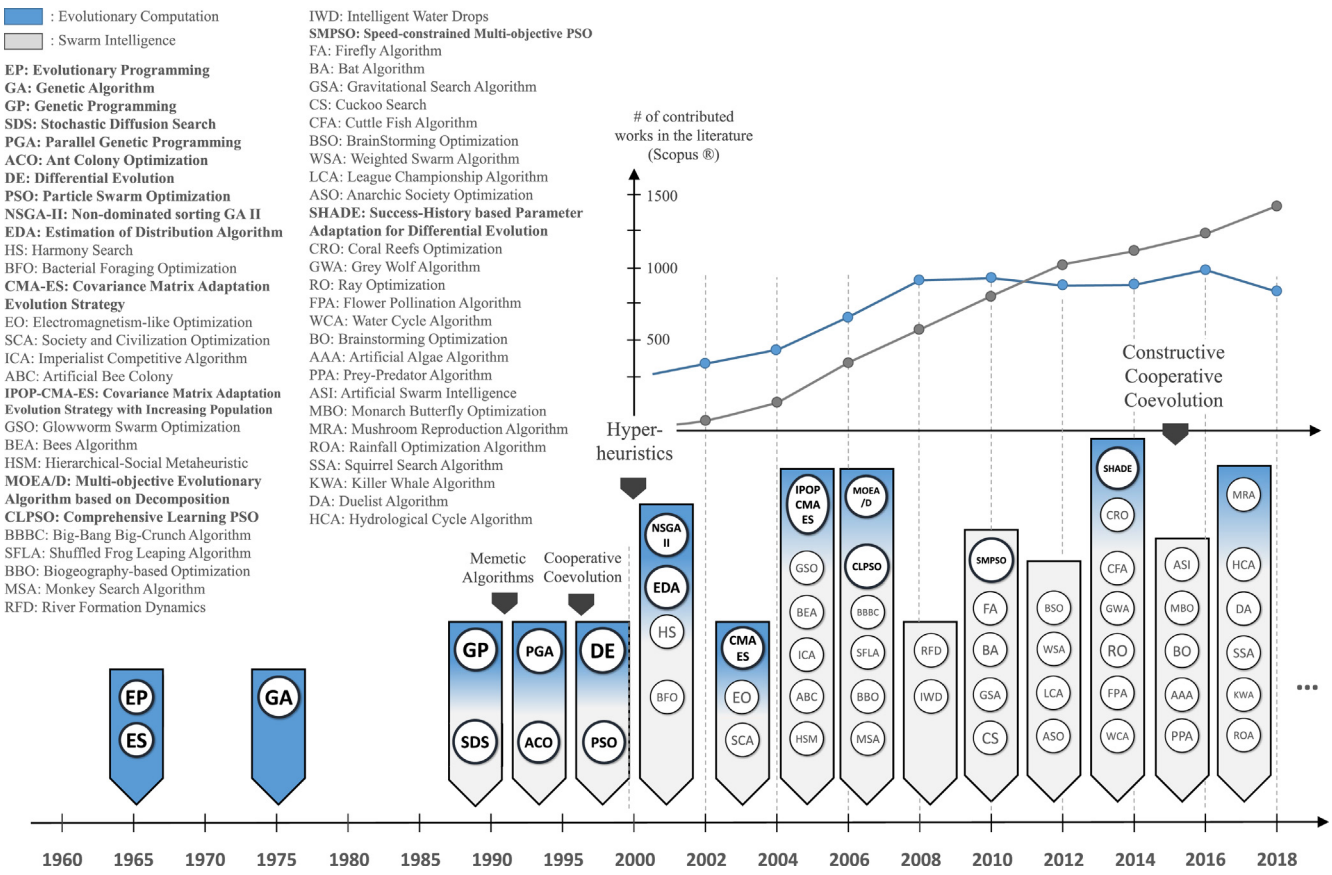

Fig. 1. Time line of the history of a representative excerpt of Nature-inspired optimization techniques (both bio-inspired and physics-based methods), where each technique is represented by its acronym inside a circle. Techniques that have posed a renowned milestone in the history of bio-inspired computation are highlighted in bold font, whereas modern nature-inspired algorithms, depicted in light gray, are included just for illustrative purposes. The figure also includes a nested plot depicting the number of published contributions related to EC and SI in the last two decades. Data retrieved from the Scopus® database (November 1st, 2018). The steady behavior of EC over recent years clashes with the rapid bibliographic growth of SI.

straints can be unstable and varying over the period in which the produced solution is applied. As such, dynamic optimization techniques are designed to deal with problems that evolve along time, a more realistic setup when the problem is defined over non-stationary scenarios [21].

In parallel to the maturity reached by EC, bio-inspired solvers evolved to algorithms with a higher level of sophistication, embracing the emulation of other processes and behaviors observed in Nature. The most acknowledged milestone in this regard is the birth of what is currently known as Swarm Intelligence (SI, [22]), a branch of bio-inspired computation based on the emergence of collective intelligence from large populations of agents with simple behavioral patterns for communication and interaction. This is essentially the fundamental principle behind Ant Colony Optimization (ACO, [23]) and Particle Swarm Optimization (PSO, [24]), arguably the first bio-inspired methods that fell within the SI umbrella. A myriad of bio-inspired SI methods have been proposed ever since, with inspirational motifs found in phenomena observed in Nature. Two branches tie together within this category. The first one relates to biological processes, such as the geographical distribution of biological organisms (Biogeography-Based Optimization [25]) or the chemotactic movement of bacteria (Bacterial Foraging [26]). The second inspirational motivation is the behavioral patterns of animals, such as bees (Artificial Bee Colony [27]) or fireflies (Firefly Algorithm [28]). The current list of bio-inspired SI is huge and grows almost on a daily basis. Exhaustive compendiums can be found in Refs. [29,30].

Social and political behaviors have also served as an inspiration for the proposal of many heuristic solvers, for instance, the adoption of political practices such as imperialism (Imperialist Competitive Algorithm [31]) or anarchy (Anarchic Society Optimization [32]). On the other hand, social attitudes have also inspired several methods, such as Society and Civilization [33] or hierarchies in which human organizations and structures are often arranged [34]. Likewise, physical processes have also laid the foundation for several optimization approaches, such as the dynamics of water (Water Cycle Algorithm [35] or Hydrological Cycle Algorithm [36]), gravitational kinematics (Gravitational Search Algorithm, [37]) or optics systems [38]. A recent survey provides a thorough overview of this literature strand which, jointly with biologically inspired solvers, gave birth to the immense field of *nature-inspired* optimization [39]. We note, however, that solvers inspired by physical processes or social behaviors have been remarkably outnumbered by those with a biological simile at their core. This being said, even though most conclusions, drawn in this manuscript also hold for optimization algorithms inspired in Physics or social behavioral patterns, we will concentrate our discussion on bio-inspired optimization techniques due to their higher prominence and relative maturity in current research.

Apart from bio-inspired EC and SI methods, which are monolithic approaches with their own intrinsic characteristics, an uprising trend has lately protruded over the literature mainstream: the hybridization of different algorithms. Since the dawn of bio-inspired computation, many researchers have focused their efforts on combining different methods and functionalities into a single solver in order to overcome the disadvantages and/or improve the performance shown by off-the-shelf methods. In this regard, Memetic Algorithms (MA) spearhead this

design principle by exploiting the synergy between a bio-inspired global optimization approach with local search procedures tightly coupled to the problem at hand [40]. The intense activity around MA has made them constantly grow to furnish more complex solvers characterized by extremely sophisticated cooperative mechanisms. In addition, Cooperative Coevolution (CC, [41]) has also played a historical role in the bio-inspired realm for its notable performance in highly complex problems, mainly by virtue of its capability to divide them into subcomponents (*species*) that are solved independently from each other.

For the sake of a better understanding of its development so far, Fig. 1 depicts a historical timeline of bio-inspired computation to date, spanning from the seminal contribution of EP and ES to the bibliographic flood of metaphor-based SI methods registered in the last couple of decades. When it comes to scientific production, EC captained the prominence of literature in the dawn of this field, with GA and DE at its foremost. A quantitative estimation supporting this statement can be made by inspecting the reputed Scopus® scientific database, which indicates that more than 13,000 works have been published in this topic since the beginning of the present century, blossoming with particular strength during the last decade (approximately 1000 papers published on average every year). Despite slightly less prolific on an aggregate basis (more than 12,000 works since year 2000), SI methods have kept on a par with their EC counterparts. However, SI has lately become the highest growing field of bio-inspired computation, thriving at a notable pace from barely 400 contributions in 2007 to more than 1200 in 2017. Indeed, the interest garnered by this branch has been *in crescendo* at such a rate that the number of published works related to SI when compared to that of EC becomes greater every year since 2012. The nested plot in Fig. 1 illustrates the yearly scientific production in which the above facts have been noted. Despite enlightening with respect to the activity in the field we note, however, that these bibliographic productivity numbers should be assessed with forewarning caution, due to the bias that the aforementioned excess of metaphor-based works may have introduced in the reported statistics.

This section may provide the reader with a general overview of bio-inspired computation, but the literature focused on this field is immense, with thorough bibliographic compendiums already contributed in the form of exhaustive surveys [42–45]. It is our intention not to review once again the existing work on the topic, but rather to identify and discuss upcoming challenges and potential research directions in the most promising hot topics of the field. We now proceed with this critical analysis.

3. Bio-inspired computation: where we stand and what's next

Bio-inspired computation is a broad field composed by multiple interconnected research areas. A thorough comprehensive review of the state of the art of all such areas would be counterproductive in our attempt at prioritizing research efforts in a global scale. For this reason, in this section, we stress on a reduced subset of research areas which, as shown in Fig. 2, have been particularly trendy in the last couple of years. Our analysis, schematically summarized in Fig. 3, emphasizes on the identification of research niches that still deserve further attention by the community, as well as on possible algorithmic synergies between different areas or between bio-inspired computation and other knowledge disciplines that could eventually disclose uncharted routes for further investigation.

3.1. Theoretical foundations

Even though bio-inspired computation has so far enjoyed great popularity, there still remains a wide gap to be bridged between the empirical performance assessment of bio-inspired optimization techniques and the rigorous mathematical understanding of some of their most important algorithmic properties such as the rate of convergence, computational complexity or statistical stability, among others. Before the modern computer era, only well-characterized problems could be tackled using mathematical programming methods and neat mathematical proofs. Meta-heuristics were developed thereafter to deal with non-rigorously defined optimization problems – even not formally defined at all, provided that a measure to compare any two solutions is available – for which traditional maths do not apply. With this prior in mind, the study of theoretical properties of bio-inspired heuristics can be thought to be, to a point, misaligned with the fundamental goal with

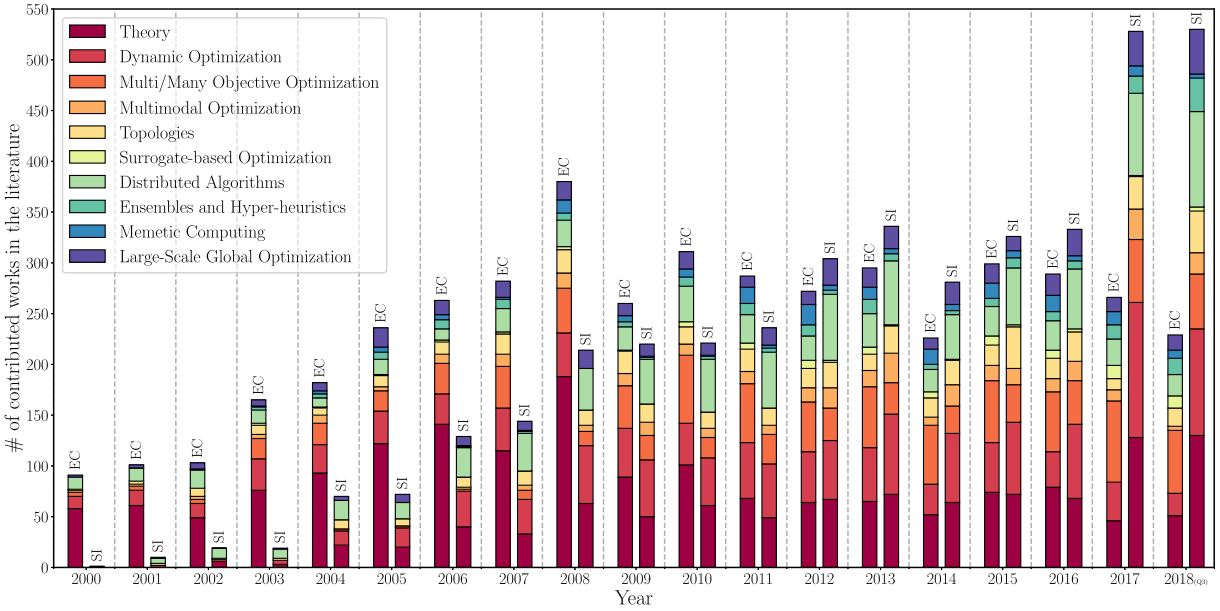

Fig. 2. Share of research contributions published within every area of bio-inspired computation addressed in Section 3. Data retrieved from the Scopus® database (November 1st, 2018). Publication counts have been produced by submitting to Scopus® the query `FIELD AND AREA`, e.g. `SWARM INTELLIGENCE AND MULTIMODAL`. Some variants of the terms have been also used for the sake of a maximum coverage per topic, such as `SWARM INTELLIGENCE AND (THEORY OR THEORETICAL)`.

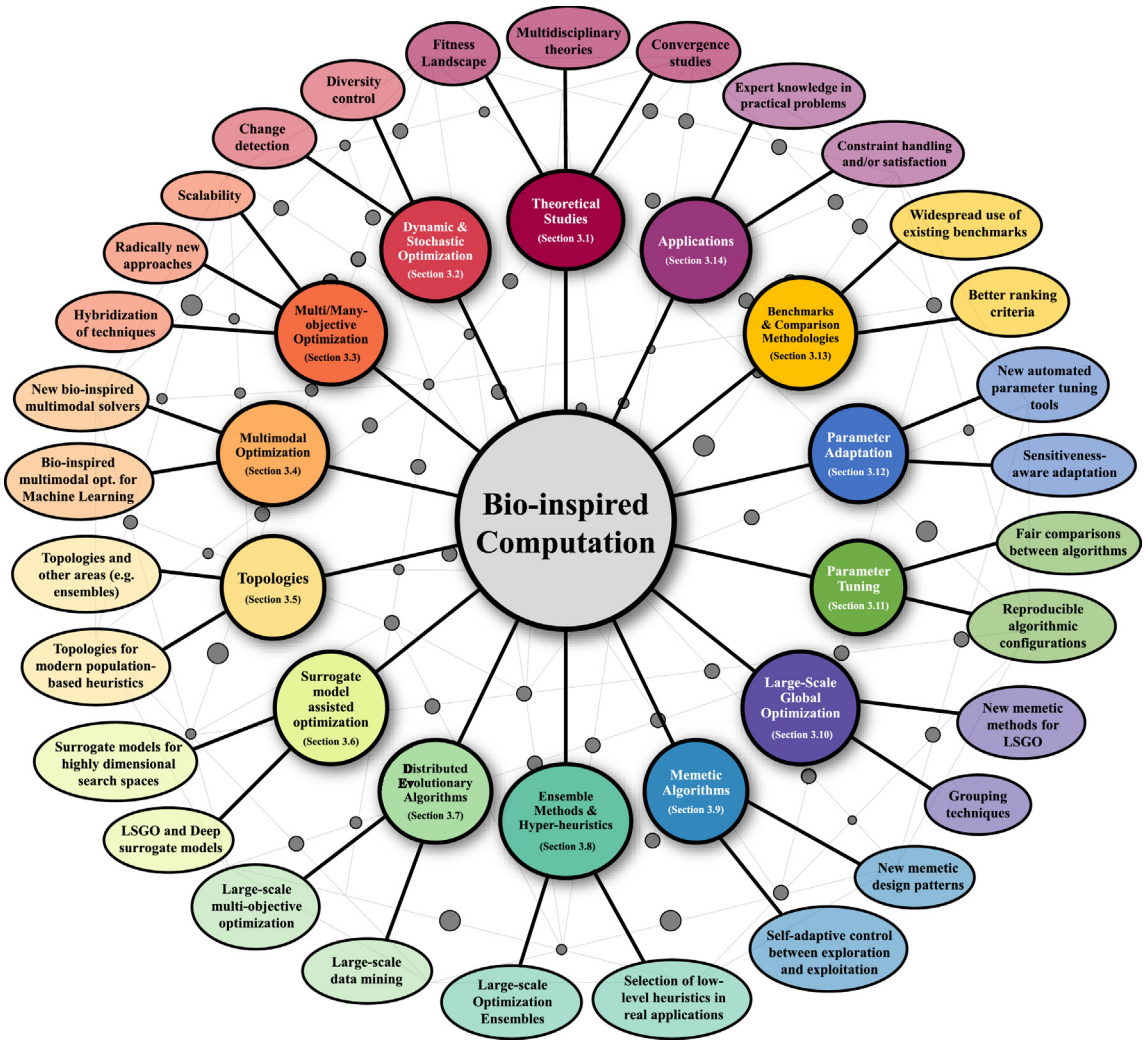

Fig. 3. Graph with first-level nodes representing the areas within bio-inspired optimization considered in our study, with some of their identified research niches stemming from each of such nodes.

which they were originally conceived [46]. Nevertheless, such analyses can yield valuable information on the fundamental reasons why a search algorithm performs empirically better than others for a certain class of problems. Indeed, the lack of a universally outperforming meta-heuristic for all kind of optimization problems yields from the well-known No Free Lunch Theorem for optimization, originally introduced in Ref. [47], further elaborated in Refs. [48–51], and recently revisited in Ref. [52]. This Theorem has been of central importance in the field of optimization meta-heuristics, as it states that no optimization algorithm can perform better than any other under any metric over all possible problems. This proven fact stimulates the need for developing theoretical studies on the properties of meta-heuristics, for which many authors have lately elaborated on different mathematical frameworks that we here overview, with the final aim at outlining future paths that should be followed in this area.

To begin with, the understanding of the working mechanisms of bio-inspired techniques has traditionally progressed by addressing fundamental questions on their convergence properties and computational complexity with different mathematical frameworks and tools [53,54]. One of the most studied topics is related to the convergence of these search algorithms, including the identification of the conditions under which such a convergence can be guaranteed. Following the insights drawn for Simulated Annealing and other heuristics by using Markov models, fixed-point theory, variance analysis, and dynamical systems, much attention has lately been paid to the extrapolation of these tools to model the agent-based behavior of population-based heuristics [45]. For example, the theory of dynamical systems has unveiled very interesting properties of PSO, such as the range of parameter values under which this algorithm can be proven to converge [55]. Logistic maps, a polynomial recurrence relation capable of modeling the non-linear demographic dynamics of a population of organisms, has been used for similar purposes [56]. An alternative approach to convergence analysis resides on the adoption of Markov Chain Monte Carlo methods to model the interactions between multiple search agents in bio-inspired solvers [57]. However, the use of this mathematical framework with more recent heuristics still remains largely unexplored for many other algorithms.

Another theoretical aspect that has undergone intense research in the last few years is the relationship between the solution space of the problem to be solved and the design of the bio-inspired optimization technique. In this regard, much attention has been devoted to the theoretical analysis of the so-called *landscape* of the problem, which is composed not only by the fitness function that assigns a metric value to a given candidate solution, but also by a neighborhood operator that relates different encoded solutions and the solution encoding strategy itself [58]. Fitness landscape analysis finds its motivation in the need for a better understanding of how a search algorithm can perform on a family of problem instances, rather than on a single problem instance.

To some extent, preliminary studies have shed light on commonalities within a given problem family that eventually lead to theoretically buttressed design directives for the search algorithm. For instance, the landscapes of many combinatorial optimization problems such as the symmetric TSP and the graph α-coloring problem are elementary on their own or can be decomposed into a number of elementary landscapes. Theoretical properties unveiled by landscape analysis regarding the suitability of neighborhood operators are favorable to the inclusion of local search methods within the bio-inspired solvers. An example of this design strategy is the work in Ref. [59], where appropriate bio-inspired meta-heuristics were selected for the protein structure prediction problem based on fitness landscape analysis with random walks. However, such approaches are often problem-specific and cannot be generalized to solve other problems.

Other interesting properties that can be inferred from landscape analysis range from the runtime performance estimation to the evolvability of a population of individuals induced by the landscape, the neutrality of the landscape, the discovery of optimal parametric settings for heuristic operators or the quantification of certain topological properties of the landscape of interest for the application of bio-inspired heuristics, such as the presence of attraction basins, barriers, or multi-modality or ruggedness (for which the concept of auto-correlation becomes essential) [60]. Reported attempts at establishing a formal methodology to decompose a problem in a series of elementary landscapes [61] are promising paths that should be continued further in the near future, for the community to argue their new bio-inspired design choices. This is the reason why landscape analysis is propelling a vibrant research activity in the last year [62,63], with recent insights being reported such as the need for a close match between the coordinate system used by the heuristic and the fitness landscape [64]. Unfortunately, such approaches can be largely problem-specific. This area is still at a very early stage of maturity to be embraced systematically by the community.

The study of the search efficiency of bio-inspired heuristics has also attracted great attention in recent times, especially in what regards to the balance between their diversification (exploration) and intensification (exploitation) capabilities [65,66]. The former refers to the generation of diverse candidate solutions within the search process leading to wide coverage of the solution space on a global scale, whereas the latter stands for the capacity of the solver to focus the search process in local regions of the solution space. In the wider context of randomized search heuristics (within which bio-inspired solvers can be thought to be a subset), many theoretical tools have been proposed to quantify numerically the aforementioned capabilities, such as the measurement of different metrics of diversity over the population of evolved solutions, that might be an indicator of the level of diversification of the algorithm along iterations. Interestingly, diversity has been identified as a driver for the success of heuristic solvers, to the point of replacing the fitness of the problem itself as the criteria to control the evolution of candidate solutions under what has been coined as *novelty search* [67]. However, it is also known that some loss of diversity is required for some heuristics to converge properly [68,69]. This suggests that more theoretical studies are needed in regards to the role of diversity and the balance of local search and global search required to undertake a certain problem efficiently. To shed light on these crucial aspects, we postulate that further developments around the study of landscapes should gradually consider behavioral aspects featured by specific bio-inspired algorithms, yielding a theoretical field that has been lately known as dynamic fitness landscape analysis [62]. Another valuable research direction for this purpose is the assessment of structural bias in population-based heuristics, namely, the limitation of certain heuristics to focus on some part of the solution space [70]. Due to their internal features, such algorithms may sample solutions more often either close to the origin, or close to the boundary, or close to any other specific part of the search space. Structural bias can degrade severely the performance of heuristic solvers, as has been already found for some important bio-inspired algorithms [71]. Similar studies alike should also be carried out with modern heuristics so as to clear up their behavior when sampling the solution space.

Some words of reflection should be posed around the fundamental comprehension of bio-inspired optimization approaches. A blossoming succession of innovative bio-inspired optimization methods has emerged during the last decade, particularly in metaphor-based SI. However, despite their inherent utility to discern the novelty of such methods with respect to the state of the art, theoretical insights on these methods have been reported at a significantly lower pace than their plain experimental performance assessment. This shortage of mathematical background is jeopardizing the discovery of new research directions in bio-inspired computation, reducing it to a desperate, senseless race for bio-inspired exoticism and creativeness. Efficient problem solving is by no doubt the ultimate goal of bio-inspired computation, goal for which Nature provides countless mechanisms for learning and self-adaptation in complex circumstances. However, the emulation of such mechanisms in the form of bio-inspired solvers should be grounded on a solid design rationale, either by showing theoretical findings that motivate design choices or by proving that the designed heuristic has improved theoretical properties in terms of convergence speed, fitness stability and other characteristics alike. For this to occur, more contributions combining empirical assessment and theoretical developments are needed in years to come so as to steer away from the '*look, it's working*' publishing frenzy witnessed in recent literature.

Finally, we believe that the theoretical and mathematical analyses of bio-inspired algorithms can be carried out from different perspectives so as to gain insight from different angles [45]. Interesting algorithmic properties can be studied using Markov chain theory, self-organized systems, filter theory, discrete and continuous dynamical systems, Bayesian statistics, computational complexity analysis and other frameworks [57]. Ultimately, truly in-depth understanding and insights will be gained by leveraging expertise, theories, and frameworks from mathematics, computer science, statistics, machine learning, control theory, complex systems, and other disciplines.

3.2. Dynamic and stochastic optimization

Most optimization problems historically addressed in the related literature built upon static fitness functions and constraint sets that do not vary along time. Furthermore, problems may rely on parameters whose values are assumed to be fixed and known a priori. However, in certain application scenarios, the dynamic characteristics of the environment where the problem is formulated do not meet these assumptions: the fitness function(s) and/or constraints are often subjected to non-stationary phenomena that make them prone to changes over time. Classical examples include the arrival of new tasks and/or machinery failures in production scheduling problems, or road accidents in traffic routing problems. These phenomena yield a stringent need for dealing with the obsolescence of the formulated problem in the heuristic search of the bio-inspired solver, as well as with the eventual uncertainty held on parameters participating in the problem definition. This dynamic context for optimization problems reflects realistically the intrinsic characteristics of avant-garde application domains such as Social Networks, Smart Cities, Industry 4.0 or Intelligent Transportation Systems, among many others. As such, dynamic optimization collectively refers to all those techniques tailored to efficiently undertake problems whose objectives and/or constraints may change along time. Likewise, stochastic optimization studies the development of solvers capable of dealing with uncertainty within the problem definition (involving e.g. random objective functions or random constraints). Such research areas are closely linked to one another, and both account for more realistic problem formulations in real setups.

