# Resilient Bioinspired Algorithms: A Computer System Design Perspective*

Carlos Cotta[1,2][0000−0001−8478−7549] and Gustavo Olague[3][0000−0001−5773−9517]

[1] Dept. Lenguajes y Ciencias de la Computación, ETSI Informática, Campus de Teatinos, Universidad de Málaga, 29071 Málaga, Spain ccottap@lcc.uma.es
[2] ITIS Software, Universidad de Málaga, Spain
[3] EvoVisión Laboratory, CICESE Research Center, Carretera Ensenada-Tijuana 3918, Col. Playitas, Ensenada 22860, B.C. México olague@cicese.mx

**Abstract.** Resilience can be defined as a system's capability for returning to normal operation after having suffered a disruption. This notion is of the foremost interest in many areas, in particular engineering. We argue in this position paper that is is a crucial property for bioinspired optimization algorithms as well. Following a computer system perspective, we correlate some of the defining requirements for attaining resilient systems to issues, features, and mechanisms of these techniques. It is shown that bioinspired algorithms do not only exhibit a notorious built-in resilience, but that their plasticity also allows accommodating components that may boost it in different ways. We also provide some relevant research directions in this area.

**Keywords:** Resilience · Bioinspired Optimization · Robustness · Computer Systems

## 1 Introduction

Stemming from the Latin word *resilire* (to jump back, or to rebound), dictionaries commonly define resilience as (1) the ability of something to return to its original shape after it has been pulled, stretched, pressed, bent, etc. and (2) the ability to become strong, healthy, or successful again after something bad happens. While the first definition is more in line with a literal Material Science interpretation, the second one has a more figurative sense that nevertheless seems more appropriate within a computational context: it captures the ability of a computer system to deliver again its functionality after a disruptive event takes place. While we shall revisit the meaning of resilience later on, let us note here in that it is commonly the case within such a computational context that resilience is identified as a synonym for fault tolerance and safety at critical applications. However, notice that fault tolerance does not necessarily implies bouncing back to normal operation, and can simply entail a well-defined

* Carlos Cotta acknowledges support by Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech.

behavior after a fault [6]. Hence, we can argue that resilience goes beyond fault tolerance, or at least that it has its own particularities, some significant overlap with fault tolerance notwithstanding.

Resilience turns out to be a fundamental feature of technological systems. Our daily life is notoriously dependent on the availability and proper functioning of many networks, computing infrastructures, and most importantly on numerous algorithms running on them. Bioinspired optimization methods are no exception to this, since not only they can be deployed on irregular, dynamic computational environments [8] but they also may have to face challenges of diverse nature during their regular operation (dynamic environments, uncertain objectives, byzantine faults, etc.). It is therefore of the foremost interest to analyze these techniques, the challenges they have to cope with, and the way they can be appropriately designed from a resilience viewpoint. This position is defended in this work, in which we try to draw some rough lines to map the ground using some lessons from other engineering fields, as well as identifying some challenges in pursue of resilience properties. The rest of this paper is organized as follows: we firstly discuss some general issues about resilience at large (Sect. 2.1), and dive into the engineering perspective on this property, and the requirements to achieve it (Sect. 2.2); then, we proceed to discuss these requirements in the context of bioinspired algorithms, i.e., what they typically entail and/or how they are often approached (Sect. 3); we close the paper with an outlook and a sketch of some challenges we believe are important in this area (Sect. 4).

## 2 Background

### 2.1 What is Resilience?

There are many different definitions of resilience [22]. The United Nations defined it in the General Assembly Resolution 71/276 as "*the ability of a system, community or society exposed to hazards to resist, absorb, accommodate, adapt to, transform and recover from the effects of a hazard in a timely and efficient manner, including through the preservation and restoration of its essential basic structures and functions through risk management*," cf. [49]. In line with this definition, the potential sources for sudden, disruptive events are numerous, and they can have natural or anthropogenic causes: natural disasters, malicious human activity, health crises, economic meltdowns, and so on [55]. If we consider a physical, a computational, or a technosocial system [51], the specificities of the disruptive events can be different. However, the bottom-line remains: they expose vulnerabilities in the corresponding systems (and in a meta-level, so can they also do in higher systems that use the former). In response and–most importantly–in anticipation of this, it is necessary to foster resilience.

