Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Real Application Challenges in Evolutionary Optimization? People!

Tobias Rodemann, Christiane Attig

2025

Preprint:

This is a post-peer-review, pre-copyedit version of an article published in EvoApplications2025. The final authenticated version is available online at: https://doi.org/10.1007/978-3-031-90062-4_29

Real Application Challenges in Evolutionary Optimization? People!

Tobias Rodemann[1 [0000−0001−6256−0060]] and Christiane Attig[2 [0000−0002−6280−2530]]

[1] Honda Research Institute Europe
Carl-Legien-Strasse 30
63073 Offenbach/Main
Germany
tobias.rodemann@honda-ri.de

[2] Institut für Multimediale und Interaktive Systeme
Universität zu Lübeck
Ratzeburger Allee 160
D-23562 Lübeck
Germany
christiane.attig@uni-luebeck.de

Abstract. The application of evolutionary optimization methods for real world problems is often far less straight-forward than expected. One of the main challenges are the people involved in real business situations that decide on whether optimization projects are successful or not. In this work we present a few insights from 20+ years of applying Evolutionary Algorithms (mostly multi- and many-objective) with in-house customers and point to some key psychological insights that can explain several of the major issues that we encountered in our work but are also reported repeatedly in application sessions on major conferences. We argue that more convincing sales messages are needed; and explain, why trust, not better objective values, might be the ultimate goal of any optimization process. We can't provide numbers or new algorithms, but maybe a different perspective on some of the most persistent practical challenges.

Keywords: Application, Evolutionary Algorithms, User Interaction, Many-objective optimization

1 Introduction

Application sessions in conferences on evolutionary algorithms (EAs) like EvoStar, GECCO, or CEC often appear to be separated from other sessions, in the sense that few of the big topics in fundamental research are actually represented in application work and many of the practical challenges are not addressed in research. In this work we want to outline some of the problems we experienced many times in our own projects. The reason these problems are not properly discussed in EA research is that they are not computer science related but social

and psychological in nature. Real-world problems are not just characterized by different objective functions but by the need to closely interact with different groups of people. In this paper we want to look at the three main groups in an optimization project:

1. The optimization specialists and how their job is different in industrial projects compared to academic projects.
2. The application team who provide application knowledge, had the idea for the optimization project, and who need to specify the details of the project.
3. The decision maker who needs to be convinced to finance the optimization project and to realize the results of the optimization process.

We will argue that the current trends in research do not improve but rather weaken the communication between these groups and make an efficient cross-group cooperation more difficult. The main practical aspects we want to address in this work are:

1. In most real world problems, there is initially no clear definition of the project target. Most of the time and budget needs to go into extracting the actual task from the application specialists, leaving little time for testing and tuning many algorithms.
2. Commercial optimization specialists are orders of magnitude more expensive than students in academic projects. Many optimization projects struggle to finance even the costs of the optimization experts.
3. Some managers require a good explanation for choices made in optimization projects, such as why a certain optimization algorithm was chosen. The explosive growth of new algorithms especially for many-objective optimization has made any justification for a specific algorithm very challenging. Increasingly complex methods (for example the integration of Machine Learning methods in surrogates [13, 9]) further increase the difficulties.
4. Decision makers typically not only consider numerical optimization results, but also require a certain level of trust in the methods and any choices made. Building this trust is often ignored and leads to mutual frustration at the end of the project when optimized solutions might not be implemented, due to low trust in the way they were generated.

Some of the issues above are actually well known from psychological research as fundamental human characteristics that need to be considered in any interaction. One of the key aspects is trust—without trust, no optimization project can be successful. Unfortunately, how to build and maintain trust has largely been ignored in computer science so far. Cognitive sciences have identified transparency and explainability as key elements for building trust. It is also known that there is threshold for the maximum cognitive load decision makers can handle and the negative impact on rational decisions and trust that results from exceeding this limit.

This article is based on a keynote speech presented at the IEEE WCCI conference 2024 in Yokohama, with several additions and modifications based on audience and reviewer comments.

2 Psychological Background

From the perspective of cognitive psychology, the optimization process can be understood as a complex human problem-solving process [14]: The status quo has been identified as an undesirable initial state by the application team, then the optimization team chooses from a number of operators (i.e., algorithms) that transform the initial state into the goal state. However, there exist barriers that impede the smooth transformation process from initial to goal state: (1) The precise goal state is unclear (i.e, ill-defined) or unknown to the optimization team, (2) the choice of operator is challenging due to the high and ever-growing number of applicable algorithms (i.e., the problem space is too vast to be fully comprehensible, leading to the necessity of using heuristics to avoid cognitive overload [21]), and (3) the algorithm selection is intransparent, likely causing low trust in the optimization process by the decision maker (see Fig. 1). A similar set of issues arises for the way the selected algorithm works and produces final candidates.

