npj | artificial intelligence

Comment

https://doi.org/10.1038/s44387-026-00089-z

Beyond metaphors: rethinking metaphors in metaheuristics algorithm design

Christian L. Camacho-Villalón, Thomas Stützle & Marco Dorigo

Although research in metaheuristics has become increasingly automated, considerable effort is still devoted to identifying new sources of inspiration for the manual design of so-called metaphor-based algorithms. In this work, we critically examine the use of metaphor-based design and its ongoing influence on the field of metaheuristics. We argue that manually designing algorithms based on metaphors belongs to the past, and that the field must move towards a more scientific and automated future.

Metaheuristics are high-level optimization frameworks that can be used to define problem-specific algorithms that provide "good" solutions to various optimization problems with relatively few adaptations. Due to their generality, flexibility, and reasonable runtime, metaheuristics are the method of choice for tackling complex optimization problems in real-world settings when exact methods fall short.

Historically, the design of new metaheuristic algorithms has been a manual process, guided by the experience and intuition of algorithm developers who use rules of thumb, common sense and mathematical principles to define consistent algorithmic frameworks that can be applied to different types of optimization problems. In a few cases, developers have also looked to nature for inspiration to design new metaheuristics—the so-called metaphor-based design approach. Although research in metaheuristics has moved towards higher levels of automation, and the approach of manually designing algorithms based on intuition and inspiration has been shown to be inefficient for meeting modern performance requirements, there are a significant number of researchers who are still trying to find new sources of inspiration for manually designing metaphor-based algorithms. Here, we discuss some of the effects, positive and negative, that metaphor-based design has had on the field of metaheuristics. We argue for a more scientific and automated future for the field, and present effective alternatives to manual design that can help us achieve these goals.

The metaphor rush: when inspiration turns into imitation

The use of natural and artificial metaphors in the design of optimization algorithms has had a profound impact on the field of metaheuristics. In the early stages of metaheuristics research, inspiration from natural phenomena was innovative because it allowed researchers to move beyond classical mathematical models and explore flexible, adaptive strategies for solving complex optimization problems. To this day, evolutionary algorithms[1–5], simulated annealing[6,7], and ant colony optimization[8,9] are examples of how inspiration from natural biological evolution, thermodynamics, and the foraging behavior of some ant species sparked the creation of truly novel optimization methods. These early metaphor-based algorithms introduced fundamentally new concepts that advanced both the theory and application of optimization. Although metaphor-based design initially offered intuitive appeal and provided developers with a framework to come up with new ideas, this successful approach gradually gave way to a problematic trend: the proliferation of metaphor-centric algorithms that lacked a substantial scientific basis[10,11].

Over the past two and a half decades, we have witnessed a constant flood of so-called "novel" optimization algorithms that rely on superficial metaphors without contributing any new algorithmic insights. From "intelligent water drops" to "cuckoos laying eggs", the metaheuristics landscape has become populated by algorithms inspired by almost every conceivable behavior—natural, artificial, and even supernatural—to the point where pseudoscience seems to have a place in the field, demonstrating poor scientific housekeeping and putting the reputation of the field at risk.

Researchers concerned by these issues have been conducting efforts to counteract the publications of papers proposing metaphor-centric algorithms. These efforts, which range from formal analyses of widespread "novel" metaphor-based algorithms to the publication of critical papers intended to create awareness in the field, have had a limited impact, as evidenced by the fact that the trend continues with increasingly far-fetched metaphors. To try and stop this trend, an open letter titled *Metaphor-based metaheuristics, a call for action: the elephant in the room* was recently published in the Swarm Intelligence journal[11]. In the letter, the authors address the editors of other scientific journals and ask them to establish guidelines to detect and stop the publication of papers with no intrinsic scientific value. Unfortunately, this open letter—signed by more than 90 researchers in the field of metaheuristics and AI—was not enough to reverse the trend, and the number of so-called "novel" metaphor-based algorithms published in the scientific literature has continued to grow steadily ever since.