Building resilience allows reverting to normal conditions after a shock. Conversely, a lack of resilience prevents the restoration of these conditions. In this sense, it is essential to distinguish between resilience and the related notion of *robustness*: while the former refers to the capacity to withstand shocks dynamically, the latter sometimes denote the ability to resist shocks without adapting

[7]. For example, it is possible to make a system robust by endowing it with redundant components to ensure continuous operation even if some of them fail. On the other hand, a system could be resilient by reconfiguring some components to keep delivering the required functionality after a shock (perhaps, having to endure a transient period of degraded performance until the reconfiguration is effective). Of course, these two possibilities are not mutually exclusive and can be ideally combined cost-effectively.

Resilience is essential to ensure *sustainability* (and conversely, absence of resilience results in unsustainability and adverse feedback loops after a crisis). Quite counterintuitively, sustainability can also be related to risk-exposure, in a phenomenon known as the volatility paradox [25], whereby systems with a low systemic risk build-up increasingly fragile, ultimately undermining sustainability; much like the immune system requires being confronted with pathogens to build up defenses, a system that endures crises at a higher frequency will develop resilience over time. Furthermore, disruptions (temporary shocks notwithstanding) are the catalysts of growth and innovation, which are essential for sustained progress instead of stagnation (biologists and evolutionary computation practitioners will identify a common theme here).

### 2.2 Resilience from an Engineering Perspective

Beyond the general definition of resilience provided in the previous subsection, it is possible to include more specific definitions withing the context of Engineering and more particularly in the context of Information, and Communications Technology. This will pave the way for attaining a better characterization of resilience in the domain of bioinspired optimization techniques. To this end, we will also indicate anthropogenic issues and malicious activities that generate a lack of trustworthiness in popular deep-learning methodologies. Also, we will provide examples in robotic technologies and machine learning industries for the interested reader.

According to [9], a resilient system acting within time, environmental, and operating conditions is that which is ready to perform its intended function, guaranteeing the absence of improper system alterations with the ability to anticipate and accommodate changes while executing and conducting servicing and inspection so that in case of a fault, quick restoration to a specified working condition must be achieved, or otherwise discontinue of the operation in a safe way. Fig. 1 shows different attributes and measures of resilience relevant to this goal. Let us briefly discuss these:

- *Reliability* is a measure of the extent to which a system can provide a continued service up to a certain time. It is therefore particularly relevant to safety critical system in which service discontinuation is not an acceptable possibility.
- *Security* encloses a subset of attributes, including integrity, maintainability, and availability.

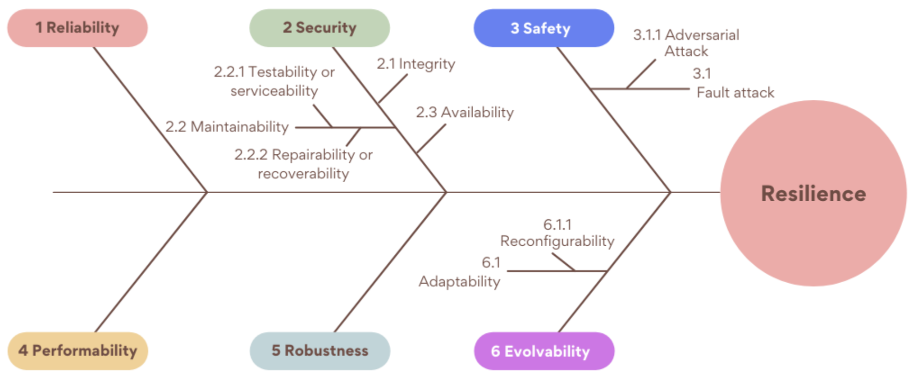


**Fig. 1.** Resilience is a term with multiple aspects such as reliability, security, safety, performability, robustness, and evolvabiliy.