Solutions to ill-defined problems such as the described optimization problem typically are not correct or incorrect but fall on a spectrum from low to high acceptability [10, 3]; moreover, solutions are context-dependent and need to be evaluated (e.g., through simulations). Consequently, the described optimization problem is characterized by at least three socio-cognitive challenges beyond algorithm quality: Insufficient goal state representation, risk of cognitive overload due to an overstraining number of possible solutions, and low trust in the optimization process. What is more, on a purely social level, communication challenges in the social interaction between the three involved groups can further complicate the optimization process during all stages.

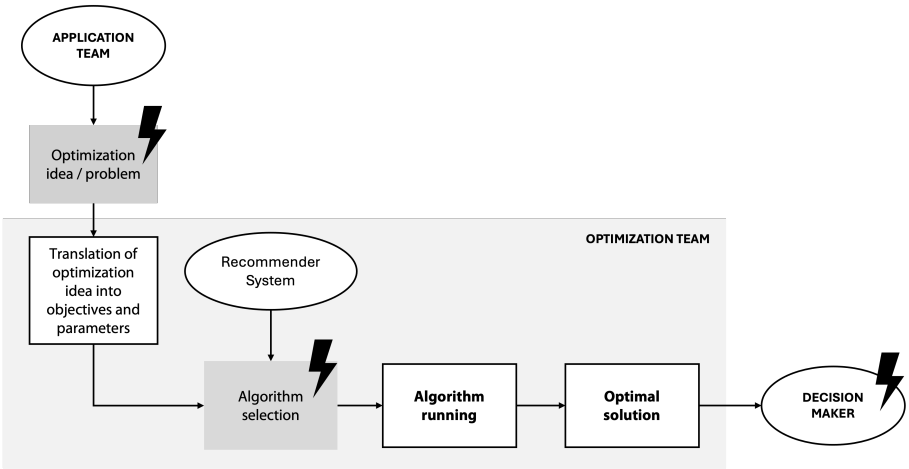

Fig. 1. Protoypical optimization process and socio-cognitive challenges in real-world applications.

In the following, we will briefly describe some of the socio-cognitive challenges with respect to the decision maker's information processing in more detail. To this end, we utilized classical control-theoretic models (e.g., [2]) to describe the psychological mechanisms in the decision maker that impede smooth optimization processes (see Fig. 2). Imagine the optimization team presenting their results (i.e., final objective values) to the decision maker. The decision maker first has to perceive and understand this information to be able to compare the technical output with the initial optimization goal (Feedback Loop 1). Only if the final objective values are sufficiently close to the optimization goal, the decision maker might decide to make investments. However, even if the reported objective values and the optimization goal are aligned, investment decision can be impeded or suspended. In fact, the decision maker not only evaluates the reported numerical values, but also the optimization process on a higher-order level (regarding e.g., number and comprehensibility of presented output values, length of the optimization process, comparison with prior optimization processes). Hence, the decision maker compares their expected optimization process performance with the actual optimization process (Feedback Loop 2). The result of this comparison is a certain subjective experience. For instance, if the presented technical output is too extensive or not adapted to the decision maker's state of expertise, cognitive overload and frustration might occur. In addition, the decision maker evaluates the traceability of the optimization process (i.e., algorithm selection and technical output; Feedback Loop 3). If they perceive the traceability as too low—maybe because the algorithm selection is based on past projects and not on the objective best fit to the optimization problem (see Section 4.3)—trust in the optimization process or team can be impaired.