To begin with, dynamic problem formulations can be found in optimization scenarios where the fitness landscape used by the solver is a function of time, thereby unchaining the need for reacting to eventual

changes in the landscape and hence, provide new optimal solutions at any time instant. This area has a long history in the literature, with surveys dating back as soon as the late past century [72], and studies contributed thereafter on different theoretical aspects, such as the behavior of evolutionary operators when dealing with dynamic problem instances [73,74]. In the last few years, dynamic optimization has ignited a great research activity, resulting in special issues, workshops and competitions [75,76] in frontline conferences and journals. In this regard, the presence of EC methods has been especially notable, among which two main algorithmic design mainstreams can be identified. In all of them, population diversity is renowned to be a design factor of pivotal importance, in a similar fashion to the well-known stability-plasticity dilemma in learning over non-stationary data streams:

- Active approaches, by which the change in the problem formulation is explicitly detected. The solver reacts upon a detected change by altering its standard search procedure (e.g. by resetting part of the population or increasing the mutation rate). Here, diversity is introduced *after* the change occurs, which makes this design approach closely dependent on the performance of the technique used to detect the change. This is the reason why many authors have lately invested efforts in enhancing the performance of change detectors when combined with bio-inspired solvers, either by evaluating periodically candidate solutions or by identifying whether key indicators of the solver's performance vary along time. In either case, both elements from EC and SI have been explored lately, with a dominance of the latter noted in the last couple of years [77]. Notable milestones in this field include the design of full or partial random restart strategies [78,79], the so-called hyper-mutation operator [80], which imprints an increase of the mutation rate once a change has been detected; or the migration of individuals among subpopulations in multi-population schemes for dynamic optimization, a strategy to actively diversify the pool of candidate solutions handled by the solver [81,82]. In this literature strand challenges reside in controlling the amount of diversity to be induced after the change (which should be coupled to the estimated characteristics of the detected change in terms of severity and speed), as well as in the development of new methods for change detection, specially for slowly-evolving and/or subtle problem changes.
- Passive approaches, in which diversity in the population/swarm is steadily maintained during the search process over time. In this case, the idea is to sacrifice search performance during those time intervals where the problem formulation can be deemed static for a better reaction of the algorithm when a change in the problem occurs. This is accomplished by injecting diversity in the population/swarm of the bio-inspired solver anyhow, to prevent it from converging to optima that might become eventually obsolete. This insertion of diversity has been so far done in many ways, including the generation of random candidate solutions, tailored crossover operators that favor the generation of offspring strongly differing from their parents, explicit convergence avoidance or the use of multiple subpopulations/swarms that are enforced to track changes of the problem with time [83]. Despite the obvious benefits of this approach with respect to its active counterpart (no need for explicit change detection, effective with sharp changes in the problem), there are still questions to be addressed in regards to the balance between diversity and performance, particularly in what refers to the coupling between the characteristics of the change (speed, severity) and the amount of diversity to be injected. As such, when the change evolves slowly and/or does not differ dramatically from the previous problem status, the diversity to be induced for optimal performance should be less than when the problem formulation changes radically in a certain instant of time. To this end, the interest has steered towards the use of self-adaptability in both active and passive approaches, always with bio-inspired solvers at their algorithmic core [77,84].

Research directions in the field of dynamic optimization are foreseen along the diverse axis: to begin with, most dynamic optimization problems tackled to date are defined over unconstrained, continuous-variable search spaces. This experimental choice finds its motivation in the lack of informed answers to the challenges still faced by the community in regards to the role played by the diversity in dynamic environments of varying characteristics. There are validated empirical pieces of evidence that align with the intuition, such as the need for ensuring a higher diversity in population-based heuristics the wider the differences between shifting problems are. While this statement has been assessed over extensive dynamic optimization benchmarks comprising continuous variable problems, this question still remains insufficiently addressed when the scope is placed on discrete problem formulations, evolving constraints and/or recurrent concepts. When this is the case, it is not clear yet how to control or even define the level of diversity injected to the solver [77].

There are other research venues around dynamic optimization that should be explored in the near future. Among them, change detection mechanisms relying on characteristics of the optimization algorithm itself are known to perform poorly when the optimization problem varies by any other reason than a change in the problem formulation itself, such as noisy objectives and/or constraints. In real environments, it is often the case that the objective evaluation is affected by external noise sources that make the fitness value vary over time. This must not be understood as a change in the problem formulation, but in this case detection mechanisms based on monitoring the fitness value might misinterpret it as a problem change and trigger a false alarm. Other dynamic scenarios where change detection mechanisms are known to fail catastrophically (e.g. slowly evolving/subtle problem changes) lead to the overall conclusion that change detection should be avoided and efforts rather invested on passive/self-adaptive schemes [77,85].

On the other hand, stochastic optimization is a research topic that came to public attention several decades ago [86–88], with comprehensive overviews contributed to the literature ever since [89,90]. This kind of optimization problems has rapidly grown in importance and led to the emergence of different trends such as Fuzzy Programming [91] or Stochastic Dynamic Programming [92]. Among them, Robust Optimization [93] has particularly protruded in last times: algorithms within this broad family of solvers define a so-called *uncertainty* set of possible realizations of the uncertain parameters underneath the problem at hand. Robust optimization focuses on optimizing against worst-case realizations within this set so that a guarantee of optimality can be given with respect to the defined uncertainty set [94]. Key design challenges in Robust Optimization reside, on one hand, in the choice of an *uncertainty* set properly suited to the problem at hand, so that a good balance between representativeness and conservativeness can be met for the problem at hand. For this purpose, several recent studies have elaborated on different methodological proposals to build good uncertainty sets, including approaches hinging on the partial availability of data for better adapting them [95,96]. On the other hand, a crescent need for robust optimization of simultaneous objectives has been noted lately in related contributions [97–99], with essential questions still to be addressed such as the definition of uncertainty in Pareto fronts. Given the paramount importance gained in recent times by EC and SI as algorithmic propellers of multi-objective solvers, we postulate that further developments should be made in deriving new bio-inspired search operators capable of handling problem uncertainty at the very core of their definition.

Another rich substrate of new research lines stems from the need for reliably modeling of real optimization scenarios where the characteristics of several of these problems hold, thereby hybridizing modeling aspects and assumptions from several classes of optimization paradigms. This is the case of large-scale stochastic optimization [100] and multi-objective stochastic optimization [101–104], which have been put to practice in different scenarios with high levels of realism and complexity such as Intelligent Transportation Systems, engi-

neering, robotics [105], power system planning [106], and hydrology [107]. Multimodal optimization in dynamic environments has been likewise addressed in Refs. [108,109]. Many sophisticated heterogeneous approaches have been proposed for addressing these demanding problems, such as Memetic Algorithms [110,111] or EC and SI solvers tailored to deal with real-world environments with noisy objectives [112]. In this context, we highlight the upturn of research around the confluence of Dynamic and Robust Optimization in what has been coined as Robust Optimization over Time (ROOT), in which the goal is to find solutions that remain acceptable (statistically robust) over the course of time [113,114]. Despite the relative age of this research area and its inherent practical interest, we foresee that significant advances can be done if an eye is kept on other sciences and disciplines dealing with uncertainty in dynamical systems, such as Risk Theory [115] or Lyapunov Optimization [116]. We expect that the hybridization of concepts and techniques from these fields with bio-inspired search operators will give birth to a new generation of robust bio-inspired solvers suited for dynamically evolving environments.

3.3. Multi- and many-objective optimization

Problems comprising several conflicting objectives have sprung a vibrant research activity during the last 20 years, with an ever-growing body of literature, competitions, and benchmark focused on deriving new bio-inspired solvers suited to produce Pareto optimal solutions with increased efficiency and efficacy. The old days in this subarea of bio-inspired computation focused on two main strategies to deal with multiple objectives [117,118]: 1) non-elitist non-Pareto-based methods, including lexicographic ordering, linear aggregating functions, VEGA, ϵ-constraint techniques and target vector approaches; and 2) non-elitist Pareto-based methods such as Pure Pareto ranking, MOGA, NSGA, NPGA, and NPGA-2. Nowadays most methods rely on some sort of elitism [119], yielding renowned schemes such as SPEA and SPEA2, NSGA-II, PAES, PESA, PESA II, μGA^2 and many others. More recent variants include MOEA/D (and its many variants), SMPSO, SMS-EMOA, HyPE and NSGA-III [120], which can be classified in three main families depending on their design strategy [121]:

- In Pareto-based methods the selection mechanism is based on Pareto optimality, for which most of them adopt some form of non-dominated sorting and a density estimator (e.g., crowding, fitness sharing, entropy, adaptive grids, parallel coordinates, etc.). A well-known limitation of this family of multi-objective solvers resides in its restricted scalability in objective function space, which is usually circumvented by using an overly large population size. Another alternative to overcome from this limiting issue is to change the density estimator, but this option has not been too popular.
- An alternative design choice often followed in multi- and many-objective optimization is to opt for decomposition-based methods. The core idea of these approaches is to transform a multi-objective problem into several single-objective optimization problems which are simultaneously solved using information from its neighboring subproblems. Unfortunately, the performance of decomposition-based MOEAs is strongly affected by the scalarizing function that they adopt, and are further sensitive to the method used to generate weights. This dependence entails a more tedious parametric tuning process when aiming to solve a certain problem, thus jeopardizing the universality of this optimization strategy. By contrast, as opposed to Pareto-based methods they are scalable in objective function space, although an increase in the number of objectives will require a higher population size and thereby, a heavier computational load of the overall solver.
- Another particularly profitable design strategy is to rely on a performance indicator for the selection of individuals during the search, as is done in indicator-based multi-objective solvers. However, the mere use of a performance indicator in the density estimator was discovered to suffice for rendering a good performance (as in e.g., SMS-EMOA in which the hypervolume is adopted as a density estimator that replaces the crowded comparison operator of NSGA-II). Unfortunately, the only performance indicator which is known to be Pareto compliant is computationally expensive in highly-dimensional objective spaces (i.e., the hypervolume). Many other performance indicators are available [122], some of which are weakly Pareto compliant (e.g., $R2$ and IGD+). However, they have not attracted much attention in recent literature.

A common practice of current research in multi-/many-objective optimization is to propose *new* algorithmic variants based on existing benchmark functions (e.g. ZDT, DTLZ, WFG, UF), or adapt bio-inspired solvers to accommodate multi-objectivity. The community will also continue producing new flavors of the most popular MOEAs in current use (i.e. MOEA/D and NSGA-III), but there is surely room for more creative enhancements and unprecedented algorithmic developments that can pave unexplored research avenues in this field. Some thoughts and niches of opportunity in this direction are outlined in what follows:

- There is an urgent need for new ideas regarding the design of multi-objective solvers that fall aside the current research mainstream represented by the three major design criteria noted above (i.e., Pareto-based, decomposition-based, and indicator-based). Looking into new ways of solving multi-objective problems may be more profitable, in the long term, than producing continuous updates to existing algorithms that, in most cases, are not adopted by an important number of researchers. The recent approach reported in Ref. [123] can serve as a simple example of the viability of this claim: in this work, a multi-objective optimization problem is transformed into a linear assignment problem using a set of uniformly scattered weight vectors. The uniform design is adopted to obtain the set of weights, and the Kuhn-Munkres (Hungarian) algorithm is used to solve the resulting assignment problem. This approach was found to perform quite well (and at a low computational cost) in many-objective optimization problems. This approach does not belong to any of the three previously indicated families of multi-objective algorithms. Besides exploring new research paths, it is important to gain a deeper understanding of the major algorithms in current use. For example, knowing that some scalarizing functions offer advantages over others [124] is very useful to design good decomposition-based and even indicator-based multi-objective solvers (algorithms based on $R2$ normally rely on decomposition).
- Another interesting idea is to combine components of different multi-objective techniques under a single framework that allows to exploit their advantages. This is the basic idea of Borg [125], which adopts ϵ-dominance, a measure of convergence speed called ϵ progress, an adaptive population size, multiple recombination operators and a steady-state selection mechanism. This hybridization of operators, in fact, can lead to the automated design of algorithms as has been already suggested by researchers from automated parameter tuning for single-objective evolutionary solvers. In this context, a very promising option is the use of hyper-heuristics to coordinate the use of several types of heuristics with the aim of combining their advantages in a wide class of problems. As we will later revisit in Subsection 3.8 (which is partly dedicated to this branch of bio-inspired computation), the idea behind hyper-heuristics is the use of a collection of basic (low-level) heuristics, which on their own do not produce good solutions to a given optimization problem, to come up with a much better solution by combining them (or by generating new heuristics from them) by means of a high-level algorithm [126]. Their main motivation is to have a more general search engine that can solve a wider variety of hard optimization problems. Hyper-heuristics have been mostly developed for discrete search spaces and have been used to solve mainly single-objective optimization problems. Surprisingly, few researchers have developed multi-objective hyper-heuristics, despite

their inherent potential to yield improved heterogeneous solvers for given families of multi-objective problems. For instance, McClymont and Keedwell [127] proposed one of the few multi-objective hyper-heuristics that has been designed for continuous optimization problems. Their approach applies a heuristic selection method modeled as a Markov chain to the DTLZ test problems. Maashi in her Ph.D. Thesis [128] proposed an online learning selection choice function based hyper-heuristic framework for multi-objective optimization. Her proposed approach controls and combines the strengths of three well-known bio-inspired multi-objective solvers (NSGA-II, SPEA2, and MOGA), which are adopted as her low-level heuristics. The indicator-based multi-objective sequence-based hyper-heuristic (MOSSHH) algorithm proposed in Ref. [129] was the first attempt to use a hyper-heuristic in many-objective problems. The study compares three indicators (one based on average rank, the hypervolume and the favor relation) with Pareto dominance in many-objective problems. In Ref. [130] a multi-objective hyper-heuristic based on a choice function was proposed to adaptively select appropriate low-level heuristics (operators) within MOEA/D. The pool of low-level heuristics consisted of five differential evolution operators. More recently, a hyper-heuristic was proposed in Ref. [131] to combine the strengths and compensate for the weaknesses of different scalarizing functions. The selection is conducted through an indicator called *s*-energy, which measures the even distribution of a set of points in *k*-dimensional manifolds. Combining different performance indicators within an indicator-based multi-objective solver is the proposal of [132], in which IGD^+, $\epsilon+$, Δ_p and $R2$ are adopted as possible density estimators (i.e., the low-level heuristics). Another strategy connected to hyper-heuristics and automatic algorithm composition is the combination of different off-the-shelf algorithms under a single control mechanism, as done in e.g. AMALGAM [133].

Despite the impulse around multi-objective hyper-heuristics exposed above, we still need more theoretical studies to better comprehend their superior performance in the reported cases so far. In fact, some work in that direction has been already done. For instance, Qian et al. provided in Ref. [134] a theoretical study on the effectiveness of selection hyper-heuristics for multi-objective optimization, concluding that selection hyper-heuristics applied to any of the three major components of a multi-objective evolutionary algorithm (selection, mutation, and acceptance) can exponentially speed up the optimization process. More theoretical findings to be contributed in the future should unveil new possibilities for the design and construction of hybrid multi-objective optimization algorithms, specially bearing in mind the emergence of new bio-inspired search operators.

- An aspect that has also attracted great attention in the last couple of years is the scalability of multi-objective techniques when addressing problems with many objectives. The reason behind this research trend lies on the fact that off-the-shelf multi-objective solvers do not scale properly under such circumstances. For instance, the number of non-dominated solutions is known to grow exponentially with the number of objectives [135], which makes the selection mechanism in Pareto-based methods completely useless since all the non-dominated solutions are considered equally good. There is also another interesting problem related to scalability: as we increase the number of objectives, the number of solutions required to sample the Pareto front also grows exponentially, further complicating an efficient exploration of the search space of the problem at hand. In this context it is also interesting to highlight the empirical work reported in Ref. [136], where it was shown that a random search is more effective than NSGA-II when dealing with more than 10 objectives. As a result, many-objective optimization deals with the design of scalable search algorithms for problems characterized by many (typically more than 3) objective functions.

In the early days of this area, two types of approaches were commonly adopted to cope with many-objective optimization problems: 1) to adopt or propose a preference relation that induces a finer grain order on the solutions than that induced by the Pareto dominance relation; or 2) to reduce the number of objectives of the problem during the search process or during the decision making process. Many other approaches are possible, including, for example, the use of machine learning techniques (as in MONEDA [137]), performance indicators (as in SMS-EMOA [138]), ϵ-dominance or the two-archive MOEA, which uses one archive for convergence and another for diversity [139].

The source of difficulty in many-objective problems has been extensively studied in recent times, with specific complexity factors that by themselves open up new challenging avenues in this area. Several works are the baseline reference in this regard: to begin with, Ishibuchi et al. considered in Ref. [140] five types of difficulties that arise in many-objective problems, including the typical ones (e.g., difficulties to generate a good approximation of the entire Pareto front) and others that are not so obvious (e.g. difficulties to assess performance). By that time, Schütze et al. had already concluded that adding more objectives for a given problem does not necessarily make it harder [141], an insight that was later empirically confirmed in Ref. [140] by showing that NSGA-II could properly solve many-objective knapsack problems whose objectives were highly correlated. However, many other challenging aspects related to many-objective optimization deserve to be studied in the near future, from density estimators (what to use and what sort of distributions do we aim to find?), the visualization of high-dimensional Pareto fronts and performance indicators suited for this family of optimization problems (at least not as expensive as the hypervolume).

- A multi-objective optimization area that has lately attracted attention is the scalability of multi-objective solvers in terms of the number of decision variables. Almost no work had been published on this topic until a few years ago, when a small study reported results in problems that were scaled up to 100 decision variables [15], later extended to more than 2000 variables and several multi-objective solvers [142]. Remarkably, in this work, PAES was found to be the most salient technique from the several compared (NSGA-II, SPEA2, MOCell, OMOPSO, and PESA-II). OMOPSO did very well up to 256 decision variables and ranked second between 512 and 1024 decision variables. Years later, the first multi-objective solver designed for large-scale multi-objective optimization was proposed in Ref. [143], where CC (Cooperative Coevolution) was used to tackle problems amounting up to 5000 decision variables.

Although other large-scale multi-objective optimization techniques have been proposed since then [144–146], several topics remain to be explored in this area. For example, there are no many sets of test problems explicitly designed for testing large-scale multi-objective problems [147]. Furthermore, large-scale many-objective optimization problems have recently entered the research arena, e.g. Ref. [148], thereby unfolding many challenging paths blending together complexities from both areas.

Clearly, the main challenge for the coming years is to continue to open new venues of research in bio-inspired multi-/many-objective optimization. This is becoming increasingly difficult, given the huge volume of research that has been conducted and is currently ongoing around the world. We need to be more creative: there are still plenty of topics to study within this field, but some of them require moving outside the main stream of the research that is currently being conducted. For example, we need new performance indicators, particularly for many-objective optimization. We lack appropriate performance indicators for assessing diversity in many-objective optimization, although there are some interesting choices (e.g. *s*-energy). It is also important to design new mechanisms (search operators, encoding strategies) realistically modeling specific features of real-world problems (e.g., heterogeneous and/or variable length encoding [149–151], expensive objective functions, uncertainty or highly-constrained search spaces [152]). Co-

evolutionary approaches can help solve complex multi-objective optimization problems, which also unleashes another interesting path for future research. Besides large scale problems, co-evolution can be useful in other domains – e.g. dynamic multi-objective optimization problems [81] – but its potential has been scarcely studied [153].

However, it is important to keep in mind that a great source of diversity regarding research ideas is the knowledge coming from other disciplines. For example, our field has adopted advanced data structures (e.g., red-black trees), concepts from computational geometry (e.g., convex hulls, quadtrees and Voronoi maps), and from economics (e.g., Game Theory) to design novel search algorithms and operators. We also need to explore more ways of bridging the gap between this field and elements from Operations Research, such as mathematical programming [154]. Another one is the use of the Karush-Kuhn-Tucker optimality conditions to estimate proximity of a solution to the Pareto optimal set [155]. All in all, diversity, heterogeneity and synergy between different disciplines must be ensured within the community working in order to attain more disruptive and scientifically valuable advances not only in multi-/many-objective optimization, but also in other areas within bio-inspired optimization. Otherwise, if we only investigate by analogy, research in this area will eventually suffer from stagnation.

3.4. Multimodal optimization

Multimodal optimization problems consist of finding multiple optimal solutions within a single algorithm run, in order to have a better knowledge about the different solutions in the search space [156]. Population-based bio-inspired algorithms are the most used approaches to solve multimodal optimization problems, specially those including diversity-preserving mechanisms, also known as *niching methods* [157,158]. Research on multimodal optimization techniques, mainly evolutionary algorithms, had a boom in the early 2000s, and has been a research topic of importance until today, as can be seen in Fig. 2 (stacked bar colored as ■). According to Ref. [158], there are several reasons why multimodal search is important in real-world problems:

- Locating multiple optimal solutions of a problem improves the knowledge of the problem domain, much more than in single-solution cases.
- Multiple solutions with optimal quality provide a decision maker with a wider portfolio of options for consideration, based on which different factors can be applied to choose the best option in each case.
- Looking for multiple optimal solutions can be good in terms of improving the search capabilities of the algorithm since the computational effort is diversified to explore different regions of the search space.
- Diversity methods involved in multimodal optimization algorithms are able to improve the quality search of the algorithm since they counteract the effect of genetic drift, which causes diversity loss in bio-inspired algorithms.

Motivated by these benefits, the literature has been rich in regards to niching approaches that have been combined with bio-inspired heuristics in order to help them properly search over multimodal fitness landscapes. For the sake of completeness, in what follows we briefly describe some of the most classical approaches for niching in evolutionary algorithms [157]:

- Fitness sharing, originally introduced in Ref. [159], consists of dividing the population of a bio-inspired algorithm into subgroups, according to a measure of similarity between solutions. Thus, an individual must share its information with individuals belonging to the same *niche* area. To this end, a rule must be defined by which every individual's fitness is decreased by an amount depending on the number of similar individuals in the population. A usual way to proceed in this matter is to modify the fitness of the i-th individual in the population as $f_i' = \frac{f_i}{m_i}$, where m_i is known as the niche count measuring the approximate number of individuals close to the i-th individual. The niche count is usually established depending on the similarity d_{ij} between the i-th and j-th individual. The similarity d_{ij} is typically measured over the genotype or the phenotype of the problem.
- The clearing method is very close to the fitness sharing approach but is instead based on the concept of limited resources of the environment [160]. Instead of sharing resources between all individuals of a single subpopulation as in fitness sharing, clearing methods assign them only to the best members of the subpopulation, in such a way that this operator preserves the fitness of the some of the best individuals (dominating individuals) of the niche, and resets the fitness of the others individuals in the population (correspondingly, dominated individuals).
- In crowding schemes for multimodal optimization [161], only a percentage of the population reproduces and dies in each generation of the evolutionary search. The newly generated individuals replace similar ones in the population. For this purpose, a subset of individuals is drawn at random from the population, whose cardinality (in % with respect to the population size) is referred to as *Crowding Factor*. Then, a new generated individual replaces the most similar element in this sample, taking into account a distance function d_{ij} similar to the one defined above. This approach was improved in Ref. [162] leading to probabilistic crowding, where different tournaments between similar individuals (parents and offspring) are carried out. When producing offspring, however, the whole population is used, hence parents can be potentially drawn from different niches. This results in a poor exploitation capability of the overall search heuristic in distinct niches, thereby motivating alternative strategies as the ones following hereafter.
- Speciation [163] and islanding [164] are other ways of dealing with multimodal optimization problems. On one hand, the speciation technique divides the population into several species according to their similarity, so each species focuses on a different solution of the problem. Islanding follows the same design principle but defines several subpopulations or islands where individuals are independently evolved.
- The induction of niching behavior within the individuals produced by a bio-inspired solver can make them converge efficiently around different basins of attraction. To this end, in addition to the general niching strategies outlined above, an alternative is to exploit the concept of neighborhood, a relationship between individuals over a distance space that can be used, for instance, to adaptively adjust the amplitude or frequency by which bio-inspired operators are applied, replace the stochastic nature of parent selection strategies, or to control the migration of individuals between different basins.

Two main niching strategies can be adopted when facing multimodal optimization problems with population-based meta-heuristics: 1) index-based topologies for niching, which define the neighborhood of an individual based on its index within the population and a predefined topology (e.g. a ring or a star); and 2) Euclidean distance based niching, which fixes the neighborhood relationship between encoded solutions in the population based on their Euclidean distance. Euclidean neighborhood was first used to generate offspring in Ref. [165] for single-objective bound constrained problems. Subsequently [166], was the first work to use Euclidean distance for generating offspring within each neighborhood, enhancing, as a result, the local exploitation around each niche. Several other contributions have since then used Euclidean distance based neighborhood niching induction for multimodal problems [167–170]. However, the use of neighborhoods and other concepts related to topology spans further possibilities in other areas of bio-inspired optimization, such as multi-objective optimization (e.g. MOEA/D resorts to the definition of a neighborhood structure to transfer information between subproblems).

In the last few years, bio-inspired solvers have thrived as the default algorithmic choice in multimodal optimization, producing new algorithms and drawing up new paradigms related to this optimization casuistry. Of special interest are the findings reported in Ref. [171], where it was shown that a single-objective multimodal problem can be reformulated as a bi-objective optimization problem, so that heuristics developed for this alternative problem statement were proven to efficiently tackle more complex multi-modal search spaces than contemporary algorithms, comprising a larger number of variables and number of optima. Beyond the relatively improved performance of its proposed approach, this work paved the way towards the application of multi-objective bio-inspired solvers to the multimodal case, and further exemplified how the synergy between different areas in bio-inspired computation can yield new efficient approaches to tackle classical problems. In fact, this intuition has been at the core of a number of recent publications, see e.g. Refs. [167,172] and citing contributions thereafter. It is also worth noticing that niching is also considered a very effective strategy to improve the performance of multi-objective algorithms [158], an observation which reinforces even further the aforementioned profitable synergy between both areas of bio-inspired optimization. Research efforts in the future should strive for exporting advances of one area to another and vice versa. This being said, late advances in many-objective optimization could serve as a source of inspiration to develop new approaches for effectively solving multimodal optimization problems.