  - *Integrity* refers to absence of improper system states (be these states physical or –as it is usually the case in algorithms– logical). Keeping integrity thus refers to preventing improper alterations to the state.
  - *Maintainability* captures the extent to which a system that has been damaged/compromised can be repaired.
  - *Availability* refers to the readiness for service. It could be defined as a measure of the how often a certain system is functional.
- *Safety* amounts to the system's ability to avoid catastrophic failures, understood as any failure that causes damage to others systems and/or compromises the safety of these. It is thus a measure of the fail-safe capabilities of the system.
- *Performability* measures the extent to which a system performs above specific functioning requirements (be these, speed, accuracy, resource consumption, etc.). It is therefore an indicator of interest for systems whose performance can be determined in a quantitative way, as it is the case of bioinspired optimization algorithms.
- *Robustness* captures whether the system can deliver correct service conditions beyond the typical domain of operation, and without fundamental changes to the original system (cf. Sect. 2.1).
- *Evolvability* measures the extent to which the system can perform changes on itself, be it decreasing its level of performance or reliability for a specific time range to compensate for faults or during exceptional circumstances

(grateful degradation) or by adapting any aspect of its functioning in order to ensure appropriate (or even improved) performance. It is therefore related to the notions of *elasticity* and *adaptability*. Specifically, engineers consider that a resilient system must have the ability to be adaptable (which is to be understood as the ability to evolve while executing; Therefore, adaptability is a subset of evolvability and may require anticipating changes prior to the resulting damage, or simply taking actions reacting to such changes).

As we may observe, resilience encompasses important attributes and measures that people use across science and engineering. Such concepts help to conceptualize different aspects needed to explain resilience. For example, resiliency naturally appears in robotics and machine learning in connection to malevolent external actions. Regarding the former, malicious attacks represent a challenging issue and preventing security vulnerabilities due to human factors is a significant subject aiming to implement and maintain effective countermeasures [53]. As to the latter, security is indeed an open issue since the technology is susceptible to adversarial attacks from hackers [1]. From the viewpoint of evolutionary computation we argue that human and data modeling are two aspects that need higher attention from the research community with interest in the development of bioinspired resilient systems.

## 3 Bioinspired Algorithms as Resilient Systems

Having laid out a general view of resilience in the previous section, as well as the requirements to attain this property in a computer system, let us discuss how these apply to the particular case of bioinspired optimization algorithms. Of course, this particular algorithmic paradigm has its own specifities, which render some of the issues defined before as only tangentially applicable to these techniques. This fact notwithstanding, the core requirements for resilience are relevant in this domain and can be actually quantified, as we shall see. Indeed, it is possible to group these requirements in a natural way into a number of feature sets which are described next.

### 3.1 Integrity and Safety

While integrity and safety may appear to be in principle some of the least applicable resilience features in this case, they actually characterize an important issue that has been extensively studied in this domain. These features can be seen as natural sides of a common issue in bioinspired optimization techniques: the normal operation of the system should not result in damage in its own state (let alone catastrophic damage), even accounting for potential external factors. Leaving aside the latter for a moment, it turns out there is indeed an internal operational factor that can cause damage to the algorithm state, namely convergence to suboptimal regions of the search space. This is typically due to the presence of deceptive features [52] in the search landscape, whereby the algorithm is led to local optima which may be in some cases far from the actual global optimum [24]. As mentioned before, the issue of premature convergence and deceptiveness has been one of the major topics studied in the literature, both from theoretical and experimental perspectives. The use of populations is widely regarded as one of the primary safeguards of bioinspired optimization techniques with respect to deceptive local optima, though their ability to escape from these will greatly depend on the effective maintenance of diversity. Fortunately, there are numerous mechanisms whereby diversity can be promoted, either proactively (e.g., use of non-panmictic populations [18], ad-hoc operators [11, 16], etc. – see also [31]) or reactively (e.g., random immigrants [48], triggered hypermutation[29], etc. – cf. Sect. 3.2). This also means that it is possible to capitalize on the knowledge available on convergence metrics (see, e.g., [12]) in order to measure the integrity $I(t)$ of the algorithm.

If external factors come into play, integrity and safety become more prominent features. Consider for example the case of volunteer computing (VC) networks. The existence of malicious agents who operate within these networks providing false results (i.e., *cheaters*) has been long documented [42]. Research suggests that distributed evolutionary algorithms running in this kind of hostile environments can indeed tolerate some degree of cheating, and would theoretically converge to the optimum given enough time [30]. Needless to say, these malicious agents can exert some other pernicious influence on distributed applications, but most of these are either implementation-dependent (e.g., overflowing buffers, injecting code, etc.) or can be better dealt with by other resilience requirements (e.g., crashes – see Sect. 3.3).