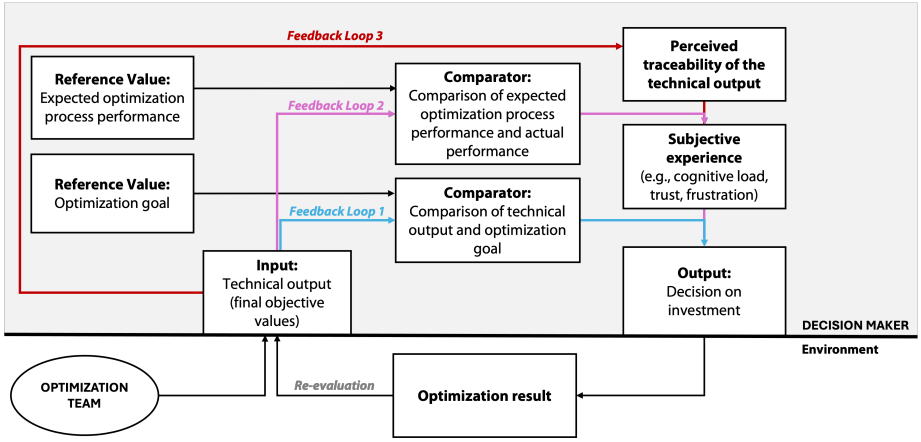

Fig. 2. Control-theoretic framework depicting psychological mechanisms in the decision-maker after presentation of the technical output by the optimization team.

Finally, the negative subjective experience of the decision maker may increase the likelihood of using heuristics that can delay or even suspend the investment decision [19]. Just to mention two well-known examples: The *feeling-as-information heuristic* [20] describes the behavioral tendency to use affective reactions (e.g., frustration, annoyance) as a basis of judgment. Furthermore, the *status quo bias* [5] is the tendency to stick with the status quo (i.e., the undesirable state from which the optimization idea arose in the first place) even though other options are feasible and better (as demonstrated by the optimization team). Particularly in instances with low trust in the optimization process, the status quo bias increases in probability because human decision makers are usually more driven to avoid losses than to make equal-sized gains [6]. When accepting the inherent socio-cognitive aspects of an optimization project, it becomes clear that a deep understanding of human psychology and cognitive limits is essential to improve the chances of a successful project.

3 Referenced Application Work

Our research group has worked intensively on different types of optimization problems in the past more than 20 years with company internal customers from different divisions. In this article we will focus on two key application areas, evolutionary design optimization (see for example [11]) and many-objective optimization of energy systems (equipment sizing optimization), see e.g., [16]. While from an optimization perspective the two applications are not too different they had a very different project setting.

In the design optimization one of the main challenges was a proper representation of shapes (e.g., turbine blades) that provides a high flexibility with few parameters. Another challenge was the evaluation of potential solutions via computational fluid dynamics (CFD) simulations. Especially in the beginning simulators had their problems and engineers did not fully trust the results. The road to success was a long-term commitment on both sides, slowly building trust in both simulation as well as optimization methods. We also put a focus on software transfer, maintenance and upskilling to ensure a long-term usability.

For the energy system optimization, the design parameters are obvious (dimensions of potential hardware purchases), but again the simulation was a cause of concern. In this application validating optimized solutions beyond a simplified simulation is very difficult and trust becomes an ever bigger issue. Even technically very reasonable results (simulations indicate strong and robust positive return on investment) did not lead to quick decisions, partially due to limited trust in the simulation and optimization process. We already elaborated on this in a prior article [18].

4 Overview of Challenges

This article has been motivated by a few occasions where cooperation partners asked questions or made comments that were difficult to answer or caused some

consternation. As with many problems of this type one tends to initially see this as distraction from the real challenges but later realizes the fundamental issues behind those questions. The remainder of this article will elaborate more on these topics, trying to explain the social issues and the people involved. In this article we want to focus on the following real challenges that we encountered:

1. Cognitive overload in multi-objective optimization: Most (not all) people don't like Pareto fronts, they don't want to choose from 100s of solutions. The extra degrees of freedom provided by a multi-objective optimization are quite often seen as a problem not a flexibility. One potential reason is the additional selection required for the decision maker (which of the many Pareto set solutions are selected). Quite obviously adding more objectives will further complicate matters.
2. Getting decisions: Requesting decisions from partners outside their domain of expertise is generally difficult, but even when staying inside the application domain, getting decisions might be challenging and lead to lower trust in the results. One example is the priorization of multiple objectives, for example as weights or reference vectors. In our experience many customers don't think in terms of explicit trade-offs and don't know how to express these in concrete numbers.
3. Concise optimization project summary: "Please summarize the results of your project in one line". In project reporting one typically needs to compress the results of the project into a few words (like "performance improved by X%"). For multi- and especially many-objective optimization the typical performance indicators like hypervolume or IGD are not well suited as they are hard to interpret.
4. Justification of algorithmic choices: "Why did you choose this method?". Decision makers often require a justification for the choice of algorithms. Especially in many-objective optimization there are so many algorithms (the popular PlatEMO [23] tool contains more than 270 EMOAs) that any choice is somewhat arbitrary, so it is difficult to come up with a good justification for any selection made.
5. Maintainability: In many cases, the project can only be called a success, if the customer is able to repeatedly employ the used optimization methods, perform minor adaptations and correctly interpret the results. Beyond software-related aspects (like DevOps) also know-how transfer has to be organized, considering the available skill-set at customer side.