An alternative research line in multimodal optimization aims at defining tailored solvers with biological inspiration at their core to when addressing these problems, among which we underscore the prominent role taken by PSO in recent times [173–175]. Indeed, the notion of memory, inherent to PSO, is exploited to induce niching behavior. Usually, a swarm is divided into two parts: 1) an *explorer* swarm (with the current particles) and 2) a *memory* swarm (with only the best particles found so far by the algorithm). The underlying idea is that best particles within the memory swarm act as solution niches, eventually locating multiple solutions to the problem. Alternative proposals to multimodal problems using PSO algorithms have gravitated on the use of multi-swarms [176], the induction of Euclidean-based niching [168] or a ring topology in the neighborhoods within the swarm [177]. Another bio-inspired solver successfully applied to multimodal optimization is DE, which has been hybridized with fitness sharing [11], speciation and islanding [178]. In light of this preceding research record, it is only a matter of time that the community will receive studies dealing with the hybridization of new bio-inspired techniques with traditional strategies to tackle multimodal problems.

Finally, Machine Learning should capitalize on bio-inspired solvers for multimodal optimization, in problems related to feature selection [179] or model calibration [180]. We do think that this is an extremely interesting research line that spans far beyond multimodal optimization, which we expect will be further growing in the near future. We will later revolve on this statement.

3.5. Topologies

Contributions related to the design of topologies within population-based heuristics elaborate on the control of the information flow among population members by specifying the nature and outreach of their interactions [181,182]. By appropriately defining topologies within the population/swarm, it becomes possible to emphasize global exploration and/or local exploitation. The rationale behind the design of topologies lies on the widely acknowledged fact that the overall search performance of evolutionary algorithms can be significantly biased by how individuals in the population are organized and interact with each other. As such, many different topological schemes have hitherto been proposed by advancing over the so-called panmictic topology, in which all population members can mate with any other member. This simple topology, which lies at the core of naive versions of well-known evolutionary algorithms such as GA, is known to foster the rapid dissemination of information among the individuals, yielding a progressive loss of diversity and ultimately, a potentially premature convergence of the overall search process. This is the reason why more elaborated topological schemes have been extensively analyzed in recent years, often relying on the definition of a structural neighborhood between individuals. A recent comprehensive overview on this topic can be found in Ref. [183].

As has been foretold in Subsection 3.4, the criterion by which the above neighborhood relationship is defined established two general categories of topologies for population-based heuristics: those based on the population index of the individuals and a predefined structure (e.g. ring, wheels, random, von Neumann or star, among others) that dictates how information flows among them; and those based on a measure of distance to dynamically determine which solutions are neighbors of each other. Furthermore, topologies can be endowed with further characteristics to make them better perform in certain problem setups. As such, topologies can be static or dynamically adapted along the run. Likewise, heterogeneous topologies [184] enforce different topological relationships (e.g. tighter or looser connectedness) between subpopulations, to the extreme of generating them in a randomized manner [185]. Once the topology is established, good solutions encountered during the search process are spread throughout the whole population or exploitation subpopulation, yielding an improved convergence and superior optimality in many flavors of optimization problems such as bound constrained single-objective optimization, multi-objective optimization and multimodal niching.

Topologies can be also defined within the subpopulations or islands (also referred to as *demes*) of a distributed EA. Although we will elaborate further on this area of bio-inspired optimization in Subsection 3.8, it is important to note that the interaction between such isolated islands (which evolve independently) takes usually the form of a migration policy, which selects which individuals are moved or copied between subpopulations. Therefore, the migration policy can be regarded as a topology between demes in distributed bio-inspired solvers. By carefully tailoring the migration policy one can balance between exploration (demes do not interact with each other and search over the solution space in isolation) and exploitation (individuals are occasionally migrated between islands). Therefore, independent intra-deme evolution and migration schemes must be devised to match a good balance between exploration and exploitation, which in turn intersects with the characteristics of the optimization problem at hand and the operators of the heuristic algorithm themselves.

Regarding this noted threefold intersection, certain population topologies have been found to perform better than others when applied to solve a given class of optimization problems. An example supporting this statement can be found in multimodal optimization problems where, as mentioned in Subsection 3.4, topology-based niching methods relying on multiple subpopulations have been extensively studied in the literature. In problems composed by single optima, however, panmictic topologies are more suitable than their sub-population based counterparts. Unfortunately, there is still no clear understanding of the theoretical foundations behind this match between problems and topologies, nor can we predict which topology performs best for a given problem. This unsolved issue triggers a wave of future research towards extensively designing new population topologies capable of dealing with problem classes that have been tackled with topology-based heuristics to a much lesser extent than multimodal and unimodal problems, such as constrained, multi-objective, large-scale, or bi-level, among others. Efforts are also foreseen to be needed towards the extrapolation of the topology concept to other heuristics than DE and PSO, which have so far monopolized the literature related to this area. Finally, a promising research direction is to jointly consider topologies and ensemble strategies (Subsection 3.8) as a means to leverage the superior explorative/exploitative powers of ensembles and delegate the

search intensification in tailored topologies set within the population of each member of the ensemble.

3.6. Surrogate model assisted optimization

In certain applications such as aerospace design, optics or biomedical engineering, problems tackled by bio-inspired optimization methods usually rely on computer simulations. Such simulations are conducted not only for testing purposes but also to properly tune the configuration of the overall system by providing a quantitative indicator of its performance. Unfortunately, accurate testing processes entail high computational efforts, with evaluation times that range from hours to days per experiment. Hence, the efficient design of bio-inspired solvers is often an extremely time-consuming task, becoming prohibitive even for the powerful computation means available nowadays. The lack of sensitivity information and numerical noise usually present in experimental outcomes can also be deemed as issues falling within this specific context. These problems, and other related ones can be mitigated by the adoption of surrogate models, which reliably portray the expensive simulation-based model in a cheaper and analytically tractable way. Simulation-driven studies supported on surrogate model assisted optimization have gained great popularity, allowing the community to address problems that could not be tackled otherwise, and dramatically reducing the computational costs of the problem-solving process. Furthermore, once built to achieve an admissible accuracy level, the surrogate model can be exploited to provide hints about where promising candidate solutions are located over the solution space, thereby serving as a driver for optimized heuristic search operators.

Several comprehensive surveys can be found in the literature, highlighting different aspects of this specific field. Two remarkable overviews are the ones in Refs. [186,187], which focus on aerospace sciences, a sector where surrogate model based optimization has been utilized in manifold design applications due to the complexity of conducting real experiments. In fact, many other recently published contributions revolve on applications within this sector: this is the case of [188], for example, which aims at showcasing that surrogate-based global optimization is feasible for aerodynamic shape optimization with a high number of design variables. Likewise, in Ref. [189] a solver based on the Kriging surrogate model and parallel infill-sampling method is proposed for the aerodynamic shape optimization of a swept natural-laminar-flow wing. Authors of that study claim that the main challenge in this study links to the trustable prediction of laminar–turbulence transitions and reasonable compromise of viscous and wave drags. Another recently presented work is [190], which explores the feasibility of a hybrid approach based on evolutionary algorithms and support vector regression to reach optimal configurations of the landing gear master cylinder. In this research study, the aerodynamic shape design problem is also approached by using surrogate models and intelligent estimation search with sequential learning (IES-SL, [191]).

Surrogate models are also appropriated for problems arisen in other disciplines and sectors. For instance, the design of different types of antennas for wireless communications has also undergone significant research with surrogate models in recent years. In Ref. [192], for instance, the design of an ultra-wideband antenna with an integrated balun is faced by an automated numerical surrogate-based optimization. Authors in Ref. [193] claim that the design of contemporary antennas requires the configuration of an unprecedentedly high number of parameters to configure, which exacerbates the complexity of the overall design problem. To alleviate it, they introduce a novel two-level method for the surrogate modeling of antenna structures using Kriging interpolation models. Another example of the prevalence of surrogates in antenna design problems is [194], where a multi-objective formulation of this problem is tackled by considering performance and robustness as conflicting objectives, and by automatically selecting the best surrogate among a portfolio of possible models (namely, polynomial regression, Gaussian process regression, and Kriging).

A stimulating general review on the area of the surrogate model assisted optimization has been recently presented in Ref. [195]. Among many other interesting aspects, this review highlights that the efficient use of surrogates entails great savings in terms of computational resources, but also pinpoints the complexity of selecting an appropriate surrogate due to the variety of available models. In this study authors discuss on frequently used approaches for obtaining surrogates, stressing particularly on recent advances in this regard. Regression models such as linear regression and support vector regression are discussed, despite an emphasis is placed on Kriging and Radial Basis Functions (RBF). These two interpolation models are among the most popular ones for feasibility analysis and optimization by virtue of their capability to provide a quantitative measure of prediction uncertainty. This has allowed these models to prevail in many applications, such as design simulation [196] and pharmaceutical process simulations [197] in the case of Kriging, and parameter estimation [198] or water pumping optimization [199] in the case of RBF.

Several important challenges can be outlined within this specific topic. To begin with, it remains unclear how surrogate models could efficiently tackle complex problems composed by a high number of dimensions. A growing amount of studies are recently echoing this issue [194,200], putting in question the efficiency of traditionally used methods like the ones mentioned previously [201,202]. New mechanisms and existing methods are being lately combined to face this issue, as in Ref. [201] where Kriging is hybridized with partial least squares. Deep Learning models could also enter this arena by incorporating techniques proposed lately to quantify the statistical reliability of their predicted output, as those exemplified by Ref. [203] or provided *off-the-shelf* by Bayesian deep models [204]. The challenge for the deep models will be to learn within a limited number of function evaluations.

A higher dimensionality of the problem under consideration also requires further improvements for the optimization algorithm itself. Sophisticated evolutionary methods are receiving attention in recent years, such as cooperative PSOs [200], hierarchical PSOs [205] or Cooperative Co-evolutionary approaches [206]. The applicability of LSGO techniques should also be inspected in this regard in combination with the aforementioned Deep Learning surrogates. We foresee that this synergy may have a groundbreaking effect in the scalability of problems addressed with surrogate models and bio-inspired optimization.

3.7. Distributed evolutionary algorithms

Since most real-world applications can be highly nonlinear and large-scale, bio-inspired algorithms that work well for small-scale or moderate-scale problems need to be modified and parallelized in implementations. Though multi-agent, population-based algorithms can be suitable for parallelization, it is not clear what is the best way to parallelize them. In addition, simple processing parallelization may not be enough to solve truly large-scale problems with thousands or millions of variables. Certain modifications and enhancements are needed, even though it is not clear yet how to achieve such effective modifications at the moment.

In this context distributed Evolutionary Algorithms (dEA) come to the scene by deploying different populations of solutions on distributed systems in order to improve the performance of sequential (traditional) evolutionary solvers [207,208]. dEAs also include co-evolution type algorithms, which tackle high dimensional problems through distributed divide-and-conquer mechanisms. A recent tutorial of the most important distributed techniques in EAs is given in Ref. [209], which classifies dEAs in population-based models (island, cellular, master-slave, hierarchical and pool) and dimension-based approaches (co-evolution and multi-agents). We review here the most important characteristics of dEAs following a similar structure for describing each model of dEA, i.e. starting with population-based dEAs and finishing with dimension-based models such as co-evolution algorithms. A

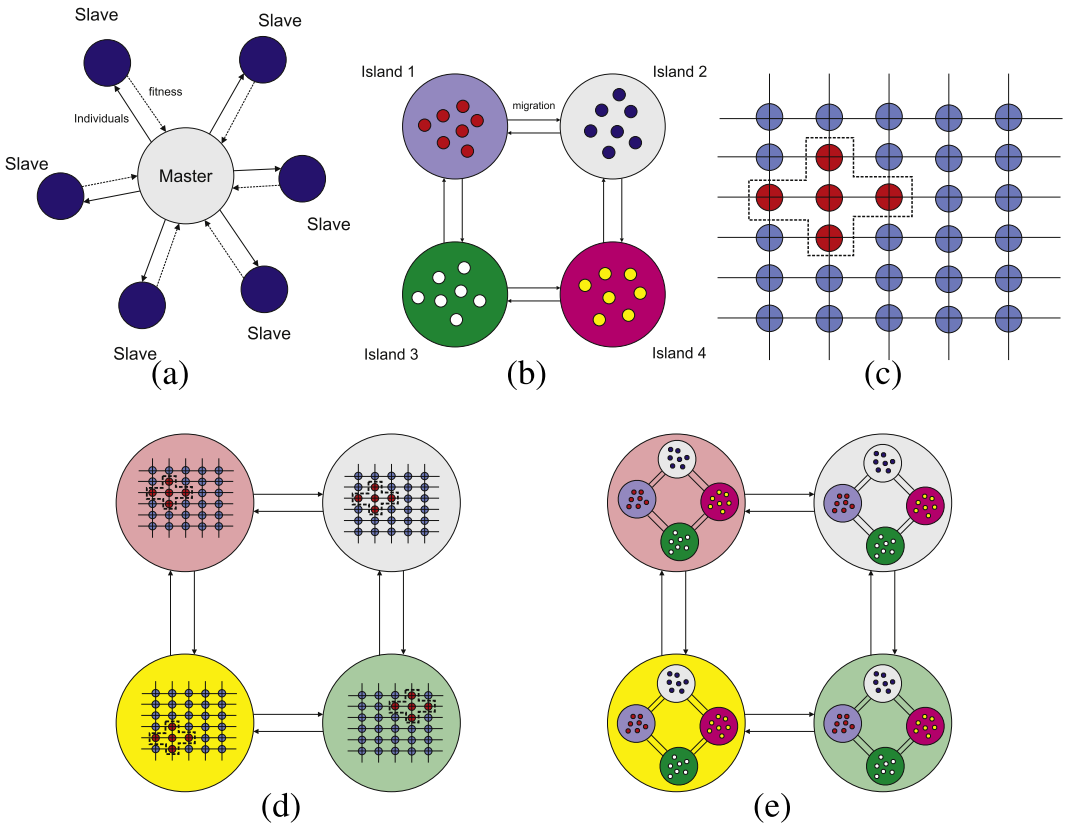

Fig. 4. Example of (a) a master-slave dEA model; (b) an island dEA model; (c) a cellular dEA model; (d) hybrid ⟨island, cellular⟩; (e) hybrid ⟨island, island⟩.

schematic representation of all these parallel computation strategies for bio-inspired computation is depicted in Fig. 4.

To begin with, the master-slave model is a simple albeit effective dEA model [209,210]. In this class of distributed optimization models, bio-inspired search operators are executed in the so-called *master* part of the algorithm, and evolved individuals are then delivered to *slave* processors for fitness evaluation. In this case, it is considered that the fitness evaluations of the individuals are independent, so there is not a need for communication among slaves processors. The distributed fitness computation featured by these dEAs makes them especially well-suited for those cases where most of the computational complexity of the problem at hand is concentrated on the computation of the fitness. Therefore, when parallel processing capabilities are available they are an efficient alternative to other methods such as surrogate-model assisted optimization.

The second dEA discussed in this subsection is the island model [211], which is a coarse-grained evolutionary algorithm where different sub-populations of solutions are considered, each processed by a different processor. Communications between islands are allowed and occur when certain individuals migrate among them at given times of the evolution. The migration mechanism is an important design part of island models, including the migration topology, its frequency, and extent, as well as the replacement policy in the destination island.

Slightly linked to the island model introduced above, a cellular dEA model [212] is a fine-grained, spatially-structured approach, which consists of just one population whose individuals are arranged on a grid of processors (*cell*). Interaction among individuals handled by the evolutionary algorithm is done by using communications paths defined by a network topology in the dEA. Thus, each individual in the population can only interact with those individuals within its neighborhood. As the neighborhoods of individuals are overlapping with each other, good individuals tend to propagate to the entire population of the algorithm.

A rather different strategy is followed in the hierarchical dEA model, also referred to as the hybrid model [213]. It combines two (or more) distributed models hierarchically, such as the ⟨island, master − slave⟩ or ⟨island, cellular⟩, among other combinations. The underlying idea of these approaches is to improve the search capabilities of the whole heuristic by embedding characteristics of both models into a single algorithm. The last population-based dEA reviewed in this subsection is the pool dEA model, in which a set of autonomous processors are deployed to work on a shared resource pool. Processors are loosely coupled, in such a way that they do not know each other's existence and only interact with the pool [214]. The pooled model provides a natural approach to asynchronization and heterogeneity in dEAs, of inherent interest for their deployment in non-controlled computing frameworks.

In what refers to dimension-based models, Cooperative Co-evolution (CC) provides a powerful divide-and-conquer architecture for computationally hard optimization problems, such as LSGO [215–217]. In CC a highly-dimensional or complex optimization problem is divided into several simpler sub-problems, which will be solved by different metaheuristics. Provided that the problem formulation allows for such a decomposition, sub-problems in which it is split can be solved independently, so that once the optimization process has been completed the solution of the problem can be obtained by assembling all produced solutions to the sub-problems together. However, when complex interdependencies exist between sub-problems, the CC model can still be applied in such a way that each computing node performs a local evolution process in a solution subspace. Then, by enforcing communication between nodes, they can adaptively adjust their search direction and cooperatively move towards regions of higher optimality for the problem at hand. A dimension-based approach is also present in multi-agent models [218], which differs from CC in that they do not require any direct coordination of agents to evolve. Instead, agents are endowed with game-theoretic behavioral rules such that they optimize

local functions and negotiate mutually to reach some stable equilibrium and thereby, complete the optimization task efficiently.

Recent activity on dEAs has been intense, with many different real-world applications having been addressed by parallel bio-inspired techniques falling within this category. Remarkable advances in this regard include [219], where new approaches for migration in dEAs based on biological invasions are proposed and proven to be particularly well suited for island genetic algorithms, showing improvements over alternative migration schemes. Another interesting work in this line is [220], where a new discretization scheme for Big Data processing based on a multivariate dEA is proposed and tested in different large-scale classification instances, such as *ECBDL14*, a highly unbalanced classification database consisting of 32 million instances with an unbalance ratio over 98%. In all the tested problems the proposed dEA-based discretizer performed best, attaining superior accuracy rates. Unsupervised classification over large-scale datasets has also been tackled by adopting dEAs, such as [221], where an island genetic algorithm is proposed for fuzzy partition problems, or [222], where dEAs are applied to improve a k-Means clustering algorithm. There has been also active research around more practical versions of dEAs, in areas such as large-scale optimization [223–225], Electromagnetism [226], Computational Fluid Mechanics [227], energy planning [228] or neural networks training [229].

Besides large-scale data mining, a multi-objective optimization is another research subarea where dEAs can yield profitable computational gains. The main motivation to develop parallel multi-objective solvers is to deal with expensive objective functions, which are common in real-world applications comprising multiple criteria. However, something surprising is that the design of new parallel approaches is very rare in the specialized literature [230]. We are lacking work on the development of asynchronous parallel multi-objective solvers, and their comparison with respect to their synchronous counterparts (quite in line with the work in Ref. [231]). We also need parallel multi-objective optimizers designed to run on GPUs, following early work in Ref. [232]. Another interesting topic is the change of granularity in a parallel multi-objective algorithm (with a unidirectional topology) with the aim of performing a more efficient search [233]. There are many other possibilities. For example, the use of asynchronous parallelism combined with the use of micro-populations was adopted in Ref. [234] to yield S-PAMICRO (PArallel MICRo Optimizer based on the S metric), a computationally efficient (parallel) version of SMS-EMOA that uses exact hypervolume contributions. Above all, a crucial task that the community should engage with is to exploit parallel architectures by designing multi-objective bio-inspired techniques that explicitly take advantage of a particular parallel architecture (e.g., grid computing or GPUs), rather than simply producing ad-hoc parallel versions of existing multi-objective evolutionary algorithms such as MOEA/D and NSGA-II. There are, however, many other research topics worth to be explored in the intersection between dEAs and multi-/many-objective optimization, such as the impact of the topology on the performance of a parallel solver and the management of diversity among distributed populations.

Future advances in the area of dEAs should be focused on better exploiting their capabilities, by inspecting radically new search methods hinging on distributed computing resources, and also assessing their applicability in general application scenarios such as LSGO and Big Data. Regarding the former, an interesting line stems from the derivation of heterogeneous search algorithms merging different encoding strategies, bio-inspired operators and constraint handling techniques [235] into a single unified search algorithm suited for parallel implementations [236]. This research has been explored in the literature recently when a heterogeneous meta-heuristic technique was proposed in Ref. [237] and later adapted in Ref. [238] for LSGO problems. In this regard, we recommend [239] for a comprehensive overview of ensemble strategies for population-based bio-inspired algorithms, with connections and prospective insights into their applicability for LSGO, multimodal, multiobjective dynamic and constrained optimization.

As for developmental application lines in dEAs, LSGO and Big Data problems in Earth and Atmospheric Sciences, Global Energy Demand or Climate Change studies are some of the most important areas where dEAs could have a deep impact due to the computationally challenging characteristic of these problems. Alternative applications in problems involving a huge amount of variables, observations and/or objectives are also of utmost interest for dEAs, as those frequently occurring in Finance, Bioengineering or Biomedicine, among others.

3.8. Ensemble methods and hyper-heuristics

The concept of *ensemble* in optimization refers to the use of multiple search strategies, subpopulations, algorithms, rules for selecting next generation population, operators and/or parameter values to tackle an optimization problem [239]. The idea is that the ensemble strategy can obtain better results than a single strategy, specifically, better than the ensemble composites working on their own, when applied to a given optimization problem. Following [239], ensembles for optimization can be classified by taking into account different characteristics of the technique at hand, mainly the type of constituent elements and the applied implementation technique. First, when the ensemble combines different types of search strategies, operators or constraint handling techniques, it is known to be a *low-level ensemble*. On the other hand, *high-level ensembles* refer to methods that adaptively select the best optimization algorithm for a given problem among a set of candidate algorithms [240,241]. Ensemble methods can be also categorized regarding their implementation structure, resulting in competitive single/population, competitive multi-population, and cooperative multi-population ensembles.

Low-level ensembles include multi-methods and multi-strategy approaches developed in the last decade. Multi-methods algorithms consider the combination of different operators or algorithms to solve an optimization problem. Examples of low-level competitive single population approaches are those proposed in Ref. [242], where different search operators are jointly applied and self-adapted in the same genetic population, or [238,243], where a set of “substrates” are defined in a Coral Reef Optimization algorithm, representing the application of different search operators defined in a single population. Multi-methods have been also applied to improve multi-objective optimization problems [244,245]. Other multi-methods operate on different sub-populations, yielding competitive multi-population approaches such as the one in Ref. [246]. Alternative low-level ensembles involving other algorithmic components have been also proposed in the literature, such as neighborhood sizes [247], constraint handling techniques [248] or niching [249] among others.

High-level ensembles have been also applied to optimization problems with success. For example [250], proposes a portfolio of different algorithms to be applied in optimization problems, choosing the best combination of algorithms depending on the problem under consideration. In Ref. [251] a comparison of different high-level multi-method ensembles is carried out. Multi-strategy ensembles choose among different versions of the same search strategy to solve optimization problems, usually in a competitive fashion in single or multiple populations. Ensembles of multi-strategy approaches based on different algorithms have been proposed in the literature, for example, based on DE [252,253], PSO [254], Artificial Bee Colony algorithms [255] or Biogeography-based optimization [256].

Closely related to the idea of ensemble methods, the *hyper-heuristic* paradigm appeared in the early 2000s as a novel computation paradigm useful to tackle hard optimization problems. As already mentioned in Subsection 3.3, hyper-heuristics have been defined as “search methods or learning procedures for selecting or generating heuristics for a given optimization problem” [257]. Hyper-heuristics operate on a set of simple heuristics – which by themselves do not render good results for a given optimization problem – and combine them via a higher-level algorithm towards achieving much better solutions [126].

In other words, a hyper-heuristic seeks the automated construction of new heuristics highly adapted to efficiently solve a given optimization problem; for this purpose, the hyper-heuristic algorithm at hand must explore a search space spanned by low-level heuristics by using hyper-heuristic operators defined for selecting, modifying, combining and/or constructing heuristic methods.

Following [126], hyper-heuristics can be classified in different categories depending on the nature of the heuristic search space. Thus, hyper-heuristics are classified into *hyper-heuristics for heuristic selection*, that comprises those methodologies focused on choosing or selecting existing heuristics, and *hyper-heuristics for heuristics generation*, that includes those methodologies focused on creating new heuristics from existing ones. Focusing on heuristic selection, the process to construct the complete algorithm is quite simple: first, it is necessary to select a set of good low-level heuristics for the optimization problem at hand. Note that this idea is quite related to the concept of low-level ensembles. It is not necessary that the heuristics on their own are very effective in solving the problem, but their number should be large enough to generate a large search space [258]. Then, a high-level approach must be selected in order to obtain the best set of low-level heuristics and how to apply them to solve the problem. In many cases, this high-level algorithm is a meta-heuristic (evolutionary algorithm, ants algorithm, particle swarm, etc.), which requires a proper encoding of the low-level encoding to perform the search. This encoding strategy stringently depends on the problem being solved, and the algorithm's performance depends also on this selection. A number of tutorial and reviews papers focused on hyper-heuristics can be found in the literature for the interested reader [257,259,260].

There are several lines which are currently hot research topics when it comes to ensembles and hyper-heuristics [239,257]. First, an important issue, currently under research in both topics, is the application of these techniques to large-scale optimization problems. In large-scale problems, the application of ensembles or hyper-heuristics becomes much more involved than usual: the encoding strategy is not straightforward, the exploration capabilities of the algorithms become inefficient, and sometimes the computational complexity of standard ensembles is too high to tackle this kind of problems. Further research is still needed to solve these issues. Different lines are currently under development related to intelligent encoding methods to alleviate the high computational complexity inherent to large-scale optimization. The second central challenge in ensemble and hyper-heuristics is the specific selection of methods to be assembled or low-level heuristics in the hyper-heuristic method. In many cases, it is possible to choose among a large set of low-level methods or heuristics, but the appropriate selection of these composing pieces of the ensembles or hyper-heuristic approaches remains an issue not fully solved when it comes to real application setups. Finally, the combination of high-level with low-level methods/heuristics is another hot topic in ensembles for optimization, also with application in hyper-heuristics. The idea is to combine multiple-methods approaches with multiple strategies in the high-level algorithm, which leads to challenging problems related to the algorithms tuning and the computational complexity of the final ensemble.