### 3.2 Evolvability and Adaptability

Evolvability and adaptability are flagship features of bioinspired optimization techniques, and surely one of their *reasons d'être*. Indeed, parameter adaptation is deeply rooted as a core principle of some bioinspired computation flavors (e.g., evolution strategies), and was taxomized well before the turn of the century [21]. Of course, adaptation (and self-adaptation) does not limit to parameter control in bioinspired methods. As a matter of fact, it can be found in other components such as population structures [15], or the definition of variation operators (e.g., local search mechanisms in memetic algorithms [44]), just to name a few. This is not surprising, since one of the keystones in practical (meta)heuristic problem-solving is the fact that tuning the optimization technique to the problem under consideration is paramount for achieving top performance, and that transferring a part of this tuning/customization effort from the human designer to the algorithm itself –i.e., by endowing it with smart mechanisms to self-adapt to the problem– has been a long pursued goal in the field of metaheuristics [10]. To a large extent, this is something that lies precisely at the root of the notion of memetic computing [32] (which is understood as the harmonic coordination of complex computational structures composed of interacting modules –memes– for problem solving, whose representation is stored and manipulated by the algorithm itself) and hyperheuristics [14] (which comprise heuristic techniques for intelligently selecting or generating a suitable heuristic for a given situation).

The previous examples underpin the amenability of bioinspired optimization methods for accommodating (self-)adaptive components and effectively taking advantage of them. From a broader perspective, these components are responsible for endowing the algorithm with self-$\star$ properties, namely any property whereby a certain system can exert advanced control on its own functioning and/or structure [4], ultimately contributing to the resilience of the former. Such properties may include self-organizing, self-healing, self-configuring or self-scaling among many others [5], and they turn out to be essential to cope with some major environmental disruptions as shown next. Before proceeding to that, let us note that evolvability is multi-faceted, and therefore quantifying $E(t)$ may depend on the particular feature subject to study (see, e.g., [45, 46]).

### 3.3 Performability and Recoverability

Performability and recoverability arguably capture the quintessential features required to develop resilience in this context. As anticipated before, it is very common that from a computational perspective resilience is equated to (or at least strongly connected to) fault tolerance. There is a large truth in this connection, at least to the extent that fault tolerance is understood as the ability to keep delivering the expected performance in the presence of failures (maybe tautologically so). Then again, fault tolerance can be more broadly assumed to mean well-defined behavior in the presence of failures [6]. Furthermore, even if we assume that delivering uninterrupted service is what defines a fault-tolerant system, an argument could be done as to whether this is achieved by means of some built-in robustness (i.e., the system is capable to withstand failures without needing to adapt or change its behaviour), or by developing resilience (i.e., the system recovers its performance after a transient degradation phase, by means of some internal adaptation or reconfiguration), cf. Sect. 2.1. As we shall see, bioinspired optimization methods can achieve fault tolerance under either interpretation, although in one case they may simply rely on its intrinsic architecture, whereas in the other the inclusion of appropriate mechanisms may be required.

Focusing on evolutionary algorithms in particular, basic fault-tolerance has been analyzed from different perspectives. It has been established that the use of populations provides some intrinsic redundancy, whereby moderate losses of individuals do not result in major performance degradation in master-slave panmictic models [27], fine-grained (cellular) decentralized models [26], and coarse-grained (island-based) decentralized models [20], and this robustness can be enhanced via standard fault-tolerance mechanisms such as checkpointing [33]. A more interesting perspective from the resilience viewpoint can be attained by endowing the algorithm with self-$\star$ properties, as mentioned in Sect. 3.2. Relevant properties in this context are self-scaling (i.e., exerting internal reconfiguration in response to changes in the computational substrate) [34] and self-healing (i.e., performing actions to correct any damage infringed by external disruptions) [35]. Performability is approached from a different angle in *brain programming* [39], a symbolic paradigm that uses the power of genetic programming combined with neuroscientific modeling and that is aimed at purposive vision. Therein, generality is a designed property where models are constantly trained in one problem and tested on a different problem, not only changing the dataset but the whole visual task [38].