The key underlying theme in most of these issues is **trust**. In the end, the decision maker, who is typically not an expert for optimization, has to evaluate not only the technical outcome of the project (the final objective values) but also the trust in the overall process. It is known [1] that trust is closely connected to transparency and explainability, meaning the ability to understand how optimal solutions were found (the process level) and why these solutions are good, respectively. These factors are increasingly considered in the field of ML and will also require more attention in EAs.

One issue often ignored is the problem of cognitive load or cognitive capacity like Miller's 7 ±2 rule [12] (a limited short-term memory capacity of around 7 items, for example objective values). Any decision setting that exceeds the cognitive limit of the decision maker (like selecting a subset of solutions out of a Pareto set with 100s of solutions and many objectives each), could trigger low-level psychological processes like fundamental decision heuristics (risk aversion, status-quo bias, etc) as described above and diminish the trust in the process. On the other hand, trust can be transferred to and from other human beings, like domain experts. Convincing an independent domain expert might be easier than convincing the final decision maker.

4.1 Project Initialization

In the academic world the optimization problem is assumed to be precisely specified, with given design variables including valid ranges, a mathematical objective function, definition of constraints, etc.

In practice, the problem is often only explained in application terminology. This means that the underlying specs of the optimization tasks need to be extracted in discussions with the application team, often leaving many aspects undefined. For example, after an initial optimization run, additional constraints or objectives might surface that were initially not considered. A substantial communication effort is required for the optimization specialist to collect all information before the "real" optimization can start.

A further problem is that in many projects the actual high-level target of the project is not well defined or not shared among team members. For example, the application team might be interested in seeing new (novel [8]) solutions as inspirations, while the optimization expert is focusing on improving objective values. It is therefore essential to clarify early on which part of the optimization results is used by whom.

Because the specification of the project objectives requires a decent amount of time, less time is available for tuning hyperparameters and benchmarking different algorithm candidates. As a result it seems that many practitioners only use a few standard algorithms out of the box.

4.2 Budget

Most optimization algorithms are developed and benchmarked by (groups of) students in university settings. In this specific situation, personnel costs are not a concern. In a business environment however, optimization experts need to be hired which can charge substantial fees, easily exceeding the monthly costs of university students in a single day.

Being able to quickly translate the business challenge into a well-defined optimization problem is essential otherwise the customer will run out of budget before the actual optimization has even started. This means that large-scale algorithmic exploration, especially if a manual supervision is required, is out of the question. Since most business applications have non-trivial fitness functions

(expensive function evaluations), only few runs with few algorithms and standard hyperparameter settings can often be done.

As argued in [15], algorithm portfolio management is also costly. Even though increasingly more algorithms are available for download, ensuring that the software is bug-free and validating the results on relevant benchmarks costs a lot of time. With hundreds of new algorithms being proposed annually (see for example the list of implemented methods in [23]), keeping up-to-date with recent research developments is next to impossible.

In addition to the factor time also the computing budgets are limited. Academic research can often access high-performance computing hardware for free, while in industry expensive cloud services might have to be procured. This means that expert and budget availability in academia can be very different from industry, leading to different preferences especially for the exploration / exploitation trade-off.

Developers of new optimization algorithms should also consider the effort to understand, implement, and tune the algorithm. Marginal performance gains at the cost of massively increased algorithm complexity might need to be reconsidered.

4.3 Choices

Due to the large number of algorithms proposed by the academic community and the increasing number of hyper-parameters for more and more complex algorithms, many choices have to be made. These choices can be hard to motivate. Let's take as an example the choice of a (many-objective) optimization algorithm. Different motivations come to mind, from best to worst:

1. **Full comparison:** "All existing algorithms have been compared on the problem function and the best one chosen". As outlined above this is next to impossible.
2. **Only fitting algorithm:** In some cases there is only one algorithm which can handle the specific requirements of your project, but in general there should be many candidate algorithms.
3. **Newest algorithm:** "The latest algorithm was used". Again very difficult, as algorithms are proposed faster than they can be tested. Also newer algorithms are not guaranteed to be better (see [17]).
4. **Everybody uses this algorithm:** Not very likely, and there is no process to check. Most academic research is based on self-made algorithms, while applied work frustratingly often uses dubious metaphor-based algorithms like the Grey Wolf (see [22] for a critical analysis) (note that this algorithm has been cited more than 15,000 times as of September 2024).
5. **Benchmark winner:** The algorithm performed well on recent benchmark competitions. One of the best options, but the test functions are probably unrelated to the application problem and only a small subset of (order of 10 vs order of 100 or 1000) algorithms has been compared.