Critical analyses: revealing the illusion of novelty

Formal analyses of highly cited metaphor-based algorithms have shown that the use of metaphorical language often serves to obscure the lack of genuine novelty in the underlying ideas. Although these algorithms may appear original on the surface, they are frequently little more than minor variations of well-established techniques. Furthermore, these algorithms tend to follow a familiar and repetitive pattern. Authors typically begin by describing some curious or visually striking behavior found in nature, society, artificial systems, or even imaginary worlds. They then construct a loose analogy between the observed behavior and an optimization process, introducing a new set of terminology that closely mimics the chosen metaphor—"wolves", "nests", "musicians", "zombies", and so on. Finally, they compare their new algorithm against outdated or poorly tuned techniques on unchallenging benchmarks and conclude that their metaphor-inspired approach outperforms existing optimization algorithms. The trend towards metaphor-based algorithms is driven less by scientific discovery than by the pressure to publish work that merely appears novel. As a result, over 500 of these metaphor-based algorithms have been published, cluttering the literature and creating confusion among newcomers to the field[12,13].

As an example, consider the gray wolf optimizer, which claims to mimic the hunting strategy and hierarchical social organization of gray wolves. Terms like "alpha", "beta" and "delta" wolves are introduced to describe the higher-quality solutions in the population. But when we look at its internal mechanisms, we find that the algorithm simply computes centroids and biases movements toward them—a behavior remarkably similar to particle swarm optimization[14]. Another well-known example is cuckoo search, which borrows terminology from the parasitic nesting behavior of cuckoo birds. "Eggs", "nests", and "flight behaviors" are mapped onto standard optimization procedures. However, a closer examination of this algorithm reveals that it is essentially a variation of evolution strategies with a differential evolution algorithm component, offering no new strategies or advantages beyond what was already available in the literature[15]. These are not isolated cases: several studies have examined "novel" metaphor-based algorithms—such as harmony search, biogeography-based optimization, gravitational search algorithm, black hole optimization, intelligent water drops, moth-flame optimization algorithm, whale optimization, firefly algorithm, bat algorithm, antlion optimizer, chimp optimization algorithm, grasshopper optimization algorithm and salp swarm optimization—and concluded that they are either trivial modifications or outright rebranding of existing methods, with metaphors that lack any solid grounding in optimization theory, scientific principles, or even reality. Even more troubling, however, is the widespread use of poor experimental practices in these papers—including unfair comparisons, biased testbeds, and a lack of reproducibility[16,17].

Finally, it should be noted that, even when a behavior observed in natural or artificial systems is rigorously abstracted and scientifically justified, the resulting method may still turn out to be essentially equivalent to previously proposed algorithms. A notable example is biogeography-based optimization[18], which was later shown by its authors to be a generalization of a genetic algorithm with global uniform recombination[19].

Meaningful innovation through automation

We believe that the field of metaheuristics must move beyond manual design, thereby rendering metaphor-based approaches obsolete. While drawing inspiration from other fields of knowledge is not inherently problematic, it must be coupled with rigorous methodology and innovation to yield algorithms that are effective. It is also essential to ensure that scientific principles are systematically applied in practice. Unfortunately, manual design—relying heavily on the intuition of human algorithm designers—makes it extremely difficult to achieve these goals. In addition to the widespread misuse of metaphors, the manual design of metaheuristics suffers from several well-known drawbacks: it is time-consuming, error-prone, and often leads to underperforming algorithms; moreover, it is a subjective, biased, and poorly documented process in which design choices are not always easy to understand and justify retrospectively.

One promising alternative to manual design is the use of automatic design methods, which treat algorithm creation as an optimization problem in itself[20,21]. These methods explore vast design spaces of algorithmic components by leveraging search strategies, learning mechanisms, and empirical feedback to discover effective combinations. By automating both the design and configuration of metaheuristics, we can reduce human bias and uncover algorithm designs that may not be apparent even to experienced practitioners[22]. In the automatic design paradigm, three main components are typically involved: (i) metaheuristic software frameworks, which provide algorithm components as modular and independent software units; (ii) algorithm design templates, which define the rules for combining these components into complete algorithms; and (iii) automatic configuration tools, such as irace, SMAC and ParamILS, which are used to search the design space and optimize algorithm performance.