The bottom-line is here that –as mentioned in Sect. 3.2– these methods can naturally accommodate self-adaptive and reactive components that enable sensibly responding to failures in a resilient way. Furthermore, the very quantitative nature in which the performance of bioinspired optimization methods can be measured leads in turn to an amenable quantification of performability $P(t)$ at time $t$, by comparing to regular undegraded performance values, or by determining the maximum level at which the algorithm yielded acceptable performance up to time $t$ (i.e., $P(t) = \max(L \mid \forall t' \leqslant t : \psi(t') \geqslant L)$, where $\psi(t)$ is the algorithm's performance at time $t$).

### 3.4 Reliability and Availability

Reliability and availability have in this particular context a significant overlap with the features previously discussed. If service continuity (understood as the effective fulfilment of the optimization purposes of the specific bioinspired method considered) is pursued in the presence of computational failures, we would be in the scenarios depicted in previous section. The previous setting is not the only possible one in order to assess reliability. In fact, there is a very interesting and relevant subfield of research that deals with dynamic optimization, namely the use of these methods in scenarios in which the optimization target changes along time [2]. Needless to say, this poses great challenges to any optimization technique, and require the use of appropriate mechanisms (such as using archives of previous solutions, diversity-preservation policies, and control mechanisms to anticipate, detect, or react to to changes in the optimization target – see [28, 54]) in order to be able to provide trustworthy operation in this kind of environments. From a different perspective, reliability aspects also appear in vision metrology systems (where thousands of simulation evaluations using complex nonlinear least-squares analysis are required, often relying in surrogate models) [37] and in the analysis of corner extraction [36].

In either case, it must be noted that reliability is often a property used to characterize systems where failures are unacceptable (e.g., safety critical systems) [9], which means that in this particular bioinspired optimization context adequate thresholds would have to be defined to characterize when the transient degraded performance does not render the optimization service discontinued. From a more quantitative point of view, the reliability $R(t)$ could be here defined as the probability of the algorithm delivering acceptable functionality at time $t$, which could be approximated as the proportion of runs in which acceptable behavior is observed. Likewise, the availability $A(t)$ could be approximated as the fraction of the time in which the algorithm delivered correct performance.

### 3.5 Sustainability

Sustainability refers to the ability to maintain a trend or a process in the long run. It is a concept that is frequently brought up in connection to human activities and the impact that these have on the environment and the toll they exert on future resource availability. As indicated in Sect. 2.1, sustainability is one of the natural consequences of resilient operation. In this sense, sustainability is not just an application goal of bioinspired methods (and any other AI method, for that matter), but also an operational requirement of these techniques. Unsurprisingly, AI methods have been identified as having a significant carbon footprint [47]. While a standard of measurement is still absent for quantifying energy consumption and carbon emission in the life cycle of AI methods [13], it is clear that monitoring the emission level of these techniques, and prioritizing energetically efficient computational platforms and algorithms is crucial. This has led to the notions of *red AI* and *green AI* [43]. The former refers to AI research that seeks to obtain state-of-the-art results through the use of massive computational power, and particularly applies to scenarios in which the computational effort scales at a substantially larger pace than the gains obtained in the results [19], whereas green AI research provides novel results without increasing computational cost or even reducing it. This is a topic that is gaining momentum in the context of machine-learning, but for which some seminal works notwithstanding, e.g., [17, 3], requires further analysis in the context of bioinspired optimization.

## 4 Outlook and Challenges

Resilience in engineering and computer science is a well-established research area, and we found a rich connection with bioinspired approaches in most aspects typically studied in other domains. We identified five research axes, namely (i) integrity and safety, (ii) evolvability and adaptability, (iii) performability and recoverability, (iv) reliability and availability, and (v) sustainability. We give a first account of the kind of problems researchers study in each compound set of attributes. An important lesson learned is that bioinspired methods have a great deal of intrinsic resilience and –most importantly– are flexible enough to admit being augmented with components to boost resilience in their different aspects.