3.9. Memetic algorithms

Memetic Algorithms were first conceived [261] and later forged [262] as a branch of bio-inspired computation characterizing a specific kind of hybrid evolutionary meta-heuristics. Initially defined as modifications of Genetic Algorithms employing local search mechanisms, the community was doubtful about Memetic Algorithms even years after their advent [40]. It is important to note that at the time of their inception, PSO, DE and other adaptations for real-valued optimization problems had not been yet contributed to the community. Memetic Algorithms were, thus, designed as an efficient workaround to endow global search heuristics at that time (e.g. binary-coded GA) with local search capabilities. However, many EC and SI heuristics reported shortly thereafter have been shown to be able to perform both global and local search provided that their search procedure is based on the difference between any pair individuals (as in the aforementioned DE and PSO). In the initial stages of the search process, global search is enforced inherently due to the diversity of the population (large differences between individuals), whereas local search is performed when the population converges around an optimum (correspondingly, small differences between individuals). Furthermore, other sophisticated bio-inspired approaches have shown an ability to perform global and local search simultaneously; this is the case of CMA-ES when configured with a highly adaptive step size [263], or the more recent HCLPSO approach [184], which defines two subpopulations (with parameters selected for one to favor exploration, the other for exploitation) so that information is exchanged only from the exploration to the exploitation subpopulation. Therefore, the separation between global and local search established in Memetic Algorithms is progressively disappearing in favor of monolithic methods jointly comprising both functionalities.

From their advent the family of Memetic Algorithms blossomed into a massive diffusion, being today one of the most prolific fields within Operations Research. Nowadays, the distinguishing concept behind Memetic Algorithms has evolved to a more generic conception of this algorithmic branch, defined as the combination of bio-inspired approaches for global optimization with separate local improvement and individual learning mechanisms, possibly incorporating domain-specific knowledge of the problem at hand.

Due to the intense activity on this specific field and with the intention of being adapted to the time's needs, Memetic Algorithms have been growing at a constant pace to yield complex techniques with extremely sophisticated cooperative mechanisms. As a result of this fertile activity, a fair amount of Memetic Algorithms coexists in the current literature, which can be classified in many different and equally appropriate ways. In our brief survey of the state of the art we embrace the taxonomy introduced in Refs. [264,265], in which three different chronological generations are distinguished based on their intrinsic characteristics and communication mechanisms:

- The first generation started with the pioneering work by Moscato and Norman mentioned above, in which the benefits of combining population-based global optimization solvers with local search procedures were first explored. Methods framed within this first category are characterized by the use of a single local search heuristic.
- The second generation of Memetic Algorithms delved into the memetic transmission and design selection, with Multi-meme evolutionary algorithms [266] and hyper-heuristics [267] leading the algorithmic streamline during this period. The main difference between these techniques and the classic MAs is the use of a group of local search methods. On the one hand, in multi-meme methods, simple inheritance mechanisms are used for the transmission of the memetic material (i.e. the choice of the local optimizer). On the other hand, in hyper-heuristics, the groups of predefined memes compete among themselves to decide which one to choose for local refinements. This competition is based on their previous performance by resorting to a reward mechanism. Interested readers on this specific generation are referred to Ref. [268], in which an extensive survey on Memetic Algorithms considering multiple learning methods inside an evolutionary strategy is presented.
- Finally, the search process of methods considered to compose the third generation of Memetic Algorithms also rely on multiple local optimizers. Additionally, memetic information is also passed on to offspring produced by crossover operators. However, the main novel ingredient of third-generation Memetic Algorithms with respect to their preceding counterparts is that the mapping between the evolutionary trajectory and the choice of the local optimizer is learned from experience. Two principal trends emerge from this third category: self-adaptive schemes [269] and co-evolving memetic methods [270].

A great upwelling of contributions has been lately noted in the literature, all focused on the discovery of new synergistic memetic approaches with biological inspiration at their core. In Ref. [271], for example, the hybridization of a Cuckoo Search with a dynamic local search is studied for multi-objective optimization problems. The same design principle is observed in Ref. [272], where a recently proposed SI approach – Brain Storm Optimization – is combined with a metric-based clustering method, which is put to practice for optimally composing flight formations in swarms of unmanned aerial vehicles. Interesting works are also noted at the junction between other areas of bio-inspired optimization, such as the one in Ref. [273], where a large-scale capacitated arc routing problem is tackled by using a Memetic Algorithm combining decomposition strategies, co-evolution, and extended neighborhood search. Another relevant study to highlight is the one introduced in Ref. [274], where a many-objective, dynamic scheduling problem is faced by using a proactive rescheduling Memetic Algorithm based on Q learning. The key concept of the designed approach is to learn the most profitable global and local search methods in an adaptive way for the dynamically changing problem environment.

Next research waypoints to be visited in this area should include, at first, the derivation of self-adaptive mechanisms to tune the balance between exploration and exploitation. For any algorithm to work well in practice, a certain balance between explorative search and exploitative search is needed. Exploration enables to search a larger area in the search space, while exploitation focuses on the local regions that can potentially speed up the overall convergence. However, too much exploitation and too little exploration can lead to premature convergence, while too much exploration and too little exploitation can slow down the convergence and increase the computational costs dramatically. Thus, a fine balance between these two-opposing components is needed. This is uniquely critical in bio-inspired Memetic Algorithms, where diversification is empowered by the bio-inspired solver and intensification is delegated on the local search method. Different mechanisms of collaboration between these two search procedures may render different degrees of exploration and exploitation. Without an in-depth understanding of when this trade-off must be tuned and how it is difficult to design algorithms that can balance these two components. In this regard, further research is needed towards achieving new procedures for detecting and quantifying the level of stagnation, as well as appropriate, configurable countermeasures (e.g. diversity induction) well suited to be inserted in classical frameworks for memetic computing.

Although we foresee an exciting future for research on this topic, the community investigating on Memetic Algorithms should avoid falling into the temptation to hybridize the myriad of bio-inspired optimization techniques reported to date with local solvers just for empirical serendipity. Though qualitative improvements may eventually be obtained, the lack of mathematical rigor or argued design decisions may hinder the discovery of memetic patterns that could bring real value to the area. In fact, it still remains unclear how to combine components from different algorithms to make a hybrid more computationally efficient and effective for a given optimization problem. Workarounds to this lack of knowledge take advantage of the availability of computational resources to automatically construct hybrid heuristics. This is the case of recent studies dealing with evolutionary hyper-heuristics deployed over large computation grids composed by volatile computing nodes [275]. Nevertheless, a detailed analysis shedding light over this worrying concept in Memetic Algorithms is urgently needed in order to gain informed intuition rather than factual observations.

3.10. Large-Scale Global Optimization

Evolutionary Algorithms are a very popular tool in the field of real coding optimization, in the industrial and scientific domain. However, sometimes these problems require complex models with hundreds, if not thousands, of real parameters. This high number of dimensions greatly worsens the behavior of algorithms designed for a smaller number of variables, because the domain search increases exponentially with the dimension (the *curse of dimensionality* [276]). This type of optimization problems, with thousands of variables, is called Large-Scale Global Optimization (LSGO).

Tackling LSGO as a special category within bio-inspired optimization allows researchers to design algorithms tailored for them, as well as to propose special benchmarks to analyze and compare their performance. This has a special interest for different reasons. On one hand, these benchmarks may, in some ways, reflect the characteristics of many real problems, where the contribution of variables to the outcome of the objective function varies greatly from one another [277,278]. On the other hand, the development of algorithms that can efficiently explore these large search spaces allows for more efficient optimization algorithms, if not more scalable with respect to the size of the problem. Indeed, this is an increasingly valued feature given the growing trend of processing large volumes of data and variables.

The community working on this particular computation paradigm has united regularly in special sessions co-located in renowned conferences since the first was held in 2008 [279,280]. In 2010 another special benchmark was proposed [281]. In 2011, there was a special issue in Soft Computing journal with another benchmark [282]. Later, in 2013 another benchmark for LSGO was proposed in Ref. [277], which was specially designed by combining functions with different levels of separability and dependency between the variables. Since 2013, many competitions have been held where these same benchmarks have been maintained. Furthermore, many research works have used them to develop their own proposals. As evinced by Fig. 2 (stacked bar colored in ■), the evolution of LSGO in last years shows a rising trend in number of contributions, with no clear predominance of EC or SI approaches. All in all, achievements as the ones outlined in what follows are symptomatic of the momentum of this subarea, with biological inspiration taking a leading role in recent contributions [20].

In fact, research in LSGO not only has allowed to design evolutionary algorithms, but also to develop further other research lines closely linked to the computational problems derived from large problem scales. First and more evident, LSGO has catalyzed the design of parallel and distributed algorithms to reduce the high processing time needed for efficiently addressing large-scale problem instances. This crossroad between areas has already been noted in the previous subsection.

An interesting research line triggering many contributed schemes in the last years is to use the aforementioned benchmarks to develop new techniques that automatically infer relationships among variables. This inference permits to identify groups of variables that could be optimized in isolation with the minimum loss of efficiency. In combination with a CC, this approach can be very effective, because grouping variables allows the algorithm to optimize a lower number of variables. These grouping variables techniques are in essence decomposition methods based on a divide-and-conquer strategy, which allows decomposing large-scale problems into multiple low-dimensional subcomponents that can later be optimized by one or several algorithms capable, in most cases, of running in parallel. This decomposition is crucial to obtain good results [283,284]. The most straightforward approaches in this regard are random [215] or dynamic grouping (i.e. changing it during the run) [285]. Later, the work in Ref. [286] proposed to adapt the subcomponents size based on the historical performance. More advanced techniques strive to detect the interaction between variables to group together those with more interactions, because they can significant improve the results [287]. An example of this strategy is the delta grouping approach proposed in Ref. [288]. Delta grouping tries to identify interactions between variables by measuring the differences in fitness when the variable values change in isolation and when they are simultaneously changed. More recently, the same authors proposed an improved delta grouping scheme in which the number of evaluations required is halved, furthermore exhibiting a

more robust behavior than its previous counterpart [284]. We refer to Ref. [289] for a more detailed description of the evolution of the LSGO area.

Anyway, even considering the improvement in computational performance yielded by these grouping methods, such gains obtained by the decomposition usually do not compensate for the cost in terms of evaluations required for detecting variable dependencies. Thus, last proposals try to reduce this high cost in evaluations. A significant new approach in this direction is to apply the decomposition in a recursive way, like in Ref. [290]. Definitely there is a research niche in this noted drawback of current grouping LSGO techniques, because when the reduction in complexity could compensate the additional evaluations in the process, it could be applied in all types of problems, not only large-scale problems. Even more, their application to many real-world problems is also expected, because these problems could encompass variables with many and few relationships with others, thus the optimization process with one grouping mechanism could be improved significantly.

Since these grouping techniques are very costly in evaluations, and they are not useful for function with overriding [287], currently they cannot be considered competitive enough in comparison with algorithms specially designed for LSGO, thus the develop of algorithms specially designed for them is still a very interesting and open research line. Among these techniques, Multiple Offspring Sampling (MOS, [291]) has been considered the *state of the art* during many years, because since its first appearance in the LSGO panorama [292] it has not been improved significantly. MOS is an algorithm that combines several generic optimization algorithms and more specific solvers for LSGO, such as local search methods that are applied to the same population in rounds with an adaptive mechanism that decides which algorithm should be applied in each round, considering the historical performance of previous applications of every algorithm. Very recently, another proposed algorithm, SHADEILS [293], surpassed results obtained by MOS (specially in more complex functions), becoming the new *state of the art* in LSGO supported by the results attained by this algorithm in the latest CEC'2018 competition [294]. SHADEILS is a memetic algorithm that combines an advanced version of DE with two local search methods and an adaptive mechanism to decide which one of the local search should be applied in each round (DE is always applied). As in other recent references [295], we observe that DE has lately prevailed as the most utilized bio-inspired search algorithm in LSGO optimization. However, additional characteristics of recent proposals like the memetic computing strategy followed in SHADEILS suggest that there is still room for more efficient LSGO algorithms.

To summarize, in a future we expect that new grouping variable techniques could reduce the cost of evaluations to the point of compensating for the use of such techniques rather than optimizing all variables together. Another promising field is to improve existing specially-designed algorithms. A third promising research line is to replace the algorithms usually used in CC, usually classical algorithms, with specific algorithms for LSGO, to avoid the possible limitations that could prevent these algorithms from reaching their full potential.

3.11. Parameter tuning

The majority of bio-inspired optimization methods are very flexible algorithms, with many parameters driving their search behavior. Thus, assigning proper values to these parameters is crucial for obtaining the best possible results for a given problem. This selection can be theoretical to meet some desired properties of the algorithm itself (as occurs in the design of CMA-ES [8]). More usually, parameters are rather set in an experimental way by e.g. using a value grid or, less regularly, by mirroring parameter settings used in similar studies to the proposed one. Unfortunately, it is often the case that bio-inspired algorithms have too many parameters, so that seeking the best value for each one of them can be regarded as another optimization problem by itself [296]. This problem, namely, to decide the best parameter values for the algorithm, is referred to as *tuning* or *off-line tuning* (because a decision is made on the parameter values before running the algorithm).

When a new bio-inspired algorithm is proposed within the community, the values of its parameters should be tailored in regards to the experimental setup under choice, for which a small study of parameter sensitivity is usually carried out aside. Depending on the number of parameters and their explored range of values, the search space can become computationally unaffordable even for very coarse value grids. One of the most important decision is to decide which parameters should be tuned, or fixed assigned based on the intuition or convention, as the recommended values by other researchers or authors of the original algorithm. When searching for the best performing parameter set over value grids recommended by other authors [297,298], the expensive cost of tuning usually restricts the number of selected parameters. Moreover, in some complex problems, it is not unusual to spend time and efforts executing the algorithm with many different combination of parameters, obtaining that only a few of them have a strong influence over the results. Therefore, many combinations can be avoided, reducing the overall computation time. A parametric sensitivity analysis, or robustness studies, can be very useful to identify the relevant parameters to tune (and hence conventional values are used for the remaining ones).

Another important question is if the improvement yielded by parameter tuning deserves the computational cost required by this process. The answer depends strongly on each particular case, but in general, it is worthwhile when real-world problems must be optimized many different times, or when small objective improvements due to parameter tuning can yield significant gains in some practical aspect, e.g., lower economic costs associated to the solution. When tuning is demanding in terms of computational burden, a very popular option is to delegate the process to tool as REVAC [299] or I-RACE [300], capable of automating the parameter tuning process. Among them I-RACE has lately become one of the most widely adopted schemes. I-RACE is a freely available software that uses racing tuning [296] to allow researchers to easily adjust the values of the parameters of an algorithm over a specific group of functions. The mechanism of I-RACE is simple: from an executable algorithm and a list of parameters (indicating, for each one, the type and range of possible values), the software samples a distribution for each parameter to be tuned, and updates it with the best configurations (as per a racing mechanism) to bias subsequent sampling stages towards parameter values in the best configurations found so far. After several sampling iterations, the best configuration is returned. In this way, although tuning still remains costly in terms of time processing, researchers are alleviated from time-consuming parameter tuning phases.

When a research work proposing a new optimization algorithm reports a benchmark with other solvers from the state of the art, it is often the case that parameter values are carefully tuned for the proposed approach. However, for the rest of algorithms the values proposed by their authors in related contributions are instead adopted, under the assumption that their optimality also holds for the problem under consideration. In some cases these adopted values were obtained under experimental conditions (e.g. dimensionality, range, or evaluation limits) that could strongly differ from the problem(s) considered in the benchmark at hand. In these cases, when the experimental conditions are very different and/or the proposal is tuned more exhaustively than its counterparts, a similar parameter tuning process should be done with each of the reference approaches in the benchmark to ensure fairness in the comparison [301]. In Ref. [298] extensive comparisons were carried out with competitive algorithms with and without automatic tuning. As a conclusion, results were shown to greatly differ based on whether default values or the tuned parameters were used, remarking the importance of a good parameter tuning (whenever it is practically feasible and affordable) in new bio-inspired algorithm proposals to come.

3.12. Parameter adaptation

To overcome the problem of setting parameter values, an uprising trend is to use adaptive or self-adaptive parameters. On one hand, adaptive parameters are changed during the run by means of an adaptation strategy that is set before the algorithm is actually executed. On the other hand, self-adaptive parameters are parameters whose values are evolved at run-time along with decision variables according to the intermediate results obtained by the algorithm during its search [302]. This could be *online tuning*, because the results obtained during the run are used as feedback to adapt current parameters values. These two techniques can be combined based on the parameter type. We consider that these adaptation mechanisms, with may also hinge on the use of ensembles or hyper-heuristics (already introduced in Section 3.8), constitute a promising research topic within bio-inspired optimization.

A current tendency in this context is to use self-adaptation mechanisms for very sensible parameters that have a strong influence on the results. Not only it reduces the parameters to tune (making the algorithm simpler to use on new problems), but it can also yield great improvements. For instance, the most successful proposals in the family of DE-based heuristics during the last few years hinge on the adaptation of the main parameters of this bio-inspired solver, namely, the F (differential weight) and CR (crossover probability) parameters [303].

One of the most difficult parameters to adapt in a bio-inspired solver is the population size. Although it is usual to resort to the same population size than in previous related works, this is a parameter with a strong influence on the trade-off between exploration and exploitation. An encouraging approach is to use an adaptive or dynamic population size: while it is well-known that a higher population size is useful for maintaining diversity in the first stages of the search process, in the last stages it is more important to foster exploitation around the best individuals within the population. Thus, enforcing a higher population size at the beginning of the search and including a reduction mechanism (in combination with elitism) could improve the performance of algorithms. This is a simple strategy that can render very good results [304,305]. Also, there are additional factors that could affect the population size, like to use a survey parameter related to each individual (reducing the lifetime for worse solutions), varying the population based on the fitness of its individuals (like [306]), or incest prevention mechanisms. Also, sometimes the population size is affected when a restart is done in the algorithm [8], or even several populations with different sizes are simultaneously used [307]. In fact, there are studies like the one in Ref. [308] that analyze in depth different criteria for the population size, concluding that the proper value of this parameter depends on the problem itself. To date, self-adaptive population size mechanisms is an open issue, with interesting modern contributions being contributed regularly in the community (see Ref. [309] for a comparison among different adaptive population size strategies).

Other parameters that possess a great influence over the search are those controlling the variation operators (like crossover or mutation) used to generate the new solutions. For long time ago it has been proposed to increase the diversity between parents and offspring when the population converges to compensate for the lack of diversity [310]. There are many proposals in GA to encourage diversity, such as an adaptive mutation [311] or and adaptation of the selection mechanism to recombine the most dissimilar individuals. As for DE, most of the contributions dealing with this heuristic propose to adapt the CR and F parameters that balance the difference between a newly produced solution and its reference solutions (encouraged by the improvement obtained). In the majority of cases the values are not fixed, but change from solution to solution by sampling a distribution with a central value. This central value is adapted based on successful solutions, considering as such those that are introduced into the population. First DE versions harnessing this design principle, like SaDE [312] or JADE [313], used only a mean value for each parameter, which is adapted along the run. However, in order to obtain a higher level of robustness, alternative versions like SHADE use a memory of several F and CR mean values; in that way, the search can be done by alternating among different mean values. Another recent trend is to consider not only the best solution to guide the search, but a group of the best p solutions (selected one random for each solution), and adapt that parameter p during the search by 1) using a reducing value to increase diversity at initial stages, and 2) centering the search around most promising solutions in later stages. All these techniques are complementary, and in many algorithms they are used in combination.

In memetic algorithms or in co-evolutionary algorithms, in which several algorithms are used in combination, there are also parameters that control how these algorithms interact with each other. The self-adaptation of these parameters makes it possible to enforce the application to each algorithm based on the improvement achieved by each one, using one reinforcement learning technique [314]. The most widespread method is to select one algorithm based on a probability, which is periodically updated as a function of the relative improvement obtained by each component. This model has been used to select the local search method [315], or to choose the mutation operator [312]. This model has been proven to be specially good in complex search spaces. For instead, proposals like [291] or [242] combine different classic algorithms and some specific local search methods, obtaining very competitive results that clearly surpass those obtained by each of its components. The excellent survey in Ref. [316] describes in depth different parameter control strategies reported in recent times.

To summarize, although the adaptation of parameters has been an active research line for years, the good results obtained by recent proposals have revived the interest in these techniques. Not only they can result in the best performance, but they also allow algorithms to better perform over a wider range of problems and, consequently, get closer to the fundamental goal pursued in meta-heuristic optimization. Additionally, making a parameter self-adaptive reduces tuning problems exposed in the previous section, and yields a more readily useable search algorithm to solve real-world problems.

3.13. Benchmarks and comparison methodologies

In order to assess the convenience of a new bio-inspired algorithm, researchers must properly gauge its performance over one or several optimization problems, and compare it with other algorithms in the literature. To this end, traditionally researchers proposing a new bio-inspired algorithm selected a group of theoretical functions to use, and the experimental conditions under which the algorithm(s) would be tested. This, however, produced an important disadvantage: results from different papers could not be compared to each other because they considered different functions and/or experimental conditions. Indeed, authors should compare the results of their proposed solvers to those obtained by other reference algorithms in the literature, over the same set of problems and under the same experimental conditions. This approach, however, posed several main drawbacks:

- In strong connection to the structural bias that has been identified in certain heuristics (see Subsection 3.1), problems could have also properties that might favor some algorithms over others. For instance, in many problems, the optimum is at the center of the domain search, favoring algorithms prone to explore this area of the search space.
- It is difficult to decide which algorithms to compare with, as it is hard to identify the current state-of-the-art algorithms.
- While the parameter values set for the newly proposed bio-inspired algorithm are rather adequate for the experimental problem, the reference algorithms are configured with the parameter values retrieved from the papers were such algorithms were originally presented (and thus, recommended by their authors). Although this is a widespread practice, it is usually very unfair because the parameter values recommended by the authors might be suitable for the origi-

nal experiments, but not for eventually new fitness functions and/or experimental conditions used in the paper of the new proposed solver. For the sake of fair comparisons, all parameter values should be automatically tuned for the new benchmark (see Section 3.11); otherwise, strongly impairing non-tuned algorithms could mislead the conclusions drawn from the benchmark, as was shown in Ref. [298].

In order to overcome the above disadvantages, in the last years the community has embraced the use of standard test suites or benchmarks. These benchmarks define a group of functions to optimize, including all their related experimental conditions, so that the authors can compare directly the obtained results for their proposed solver. There are many benchmarks available in the literature, most of them proposed by organizers of specific special sessions and competitions in conferences from the field. Such benchmarks have also been widely adopted in other fora (e.g. journal papers) as standard benchmarks. Thus, these benchmarks not only allow researchers to compare directly their results, but also give a ranking of the best algorithms for the benchmark (ranking that can be completed by the proposals published in journals). This makes it very easy to identify the current state-of-the-art for any given benchmark, thereby becoming a clear reference algorithm for comparing optimization techniques proposed in the future [317].

Unfortunately, more than ten years after the first benchmarks too many proposals are still published without a right comparison methodology, mainly because the proposals were not compared against the considered state-of-the-art algorithms in many proposals in the literature. Authors of many contributions do not compare with algorithms identified as the state-of-art, but most worryingly: they limit their benchmark to classic algorithms (such as off-the-shelf versions of PSO or DE) proposed more than twenty years ago. Such deceptive works thus ignore a whole series of variants that have been proposed over the years, and that have been found to be much more competitive than their original counterparts and variants of other bio-inspired algorithms. In Ref. [318] reference algorithms selected in new proposals published in well-known and recognised journals were analyzed, concluding that in the majority of cases, such reference algorithms did not perform competitively enough to be selected as such.

We must also pay special attention to the huge number of new bio-inspired proposals contributed in last years [319]. The previous lack of right comparative against competitive and modern algorithms is specially relevant for these new proposals. In many times they are very innovative in its biological inspiration, and so is it remarked in their description and terminology. However, it is mostly the case that similarities with respect to previous algorithms are not highlighted – even not acknowledged at all. Furthermore, no proper performance comparison is done with state-of-the-art variants of currently available heuristics. When addressing real-world optimization problems, achieving competitive results and a good searching behavior are the most important factors, not the originality of their source of inspiration [2]. Thus, when new bio-inspired algorithms are introduced, the proposing authors should incorporate experiments by using any of the proposed benchmarks, and thereby check competition websites to find state-of-the-art algorithms to use them as a baseline in their selected benchmarks. Responsible research when proposing new bio-inspired optimization methods should be also enforced by reviewers and other stakeholders along the editorial workflow, ensuring that quantitative, statistically reliable evidences of the claims held in upcoming publications are provided by the authors.

In this context we summarize in Table 1 the main proposed benchmarks in the literature related to different type of optimization problems, remarking some of their characteristics. For all these benchmarks:

- A group of optimization functions is considered, with different levels of difficulty.
- A measure of error is computed for each function.
- A different number of dimensions is considered for each function.

Table 1
Benchmarks commonly used in selected areas of bio-inspired optimization.

Area	Benchmark [Reference]	Main Characteristics
Continuous (real-valued) optimization	CEC'2005 [320]	First benchmark, all functions shifted
	CEC'2011 [321]	Real-world problems, small dimensions
	CEC'2013 [322]	Rotated and shifted functions
	CEC'2014 [323]	More multimodal functions
	CEC'2014 Expensive [324]	Reduced number of evaluations
	CEC'2015 [325]	Allowed specific parameter values for functions
	CEC'2017 [326]	Composed test problems by extracting features dimension-wise for several problems
	BBOB [327]	Functions with increasing dimensionality (from small dimensions)
Multi-objective optimization	CEC'2009 [328] and editions ever since, ZDT [329], DTLZ [330], WFG [331], LZ [332], UF [328] and others [333]	From classical functions to dynamic benchmarks
Many-objective optimization	CEC'2017 [334], CEC'2018 [335] and others [336,337]	Many more targets than in multi-objective benchmarks
Dynamic optimization	Moving peaks [338,339]	Landscape change during the runtime, recently dynamic multi-objective benchmarks
Multimodal optimization	CEC'2013 [340] and editions ever since, new benchmark functions featuring linkage across dimensions [341]	Measure performance is the ratio of found optima for different thresholds
	Newer benchmarks for multimodal multi-objective optimization [342]	Performance assessed in terms of multi-objective quality indicators and Pareto set retention capability
LSGO	CEC'2013 [277] and editions ever since	Dimension equal to 1000, different degrees of relationship between variables

- Each algorithm is run several times (usually no less than 25) for each function and dimension. The more times the algorithm is executed, the better the statistical characterization of the algorithm's performance will be.
- A stopping criterion is set, which can be either a maximum number of function evaluations, that linearly depends on the dimension value, or a required accuracy level, so that the performance of each algorithm is given by the number of evaluations needed to achieve it.
- A criterion is defined to rank the algorithms. In multimodal optimization, the measure to optimize is the ratio of local optima found by the algorithm at hand; in multi- and many-objective there are specific measures related to the Pareto optimality and diversity; in real optimization the criterion relies in a weighted global ranking measure, granting more importance to the results obtained for more difficult functions (higher dimensions). This primary ranking criterion is complemented by a feasibility rate should the problem include constraints in its definition.