While metaheuristic software frameworks and algorithm templates provide researchers with a clear set of design choices aligned with established practices in the literature, automatic configuration tools enable the automated exploration of numerous designs by combining metaheuristic components, fine-tuning their parameters, and evaluating configurations across diverse problem sets. Metaheuristic software frameworks such as ParadisEO, HeuristicLab, and jMetal offer modular environments that support both traditional and machine learning-based optimization[22]. Among the growing number of approaches for automating the creation of metaheuristic algorithms, automatic design has proven particularly effective in producing state-of-the-art implementations.

Many of these advances are closely tied to developments in machine learning. Indeed, machine learning offers a powerful set of tools for advancing the field of metaheuristics[23,24]. Techniques such as reinforcement learning, surrogate modeling, and automated parameter configuration can enhance both the applicability and the performance of metaheuristics. Rather than starting from predefined algorithm design or searching for new metaphors, we can leverage data to address key questions: What patterns emerge during the optimization process? Which strategies are most effective for solving a particular class of problems and under what conditions? How can we adapt dynamically to new problem features by modifying algorithm components–or even the algorithms themselves—within a metaheuristic portfolio? There are multiple levels at which machine learning can be integrated into the design of metaheuristics:

- Problem-level: using machine learning to model objective functions or to characterize fitness landscapes.
- Algorithm-level: selecting or designing entire algorithms based on historical performance data.
- Component-level: selecting and configuring algorithmic building blocks based on past data.

Together, these approaches enable the development of optimization algorithms that are not only more effective, but also more interpretable, reproducible, and adaptable[25,26].

Another recent approach is the use of large language models (LLMs) to automate the design of metaheuristics. One example is the LLM Evolutionary Algorithm Framework (LLaMEA), which leverages generative pre-trained transformers (GPTs) to evolve black-box optimizers[27]. LLaMEA has produced competitive algorithms that outperform state-of-the-art baselines. Another example is the multi-objective evolution of heuristics (MEoH), which employs LLMs in a zero-shot fashion to generate diverse, Pareto-efficient heuristics that balance multiple design criteria, such as performance, efficiency, and scalability[28]. Recent work by Senkerik[29] further shows that both closed- and open-weight LLMs can generate effective algorithms for engineering optimization problems, demonstrating the practical viability of LLMs across model families such as `GPT-4o`, `GPT-4o Mini`, `Claude Sonnet 3.5`, and `Llama 3.1`. Together, these studies provide broader evidence that LLMs can not only refine existing algorithmic steps but also design new ones, acting as both optimizers and designers of optimization algorithms[30].

Toward the scientific era of metaheuristics

We have argued before for steering the field of metaheuristics towards a more scientific and automated future[11,22], a position also taken by others in the community. One such initiative is the Metaheuristics "in the Large" community project[31], which advocates the use of standardized abstractions and common protocols to enable: (i) modular algorithm composition from reusable components, (ii) large-scale comparisons that yield analytical insights to guide metaheuristics design and selection, and (iii) cumulative scientific progress through service-oriented architectures that support the broad reuse of data and software. Recently proposed component-based frameworks bring this vision into practice by providing scaffolding for automation. For example, MAHF, implemented in `Rust`, offers uniform interfaces for components and conditions, with a shared blackboard state for communication, explicitly targeting components combinability and experimental analysis[32]; MetaGen provides a standardized `Python` interface for solution representations and operators, simplifying the rapid assembly and benchmarking of metaheuristic across problem classes[33]; and METAFOR, implemented in `C++`, can be coupled with automatic algorithm configuration tools to support modular hybridization of different metaheuristic paradigms, thus enabling users to generate families of hybrids tailored to specific problem classes[34].