We envision that future research will not just focus on studying the resilience of bioinspired optimization methods, but should also exploit resilience as a stimulus for optimization. In this sense, it must be noted that exposure to disruptions is very often the catalyst for breakthroughs. This has been observed in many systems, including bioinspired methods (e.g., see [50, 34]), and lies in the spirit of some long-known approaches such as competitive coevolution [41] and, more recently, adversarial attacks (e.g., [23, 40]). The latter have a longer history in the area of computer vision and machine learning, and are bound to have an important impact in bioinspired optimization as well. Needless to say, these strategies are targeted to disrupt evolutionary equilibrium, hence exerting a continuous and directed selective pressure, but analogous strategies aimed to strategically attack other aspects of the algorithm are not inconceivable, and would provide the playground for the evolution of resilience, and indirectly for the improvement of the underlying optimization process. Another line of research which we envision will gain momentum in the near future is the development of green bioinspired optimization methods, following the trend of the machine-learning community.

## References


1. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access **6**, 14410–14430 (2018)
2. Alba, E., Nakib, A., Siarry, P. (eds.): Metaheuristics for Dynamic Optimization, Studies in Computational Intelligence, vol. 433. Springer (2013)
3. Álvarez, J.D., Chávez, F., Castillo, P.A., García Martínez and Francisco J. Rodriguez, J.Á., Fernández de Vega, F.: A fuzzy rule-based system to predict energy consumption of genetic programming algorithms. Comput. Sci. Inf. Syst. **15**(3), 635–654 (2018)
4. Babaoglu, O., Jelasity, M., Montresor, A., Fetzer, C., Leonardi, S., van Moorsel, A., van Steen, M. (eds.): Self-star Properties in Complex Information Systems, Lecture Notes in Computer Science, vol. 3460. Springer-Verlag, Berlin Heidelberg (2005)
5. Berns, A., Ghosh, S.: Dissecting self-$\star$ properties. In: Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems - SASO 2009. pp. 10–19. IEEE Press, San Francisco, CA (2009)
6. Bouvry, P., Mayer, R., Muszynski, J., Petcu, D., Rauber, A., Tempesti, G., Trinh, T., Varrette, S.: Resilience within ultrascale computing system: Challenges and opportunities from Nesus project. Supercomputing Frontiers and Innovations **2**(2), 46–63 (apr 2015)
7. Brunnermeier, M.: The Resilient Society. Endeavor Literary Press, Colorado Springs (2021)
8. Camacho, D., Lara-Cabrera, R., Merelo-Guervós, J., Castillo, P.A., Cotta, C., Fernández-Leiva, A.J., Fernández de Vega, F., Chávez, F.: From ephemeral computing to deep bioinspired algorithms: New trends and applications. Future Generation Computer Systems **88**, 735–746 (2018)
9. Castano, V., Schagaev, I.: Resilient Computer System Design. Springer, Cham (2015)
10. Cotta, C., Sevaux, M., Sörensen, K. (eds.): Adaptive and Multilevel Metaheuristics, Studies in Computational Intelligence, vol. 136. Springer-Verlag, Berlin Heidelberg (2008)
11. Cotta, C., Troya, J.: Using dynastic exploring recombination to promote diversity in genetic search. In: Schoenauer, M., et al. (eds.) Parallel Problem Solving From Nature VI, Lecture Notes in Computer Science, vol. 1917, pp. 325–334. Springer-Verlag, Paris (2000)
12. Derrac, J., García, S., Hui, S., Suganthan, P.N., Herrera, F.: Analyzing convergence performance of evolutionary algorithms: A statistical approach. Information Sciences **289**, 41–58 (2014)
13. Dhar, P.: The carbon impact of artificial intelligence. Nature Machine Intelligence **2**, 423–425 (2020)
14. Drake, J.H., Kheiri, A., Özcan, E., Burke, E.K.: Recent advances in selection hyperheuristics. European Journal of Operational Research **285**(2), 405–428 (2020)