We refer to Ref. [317] for further information about optimization benchmarks and its evolution and winners through the years of competitions.

Specifically for continuous (real-valued) optimization two different benchmarks have evolved along the years: 1) benchmarks proposed within the IEEE Congress on Evolutionary Computation (CEC), and 2) the Black-Box Competition within the Genetic and Evolutionary Computation (GECCO) conference, with an increasing difficulty of the majority of functions. They also differ in the criterion adopted to compare algorithms: in CEC competitions the maximum number of evaluations is fixed, and the error is used to compare the algorithms (*fixed cost*). By contrast, in BBOB competitions the expected accuracy is indicated, and the algorithms are compared by considering the number of evaluations needed by a competing algorithm to achieve a given level of precision (correspondingly, *fixed target*). Lately *expensive optimization* benchmarks have been also proposed in which the number of evaluations is significantly reduced to target algorithms suited for realistic situations where function evaluations could be very costly in time (e.g. function evaluations provided by computationally intensive simulations).

The criterion used to measure the performance of each benchmark strongly depends on the area. In real-parameter optimization, the average error or the average number of evaluations is obtained, and in multimodal-optimization is used the ratio of found optima. For multi- and many-objective problems, there are special measures for the Pareto front. The benchmarks, to compare algorithms, usually only the average measure (and occasionally the median, because average is very sensible to data) is used to compare algorithms, remarking which algorithm obtain best results by each function. Since in some cases highlighting which algorithm is better in each functions is not informative enough, it is very common to calculate the average ranking in order to sort all algorithms as per a single global ranking. Recently, rankings are transformed into a certain number of points (more points to better ones), combining the total ranking for each dimension when there are several dimension values (giving more importance to results with greater dimension value, because the difficulty increases with the dimension). This is actually mainly used for the organizers of competitions rather than in comparisons made in research papers. We believe that this comparison methodology could be more widely adopted in the future, not only for competitions and other workshops alike.

Nowadays, in order to compare several algorithms it is crucial to ensure that improvement in results are not due to stochastic differences in runs. Thus, it is mandatory to apply statistical tests to clarify the statistical significance of the performance gaps found among algorithms [343]. However, the majority of benchmarks in Table 1 do not consider the application of statistical testing, neither in their experimental setting (the number of runs is lower than those recommended for them), nor in the comparison criterion established by the organizers. About statistical testing, it has been proven that required assumptions were not fulfilled in some benchmarks [344], so in general these statistical tests are not recommended. Non-parametric tests can be used instead which, despite less powerful, do not require any assumption to be met by the sample beforehand. Among them, the most popular is arguably the Wilcoxon test, that allow comparing results achieved by two algorithms. Since in these non-parametric tests the expected error increase with each one-by-one comparison, for multiple comparisons post-hoc tests such as Holm/Hochberg/Hommel (they are very similar to each other) are rather adopted to maintain the expected error controlled. A prescription of good practices on statistical tests for comparison among EC and SI algorithms can be found in Refs. [345,346]. Recently, new trends in the field of statistical comparison of algorithms indicate that Bayesian tests, which provides a distribution over the parameter of interest, are a promising approach in this regard. The meaning of p-value in null hypothesis statistical tests (parametric and non-parametric) is usually misinterpreted, as if it were the probability of not complying with the null hypothesis. Bayesian tests actually return the probability distribution of the null hypothesis. Therefore, while in the hypothesis statistical tests when null hypothesis is not rejected there is no information at all, Bayesian tests provide useful information. In addition, they offer more robust results, as they are not as influenced by the number of observations as the previous ones [347].

There is an open challenge that has become increasingly entangled through the years [317]. In theory these benchmarks should serve to identify which bio-inspired algorithms offer the most promising behavior when optimizing different problems. Their ultimate goal should be to dictate which algorithms to apply to a new real-world problem, choosing from the most promising alternatives tested in functions and problems similar (or at least, related) to the problem at hand. However, over the last decade benchmarks have undergone a sharp evolution mainly in terms of an increasing level of difficulty, reaching a situation where it is possible that the functions included in these benchmarks are overly complex when compared to real-world problems. Thus, an unsolved issue for benchmarks is to bridge the computational gap between part of their constituent functions and the characteristics of real-world problems, towards improving the relevance and practicality of these artificially-constructed benchmarks.

Another important disadvantage in these benchmarks is that the used stopping criterion is not realistic. In real-world problems it is not only interesting to discern which algorithm performs best with a fixed and non-realistic number of evaluations, but it is also relevant to assess which algorithm attains an acceptable error within the minimum time/number of evaluations. Thus, benchmarks measuring the evaluations to achieve a certain accuracy are more realistic, when the expected accuracy is adequate. In general the scalability of algorithms does not receive enough attention in the competition benchmarks. Moreover, a minority of contributions has studied how evaluation numbers influence the convenience of one type of algorithm over another. In Ref. [348] it is observed that for a low number of evaluations, mathematical methods could be better suited than evolutionary algorithms. However, extensive competitions have shown also that for a reduced number of evaluations, specifically designed evolutionary algorithms are also a good option. This conclusion goes in line with recent studies about how bio-inspired optimization algorithms perform over budget-limited problem instances with respect to deterministic global optimization methods [349]. In Ref. [350] the performance of many algorithms is studied, concluding that PSO algorithms are more adequate for a low number of evaluations, while DE algorithms can obtain more precise results but they often require a higher number of evaluations. These type of studies are very useful and practical, yet lacking in the current literature. This promising research line could reduce the gaps between theoretical benchmarks and the applicability of bio-inspired heuristics to real setups.

To summarize, even while the proposal and use of benchmarks have become a *de-facto* standard in almost all areas of bio-inspired optimization, there is still room for improvement and open issues to be addressed, such as new benchmarks with reduced gap among them and real-world problems [351], and new and better comparison methodologies, including more attention to scalability and new statistical testing approaches such as the use of Bayesian tests.

3.14. Applications of bio-inspired optimization

Bio-inspired algorithms and methods have been successfully applied into a wide variety of domains [30]. A comprehensive and detailed analysis of the current applications based on bio-inspired methods is outside the scope of this article. However, it is interesting to delve into some highly relevant areas of applications, such as social data analysis, medicine and health, cybersecurity, or video games (among many others), where these methods have been profusely utilized in recent times. Next we provide a short description of these selected application areas, along with a brief digest of the activity therein related to bio-inspired methods:

- *Social Network Analysis* (SNA), which comprises topics such as social-based applications for data mining, data analysis, community detection or social mining, has received an increasing attention from the research community. The main goal targeted in SNA is to identify structures and patterns in social-based information sources [352]. This area is inherently interdisciplinary, and covers areas such as data mining, machine learning, statistics, complex systems, graph theory, information retrieval, natural language processing, semantic web, and big data computing, among others. When it comes to bio-inspired algorithms and SNA, several EA and SI approaches have been utilized over single- and multi-objective problems modeled over social network structures [353,354]. Bio-inspired methods have been used to guide the search process for inner structures, such as clusters or communities, towards finding the most representative nodes (e.g. authorities) in the network [355], analyzing and optimizing the diffusion of information throughout the network [356], isolating the so-called ego network of selected central nodes [357], or studying the dynamics of a network when modeling a non-stationary information source [358]. Besides general studies reported on generic graph structures, many practical applications can harness the insights provided by bio-inspired optimization techniques, such as e-health [359], smart cities [360] or energy transmission networks [361]. An exhaustive survey of evolutionary algorithms for community detection in networks has been recently presented in Ref. [362].
- *Medicine and health systems* have benefited from Artificial Intelligence methods since the advent of this research field, as exposed by the development of the first expert systems for disease diagnosis. The increasingly complex challenges faced by medical and health systems in the last years have grown the need for new decision support systems aimed to help health experts improve the diagnosis accuracy. Several recent approaches have elucidated that bio-inspired optimization techniques can play a crucial role in this regard. For instance, in Ref. [363] a technique based on the Artificial Bee Colony (ABC) algorithm is proposed to efficiently determine the IIR filter coefficients capable of eliminating Doppler noise present in the aortic valve. In Ref. [364], an enhanced version of the aforementioned ABC solver is proposed to diagnose breast cancer: it is used to automatically detect the breast border and nipple position, so that the suspicious regions are identified using bilateral subtraction. In Ref. [365] an improved ACO algorithm is used to segment MRI brain images, which is indeed a particularly prevailing medical application addressed by the bio-inspired optimization community (see e.g. Refs. [366,367] and references thereafter). Likewise, in Ref. [368] a novel method relying on adaptive bio-inspired algorithms – namely, ACO, Bee Colonies Optimization (BCO) and GA – is introduced for selecting features extracted from a mammogram image. Several other approaches hybridize PSO with other methods so as to improve the current state of the art of several medical problems. In Ref. [369], for instance, a technique using finite-difference frequency domain is hybridized with PSO for reconstructing cell dimension in breast cancer and to find its position using 2-D and 3-D breast models. In Ref. [370], a method for segmenting breast tumor images using modified automatic seeded region growing based on PSO-based image clustering is proposed. The hybrid approach in Ref. [371] combines together SI and neural networks for the detection and classification of micro calcifications in mammogram images. Finally, modern bio-inspired approaches are lately entering the health arena, such as [372] which resorts to the Firefly Algorithm for breast cancer classification, or [373], where the Bat Algorithm is used to isolate skin lesions in medical images. Compendiums on the applications of bio-inspired methods to medicine and health can be accessed in Refs. [374–376].
- *Cybersecurity*, which recurrently informed security breach incidents have spurred on the interest in methods aimed to increase the resilience of computer systems and processes. This field currently spreads beyond local security threats such as virus detection and intrusion detection to also consider more elaborated, global phenomena such as cyberterrorism [377] or cybercrime [378]. Whatever the threat may be, bio-inspired methods have been lately applied to improve the overall response and resiliency of computer-based systems, thereby facing with success common computer attacks such as phishing [379], eavesdropping [380], Denial-of-Service or spoofing [381]. Closely related to these exemplified cases, the automatic detection of malware has also leveraged the application of bio-inspired algorithms in recent contributions. For instance, in Ref. [382] a hybrid method encompassing an adaptive neuro fuzzy inference system and PSO is proposed to find the optimum parameters that can be used to facilitate mobile malware identification. The work presented in Ref. [383] reports the development of an innovative active security system that acts as an extension on the ART (Android Run Time) Virtual Machine architecture, and uses a Biogeography-Based bio-inspired solver for training a Multi-Layer Perceptron that classifies Java classes of a software application as benign or malicious. Finally, in Ref. [384], a novel method that employs Deep Learning models to improve the detection of malware is proposed. In this approach a Bat Algorithm addresses the problem of class imbalance among different malware families, which is a frequent problem in data-based studies related to cybersecurity. We refer to Refs. [385–389] for baseline material about the past history of bio-inspired computation applied to problems related to cybersecurity, as well as a prospect of the research directions in this matter.
- *Video games* have massively embraced Artificial Intelligence in a large number of problems and challenges, such as Procedural Content Generation (PCG), bots generation or the development of virtual players [390]. To begin with, the automated creation of content for video games via PCG techniques [391] is a critical aspect. The industrial benefits, economically speaking, are clear: PCG can reduce development costs and enlarge the life of commercial video games (with the corresponding earnings increase). For this reason, most PCG methods are focused on the generation of a specific type of contents (e.g. maps o non-player characters strategies). Different approaches, such as co-evolutionary competitive bio-inspired algorithms, have been proposed to generate simultaneously two different kind of contents, namely maps and game AI [392]. The development of game bots that can dynamically adapt to different difficulty levels as well as variable game environments is another research line with AI methods at its core. An example is [393], which focuses on developing a generic framework, called *AntBot*, that builds on SI methods (in particular, ACO heuristics), to imple-

ment real-time gaming bots for dynamic game environments. In Ref. [394], the performance of ACO and GA was assessed when dealing with graph-constrained models in video games problems. In Ref. [395], a naive PSO solver was shown to automatically create complex two-dimensional graphic characters by evolving modifications made to a base character. Overviews on the applicability of bio-inspired optimization to games have been published during the last decade [396,397], but new paradigms such as *serious games* have lately rekindled the interest in this topic and enlarged its application scope far beyond leisure (see e.g. Ref. [398] for rehabilitation robotics).

The above list of selected topics must be regarded as an exemplifying excerpt of the plethora of applications where bio-inspired optimization is becoming a relevant technological enabler. Many other fields such as intelligent transportation systems [399–401], automated manufacturing [402–405] or energy systems and smart grids [406–409] have also been bolstered by bio-inspired optimization methods. In all these sectors we discriminate several challenges related to the practicality of bio-inspired solvers when deployed in real problem setups. First, real-world scenarios are often subject to constraints that must be modeled and included in the mathematical problem statement but, most importantly, considered by the optimization algorithm during its search process. When this is the case, many strategies to handle different types of constraints have been reported by the community [235,410–412]. However, there is no clear consensus about how to deal with practical problems that undergo a high number of restrictions. When this is the case, it might not be of practical value to face it as an *optimization* problem, but rather as a *constraint satisfaction* problem [413,414]. Different design patterns can be followed depending on how the problem is modeled, but it is the added value to the application what should drive the decision whether to opt for one modeling approach or the other. Just finding a solution satisfying all imposed constraints is enough in many practical cases. From a research perspective, the community lacks a formal, methodological study on the conditions that should be met to select a good modeling approach for a practical optimization problem, capable of realistically reflecting its particularities and constraints (such as latency, complexity or scalability), yet properly balancing the cost per value of the solver designed to solve it efficiently.

Aligned with the above objective, the expert knowledge accumulated by users of the system to be optimized along years of operation is a very valuable source of inspiration for the design of efficient ad-hoc heuristics [415]. In a sense, it can be regarded as the byproduct of an adaptation of a learning machine (the human) to a problem as a result of a series of repeated trial-and-measure iterations over time. Unfortunately, expert knowledge is often overseen in practice; meta-heuristic algorithms are instead preferred to expedite the achievement of good solutions at the cost of a slower convergence or a lower optimality. In problems of increasing complexity and/or when convergence speed is a critical design factor, expert knowledge should be exploited during the search process by including specialized operators that model the solving patterns followed traditionally in the scenario at hand. In most practical situations information revealed by experts can yield better search operators or higher convergence rates; in short, a better overall search performance than those of the most sophisticated bio-inspired optimization method.

4. Bio-inspired computation: a curly road ahead

In the previous section we have identified possible research paths to follow in important research areas within bio-inspired computation. Unfortunately, the field still undergoes general issues that threaten to jeopardize true advances in years to come. A synergistic push from the community should be made towards addressing these issues for the benefit of Science. We herein provide some thoughts so as to constructively foster research efforts in such directions:

4.1. More is not always better

The first big issue within bio-inspired computation is to decide whether we need to improve methods discovered so far by the community, or instead look for new biological sources of inspiration to conceive new algorithmic developments. The recently witnessed controversy around metaphor-based approaches has not achieved any common grounds in regards to the strategy that the community should embrace in regards to this field, nor has it stopped the emergence of more and more optimization techniques relying on allegedly innovative bio-inspired methods [416]. It is sadly concerning that part of this literature outbreak is motivated by a lack of perspective about the real needs of the field. However, without a consensus on how new algorithms should be evaluated both theoretically (novelty, properties) and empirically (comparison methodologies, benchmark problems), it is absolutely unfeasible to separate the *wheat* from the *chaff*. We herein advocate for starting over a clean sheet in the field, and thus focus on the fundamental paradigms that underlay bio-inspired computation as a whole, so that new advances to come will help clear up controversial questions within the community.

4.2. Towards a unified notation and description of bio-inspired algorithms

Closely related to the above, much has been discussed around the analogy between *old* and *new* bio-inspired heuristics, specially within SI. Notable are the cases of Harmony Search and Evolution Strategies [417,418]; Particle Swarm Optimization and Firefly Algorithm [419]; and Ant Colony Optimization [420] and Intelligent Water Drops [421]. In most cases disagreements could have been avoided by unifying the semantics by which bio-inspired heuristics are described, so that their novel ingredients can be put on relevance in a more cohesive manner. As a matter of fact, the myriad of new bio-inspired algorithms that are reported on a regular basis justifies by itself the adoption of a standard notation focused on the domain-agnostic description of new algorithmic operators and design patterns of heuristics and meta-heuristics. Such a standardized, metaphor-free vocabulary would prevent the community from obscure mathematical formulae widely employed nowadays to obfuscate the real mechanisms of newly proposed methods. By virtue of this metaphor-free description, for instance, a candidate solution would be identified explicitly as such and not as e.g. an egg, a water drop or a bee's nest. This need for notational uniqueness was highlighted recently in Ref. [422], where metaphorical aspects characterizing different honey bee inspired solvers were decoded to standardized optimization terms. Other benefits would stem from this standardization process beyond the assessment of differences and similarity between bio-inspired solvers, such as a higher modularity and reusability of heuristic components, a better detection of possible sources of unnecessary algorithmic complexity, and a more straightforward and reliable reproducibility of results. Even though a manifesto for transparency through descriptive standardization was already published years ago [423], no significant steps have been taken ever since in this direction, despite the significant impact any minor advance could incur in the field.

4.3. New theoretical directions to better understand bio-inspired algorithms

Different empirical observations and numerical simulations have elucidated that bio-inspired algorithms can work surprisingly well in practice, but in most cases we rarely understand why they work under the given conditions for a given type of problem [424,425]. Though there are some progress in theoretical analysis as we have seen earlier, it is highly needed to use more systematic approaches – ideally, a unified framework – to analyze bio-inspired algorithms so as to gain mathematical insight into their working mechanisms, to estimate their rates of convergence and assess their conditions for stability. In this regard, fresh theoretical studies lately proposed within the community seem to

be promising for analyzing and understanding bio-inspired heuristics, such as the use of network science to characterize swarm-based algorithms [426]. Ultimately, we hope that such insights can help to choose the right algorithms for a given set of problems so as to solve them efficiently.

4.4. From efficacy to efficiency in bio-inspired optimization

When it comes to performance assessment, most contributions to date revolve around effectiveness (e.g. fitness statistics) for a given complexity level (correspondingly, number of fitness evaluation) when comparing among different bio-inspired approaches. Nonetheless, there is an increasing global concern with energy efficiency, which has lately coined the so-called Green Computing concept [427]. Algorithms under this umbrella are designed with environmental sustainability as a design goal itself, imprinting severe constraints in several steps of their execution thread. Important modifications result from the adoption of this design directive, in aspects such as resource allocation, memory indexing or processing time. Most importantly, the way an algorithm is implemented also renders a significant effect on the its actual efficiency, which calls for the adoption of Green Computing from the very beginning of the algorithm design procedure. Good practices aligned with this range should enforce the community to always perform a complexity analysis of novel algorithmic proposals (by quantifying it in terms of number of sums/products and other similar implementation-agnostic indicators). In any case, studies on the algorithmic efficiency of bio-inspired approaches should avoid reporting on measurements closely linked to the implementation and deployment of the algorithm itself, such as timing logs or net memory consumption, which are strongly biased by non-algorithmic matters. A correspondence between the characteristics of algorithmic components and their expected carbon footprint should be also derived in the future.

4.5. Bio-inspired machine learning and deep neural processing

Traditionally the useful intersection between bio-inspired optimization and Machine Learning (ML) has drawn close attention in the literature, combining both fields of knowledge into single data-based models for manifold reasons: to mention a few, bio-inspired algorithms have been profusely exploited to expedite the learning process of ML models, with particular emphasis on neural networks of different kind [428,429]. Correspondingly, ML models lie at the core of surrogate-model assisted optimization in which, as introduced in Subsection 3.6, computationally expensive objectives are replaced by cheaper regression models built on a few evaluated individuals [430]. Likewise, bio-inspired heuristics are often utilized for prescribing near-optimal actions based on the predictions produced by ML models, completing what has lately been known as actionable data science [431]. Examples of the mutually rewarding relationship between Markov Chain Monte Carlo and bio-inspired optimization are also enlightening: the efficiency of complex MCMC simulations has been shown to improve by leveraging DE-based heuristics [432], whereas DE solvers encompassing MCMC elements have been shown to better choose the scale and orientation of the distribution modeling the underlying mutation operator [433]. Recently Wang et al. [434] evolved a population of generators which competed against the discriminator network in the framework of Generative Adversarial Networks (GANs) through an adversarial game and by using specially defined mutation operators. The resulting evolutionary GAN not only alleviated some of the inherent problems associated with the training of conventional GANs, but also exhibited improved generative capabilities.

Besides other multiple scenarios leveraging this profitable complementarity (such as feature selection/construction [435], opposition-based learning for bio-inspired optimization methods [436] or their hybridization with elements from Reinforcement Learning [437]), possibilities for the future are foreseen to sprout sharply given the enormous number of parameters featured by the family of Deep Learning models, which are lately prospecting the adoption of bio-inspired LSGO techniques as an efficient replacement for gradient back-propagation [438,439]. Another ML research niche with a noted predominance of bio-inspired heuristics is automated ML which, despite being known for decades under different denominations (e.g. neuro-evolution [440]), it is now when the area has been reforged as a result of the increasing design complexity and variety of algorithmic components of Deep Learning models [441–443]. Surely ML and bio-inspired computation will enjoy a mutually rewarding marriage in years to come.

4.6. New challenges of bio-inspired techniques for human-centric applications

Human-centric applications such as Video Games or Virtual Reality/Augmented Reality (VR/AR) are currently at the forefront of scenarios where bio-inspired optimization can provide unprecedented levels of machine intelligence [444]. Such applications are often characterized by a continuous interaction between the user and the machine, thereby requiring superior capabilities of the underlying algorithms for dynamic adaptation, incremental learning and bounded complexity. Despite these computational constraints, the countless opportunities that bio-inspired computation may bring to these application domains – from improved self-localization to the optimized simulation of crowds or the optimal manipulation of virtual objects– can drive significantly efforts towards adapting them to these computing environments. For instance, latency in VR/AR is known to be severely limited by the so-called motion to photon threshold (~20 ms), which poses challenging design constraints on any bio-inspired optimization method designed to e.g. improve the user experience or optimize the rendering process of media content from streaming servers. Here we venture a vast algorithmic territory to be explored in the future, unleashing new interesting paradigms such as continual optimization for multiple, recurrently varying problem statements (in clear connection to Continual/Lifelong Machine Learning [445]) and self-impacting models, in which actions taken on the basis of their predictions may affect subsequent predictive outcomes thereafter.

4.7. Bio-inspired optimization and emerging computing paradigms

In addition to the many-sided research avenues outlined above, the entire community should keep a close eye at the impending arrival of new computing paradigms, from the Map/Reduce model that lies underneath Big Data architectures to Ephemeral Computing, Exascale Computing and Quantum Computing. Roughly a decade ago we did not expect computing technology to evolve as fast as we have witnessed ever since, developing data-intensive technologies capable of ingesting, storing and handling huge amounts of data. Nowadays Map/Reduce implementations of bio-inspired algorithms are available for their deployment on Big Data platforms [446–448]. In terms of processing power a similar trend can be noted nowadays in the form of Ephemeral Computing [449] and Exascale Computing [450], both providing efficient means for scaling up complex bio-inspired algorithms. Yet still at its infancy, Quantum Computing is already spanning its applicability towards the ML realm and anticipating astonishing gains in terms of processing throughput for many other fields, including ML and large-scale optimization [451,452].

We do not know which other computing paradigms we will encounter in the future and most importantly, how they will impair the design and deployment of bio-inspired optimization methods. Nevertheless, we must prepare ourselves for their eventual arrival by conducting research efforts along valuable directions in this field. Unless we all acknowledge this pressing need for joining forces and agree on the real priorities of bio-inspired computation, we will wander erratically and blind through incremental research paths that lead nowhere in this field.

5. Conclusion: an exciting future for bio-inspired computation

In this manuscript we have shared our envisioned status of bio-inspired computation, which calls for a profound reflection on the research paths that the community should follow in the future. To this end, we have briefly reviewed the history of this field from the very advent of EC to the plethora of new SI methods appearing in the late literature. Grounded on this historical perspective we have identified research paths for a number of selected areas within bio-inspired optimization that should congregate most of the global research efforts in years to come.

Nature is truly fascinating and full of intriguing phenomena still to be understood. Unrevealed paradigms underneath this science will surely continue fostering new advances in bio-inspired optimization, featuring unseen levels of performance and computational efficiency. Today we, the research community, have the chance to leave misleading research paths behind, and face together an overwhelming future for this field in a harmonized, principled and scientifically enriching manner.

Acknowledgements

Javier Del Ser and Eneko Osaba would like to thank the Basque Government for funding support received through the EMAITEK and ELKARTEK funding programs. The work of Sancho Salcedo-Sanz and David Camacho is partially supported by the Ministerio de Economía y Competitividad (MINECO) of Spain (grant no. TIN2017-85887-C2-2-P and TIN2017-85727-C4-3-P, respectively). The work of Daniel Molina and Francisco Herrera is supported by the Spanish Ministry of Science (grants TIN2016-8113-R and TIN2017-89517-P). Carlos A. Coello Coello acknowledges support from CONACYT grant no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*). Javier Del Ser would also like to thank Prof. Josu Ceberio from the University of the Basque Country (UPV/EHU) and Dr. Esther Villar-Rodriguez from TECNALIA for fruitful discussions that improved this manuscript. All the authors are grateful to the anonymous reviewers for their insightful feedback.