In this paper, we emphasize the next steps in the foundation of the automated generation, selection, and evaluation of metaheuristic implementations via ML- and LLM-driven approaches, thereby unifying the Metaheuristics "in the Large" perspective with the automation we advocate. However, to achieve this goal, we must critically examine the current state of the field. While the metaphor-based approach initially sparked creativity and progress, its overuse has led to a crisis of confusion, diminished credibility, and, at times, pseudoscience. The publication of hundreds of metaphor-centric algorithms has fragmented the literature, misled newcomers, and hindered meaningful progress. To move forward, we must prioritize scientific integrity over novelty for its own sake. Although metaphors can still be valuable, their contribution is likely to remain marginal. This is especially true in a field that is now rapidly advancing through automation and increasingly converging with other branches of artificial intelligence. In our view, the future lies in data-driven, systematically automated, and machine-learning-based approaches. These paradigms enable the automatic creation, configuration, and evaluation of high-performing metaheuristics while significantly reducing reliance on manual, human-guided design.

Data availability

No datasets were generated or analyzed during the current study.

Christian L. Camacho-Villalón[1,2] ✉, Thomas Stützle[2] & Marco Dorigo[2]
[1]Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia. [2]Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA), Université Libre de Bruxelles, Bruxelles, Belgium. ✉e-mail: christian.camacho.villalon@ijs.si

Received: 13 June 2025; Accepted: 24 February 2026;
Published online: 11 March 2026