15. Fernandes, C.M., Rosa, A.C., Laredo, J.L., Merelo, J., Cotta, C.: Dynamic models of partially connected topologies for population-based metaheuristics. In: 2018 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8. IEEE, Rio de Janeiro, Brazil (2018)
16. Fernandes, C., Laredo, J., Rosa, A., Merelo, J.: The sandpile mutation genetic algorithm: an investigation on the working mechanisms of a diversity-oriented and self-organized mutation operator for non-stationary functions. Applied Intelligence **39**(2), 279–306 (2013)
17. Fernández de Vega, F., Chávez de la O, F., Díaz, J., García, J.Á., Castillo, P.A., Merelo Guervós, J.J., Cotta, C.: A cross-platform assessment of energy consumption in evolutionary algorithms - towards energy-aware bioinspired algorithms. In: Handl, J., Hart, E., Lewis, P.R., López-Ibáñez, M., Ochoa, G., Paechter, B. (eds.) Parallel Problem Solving from Nature - PPSN XIV. Lecture Notes in Computer Science, vol. 9921, pp. 548–557. Springer (2016)
18. Harada, T., Alba, E.: Parallel genetic algorithms: A useful survey. ACM Comput. Surv. **53**(4), 86:1–86:39 (2020)
19. Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., Pineau, J.: Towards the systematic reporting of the energy and carbon footprints of machine learning. Journal of Machine Learning Research **21**(248), 1–43 (2020)
20. Hidalgo, J., Lanchares, J., Fernández de Vega, F., Lombraña, D.: Is the island model fault tolerant? In: Genetic and Evolutionary Computation - GECCO 2007. pp. 2737 – 2744. ACM, New York (2007)
21. Hinterding, R., Michalewicz, Z., Eiben, A.: Adaptation in evolutionary computation: A survey. In: Fourth IEEE Conference on Evolutionary Computation. pp. 65–69. IEEE Press, Piscataway, New Jersey (1997)
22. Huq, M., Sarker, M., Prasad, R., et al.: Resilience for disaster management: Opportunities and challenges. In: Alam, G., et al. (eds.) Climate Vulnerability and Resilience in the Global South, pp. 425–442. Springer, Cham (2021)
23. Ibarra-Vazquez, G., Olague, G., Chan-Ley, M., Puente, C., Soubervielle-Montalvo, C.: Brain programming is immune to adversarial attacks: Towards accurate and robust image classification using symbolic learning (2021)
24. Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Eshelman, L.J. (ed.) Proceedings of the Sixth International Conference on Genetic Algorithms. pp. 184–192. Morgan Kaufmann (1995)
25. Kubitza, C.: Tackling the volatility paradox: Spillover persistence and systemic risk. ECONtribute Discussion Paper No. 079 (2021), `https://dx.doi.org/10.2139/ssrn.2858763`
26. Laredo, J., Castillo, P., Mora, A., Merelo, J., Fernandes, C.: Resilience to churn of a peer-to-peer evolutionary algorithm. International Journal of High Performance Systems Architecture **1**(4), 260–268 (2008)
27. Lombraña González, D., Jiménez Laredo, J., Fernández de Vega, F., Guervós, J.M.: Characterizing fault-tolerance in evolutionary algorithms. In: Parallel Architectures and Bioinspired Algorithms. Springer-Verlag (2012)
28. Mavrovouniotis, M., Li, C., Yang, S.: A survey of swarm intelligence for dynamic optimization: Algorithms and applications. Swarm and Evolutionary Computation **33**, 1–17 (2017)
29. Morrison, R., De Jong, K.: Triggered hypermutation revisited. In: Proceedings of the 2000 Congress on Evolutionary Computation. vol. 2, pp. 1025–1032. IEEE (2000)