References

[1] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, 1975.
[2] K. Sörensen, Metaheuristics – the metaphor exposed, Int. Trans. Oper. Res. 22 (1) (2015) 3–18.
[3] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley, 1966.
[4] H.-P. Schwefel, Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie, vol. 26, ISR, Birkhaeuser, 1977.
[5] I. Rechenberg, Evolution Strategy: Optimization of Technical Systems by Means of Biological Evolution, vol. 104, Fromman-Holzboog, Stuttgart, 1973, pp. 15–16.
[6] T. Bäck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, CRC Press, 1997.
[7] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[8] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 1 (11) (2003) 1–18.
[9] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: 2005 IEEE Congress on Evolutionary Computation, vol. 2, IEEE, 2005, pp. 1769–1776.
[10] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 71–78.
[11] R. Thomsen, Multimodal optimization using crowding-based differential evolution, in: IEEE Congress on Evolutionary Computation, vol. 2, 2004, pp. 1382–1389.
[12] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[13] X.-S. Yang, Firefly algorithms for multimodal optimization, in: International Symposium on Stochastic Algorithms, 2009, pp. 169–178.
[14] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[15] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[16] A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.C. Coello, F. Luna, E. Alba, SMPSO: A new PSO-based metaheuristic for multi-objective optimization, in: 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM), IEEE, 2009, pp. 66–73.
[17] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, vol. 16, John Wiley & Sons, 2001.
[18] C. Coello Coello, Evolutionary multi-objective optimization: A historical view of the field, IEEE Comput. Intell. Mag. 1 (1) (2006) 28–36.
[19] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: a short review, in: IEEE Congress on Evolutionary Computation, 2008, pp. 2419–2426.
[20] S. Mahdavi, M.E. Shiri, S. Rahnamayan, Metaheuristics in large-scale global continues optimization: A survey, Inf. Sci. 295 (2015) 407–428.
[21] E. Alba, A. Nakib, P. Siarry, Metaheuristics for Dynamic Optimization, vol. 433, Springer, 2013.
[22] E. Bonabeau, M. Dorigo, G. Théraulaz, Swarm Intelligence: from Natural to Artificial Systems, Oxford University press, 1999.
[23] M. Dorigo, Optimization, Learning and Natural Algorithms. (Ph.D. thesis), Politecnico di Milano, 1992.
[24] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: IEEE International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[25] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[26] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. 22 (3) (2002) 52–67.
[27] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Tech. rep., Technical report-tr06, Erciyes University, 2005.
[28] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84.
[29] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, Metaheuristic research: A comprehensive survey, Artif. Intell. Rev. (2018) 1–43.
[30] A.K. Kar, Bio inspired computing–a review of algorithms and scope of applications, Expert Syst. Appl. 59 (2016) 20–32.
[31] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: IEEE Congress on Evolutionary Computation, 2007, pp. 4661–4667.
[32] A. Ahmadi-Javid, Anarchic society optimization: a human-inspired method, in: IEEE Congress on Evolutionary Computation, 2011, pp. 2586–2592.
[33] T. Ray, K.M. Liew, Society and civilization: an optimization algorithm based on the simulation of social behavior, IEEE Trans. Evol. Comput. 7 (4) (2003) 386–396.
[34] A. Duarte, F. Fernández, Á. Sánchez, A. Sanz, A hierarchical social metaheuristic for the max-cut problem, in: European Conference on Evolutionary Computation in Combinatorial Optimization, 2004, pp. 84–94.
[35] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110 (2012) 151–166.
[36] A. Wedyan, J. Whalley, A. Narayanan, Hydrological cycle algorithm for continuous optimization problems, J. Optim. (2017) 25 Article ID 3828420.
[37] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[38] A.H. Kashan, A new metaheuristic for optimization: Optics inspired optimization (OIO), Comput. Oper. Res. 55 (2015) 99–125.
[39] S. Salcedo-Sanz, Modern meta-heuristics based on nonlinear physics processes: a review of models and design procedures, Phys. Rep. 655 (2016) 1–70.
[40] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: a literature review, Swarm Evolut. Comput. 2 (2012) 1–14.
[41] M.A. Potter, The Design and Analysis of a Computational Model of Cooperative Coevolution. (Ph.D. thesis), George Mason University, 1997.
[42] R.S. Parpinelli, H.S. Lopes, New inspirations in swarm intelligence: A survey, Int. J. Bio-Inspired Comput. 3 (1) (2011) 1–16.
[43] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evolut. Comput. 1 (1) (2011) 32–49.
[44] S. Binitha, S.S. Sathya, et al., A survey of bio inspired optimization algorithms, Int. J. Soft Comput. Eng. 2 (2) (2012) 137–151.
[45] X.-S. Yang, S. Deb, Y.-X. Zhao, S. Fong, X. He, Swarm intelligence: Past, present and future, Soft Comput. (2018) 1–11.
[46] P.N. Suganthan, Numerical optimization by DE and PSO, in: Tutorial Lecture at IEEE SDE/SIS/SSCI 2018, Bengaluru, India, 2018, Available at: http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/PDFs/Numerical%20Optimization.pdf.
[47] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[48] C. Schumacher, M.D. Vose, L.D. Whitley, The no free lunch and problem description length, in: 3rd Annual Conference on Genetic and Evolutionary Computation, 2001, pp. 565–570.
[49] C. Igel, M. Toussaint, A no-free-lunch theorem for non-uniform distributions of target functions, J. Math. Model. Algorithms 3 (4) (2005) 313–322.
[50] D. Whitley, J. Rowe, Focused no free lunch theorems, in: Annual Conference on Genetic and Evolutionary Computation, 2008, pp. 811–818.
[51] C. Igel, No free lunch theorems: limitations and perspectives of metaheuristics, in: Theory and Principled Methods for the Design of Metaheuristics, 2014, pp. 1–23.

[52] T. Joyce, J.M. Herrmann, A review of no free lunch theorems, and their implications for metaheuristic optimisation, in: Nature-Inspired Algorithms and Applied Optimization, 2018, pp. 27–51.
[53] F. Neumann, C. Witt, Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity, Springer Science & Business Media, 2010.
[54] A. Auger, B. Doerr, Theory of Randomized Search Heuristics: Foundations and Recent Developments, vol. 1, 2011.
[55] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.
[56] X.-S. Yang, Metaheuristic optimization: algorithm analysis and open problems, in: International Symposium on Experimental Algorithms, 2011, pp. 21–32.
[57] X.-S. Yang, Mathematical analysis of nature-inspired algorithms, in: Nature-Inspired Algorithms and Applied Optimization, 2018, pp. 1–25.
[58] E. Pitzer, M. Affenzeller, A comprehensive survey on fitness landscape analysis, in: Recent Advances in Intelligent Engineering Systems, 2012, pp. 161–191.
[59] N.D. Jana, J. Sil, S. Das, Selection of appropriate metaheuristic algorithms for protein structure prediction in ab off-lattice model: a perspective from fitness landscape analysis, Inf. Sci. 391 (2017) 28–64.
[60] T. Jones, Evolutionary Algorithms, Fitness Landscapes and Search. (Ph.D. thesis), The University of New Mexico, Albuquerque, New Mexico, 1995.
[61] F. Chicano, L.D. Whitley, E. Alba, A methodology to find the elementary landscape decomposition of combinatorial optimization problems, Evol. Comput. 19 (4) (2011) 597–637.
[62] M. Wang, B. Li, G. Zhang, X. Yao, Population evolvability: dynamic fitness landscape analysis for population-based metaheuristic algorithms, IEEE Trans. Evol. Comput. 22 (4) (2018) 550–563.
[63] H. Lu, R. Zhou, Z. Fei, C. Guan, Spatial-domain fitness landscape analysis for combinatorial optimization, Inf. Sci. 472 (2019) 126–144.
[64] Z.-Z. Liu, Y. Wang, S. Yang, K. Tang, An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms, IEEE Trans. Cybern. 99 (2018) 1–14.
[65] X.-S. Yang, Efficiency analysis of swarm intelligence and randomization techniques, J. Comput. Theor. Nanosci. 9 (2) (2012) 189–198.
[66] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. (CSUR) 45 (3) (2013) 35.
[67] J. Lehman, K.O. Stanley, Novelty search and the problem with objectives, in: Genetic Programming Theory and Practice IX, 2011, pp. 37–56.
[68] M. Weber, V. Tirronen, F. Neri, Scale factor inheritance mechanism in distributed differential evolution, Soft Comput. 14 (11) (2010) 1187–1207.
[69] F. Vitaliy, Differential Evolution–In Search of Solutions, 2006.
[70] A.V. Kononova, D.W. Corne, P. De Wilde, V. Shneer, F. Caraffini, Structural bias in population-based algorithms, Inf. Sci. 298 (2015) 468–490.
[71] A.P. Piotrowski, J.J. Napiorkowski, Some metaheuristics should be simplified, Inf. Sci. 427 (2018) 32–62.
[72] J. Branke, Evolutionary Approaches to Dynamic Optimization Problems – A Survey, Technical Report 387, University of Karlsruhe, 1999.
[73] D.V. Arnold, H.-G. Beyer, Random dynamics optimum tracking with evolution strategies, in: International Conference on Parallel Problem Solving from Nature, 2002, pp. 3–12.
[74] D.V. Arnold, H.-G. Beyer, Optimum tracking with evolution strategies, Evol. Comput. 14 (3) (2006) 291–308.
[75] S. Jiang, S. Yang, X. Yao, K.C. Tan, M. Kaiser, N. Krasnogor, Benchmark Functions for the CEC'2018 Competition on Dynamic Multiobjective Optimization, Tech. rep., Newcastle University, 2018.
[76] C. Cruz, J.R. González, D.A. Pelta, Optimization in dynamic environments: A survey on problems, methods and measures, Soft Comput. 15 (7) (2011) 1427–1448.
[77] M. Mavrovouniotis, C. Li, S. Yang, A survey of swarm intelligence for dynamic optimization: algorithms and applications, Swarm Evolut. Comput. 33 (2017) 1–17.
[78] X. Hu, R.C. Eberhart, Adaptive particle swarm optimization: Detection and response to dynamic systems, in: IEEE Congress on Evolutionary Computation, vol. 2, 2002, pp. 1666–1670.
[79] S. Janson, M. Middendorf, A hierarchical particle swarm optimizer for noisy and dynamic environments, Genet. Program. Evolvable Mach. 7 (4) (2006) 329–354.
[80] H.G. Cobb, An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-dependent Nonstationary Environments, Tech. rep., Naval Research Lab, Washington DC, 1990.
[81] C.-K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 13 (1) (2009) 103–127.
[82] H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, H. Zhou, Multi-population techniques in nature inspired optimization algorithms: a comprehensive survey, Swarm Evolut. Comput. 44 (2019) 365–387.
[83] D. Yazdani, T.T. Nguyen, J. Branke, J. Wang, A new multi-swarm particle swarm optimization for robust optimization over time, in: European Conference on the Applications of Evolutionary Computation, 2017, pp. 99–109.
[84] P. Novoa-Hernández, C.C. Corona, D.A. Pelta, Self-adaptation in dynamic environments-a survey and open issues, Int. J. Bio-Inspired Comput. 8 (1) (2016) 1–13.
[85] T.T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: a survey of the state of the art, Swarm Evolut. Comput. 6 (2012) 1–24.
[86] G. Tintner, Stochastic linear programming with applications to agricultural economics, in: Symposium in Linear Programming, 1955, pp. 197–228.
[87] E.M.L. Beale, On minimizing a convex function subject to linear inequalities, J. R. Stat. Soc. Ser. B (Methodological) (1955) 173–184.
[88] N.V. Sahinidis, Optimization under uncertainty: State-of-the-art and opportunities, Comput. Chem. Eng. 28 (6–7) (2004) 971–983.
[89] G.B. Dantzig, Linear programming under uncertainty, Manag. Sci. 1 (3–4) (1955) 197–206.
[90] Z.B. Zabinsky, Stochastic adaptive search methods: Theory and implementation, in: Handbook of Simulation Optimization, Springer, 2015, pp. 293–318.
[91] M. Inuiguchi, H. Ichihashi, H. Tanaka, Fuzzy programming: A survey of recent developments, in: Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty, Springer, 1990, pp. 45–68.
[92] S.M. Ross, Introduction to Stochastic Dynamic Programming, Academic press, 2014.
[93] H.-G. Beyer, B. Sendhoff, Robust optimization–a comprehensive survey, Comput. Methods Appl. Mech. Eng. 196 (33–34) (2007) 3190–3218.
[94] A. Ben-Tal, L. El Ghaoui, A. Nemirovski, Robust Optimization, vol. 28, Princeton University Press, 2009.
[95] İhsan Yanıkoğlu, Bram L. Gorissen, Dick den Hertog, A survey of adjustable robust optimization, Eur. J. Oper. Res. 277 (3) (2019) 799–813.
[96] D. Bertsimas, V. Gupta, N. Kallus, Data-driven robust optimization, Math. Program. 167 (2) (2018) 235–292.
[97] G. Eichfelder, C. Krüger, A. Schöbel, Decision uncertainty in multiobjective optimization, J. Glob. Optim. 69 (2) (2017) 485–510.
[98] K. Kuhn, A. Raith, M. Schmidt, A. Schöbel, Bi-objective robust optimisation, Eur. J. Oper. Res. 252 (2) (2016) 418–431.
[99] R. Bokrantz, A. Fredriksson, Necessary and sufficient conditions for pareto efficiency in robust multiobjective optimization, Eur. J. Oper. Res. 262 (2) (2017) 682–692.
[100] Z. Zhao, Q. Quan, K.-Y. Cai, A health performance prediction method of large-scale stochastic linear hybrid systems with small failure probability, Reliab. Eng. Syst. Saf. 165 (2017) 74–88.
[101] F. Tricoire, A. Graf, W.J. Gutjahr, The bi-objective stochastic covering tour problem, Comput. Oper. Res. 39 (7) (2012) 1582–1592.
[102] J. Yazdi, E. Lee, J. Kim, Stochastic multiobjective optimization model for urban drainage network rehabilitation, J. Water Resour. Plan. Manag. 141 (8) (2014) 04014091.
[103] W.J. Gutjahr, A. Pichler, Stochastic multi-objective optimization: A survey on non-scalarizing methods, Ann. Oper. Res. 236 (2) (2016) 475–499.
[104] A.A. Malikopoulos, A multiobjective optimization framework for online stochastic optimal control in hybrid electric vehicles, IEEE Trans. Control Syst. Technol. 24 (2) (2016) 440–450.
[105] L. Janson, E. Schmerling, M. Pavone, Monte Carlo motion planning for robot trajectory optimization under uncertainty, in: Robotics Research, 2018, pp. 343–361.
[106] L. Baringo, A. Baringo, A stochastic adaptive robust optimization approach for the generation and transmission expansion planning, IEEE Trans. Power Syst. 33 (1) (2018) 792–802.
[107] X. Zhuang, Y. Li, S. Nie, Y. Fan, G. Huang, Analyzing climate change impacts on water resources under uncertainty using an integrated simulation-optimization approach, J. Hydrol. 556 (2018) 523–538.
[108] R.K. Ursem, Multinational gas: multimodal optimization techniques in dynamic environments, in: Annual ACM Conference on Genetic and Evolutionary Computation, 2000, pp. 19–26.
[109] F.O. de França, F.J. Von Zuben, L.N. de Castro, An artificial immune network for multimodal function optimization on dynamic environments, in: Annual ACM Conference on Genetic and Evolutionary Computation, 2005, pp. 289–296.
[110] Y. Fu, H. Wang, G. Tian, Z. Li, H. Hu, Two-agent stochastic flow shop deteriorating scheduling via a hybrid multi-objective evolutionary algorithm, J. Intell. Manuf. (2018) 1–16.
[111] J.T. Lin, C.-C. Chiu, A hybrid particle swarm optimization with local search for stochastic resource allocation problem, J. Intell. Manuf. 29 (3) (2018) 481–495.
[112] P. Rakshit, A. Konar, S. Das, Noisy evolutionary optimization algorithms–a comprehensive survey, Swarm Evolut. Comput. 33 (2017) 18–45.
[113] X. Yu, Y. Jin, K. Tang, X. Yao, Robust optimization over time—a new perspective on dynamic optimization problems, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–6.
[114] Y. Jin, K. Tang, X. Yu, B. Sendhoff, X. Yao, A framework for finding robust optimal solutions over time, Memet. Comput. 5 (1) (2013) 3–18.
[115] J. Grandell, Aspects of Risk Theory, Springer Science & Business Media, 2012.
[116] L. Tassiulas, A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, IEEE Trans. Autom. Control 37 (12) (1992) 1936–1948.
[117] D. Kalyanmoy, Multi Objective Optimization Using Evolutionary Algorithms, John Wiley and Sons, 2001.
[118] Y. Collette, P. Siarry, Multiobjective Optimization: Principles and Case Studies, Springer Science & Business Media, 2013.
[119] C.A. Coello Coello, G.B. Lamont, D.A. Van Veldhuizen, et al., Evolutionary Algorithms for Solving Multi-Objective Problems, vol. 5, Springer, 2007.
[120] J.K. Mandal, S. Mukhopadhyay, P. Dutta, Multi-Objective Optimization: Evolutionary to Hybrid Framework, Springer, 2018.
[121] S. Bechikh, R. Datta, A. Gupta, Recent Advances in Evolutionary Multi-Objective Optimization, vol. 20, Springer, 2016.

[122] M. Laszczyk, P.B. Myszkowski, Survey of quality measures for multi-objective optimization. construction of complementary set of multi-objective quality measures, Swarm Evolut. Comput. 48 (2019) 109–133.
[123] L.M. Antonio, J.A.M. Berenguer, C.A. Coello Coello, Evolutionary many-objective optimization based on linear assignment problem transformations, Soft Comput. 22 (16) (2018) 5491–5512.
[124] M. Pescador-Rojas, R.H. Gómez, E. Montero, N. Rojas-Morales, M.-C. Riff, C.A. Coello Coello, An overview of weighted and unconstrained scalarizing functions, in: International Conference on Evolutionary Multi-Criterion Optimization, 2017, pp. 499–513.
[125] D. Hadka, P. Reed, Borg: an auto-adaptive many-objective evolutionary computing framework, Evol. Comput. 21 (2) (2013) 231–259.
[126] E.K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, J.R. Woodward, A classification of hyper-heuristic approaches, in: Handbook of Metaheuristics, Springer, 2010, pp. 449–468.
[127] K. McClymont, E.C. Keedwell, Markov chain hyper-heuristic (MCHH): An online selective hyper-heuristic for multi-objective continuous problems, in: Annual ACM Conference on Genetic and Evolutionary Computation, 2011, pp. 2003–2010.
[128] M. Maashi, An Investigation of Multi-Objective Hyper-Heuristics for Multi-Objective Optimisation. (Ph.D. thesis), University of Nottingham, 2014.
[129] D.J. Walker, E. Keedwell, Towards many-objective optimisation with hyper-heuristics: identifying good heuristics with indicators, in: International Conference on Parallel Problem Solving from Nature, 2016, pp. 493–502.
[130] R.A. Gonçalves, J.N. Kuk, C.P. Almeida, S.M. Venske, MOEA/D-HH: a hyper-heuristic for multi-objective problems, in: International Conference on Evolutionary Multi-Criterion Optimization, 2015, pp. 94–108.
[131] R.H. Gómez, C.A. Coello Coello, A hyper-heuristic of scalarizing functions, in: Genetic and Evolutionary Computation Conference, 2017, pp. 577–584.
[132] J.G. Falcón-Cardona, C.A. Coello Coello, A multi-objective evolutionary hyper-heuristic based on multiple indicator-based density estimators, in: Genetic and Evolutionary Computation Conference, 2018, pp. 633–640.
[133] J.A. Vrugt, B.A. Robinson, Improved evolutionary optimization from genetically adaptive multimethod search, Natl. Acad. Sci. 104 (3) (2007) 708–711.
[134] C. Qian, K. Tang, Z.-H. Zhou, Selection hyper-heuristics can provably be helpful in evolutionary multi-objective optimization, in: International Conference on Parallel Problem Solving from Nature, 2016, pp. 835–846.
[135] M. Farina, P. Amato, A fuzzy definition of "optimality" for many-criteria optimization problems, IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 34 (3) (2004) 315–326.
[136] S. Mostaghim, H. Schmeck, Distance based ranking in many-objective particle swarm optimization, in: International Conference on Parallel Problem Solving from Nature, 2008, pp. 753–762.
[137] L. Martí, J. García, A. Berlanga, J.M. Molina, Introducing MONEDA: scalable multiobjective optimization with a neural estimation of distribution algorithm, in: Annual ACM Conference on Genetic and Evolutionary Computation, 2008, pp. 689–696.
[138] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, Eur. J. Oper. Res. 181 (3) (2007) 1653–1669.
[139] K. Praditwong, X. Yao, A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm, in: International Conference on Computational Intelligence and Security, vol. 1, 2006, pp. 286–291.
[140] H. Ishibuchi, N. Akedo, Y. Nojima, Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems, IEEE Trans. Evol. Comput. 19 (2) (2015) 264–283.
[141] O. Schutze, A. Lara, C.A. Coello Coello, On the influence of the number of objectives on the hardness of a multiobjective optimization problem, IEEE Trans. Evol. Comput. 15 (4) (2011) 444–455.
[142] J.J. Durillo, A.J. Nebro, C.A. Coello Coello, F. Luna, E. Alba, A comparative study of the effect of parameter scalability in multi-objective metaheuristics, in: IEEE Congress on Evolutionary Computation, 2008, pp. 1893–1900.
[143] L.M. Antonio, C.A. Coello Coello, Use of cooperative coevolution for solving large scale multiobjective optimization problems, in: IEEE Congress on Evolutionary Computation, 2013, pp. 2758–2765.
[144] W. Hong, K. Tang, A. Zhou, H. Ishibuchi, X. Yao, A scalable indicator-based evolutionary algorithm for large-scale multi-objective optimization, IEEE Trans. Evol. Comput. (2018) 1, https://doi.org/10.1109/TEVC.2018.2881153 (in press).
[145] H. Zille, S. Mostaghim, Comparison study of large-scale optimisation techniques on the LSMOP benchmark functions, in: IEEE Symposium Series on Computational Intelligence, 2017, pp. 1–8.
[146] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, A framework for large-scale multiobjective optimization based on problem transformation, IEEE Trans. Evol. Comput. 22 (2) (2018) 260–275.
[147] R. Cheng, Y. Jin, M. Olhofer, et al., Test problems for large-scale multiobjective and many-objective optimization, IEEE Trans. Cybern. 47 (12) (2017) 4108–4121.
[148] B. Cao, J. Zhao, Z. Lv, X. Liu, S. Yang, X. Kang, K. Kang, Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization, IEEE Access 5 (2017) 8214–8221.
[149] H. Li, K. Deb, Challenges for evolutionary multiobjective optimization algorithms for solving variable-length problems, in: IEEE Congress on Evolutionary Computation, 2017, pp. 2217–2224.
[150] A. González-Pardo, J.J. Jung, D. Camacho, ACO-based clustering for ego network analysis, Future Gener. Comput. Syst. 66 (2017) 160–170.
[151] G.B. Orgaz, S. Salcedo-Sanz, D. Camacho, A multi-objective genetic algorithm for overlapping community detection based on edge encoding, Inf. Sci. 462 (2018) 290–314.
[152] C.R. Atencia, J. Del Ser, D. Camacho, Weighted strategies to guide a multi-objective evolutionary algorithm for multi-UAV mission planning, Swarm Evolut. Comput. 44 (2019) 480–495.
[153] L.M. Antonio, C.A. Coello Coello, Coevolutionary multi-objective evolutionary algorithms: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 22 (6) (2018) 851–865.
[154] A. Lara, G. Sanchez, C.A. Coello Coello, O. Schutze, HCS: a new local search strategy for memetic multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 14 (1) (2010) 112–132.
[155] M. Abouhawwash, H. Seada, K. Deb, Towards faster convergence of evolutionary multi-criterion optimization algorithms using Karush-Kuhn-Tucker optimality based local search, Comput. Oper. Res. 79 (2017) 331–346.
[156] S. Das, S. Maity, B.-Y. Qu, P.N. Suganthan, Real-parameter evolutionary multimodal optimization – A survey of the state-of-the-art, Swarm Evolut. Comput. 1 (2) (2011) 71–88.
[157] B. Sareni, L. Krahenbuhl, Fitness sharing and niching methods revisited, IEEE Trans. Evol. Comput. 2 (3) (1998) 97–106.
[158] X. Li, M.G. Epitropakis, K. Deb, A. Engelbrecht, Seeking multiple solutions: An updated survey on niching methods and their applications, IEEE Trans. Evol. Comput. 21 (4) (2017) 518–538.
[159] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, 1987, pp. 41–49.
[160] A. Pétrowski, A clearing procedure as a niching method for genetic algorithms, in: IEEE International Conference on Evolutionary Computation, 1996, pp. 798–803.
[161] K.A. De Jong, Analysis of the Behavior of a Class of Genetic Adaptive Systems, University of Michigan, 1975 (Ph.D. thesis).
[162] S.W. Mahfoud, Niching Methods for Genetic Algorithms, University of Illinois, Urbana-Champaign, 1995 (Ph.D. thesis).
[163] J.-P. Li, M.E. Balazs, G.T. Parks, P.J. Clarkson, A species conserving genetic algorithm for multimodal function optimization, Evol. Comput. 10 (3) (2002) 207–234.
[164] M. Bessaou, A. Pétrowski, P. Siarry, Island model cooperating with speciation for multimodal optimization, in: International Conference on Parallel Problem Solving from Nature, 2000, pp. 437–446.
[165] P.N. Suganthan, Particle swarm optimiser with neighbourhood operator, in: IEEE Congress on Evolutionary Computation, vol. 3, 1999, pp. 1958–1962.
[166] B.-Y. Qu, P.N. Suganthan, J.-J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (5) (2012) 601–614.
[167] A. Basak, S. Das, K.C. Tan, Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection, IEEE Trans. Evol. Comput. 17 (5) (2013) 666–685.
[168] B.-Y. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, IEEE Trans. Evol. Comput. 17 (3) (2013) 387–402.
[169] S. Biswas, S. Kundu, S. Das, An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution, IEEE Trans. Cybern. 44 (10) (2014) 1726–1737.
[170] S. Biswas, S. Kundu, S. Das, et al., Inducing niching behavior in differential evolution through local information sharing, IEEE Trans. Evol. Comput. 19 (2) (2015) 246–263.
[171] K. Deb, A. Saha, Multimodal optimization using a bi-objective evolutionary algorithm, Evol. Comput. 20 (1) (2012) 27–62.
[172] Y. Wang, H.-X. Li, G.G. Yen, W. Song, MOMMOP: multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems, IEEE Trans. Cybern. 45 (4) (2015) 830–843.
[173] A.P. Engelbrecht, B. Masiye, G. Pampard, Niching ability of basic particle swarm optimization algorithms, in: IEEE Swarm Intelligence Symposium, 2005, pp. 397–400.
[174] J. Barrera, C.A. Coello Coello, A review of particle swarm optimization methods used for multimodal optimization, in: Innovations in Swarm Intelligence, 2009, pp. 9–37.
[175] Y. Liu, X. Ling, Z. Shi, M. Lv, J. Fang, L. Zhang, A survey on particle swarm optimization algorithms for multimodal function optimization, J. Softw. 6 (12) (2011) 2449–2455.
[176] T. Blackwell, J. Branke, Multiswarms, exclusion, and anti-convergence in dynamic environments, IEEE Trans. Evol. Comput. 10 (4) (2006) 459–472.
[177] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Trans. Evol. Comput. 14 (1) (2010) 150–169.
[178] X. Li, Efficient differential evolution using speciation for multimodal function optimization, in: Annual ACM Conference on Genetic and Evolutionary Computation, 2005, pp. 873–880.
[179] S. Kamyab, M. Eftekhari, Feature selection using multimodal optimization techniques, Neurocomputing 171 (2016) 586–597.
[180] M. Chica, J. Barranquero, T. Kajdanowicz, S. Damas, O. Cordón, Multimodal optimization: an effective framework for model calibration, Inf. Sci. 375 (2017) 79–97.
[181] M. Tomassini, Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time, Springer, 2006.