References

1. Fogel, D. B., Owens, A. J. & Walsh, M. J. *Artificial Intelligence Through Simulated Evolution* (John Wiley & Sons, 1966).
2. Holland, J. H. *Adaptation in Natural and Artificial Systems* (University of Michigan Press, 1975).
3. Rechenberg, I. *Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution* (Frommann-Holzboog, 1973).
4. Schwefel, H.-P. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie* (Birkhäuser, 1977).
5. Schwefel, H.-P. *Numerical Optimization of Computer Models* (John Wiley & Sons, 1981).
6. Kirkpatrick, S., Gelatt, C. D. J. & Vecchi, M. P. Optimization by simulated annealing: quantitative studies. *J. Stat. Phys.* **34**, 975–986 (1984).
7. Černý, V. A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J. Optim. Theory Appl.* **45**, 41–51 (1985).
8. Dorigo, M., Maniezzo, V. & Colorni, A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B* **26**, 29–41 (1996).
9. Dorigo, M. & Stützle, T. *Ant Colony Optimization* (MIT Press, 2004).
10. Sörensen, K. Metaheuristics-the metaphor exposed. *Int. Trans. Oper. Res.* **22**, 3–18 (2015).
11. Aranha, C. et al. Metaphor-based metaheuristics, a call for action: the elephant in the room. *Swarm Intell.* **16**, 1–6 (2022).
12. Campelo, F. & Aranha, C. Evolutionary computation bestiary. Version visited 26 Sep 2025. https://github.com/fcampelo/EC-Bestiary (2016).
13. Ma, Z., Wu, G., Suganthan, P. N., Song, A. & Luo, X. Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms. *Swarm Evol. Comput.* **77**, 101248 (2023).
14. Camacho-Villalón, C. L., Dorigo, M. & Stützle, T. Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: six misleading optimization techniques inspired by bestial metaphors. *Int. Trans. Oper. Res.* **30**, 2945–2971 (2023).
15. Camacho-Villalón, C. L., Dorigo, M. & Stützle, T. An analysis of why cuckoo search does not bring any novel ideas to optimization. *Comput. Oper. Res.* **142**, 105747 (2022).
16. García-Martínez, C., Gutiérrez, P. D., Molina, D., Lozano, M. & Herrera, F. Since CEC 2005 competition on real-parameter optimisation: a decade of research, progress and comparative analysis's weakness. *Soft Comput.* **21**, 5573–5583 (2017).
17. Kudela, J. A critical problem in benchmarking and analysis of evolutionary computation methods. *Nat. Mach. Intell.* **4**, 1238–1245 (2022).
18. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **12**, 702–713 (2008).
19. Simon, D., Rarick, R., Ergezer, M. & Du, D. Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. *Inf. Sci.* **181**, 1224–1248 (2011).
20. López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Stützle, T. & Birattari, M. The irace package: Iterated racing for automatic algorithm configuration. *Oper. Res. Perspect.* **3**, 43–58 (2016).
21. Stützle, T. & López-Ibáñez, M. Automated design of metaheuristic algorithms. In *Handbook of Metaheuristics*, vol. 272 of *International Series in Operations Research & Management Science* (eds Gendreau, M. & Potvin, J.-Y.) 541–579 (Springer, New York, 2019).
22. Camacho-Villalón, C. L., Stützle, T. & Dorigo, M. Designing new metaheuristics: manual versus automatic approaches. *Intell. Comput.* **2**, 0048 (2023).
23. Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, M. & Talbi, E.-G. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: a state-of-the-art. *Eur. J. Oper. Res.* **296**, 393–422 (2022).
24. Talbi, E.-G. Machine learning into metaheuristics: a survey and taxonomy. *ACM Comput. Surv.* **54**, 1–32 (2021).
25. Kostovska, A. et al. Using machine learning methods to assess module performance contribution in modular optimization frameworks. *Evol. Comput.* **32**, 1–28 (2024).
26. Nikolikj, A., Munoz, M. A. & Eftimov, T. Benchmarking footprints of continuous black-box optimization algorithms: explainable insights into algorithm success and failure. *Swarm Evol. Comput.* **94**, 101895 (2025).
27. van Stein, N. & Bäck, T. Llamea: A large language model evolutionary algorithm for automatically generating metaheuristics. *IEEE Trans. Evol. Comput.* **29**, 331–345 (2025).
28. Yao, S. et al. Multi-objective evolution of heuristic using large language model. *Proc. AAAI Conf. Artif. Intell.* **39**, 27144–27152 (2025).
29. Senkerik, R. et al. Open and closed-source models for LLM-generated metaheuristics solving engineering optimization problem. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* 372–385 (Springer Nature Switzerland, Cham, 2025).
30. Liu, F. et al. A systematic survey on large language models for algorithm design. *ACM Comput. Surv.* **58**, 1–32 (2026).
31. Swan, J. et al. Metaheuristics "in the large". *Eur. J. Oper. Res.* **297**, 393–406 (2022).
32. Stegherr, H., Luley, L., Wurth, J., Heider, M. & Hähner, J. *A Framework for Modular Construction and Evaluation of Metaheuristics*. Technical report. https://opus.bibliothek.uni-augsburg.de/opus4/103452 (University of Augsburg, 2023).
33. Gutiérrez-Avilés, D., Jiménez-Navarro, M. J., Torres, J. F. & Martínez-Álvarez, F. Metagen: a framework for metaheuristic development and hyperparameter optimization in machine and deep learning. *Neurocomputing* **637**, 130046 (2025).
34. Camacho-Villalón, C. L., Dorigo, M. & Stützle, T. Metafor: a hybrid metaheuristics software framework for single-objective continuous optimization problems. arXiv preprint arXiv. https://arxiv.org/abs/2502.11225 (2025).

Acknowledgements

C.L.C.V. acknowledges support from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Postdoctoral Fellowship Programme, SMASH co-funded under the grant agreement No 101081355. The operation (SMASH project) is co-funded by the

Republic of Slovenia and the European Union from the European Regional Development Fund. T.S. and M.D. acknowledge support from the Belgian F.R.S.-FNRS of which they are research directors.

Author contributions

C.L.C.V. drafted the initial manuscript. T.S. and M.D. supervised the work and offered guidance throughout the writing process. All authors critically reviewed, edited, and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Christian L. Camacho-Villalón.