30. Muszyński, J., Varrette, S., Bouvry, P., Seredyński, F., Khan, S.U.: Convergence analysis of evolutionary algorithms in the presence of crash-faults and cheaters. Computers & Mathematics with Applications **64**(12), 3805–3819 (2012)
31. Neri, F.: Diversity management in memetic algorithms. In: Neri, F., Cotta, C., Moscato, P. (eds.) Handbook of Memetic Algorithms, Studies in Computational Intelligence, vol. 379, pp. 155–167. Springer-Verlag, Berlin Heidelberg (2012)
32. Neri, F., Cotta, C.: Memetic algorithms and memetic computing optimization: A literature review. Swarm and Evolutionary Computation **2**, 1–14 (2012)
33. Nogueras, R., Cotta, C.: Studying fault-tolerance in island-based evolutionary and multimemetic algorithms. Journal of Grid Computing **13**, 351–374 (2015)
34. Nogueras, R., Cotta, C.: Studying self-balancing strategies in island-based multimemetic algorithms. Journal of Computational and Applied Mathematics **293**, 180–191 (2016)
35. Nogueras, R., Cotta, C.: Self-healing strategies for memetic algorithms in unstable and ephemeral computational environments. Natural Computing **16**(2), 189–200 (2017)
36. Olague, G., Hernandez, B., Dunn, E.: Hybrid evolutionary ridge regression approach for high-accurate corner extraction. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. vol. 1, pp. I–I (2003)
37. Olague, G.: Automated photogrammetric network design using genetic algorithms. Photogrammetric Enginering and Remote Sensing **68**(5), 423–431 (2002)
38. Olague, G., Clemente, E., Hernández, D.E., Barrera, A., Chan-Ley, M., Bakshi, S.: Artificial visual cortex and random search for object categorization. IEEE Access **7**, 54054–54072 (2019). https://doi.org/10.1109/ACCESS.2019.2912792
39. Olague, G., Hernández, D.E., Clemente, E., Chan-Ley, M.: Evolving head tracking routines with brain programming. IEEE Access **6**, 26254–26270 (2018)
40. Qiu, H., Custode, L.L., Iacca, G.: Black-box adversarial attacks using evolution strategies. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. p. 1827–1833. GECCO '21, Association for Computing Machinery, New York, NY, USA (2021)
41. Rosin, C.D., Belew, R.K.: New Methods for Competitive Coevolution. Evolutionary Computation **5**(1), 1–29 (1997)
42. Sarmenta, L.F.: Sabotage-tolerance mechanisms for volunteer computing systems. Future Generation Computer Systems **18**(4), 561–572 (2002)
43. Schwartz, R., Dodge, J., Smith, N., Etzioni, O.: Green ai. Communications of the ACM **63**(12), 54–63 (2020)
44. Smith, J.E.: Self-adaptative and coevolving memetic algorithms. In: Neri, F., Cotta, C., Moscato, P. (eds.) Handbook of Memetic Algorithms, Studies in Computational Intelligence, vol. 379, pp. 167–188. Springer (2012)
45. Smith, T., Husbands, P., O'Shea, M.: Fitness Landscapes and Evolvability. Evolutionary Computation **10**(1), 1–34 (2002)
46. Soria-Alcaraz, J.A., Espinal, A., Sotelo-Figueroa, M.A.: Evolvability metric estimation by a parallel perceptron for on-line selection hyper-heuristics. IEEE Access **5**, 7055–7063 (2017)
47. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for modern deep learning research. Proceedings of the AAAI Conference on Artificial Intelligence **34**(09), 13693–13696 (2020)
48. Tinós, R., Yang, S.: A self-organizing random immigrants genetic algorithm for dynamic optimization problems. Genetic Programmming & Evolvable Machines **8**, 255–286 (2007)

49. United Nations High-level Political Forum on Sustainable Development (HLPF): 2018 HPLF thematic review: Transformation towards sustainable and resilient societies – building resilience (2018), `sustainabledevelopment.un.org/hlpf/2018`
50. Fernández de Vega, F., Cantú-Paz, E., López, J., Manzano, T.: Saving resources with plagues in genetic algorithms. In: Yao, X., et al. (eds.) Parallel Problem Solving from Nature - PPSN VIII. Lecture Notes in Computer Science, vol. 3242, pp. 272–281. Springer-Verlag, Berlin Heidelberg (2004)
51. Vespignani, A.: Predicting the behavior of techno-social systems. Science **325**, 425–428 (2009)
52. Whitley, D.: Mk landscapes, NK landscapes, MAX-KSAT: A proof that the only challenging problems are deceptive. In: Silva, S., Esparcia-Alcázar, A.I. (eds.) Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. p. 927–934. ACM, New York, NY (2015)
53. Yaacoub, J.P.A., Noura, H.N., Salman, O., Chehab, A.: Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. International Journal of Information Security (2021)
54. Yazdani, D., Cheng, R., Yazdani, D., Branke, J., Jin, Y., Yao, X.: A survey of evolutionary continuous dynamic optimization over two decades—part a. IEEE Transactions on Evolutionary Computation **25**(4), 630–650 (2021)
55. Zobel, C., Khansa, L.: Characterizing multi-event disaster resilience. Computers and Operations Research **42**, 83–94 (2014)