[182] J.L. Payne, M. Giacobini, J.H. Moore, Complex and dynamic population structures: synthesis, open questions, and future directions, Soft Comput. 17 (7) (2013) 1109–1120.
[183] N. Lynn, M.Z. Ali, P.N. Suganthan, Population topologies for particle swarm optimization and differential evolution, Swarm Evolut. Comput. 39 (2018) 24–35.
[184] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evolut. Comput. 24 (2015) 11–24.
[185] Q. Ni, J. Deng, A new logistic dynamic particle swarm optimization algorithm based on random topology, Sci. World J. (2013) 8 Article ID 409167.
[186] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P.K. Tucker, Surrogate-based analysis and optimization, Prog. Aero. Sci. 41 (1) (2005) 1–28.
[187] A.I. Forrester, A.J. Keane, Recent advances in surrogate-based optimization, Prog. Aero. Sci. 45 (1–3) (2009) 50–79.
[188] E. Andrés-Pérez, D. González-Juarez, M. Martin, E. Iuliano, D. Cinquegrana, G. Carrier, J. Peter, D. Bailly, O. Amoignon, P. Dvorak, et al., Garteur ad/ag-52: surrogate-based global optimization methods in preliminary aerodynamic design, in: Evolutionary and Deterministic Methods for Design Optimization and Control with Applications to Industrial and Societal Problems, Springer, 2019, pp. 195–210.
[189] Z.-H. Han, J. Chen, K.-S. Zhang, Z.-M. Xu, Z. Zhu, W.-P. Song, Aerodynamic shape optimization of natural-laminar-flow wing using surrogate-based approach, AIAA J. (2018) 1–15.
[190] E. Andrés-Pérez, D. González-Juárez, M.J. Martin-Burgos, L. Carro-Calvo, S. Salcedo-Sanz, Obtaining minimum-drag shapes through surrogate-based global optimization: an application to the aerodynamic shape design of the landing gear master cylinder, Proc. IME Part G J. Aero. Eng. 232 (15) (2018) 2858–2869.
[191] E. Andres, E. Iuliano, Application of Surrogate-Based Global Optimization to Aerodynamic Design, Springer, 2016.
[192] S. Koziel, S. Ogurtsov, W. Zieniutycz, A. Bekasiewicz, Design of a planar UWB dipole antenna with an integrated balun using surrogate-based optimization, IEEE Antennas Wirel. Propag. Lett. 14 (2015) 366–369.
[193] S. Koziel, A. Pietrenko-Dabrowska, Performance-based nested surrogate modeling of antenna input characteristics, IEEE Trans. Antennas Propag. (2019) 1 (in press).
[194] J.A. Easum, J. Nagar, P.L. Werner, D.H. Werner, Efficient multiobjective antenna optimization with tolerance analysis through the use of surrogate models, IEEE Trans. Antennas Propag. 66 (12) (2018) 6706–6715.
[195] A. Bhosekar, M. Ierapetritou, Advances in surrogate based modeling, feasibility analysis, and optimization: a review, Comput. Chem. Eng. 108 (2018) 250–267.
[196] P. Prebeg, V. Zanic, B. Vazic, Application of a surrogate modeling to the ship structural design, Ocean. Eng. 84 (2014) 259–272.
[197] Z. Wang, M.S. Escotet-Espinoza, R. Singh, M. Ierapetritou, Surrogate-based optimization for pharmaceutical manufacturing processes, in: Computer Aided Chemical Engineering, vol. 40, Elsevier, 2017, pp. 2797–2802.
[198] J. Müller, R. Paudel, C. Shoemaker, J. Woodbury, Y. Wang, N. Mahowald, Ch 4 parameter estimation in clm4. 5bgc using surrogate global optimization, Geosci. Model Dev. 8 (10) (2015) 3285–3310.
[199] V. Christelis, A. Mantoglou, Pumping optimization of coastal aquifers assisted by adaptive metamodelling methods and radial basis functions, Water Resour. Manag. 30 (15) (2016) 5845–5859.
[200] C. Sun, Y. Jin, R. Cheng, J. Ding, J. Zeng, Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, IEEE Trans. Evol. Comput. 21 (4) (2017) 644–660.
[201] M. Amine Bouhlel, N. Bartoli, R.G. Regis, A. Otsmane, J. Morlier, Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method, Eng. Optim. 50 (12) (2018) 2038–2053.
[202] M.A. Bouhlel, N. Bartoli, A. Otsmane, J. Morlier, Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction, Struct. Multidiscip. Optim. 53 (5) (2016) 935–952.
[203] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: representing model uncertainty in deep learning, in: International Conference on Machine Learning, 2016, pp. 1050–1059.
[204] H. Wang, D.-Y. Yeung, Towards bayesian deep learning: a framework and some existing methods, IEEE Trans. Knowl. Data Eng. 28 (12) (2016) 3395–3408.
[205] H. Yu, Y. Tan, J. Zeng, C. Sun, Y. Jin, Surrogate-assisted hierarchical particle swarm optimization, Inf. Sci. 454 (2018) 59–72.
[206] J. Blanchard, C. Beauthier, T. Carletti, A surrogate-assisted cooperative co-evolutionary algorithm for solving high dimensional, expensive and black box optimization problems, in: International Conference on Engineering Optimization, Springer, 2018, pp. 41–52.
[207] E. Alba, M. Tomassini, Parallelism and evolutionary algorithms, IEEE Trans. Evol. Comput. 6 (5) (2002) 443–462.
[208] E. Alba, Parallel Metaheuristics: A New Class of Algorithms, vol. 47, John Wiley & Sons, 2005.
[209] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, J.-J. Li, Distributed evolutionary algorithms and their models: a survey of the state-of-the-art, Appl. Soft Comput. 34 (2015) 286–300.
[210] A. Mendiburu, J.A. Lozano, J. Miguel-Alonso, Parallel implementation of EDAs based on probabilistic graphical models, IEEE Trans. Evol. Comput. 9 (4) (2005) 406–423.
[211] D. Whitley, T. Starkweather, Genitor II: a distributed genetic algorithm, J. Exp. Theor. Artif. Intell. 2 (3) (1990) 189–214.
[212] E. Alba, B. Dorronsoro, Cellular Genetic Algorithms, vol. 42, Springer Science & Business Media, 2009.
[213] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, B.-S. Lee, Efficient hierarchical parallel genetic algorithms using grid computing, Future Gener. Comput. Syst. 23 (4) (2007) 658–670.
[214] G. Roy, H. Lee, J.L. Welch, Y. Zhao, V. Pandey, D. Thurston, A distributed pool architecture for genetic algorithms, in: IEEE Congress on Evolutionary Computation, 2009, pp. 1177–1184.
[215] M.A. Potter, K.A. De Jong, Cooperative coevolution: an architecture for evolving coadapted subcomponents, Evol. Comput. 8 (1) (2000) 1–29.
[216] R. Subbu, A.C. Sanderson, Modeling and convergence analysis of distributed coevolutionary algorithms, IEEE Trans. Syst., Man, Cybern., Part B (Cybernetics) 34 (2) (2004) 806–822.
[217] R. Subbu, A.C. Sanderson, Network-based distributed planning using coevolutionary agents: architecture and evaluation, IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 34 (2) (2004) 257–269.
[218] F. Seredynski, Competitive coevolutionary multi-agent systems: the application to mapping and scheduling problems, J. Parallel Distrib. Comput. 47 (1) (1997) 39–57.
[219] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, Biological invasion–inspired migration in distributed evolutionary algorithms, Inf. Sci. 207 (2012) 50–65.
[220] S. Ramírez-Gallego, S. García, J. Benítez, F. Herrera, A distributed evolutionary multivariate discretizer for big data processing on Apache spark, Swarm Evolut. Comput. 38 (2018) 240–250.
[221] S. Salcedo-Sanz, J. Del Ser, Z. Geem, An island grouping genetic algorithm for fuzzy partitioning problems, Sci. World J. (2014) 15 Article ID 916371.
[222] M.C. Naldi, R.J. Campello, Comparison of distributed evolutionary k-means clustering algorithms, Neurocomputing 163 (2015) 78–93.
[223] B. Cao, J. Zhao, Z. Lv, X. Liu, A distributed parallel cooperative coevolutionary multiobjective evolutionary algorithm for large-scale optimization, IEEE Trans. Ind. Inf. 13 (4) (2017) 2030–2038.
[224] X.-M. Hu, F.-L. He, W.-N. Chen, J. Zhang, Cooperation coevolution with fast interdependency identification for large scale optimization, Inf. Sci. 381 (2017) 142–160.
[225] X. Peng, Y. Wu, Large-scale cooperative co-evolution using niching-based multi-modal optimization and adaptive fast clustering, Swarm Evolut. Comput. 35 (2017) 65–77.
[226] J. Starzynski, R. Szmurlo, J. Kijanowski, B. Dawidowicz, B. Sawicki, S. Wincenciak, Distributed evolutionary algorithm for optimization in electromagnetics, IEEE Trans. Magn. 42 (4) (2006) 1243–1246.
[227] J. Leskinen, J. Périaux, Distributed evolutionary optimization using nash games and GPUs – applications to CFD design problems, Comput. Fluids 80 (2013) 190–201.
[228] S. Bucking, V. Dermardiros, Distributed evolutionary algorithm for co-optimization of building and district systems for early community energy masterplanning, Appl. Soft Comput. 63 (2018) 14–22.
[229] M.G. Epitropakis, V.P. Plagianakos, M.N. Vrahatis, Hardware-friendly higher-order neural network training using distributed evolutionary algorithms, Appl. Soft Comput. 10 (2) (2010) 398–408.
[230] E.-G. Talbi, S. Mostaghim, T. Okabe, H. Ishibuchi, G. Rudolph, C.A. Coello Coello, Parallel approaches for multiobjective optimization, in: Multiobjective Optimization, 2008, pp. 349–372.
[231] S. Wessing, G. Rudolph, D.A. Menges, Comparing asynchronous and synchronous parallelization of the SMS-EMOA, in: International Conference on Parallel Problem Solving from Nature, 2016, pp. 558–567.
[232] W. Zhu, A. Yaseen, Y. Li, DEMCMC-GPU: An efficient multi-objective optimization method with gpu acceleration on the fermi architecture, New Generat. Comput. 29 (2) (2011) 163–184.
[233] A.L. Jaimes, C.A. Coello Coello, MRMOGA: A new parallel multi-objective evolutionary algorithm based on the use of multiple resolutions, Concurrency Comput. Pract. Ex. 19 (4) (2007) 397–441.
[234] R.H. Gómez, C.A. Coello Coello, E. Alba, A parallel version of SMS-EMOA for many-objective optimization problems, in: International Conference on Parallel Problem Solving from Nature, 2016, pp. 568–577.
[235] S. Salcedo-Sanz, A survey of repair methods used as constraint handling techniques in evolutionary algorithms, Comput. Sci. Rev. 3 (3) (2009) 175–192.
[236] F. Rothlauf, Representations for Genetic and Evolutionary Algorithms, Springer Science & Business Media, 2006.
[237] S. Salcedo-Sanz, J. Muñoz-Bulnes, M.J. Vermeij, New coral reefs-based approaches for the model type selection problem: A novel method to predict a nation's future energy demand, Int. J. Bio-Inspired Comput. 10 (3) (2017) 145–158.
[238] S. Salcedo-Sanz, C. Camacho-Gómez, D. Molina, F. Herrera, A coral reefs optimization algorithm with substrate layers and local search for large scale global optimization, in: IEEE Congress on Evolutionary Computation, 2016, pp. 3574–3581.
[239] G. Wu, R. Mallipeddi, P.N. Suganthan, Ensemble strategies for population-based optimization algorithms–a survey, Swarm Evolut. Comput. 44 (2019) 695–711.
[240] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55 (C) (2017) 533–548.
[241] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, P.N. Suganthan, Ensemble of differential evolution variants, Inf. Sci. 423 (C) (2018) 172–186.
[242] J.A. Vrugt, B.A. Robinson, J.M. Hyman, Self-adaptive multimethod search for global optimization in real-parameter spaces, IEEE Trans. Evol. Comput. 13 (2) (2009) 243–259.

[243] E. Bermejo, M. Chica, S. Damas, S. Salcedo-Sanz, O. Cordón, Coral reef optimization with substrate layers for medical image registration, Swarm Evolut. Comput. 42 (2018) 138–159.
[244] J.A. Vrugt, B.A. Robinson, Improved evolutionary optimization from genetically adaptive multimethod search, Proc. Natl. Acad. Sci. Unit. States Am. 104 (3) (2007) 708–711.
[245] W.K. Mashwani, A. Salhi, Multiobjective evolutionary algorithm based on multimethod with dynamic resources allocation, Appl. Soft Comput. 39 (2016) 292–309.
[246] Y. Xue, S. Zhong, Y. Zhuang, B. Xu, An ensemble algorithm with self-adaptive learning techniques for high-dimensional numerical optimization, Appl. Math. Comput. 231 (2014) 329–346.
[247] S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes, IEEE Trans. Evol. Comput. 16 (3) (2012) 442–446.
[248] R. Mallipeddi, P.N. Suganthan, Ensemble of constraint handling techniques, IEEE Trans. Evol. Comput. 14 (4) (2010) 561–579.
[249] E.L. Yu, P.N. Suganthan, Ensemble of niching algorithms, Inf. Sci. 180 (15) (2010) 2815–2833.
[250] F. Peng, K. Tang, G. Chen, X. Yao, Population-based algorithm portfolios for numerical optimization, IEEE Trans. Evol. Comput. 14 (5) (2010) 782–800.
[251] J. Grobler, A.P. Engelbrecht, G. Kendall, V. Yadavalli, Multi-method algorithms: investigating the entity-to-algorithm allocation problem, in: IEEE Congress on Evolutionary Computation, 2013, pp. 570–577.
[252] W. Gong, Á. Fialho, Z. Cai, H. Li, Adaptive strategy selection in differential evolution for numerical optimization: an empirical study, Inf. Sci. 181 (24) (2011) 5364–5386.
[253] R. Mallipeddi, P.N. Suganthan, Q.-K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2) (2011) 1679–1696.
[254] W. Du, B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, Inf. Sci. 178 (15) (2008) 3096–3109.
[255] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, J.-S. Pan, Multi-strategy ensemble artificial bee colony algorithm, Inf. Sci. 279 (2014) 587–603.
[256] G. Xiong, D. Shi, X. Duan, Multi-strategy ensemble biogeography-based optimization for economic dispatch problems, Appl. Energy 111 (2013) 801–811.
[257] E.K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, R. Qu, Hyper-heuristics: a survey of the state of the art, J. Oper. Res. Soc. 64 (12) (2013) 1695–1724.
[258] S. Remde, P. Cowling, K. Dahal, N. Colledge, E. Selensky, An empirical study of hyperheuristics for managing very large sets of low level heuristics, J. Oper. Res. Soc. 63 (3) (2012) 392–405.
[259] E. Özcan, B. Bilgin, E.E. Korkmaz, A comprehensive analysis of hyper-heuristics, Intell. Data Anal. 12 (1) (2008) 3–23.
[260] P. Ross, Hyper-heuristics, in: Search Methodologies, Springer, 2005, pp. 529–556.
[261] P. Moscato, On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: towards Memetic Algorithms, Tech. Rep. 826, California Institute of Technology, 1989.
[262] P. Moscato, M. Norman, A Competitive and Cooperative Approach to Complex Combinatorial Search, Tech. Rep. 790, Caltech Concurrent Computation Program, 1989.
[263] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (1) (2003) 1–18.
[264] Q.H. Nguyen, Y.S. Ong, M.H. Lim, Non-genetic transmission of memes by diffusion, in: Annual ACM Conference on Genetic and Evolutionary Computation, 2008, pp. 1017–1024.
[265] R. Meuth, M.-H. Lim, Y.-S. Ong, D.C. Wunsch, A proposition on memes and meta-memes in computing for higher-order learning, Memet. Comput. 1 (2) (2009) 85–100.
[266] N. Krasnogor, Studies on the Theory and Design Space of Memetic Algorithms. (Ph.D. thesis), University of the West of England at Bristol, 2002.
[267] P. Cowling, G. Kendall, E. Soubeiga, A hyperheuristic approach to scheduling a sales summit, in: International Conference on the Practice and Theory of Automated Timetabling, 2000, pp. 176–190.
[268] Y.-S. Ong, M.-H. Lim, N. Zhu, K.-W. Wong, Classification of adaptive memetic algorithms: A comparative study, IEEE Trans. Syst., Man, Cybern. – Part B (Cybernetics) 36 (1) (2006) 141–152.
[269] N. Krasnogor, S. Gustafson, A study on the use of “self-generation” in memetic algorithms, Nat. Comput. 3 (1) (2004) 53–76.
[270] J.E. Smith, Coevolving memetic algorithms: a review and progress report, IEEE Trans. Syst., Man, Cybern. – Part B (Cybernetics) 37 (1) (2007) 6–17.
[271] M. Zhang, H. Wang, Z. Cui, J. Chen, Hybrid multi-objective cuckoo search with dynamical local search, Memet. Comput. 10 (2) (2018) 199–208.
[272] H. Duan, D. Zhang, Y. Shi, Y. Deng, Close formation flight of swarm unmanned aerial vehicles via metric-distance brain storm optimization, Memet. Comput. 10 (4) (2018) 369–381.
[273] R. Shang, B. Du, K. Dai, L. Jiao, Y. Xue, Memetic algorithm based on extension step and statistical filtering for large-scale capacitated arc routing problems, Nat. Comput. 17 (2) (2018) 375–391.
[274] X.-N. Shen, L.L. Minku, N. Marturi, Y.-N. Guo, Y. Han, A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling, Inf. Sci. 428 (2018) 1–29.
[275] A. Martinez, E. Osaba, M.N. Bilbao, J. Del Ser, Let nature decide its nature: on the design of collaborative hyperheuristics for decentralized ephemeral environments, Future Gener. Comput. Syst. 88 (2018) 792–805.
[276] S. Chen, J. Montgomery, A.B. Röhler, Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution, Appl. Intell. 42 (3) (2015) 514–526.
[277] X. Li, K. Tang, M.N. Omidvar, Z. Yang, K. Quin, Benchmark Functions for the CEC’2013 Special Session and Competition on Large Scale Global Optimization, Tech. rep., RMIT University, 2013.
[278] M.N. Omidvar, X. Li, K. Tang, Designing benchmark problems for large-scale continuous optimization, Inf. Sci. 316 (2015) 419–436.
[279] K. Tang, X. Yao, P.N. Suganthan, C. MacNish, Y.P. Chen, C.M. Chen, Z. Yang, Benchmark Functions for the CEC’2008 Special Session and Competition on Large Scale Global Optimization, Tech. rep., Nature Inspired Computation and Applications Laboratory, 2007.
[280] K. Tang, Summary of Results on CEC’08 Competition on Large Scale Global Optimization, Tech. rep., Nature Inspired Computation and Applications Laboratory, 2008.
[281] K. Tang, X. Li, P.N. Suganthan, Z. Yang, T. Weise, Benchmark Functions for the CEC’2010 Special Session and Competition on Large Scale Global Optimization, Tech. rep., Nature Inspired Computation and Applications Laboratory, 2010.
[282] M. Lozano, D. Molina, F. Herrera, Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems, Soft Comput. - A Fusion Found. Methodol. Appl. 15 (2011) 2085–2087.
[283] W. Chen, K. Tang, Impact of problem decomposition on cooperative coevolution, in: IEEE Congress on Evolutionary Computation, 2013, pp. 733–740.
[284] M.N. Omidvar, M. Yang, Y. Mei, X. Li, X. Yao, DG2: a faster and more accurate differential grouping for large-scale black-box optimization, IEEE Trans. Evol. Comput. 21 (6) (2017) 929–942.
[285] Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution, Inf. Sci. 178 (15) (2008) 2985–2999.
[286] Z. Yang, K. Tang, X. Yao, Multilevel cooperative coevolution for large scale optimization, in: IEEE Congress on Evolutionary Computation, 2008, pp. 1663–1670.
[287] H. Liu, Y. Wang, X. Liu, S. Guan, Empirical study of effect of grouping strategies for large scale optimization, in: International Joint Conference on Neural Networks, 2016, pp. 3433–3439.
[288] M.N. Omidvar, X. Li, X. Yao, Cooperative co-evolution with delta grouping for large scale non-separable function optimization, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.
[289] S. Mahdavi, M.E. Shiri, S. Rahnamayan, Metaheuristics in large-scale global continues optimization: a survey, Inf. Sci. 295 (2015) 407–428.
[290] Y. Sun, M. Kirley, S.K. Halgamuge, A recursive decomposition method for large scale continuous optimization, IEEE Trans. Evol. Comput. 22 (5) (2018) 647–661.
[291] A. LaTorre, S. Muelas, J.M. Peña, A comprehensive comparison of large scale global optimizers, Inf. Sci. 316 (2014) 517–549.
[292] A. LaTorre, S. Muelas, J.M. Peña, Large scale global optimization: experimental results with mos-based hybrid algorithms, in: IEEE Congress on Evolutionary Computation, 2013, pp. 2742–2749.
[293] A. Molina, D. LaTorre, F. Herrera, SHADE with iterative local search for large-scale global optimization, in: IEEE Congress on Evolutionary Computation, 2018, pp. 1252–1259.
[294] D. Molina, A. La Torre, CEC’2018 Special Session and Competition in Large-Scale Global Optimization, 2018. Available at: http://www.tflsgo.org/special_sessions/cec2018.html#benchmark-competition.
[295] M.S. Maučec, J. Brest, B. Bošković, Z. Kačič, Improved differential evolution for large-scale black-box optimization, IEEE Access 6 (2018) 29516–29531.
[296] S.K. Smit, A.E. Eiben, Comparing parameter tuning methods for evolutionary algorithms, in: IEEE Congress on Evolutionary Computation, 2009, pp. 399–406.
[297] S. Smit, A. Eiben, Beating the ‘world champion’ evolutionary algorithm via REVAC tuning, in: IEEE Congress on Evolutionary Computation (CEC), IEEE, 2010, pp. 1–8.
[298] T. Liao, D. Molina, T. Stützle, Performance evaluation of automatically tuned continuous optimizers on different benchmark sets, Appl. Soft Comput. 27 (2015) 490–503.
[299] V. Nannen, A.E. Eiben, Efficient relevance estimation and value calibration of evolutionary algorithm parameters, in: IEEE Congress on Evolutionary Computation, 2007, pp. 103–110.
[300] M. López-Ibáñez, J. Dubois-Lacoste, L.P. Cáceres, M. Birattari, T. Stützle, The I-RACE package: iterated racing for automatic algorithm configuration, Oper. Res. Perspect. 3 (2016) 43–58.
[301] A. Arcuri, G. Fraser, Parameter tuning or default values? An empirical investigation in search-based software engineering, Empir. Softw Eng. 18 (3) (2013) 594–623.
[302] A. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, IEEE Trans. Evol. Comput. 3 (1999) 124–141.
[303] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution – An updated survey, Swarm Evolut. Comput. 27 (2016) 1–30.
[304] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014, 2014, pp. 1658–1665.
[305] J. Brest, M. Sepesy Maučec, Population size reduction for the differential evolution algorithm, Appl. Intell. 29 (3) (2008) 228–247.
[306] A. Eiben, T. Baeck, N. van der Vaart, An empirical study on gas “without parameters”, in: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature, PPSN VI, Springer, 2000, pp. 315–324.

[307] N. Hansen, Benchmarking a bi-population CMA-ES on the BBOB-2009 function testbed, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, GECCO '09, ACM, New York, NY, USA, 2009, pp. 2389–2396.
[308] A. Piotrowski, Review of differential evolution population size, Swarm Evolut. Comput. 32 (2017) 1–24.
[309] R. Polakova, P. Bujok, Adaptation of population size in differential evolution algorithm: an experimental comparison, in: IEEE International Conference on Systems, Signals and Image Processing (IWSSIP), 2018, pp. 1–5.
[310] M. Srinivas, L.M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Trans. Syst., Man, Cybern. 24 (4) (1994) 656–667.
[311] F. Vafaee, P.C. Nelson, An explorative and exploitative mutation scheme, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.
[312] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: 2005 IEEE Congress on Evolutionary Computation, vol. 2, 2005, pp. 1785–1791.
[313] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[314] R.S. Sutton, A.G. Barto, Introduction to Reinforcement Learning, first ed., MIT Press, Cambridge, MA, USA, 1998.
[315] Y. Ong, A.J. Keane, Meta-lamarckian learning in memetic algorithms, IEEE Trans. Evol. Comput. 8 (2) (2004) 99–110.
[316] G. Karafotias, M. Hoogendoorn, A.E. Eiben, Parameter control in evolutionary algorithms: trends and challenges, IEEE Trans. Evol. Comput. 19 (2) (2015) 167–187.
[317] D. Molina, A. LaTorre, F. Herrera, An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions, Cognit. Comput. 10 (4) (2018) 517–544.
[318] C. García-Martínez, P.D. Gutiérrez, D. Molina, M. Lozano, F. Herrera, Since CEC 2005 competition on real-parameter optimisation: A decade of research, progress and comparative analysis's weakness, Soft Comput. 21 (19) (2017) 5573–5583.
[319] A.K. Kar, Bio inspired computing – a review of algorithms and scope of applications, Expert Syst. Appl. 59 (2016) 20–32.
[320] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Kangal report 2005005, Nanyang Technological University Singapore, 2005.
[321] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria for CEC'2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Tech. rep., Jadavpur University, 2010.
[322] J. Liang, B.-Y. Qu, P.N. Suganthan, A. Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC'2013 Special Session and Competition on Real-Parameter Optimization, Tech. Rep., Zhengzhou University and Nanyang Technological University, 2013.
[323] J. Liang, B.-Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Tech. Rep., Zhengzhou University and Nanyang Technological University, 2013.
[324] B. Liu, Q. Chen, Q. Zhang, J.J. Liang, P.N. Suganthan, B.Y. Qu, Problem Definitions and Evaluation Criteria for Computationally Expensive Single Objective Numerical Optimization, Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, 2013.
[325] J. Liang, Q. Chen, B. Qu, B. Liu, P.N. Suganthan, Q. Chen, Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-Based Real-Parameter Single Objective Optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Nov 2014.
[326] N. Awad, M. Ali, P.N. Suganthan, J. Liang, B. Qu, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Tech. Rep., Nanyang Technological University, Jordan University of Science and Technology, and Zhengzhou University, 2016.
[327] N. Hansen, A. Auger, O. Mersmann, T. Tusar, D. Brockhoff, COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting, 2016, arxiv:1603.08785.
[328] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, Tech. Rep., 2008.
[329] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[330] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: IEEE Congress on Evolutionary Computation, vol. 1, 2002, pp. 825–830.
[331] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, in: International Conference on Evolutionary Multi-Criterion Optimization, 2005, pp. 280–295.
[332] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302.
[333] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evolut. Comput. 1 (1) (2011) 32–49.
[334] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, X. Yao, Benchmark Functions for the CEC'2017 Competition on Many-Objective Optimization, Technical Report No. CSR-17-01, School of Computer Science, University of Birmingham, 2017.
[335] R. Cheng, M. Li, Y. Tian, X. Xiang, X. Zhang, S. Yang, Y. Jin, X. Yao, Benchmark Functions for the CEC'2018 Competition on Many-Objective Optimization, Tech. Rep., 2018.
[336] S. Mane, M. Narasinga Rao, Many-objective optimization: problems and evolutionary algorithms – a short review, Int. J. Appl. Eng. Res. 12 (20) (2017) 9774–9793.
[337] H. Li, K. Deb, Q. Zhang, P.N. Suganthan, L. Chen, Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties, Swarm Evolut. Comput. 46 (2019) 104–117.
[338] I. Moser, R. Chiong, Dynamic function optimization: the moving peaks benchmark, Stud. Comput. Intell. 433 (2013) 35–59.
[339] S. Jiang, S. Yang, Evolutionary dynamic multiobjective optimization: benchmarks and algorithm comparisons, IEEE Trans. Cybern. 47 (1) (2017) 198–211.
[340] X. Li, A. Engelbrecht, M.G. Epitropakis, Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization, Tech. Rep., RMIT University, 2013.
[341] B.Y. Qu, J.J. Liang, Z.Y. Wang, Q. Chen, P.N. Suganthan, Novel benchmark functions for continuous multimodal optimization with comparative results, Swarm Evolut. Comput. 26 (2016) 23–34.
[342] C. Yue, B. Qu, K. Yu, J. Liang, X. Li, A novel scalable test problem suite for multimodal multiobjective optimization, Swarm Evolut. Comput. 48 (2019) 62–71.
[343] M. Hellwig, H.-G. Beyer, Benchmarking evolutionary algorithms for single objective real-valued constrained optimization – a critical review, Swarm Evolut. Comput. 44 (2019) 927–944.
[344] S. Garcia, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 15 (6) (2009) 617–644.
[345] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evolut. Comput. 1 (1) (2011) 3–18.
[346] S. Das, Evaluating the evolutionary algorithms - classical perspectives and recent trends, in: Encyclopedia of Life Support Systems (EOLSS), 2013.
[347] J. Carrasco, S. García, M. del Mar Rueda, F. Herrera, rNPBST: An R package covering non-parametric and Bayesian statistical tests, in: Hybrid Artificial Intelligent Systems, Springer International Publishing, 2017, pp. 281–292.
[348] P. Pošík, W. Huyer, L. Pál, A comparison of global search algorithms for continuous black box optimization, Evol. Comput. 20 (4) (2012) 509–541.
[349] Y.D. Sergeyev, D. Kvasov, M. Mukhametzhanov, On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget, Sci. Rep. 8 (1) (2018) 453.
[350] A.P. Piotrowski, M.J. Napiorkowski, J.J. Napiorkowski, P.M. Rowinski, Swarm intelligence and evolutionary algorithms: performance versus speed, Inf. Sci. 384 (2017) 34–85.
[351] A.P. Piotrowski, Regarding the rankings of optimization heuristics based on artificially-constructed benchmark functions, Inf. Sci. 297 (2015) 191–201.
[352] A. Hussain, E. Cambria, Semi-supervised learning for big social data analysis, Neurocomputing 275 (2018) 1662–1673.
[353] K. Kim, R.I. McKay, B.-R. Moon, Multiobjective evolutionary algorithms for dynamic social network clustering, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM, 2010, pp. 1179–1186.
[354] M. Mazzara, L. Biselli, P.P. Greco, N. Dragoni, A. Marraffa, N. Qamar, S. de Nicola, Social networks and collective intelligence: A return to the agora, in: Social Network Engineering for Secure Web Data and Services, IGI global, 2013, pp. 88–113.
[355] C. Gao, X. Lan, X. Zhang, Y. Deng, A bio-inspired methodology of identifying influential nodes in complex networks, PLoS One 8 (6) (2013) e66732.
[356] T. Komatsu, A. Namatame, Dynamic diffusion in evolutionary optimised networks, Int. J. Bio-Inspired Comput. 3 (6) (2011) 384–392.
[357] A. Gonzalez-Pardo, J.J. Jung, D. Camacho, ACO-based clustering for ego network analysis, Future Gener. Comput. Syst. 66 (2017) 160–170.
[358] D. Greene, D. Doyle, P. Cunningham, Tracking the evolution of communities in dynamic social networks, in: 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2010, pp. 176–183.
[359] G. Bello-Orgaz, J. Hernandez-Castro, D. Camacho, Detecting discussion communities on vaccination in Twitter, Future Gener. Comput. Syst. 66 (2017) 125–136.
[360] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, P. Lió, K.-W. Lee, Bio-inspired multi-agent data harvesting in a proactive urban monitoring environment, Ad Hoc Netw. 7 (4) (2009) 725–741.
[361] L. Cuadra, S. Salcedo-Sanz, J. Del Ser, S. Jiménez-Fernández, Z. Geem, A critical review of robustness in power grids using complex networks concepts, Energies 8 (9) (2015) 9211–9265.
[362] C. Pizzuti, Evolutionary computation for community detection in networks: A review, IEEE Trans. Evol. Comput. 22 (3) (2018) 464–483.
[363] T. Koza, N. Karaboga, S. Koçkanat, Aort valve Doppler signal noise elimination using IIR filter designed with ABC algorithm, in: International Symposium on Innovations in Intelligent Systems and Applications (INISTA), IEEE, 2012, pp. 1–5.
[364] R. Sivakumar, M. Karnan, Diagnose breast cancer through mammograms using EABCO algorithm, Int. J. Eng. Technol. 4 (2012) 302–307.
[365] M. Karnan, T. Logheshwari, Improved implementation of brain MRI image segmentation using ant colony system, in: IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), IEEE, 2010, pp. 1–4.

[366] T. Logeswari, M. Karnan, An improved implementation of brain tumor detection using segmentation based on hierarchical self organizing map, Int. J. Comput. Theor. Eng. 2 (4) (2010) 591.

[367] V. Soleimani, F.H. Vincheh, Improving ant colony optimization for brain MRI image segmentation and brain tumor diagnosis, in: Conference on Pattern Recognition and Image Analysis (PRIA), IEEE, 2013, pp. 1–6.

[368] K. Thangavel, C. Velayutham, Mammogram image analysis: bio-inspired computational approach, in: International Conference on Soft Computing for Problem Solving (SocProS), Springer, 2012, pp. 941–955.

[369] S.H. Zainud-Deen, W.M. Hassen, E. Ali, K.H. Awadalla, H. Sharshar, Breast cancer detection using a hybrid finite difference frequency domain and particle swarm optimization techniques, in: Radio Science Conference, IEEE, 2008, pp. 1–8.

[370] A.Q. Al-Faris, U.K. Ngah, N.A.M. Isa, I.L. Shuaib, Breast MRI tumour segmentation using modified automatic seeded region growing based on particle swarm optimization image clustering, in: Soft Computing in Industrial Applications, Springer, 2014, pp. 49–60.

[371] M. Suganthi, M. Madheswaran, An improved medical decision support system to identify the breast cancer using mammogram, J. Med. Syst. 36 (1) (2012) 79–91.

[372] M.R. Senapati, P.K. Dash, Intelligent system based on local linear wavelet neural network and recursive least square approach for breast cancer classification, Artif. Intell. Rev. 39 (2) (2013) 151–163.

[373] A. Gálvez, I. Fister, E. Osaba, J. Del Ser, A. Iglesias, Automatic fitting of feature points for border detection of skin lesions in medical images with bat algorithm, in: International Symposium on Intelligent and Distributed Computing, 2018, pp. 357–368.

[374] S.L. Smith, S. Cagnoni, Genetic and Evolutionary Computation: Medical Applications, John Wiley & Sons, 2011.

[375] A. Ghaheri, S. Shoar, M. Naderan, S.S. Hoseini, The applications of genetic algorithms in medicine, Oman Med. J. 30 (6) (2015) 406.

[376] C.-W. Tsai, M.-C. Chiang, A. Ksentini, M. Chen, Metaheuristic algorithms for healthcare: open issues and challenges, Comput. Electr. Eng. 53 (2016) 421–434.

[377] M. Choraś, R. Kozik, Machine learning techniques for threat modeling and detection, in: Security and Resilience in Intelligent Data-Centric Systems and Communication Networks, Elsevier, 2018, pp. 179–192.

[378] S. Dilek, H. Çakir, M. Aydin, Applications of artificial intelligence techniques to combating cyber crimes: a review, Int. J. Artif. Intell. Appl. 6 (1) (2015) 21.

[379] X.-S. Yang, X. He, Bat algorithm: literature review and applications, Int. J. Bio-Inspired Comput. 5 (3) (2013) 141–149.

[380] Y. Xiao, Bio-inspired Computing and Networking, CRC Press, 2016.

[381] A. Awais, M. Farooq, M.Y. Javed, Attack analysis & bio-inspired security framework for ipmultimedia subsystem, in: Proceedings of the 10th Annual Conference Companion on Genetic and Evolutionary Computation, ACM, 2008, pp. 2093–2098.

[382] F. Afifi, N.B. Anuar, S. Shamshirband, K.-K.R. Choo, Dyhap: dynamic hybrid ANFIS-PSO approach for predicting mobile malware, PLoS One 11 (9) (2016) e0162627.

[383] K. Demertzis, L. Iliadis, Bio-inspired hybrid intelligent method for detecting android malware, in: Knowledge, Information and Creativity Support Systems, Springer, 2016, pp. 289–304.

[384] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-g. Wang, J. Chen, Detection of malicious code variants based on deep learning, IEEE Trans. Ind. Inf. 14 (7) (2018) 3187–3196.

[385] J. Kusyk, M.U. Uyar, C.S. Sahin, Survey on evolutionary computation methods for cybersecurity of mobile ad hoc networks, Evolut. Intell. 10 (3–4) (2018) 95–117.

[386] H. He, C. Maple, T. Watson, A. Tiwari, J. Mehnen, Y. Jin, B. Gabrys, The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence, in: IEEE Congress on Evolutionary Computation, IEEE, 2016, pp. 1015–1021.

[387] S. Elsayed, R. Sarker, D. Essam, Survey of uses of evolutionary computation algorithms and swarm intelligence for network intrusion detection, Int. J. Comput. Intell. Appl. 14 (04) (2015) 1550025.

[388] S. Al Amro, D.A. Elizondo, A. Solanas, A. Martínez-Ballesté, Evolutionary computation in computer security and forensics: an overview, in: Computational Intelligence for Privacy and Security, Springer, 2012, pp. 25–34.

[389] A. Abraham, C. Grosan, C. Martin-Vide, Evolutionary design of intrusion detection programs, IJ Netw. Secur. 4 (3) (2007) 328–339.

[390] G.N. Yannakakis, J. Togelius, Artificial Intelligence and Games, vol. 2, Springer, 2018.

[391] N. Shaker, J. Togelius, M.J. Nelson, Procedural Content Generation in Games, Computational Synthesis and Creative Systems, Springer, 2016.

[392] M. Nogueira Collazo, C. Cotta, A.J. Fernández Leiva, Competitive algorithms for coevolving both game content and AI. A case study: planet wars, IEEE Trans. Comput. Intell. AI Games 8 (4) (2016) 325–337.

[393] G. Recio, E. Martin, C. Estébanez, Y. Saez, Antbot: ant colonies for video games, IEEE Trans. Comput. Intell. AI Games 4 (4) (2012) 295–308.

[394] A. Gonzalez-Pardo, F. Palero, D. Camacho, An empirical study on collective intelligence algorithms for video games problem-solving, Comput. Inf. 34 (1) (2015) 233–253.

[395] I. Fister Jr., M. Perc, K. Ljubič, S.M. Kamal, A. Iglesias, I. Fister, Particle swarm optimization for automatic creation of complex graphic characters, Chaos, Solit. Fractals 73 (2015) 29–35.

[396] S.M. Lucas, G. Kendall, Evolutionary computation and games, IEEE Comput. Intell. Mag. 1 (1) (2006) 10–18.

[397] E. Melcer, T.-H.D. Nguyen, Z. Chen, A. Canossa, M.S. El-Nasr, K. Isbister, Games research today: analyzing the academic landscape 2000–2014, Network 17 (2015) 20.

[398] K.O. Andrade, R.C. Joaquim, G.A. Caurin, M.K. Crocomo, Evolutionary algorithms for a better gaming experience in rehabilitation robotics, Comput. Entertain. (CIE) 16 (2) (2018) 4.

[399] J. Del Ser, E. Osaba, J.J. Sanchez-Medina, I. Fister Jr., I. Fister, Bioinspired computational intelligence and transportation systems: A long road ahead, IEEE Trans. Intell. Transp. Syst. (2019) 1–30, https://doi.org/10.1109/TITS.2019.2897377 (in press).

[400] D. Zhao, Y. Dai, Z. Zhang, Computational intelligence in urban traffic signal control: A survey, IEEE Trans. Syst., Man, Cybern., Part C (Applications and Reviews) 42 (4) (2012) 485–494.

[401] M.R. Jabbarpour, H. Zarrabi, R.H. Khokhar, S. Shamshirband, K.-K.R. Choo, Applications of computational intelligence in vehicle traffic congestion problem: A survey, Soft Comput. 22 (7) (2018) 2299–2320.

[402] A. Diez-Olivan, J. Del Ser, D. Galar, B. Sierra, Data fusion and machine learning for industrial prognosis: trends and perspectives towards industry 4.0, Inf. Fusion 50 (2019) 92–111.

[403] M. Gen, W. Zhang, L. Lin, Y. Yun, Recent advances in hybrid evolutionary algorithms for multiobjective manufacturing scheduling, Comput. Ind. Eng. 112 (2017) 616–633.

[404] E. Wari, W. Zhu, A survey on metaheuristics for optimization in food manufacturing industry, Appl. Soft Comput. 46 (2016) 328–343.

[405] M. Gen, L. Lin, Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey, J. Intell. Manuf. 25 (5) (2014) 849–866.

[406] P. Verma, K. Sanyal, D. Srinivasan, K. Swarup, R. Mehta, Computational intelligence techniques in smart grid planning and operation: a survey, in: IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia), 2018, pp. 891–896.

[407] B.Y. Qu, Y.S. Zhu, Y.C. Jiao, M.Y. Wu, P.N. Suganthan, J.J. Liang, A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems, Swarm Evolut. Comput. 38 (2018) 1–11.

[408] C. Gamarra, J.M. Guerrero, Computational optimization techniques applied to microgrids planning: a review, Renew. Sustain. Energy Rev. 48 (2015) 413–424.

[409] T. Logenthiran, D. Srinivasan, T.Z. Shun, Demand side management in smart grid using heuristic optimization, IEEE Trans. Smart Grid 3 (3) (2012) 1244–1252.

[410] A.R. Jordehi, A review on constraint handling strategies in particle swarm optimisation, Neural Comput. Appl. 26 (6) (2015) 1265–1275.

[411] Y.G. Woldesenbet, G.G. Yen, B.G. Tessema, Constraint handling in multiobjective evolutionary optimization, IEEE Trans. Evol. Comput. 13 (3) (2009) 514–525.

[412] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art, Comput. Methods Appl. Mech. Eng. 191 (11–12) (2002) 1245–1287.

[413] E. Tsang, Foundations of Constraint Satisfaction: The Classic Text, BoD–Books on Demand, 2014.

[414] R. Barták, M.A. Salido, F. Rossi, New trends in constraint satisfaction, planning, and scheduling: A survey, Knowl. Eng. Rev. 25 (3) (2010) 249–279.

[415] Y. Jin, Knowledge Incorporation in Evolutionary Computation, vol. 167, Springer, 2013.

[416] I. Fister Jr., U. Mlakar, J. Brest, I. Fister, A new population-based nature-inspired algorithm every month: Is the current era coming to the end, in: Student Computer Science Research Conference, 2016, pp. 33–37.

[417] D. Weyland, A critical analysis of the harmony search algorithm – how not to solve sudoku, Oper. Res. Perspect. 2 (2015) 97–105.

[418] M.P. Saka, O. Hasançebi, Z.W. Geem, Metaheuristics in structural optimization and discussions on harmony search algorithm, Swarm Evolut. Comput. 28 (2016) 88–97.

[419] M.A. Lones, Metaheuristics in nature-inspired algorithms, in: Annual ACM Conference on Genetic and Evolutionary Computation, 2014, pp. 1419–1422.

[420] A. González-Pardo, J. Del Ser, D. Camacho, Comparative study of pheromone control heuristics in ACO algorithms for solving RCPSP problems, Appl. Soft Comput. 60 (2017) 241–255.

[421] C.L. Camacho-Villalón, M. Dorigo, T. Stützle, Why the intelligent water drops cannot be considered as a novel algorithm, in: International Conference on Swarm Intelligence, Springer, 2018, pp. 302–314.

[422] A. Rajasekhar, N. Lynn, S. Das, P.N. Suganthan, Computing with the collective intelligence of honey bees–a survey, Swarm Evolut. Comput. 32 (2017) 25–48.

[423] J. Swan, S. Adriaensen, M. Bishr, E.K. Burke, J.A. Clark, P. De Causmaecker, J. Durillo, K. Hammond, E. Hart, C.G. Johnson, Z.A. Kocsis, B. Kovitz, K. Krawiec, S. Martin, J.J. Merelo, L.L. Minku, E. Özcan, G.L. Pappa, E. Pesch, P. Garcia-Sanchez, A. Schaerf, K. Sim, J.E. Smith, T. Stützle, S. Voß, S. Wagner, X. Yao, A research agenda for metaheuristic standardization, in: XI Metaheuristics International Conference, 2015, pp. 1–5.

[424] X.-S. Yang, M. Karamanoglu, Swarm intelligence and bio-inspired computation: An overview, in: X.-S. Yang, Z. Cui, R. Xiao, A.H. Gandomi, M. Karamanoglu (Eds.), Swarm Intelligence and Bio-Inspired Computation, Elsevier, 2013, pp. 3–23. https://doi.org/10.1016/B978-0-12-405163-8.00001-6.

[425] P.K. Lehre, P.S. Oliveto, Theoretical analysis of stochastic search algorithms, in: Handbook of Heuristics, 2018, pp. 849–884.

[426] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-Filho, R. Menezes, Unveiling Swarm Intelligence with Network Science – the Metaphor Explained, 2018, arXiv:1811.03539.

[427] A. Hooper, Green computing, Commun. ACM 51 (10) (2008) 11–13.

[428] D.J. Montana, L. Davis, Training feedforward neural networks using genetic algorithms, in: International Joint Conference on Artificial Intelligence, vol. 89, 1989, pp. 762–767.

[429] E. Alba, R. Martí, Metaheuristic Procedures for Training Neural Networks, vol. 35, Springer Science & Business Media, 2006.
[430] Y. Jin, Surrogate-assisted evolutionary computation: recent advances and future challenges, Swarm Evolut. Comput. 1 (2) (2011) 61–70.
[431] L. Cao, Data science: challenges and directions, Commun. ACM 60 (8) (2017) 59–68.
[432] J.A. Vrugt, C. Ter Braak, C. Diks, B.A. Robinson, J.M. Hyman, D. Higdon, Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling, Int. J. Nonlinear Sci. Numer. Stimul. 10 (3) (2009) 273–290.
[433] C.J. Ter Braak, A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces, Stat. Comput. 16 (3) (2006) 239–249.
[434] C. Wang, C. Xu, X. Yao, D. Tao, Evolutionary generative adversarial networks, IEEE Trans. Evol. Comput. (2019), https://doi.org/10.1109/TEVC.2019.2895748.
[435] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626.
[436] S. Mahdavi, S. Rahnamayan, K. Deb, Opposition based learning: a literature review, Swarm Evolut. Comput. 39 (2018) 1–23.
[437] M.M. Drugan, Reinforcement learning versus evolutionary computation: a survey on hybrid algorithms, Swarm Evolut. Comput. 44 (2019) 228–246.
[438] F.P. Such, V. Madhavan, E. Conti, J. Lehman, K.O. Stanley, J. Clune, Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning, 2017, arXiv:1712.06567.
[439] N. Müller, T. Glasmachers, Challenges in high-dimensional reinforcement learning with evolution strategies, in: International Conference on Parallel Problem Solving from Nature, 2018, pp. 411–423.
[440] K.O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies, Evol. Comput. 10 (2) (2002) 99–127.
[441] T. Maul, A. Bargiela, S.Y. Chong, A.S. Adamu, F. Squazzoni, F. Baronio, C. Archetti, M. Castellani, Towards evolutionary deep neural networks, in: European Conference on Modelling and Simulation, 2014, pp. 319–325.
[442] M. Suganuma, S. Shirakawa, T. Nagao, A genetic programming approach to designing convolutional neural network architectures, in: ACM Genetic and Evolutionary Computation Conference, 2017, pp. 497–504.
[443] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, et al., Evolving deep neural networks, in: Artificial Intelligence in the Age of Neural Networks and Brain Computing, 2019, pp. 293–312.
[444] R. Lara-Cabrera, C. Cotta, A.J. Fernández-Leiva, A review of computational intelligence in RTS games, in: IEEE Symposium on Foundations of Computational Intelligence, 2013, pp. 114–121.
[445] Z. Chen, B. Liu, Lifelong machine learning, Synth. Lect. Artif. Intell. Mach. Learn. 10 (3) (2016) 1–145.
[446] A. Verma, X. Llorà, D.E. Goldberg, R.H. Campbell, Scaling genetic algorithms using mapreduce, in: IEEE International Conference on Intelligent Systems Design and Applications, 2009, pp. 13–18.
[447] P. Fazenda, J. McDermott, U.-M. O'Reilly, A library to run evolutionary algorithms in the cloud using MapReduce, in: European Conference on the Applications of Evolutionary Computation, 2012, pp. 416–425.
[448] B. Jena, M.K. Gourisaria, S.S. Rautaray, M. Pandey, A survey work on optimization techniques utilizing map reduce framework in hadoop cluster, Int. J. Intell. Syst. Appl. 9 (4) (2017) 61.
[449] D. Camacho, R. Lara-Cabrera, J. Merelo-Guervós, P.A. Castillo, C. Cotta, A.J. Fernández-Leiva, F.F. de Vega, F. Chávez, From ephemeral computing to deep bioinspired algorithms: new trends and applications, Future Gener. Comput. Syst. 88 (2018) 735–746.
[450] S. Fiore, M. Bakhouya, W.W. Smari, On the Road to Exascale: Advances in High Performance Computing and Simulations – an Overview and Editorial, 2018.
[451] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549 (7671) (2017) 195.
[452] C. Ciliberto, M. Herbster, A.D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, L. Wossnig, Quantum machine learning: A classical perspective, Proc. R. Soc. A 474 (2209) (2018) 20170551.