6. **Reputation:** "The algorithm was developed by a prestigious research group in a famous university". In practice, one of the best arguments (trust in an algorithm is taking from trust in experts and institutions).
7. **In-house development:** "We developed the algorithm ourselves and we believe it to be really well designed". Again this is replacing performance evaluation by trust in people.
8. **Tested for application:** The same algorithm has been used before for the same application (but often without a comparison to many algorithms). We found this often in application papers that used the same optimization algorithm that others used on the same domain before.
9. **Past projects:** The algorithm worked well in past projects and so we use it again. The obvious question is why the algorithm was chosen in the prior work.
10. **Marketing impact:** "It is quite famous! And featured in a Youtube video!". We hope that the minimal scientific merit of this statement does not require further explanation.
11. **Fatalism:** "The choice of algorithms does not matter anyways!" This is probably not true and would be very hard to digest leading to the question which problem the mass of publications on optimization algorithms is addressing.
12. **Easy to get:** "We could download it for free from the internet". This is unfortunately not so rare as a motivation, especially for users unaware of the number of alternative algorithms.

Without a good motivation for key algorithmic choices, decision makers might be hesitant to trust the results of the optimizer. If you think this is exaggerated, consider that multi-million Euro investments or human lives might depend on the output of the optimizer. Not being able to concisely argue for key algorithmic decisions can be a major obstacle. Unfortunately, even for articles in high-quality journals, the choice of algorithms for benchmarking is rarely motivated concisely.

Having well-established best practice solutions (standard algorithms to try first for an unknown application problem, accepted by the scientific community) would provide a stronger justification.

Obviously, if it is possible to cheaply and reliably assess the quality of an optimized solution and when reference solutions (like current best) are available, trust can be built directly from the evaluation and comparison to other methods. But in many cases, evaluating a solution might require building expensive prototypes (for example jet turbines) or is only feasible by building the final system (as for renewable energy systems).

4.4 Many-objective Optimization & Performance Indicators

One of the hottest topics in evolutionary algorithms in the past decades is many-objective optimization (MAO). The potential to consider a larger set of objectives and analyze the trade-off between those objectives is very appealing. In our experience, however, customers' enthusiasm was limited. MAO, especially

for more than two objectives, is hard to understand and selecting one of the Pareto set solutions more difficult than expected.

A special problem is performance indicators. In engineering projects numerical progress indicators are highly valued. For single objective optimization one can simply use the objective value (or some indicator derived from that). This is not possible for MAO, so special performance indicators like Hypervolume (HV) or IGD are used. For a computer scientist, these indicators are relatively easy to understand (but contain a number of non-obvious issues that might affect algorithmic performance, see e.g. [4]). For the application team and decision makers this is far more difficult and often outright confusing.

Let's imagine we want to optimize an engine for costs (in Euro) and CO_2 emissions (in g/km). Both objectives are easy to understand individually, but using HV in MAO would generate a unit-free number, say initially 0.02 and improving to 0.3 over the course of the optimization. Understanding this value requires multiple steps of thought. Firstly, one needs to understand that HV does not rate individual solutions but the quality of a population. Secondly, all values are relative to a (somewhat) arbitrary reference point. This leads to numbers that have no interpretability any more, there is no physical unit attached, and HV should increase even when we want to decrease the individual objectives. The numerical improvement in HV is also hard to interpret.

A potential remedy is to use desirabilities [24] instead of objective values. One could then report for example an increase in average desirability for all objectives over the entire population.

4.5 Maintenance

When finally a good solution has been found and accepted by the decision maker, a new question might arise: How to make the optimization process available to the customer on a continuous basis? For example customers might want to repeat the optimization once new data arrives or constraints and objectives change.

Again, one needs to consider the involved people. Which skills are available in the team, what complexity of software, algorithms and visualization can they handle? Using very sophisticated methods to derive optimal solutions only to discover that the customer is unable to maintain the software (assuming that at least some minor changes might be required), is ultimately a failure. Here again, discussing the project to the very end at the beginning is a key to success.

The Machine Learning (ML) community has come up with a variety of tools like MLOps [7], that formalize some of the mentioned challenges and provide supporting tools. With increasing contributions from ML like surrogate models this might become more relevant.

5 Recommendations

As stated initially, real world optimization projects are an inherently social activity. It is essential to clarify right from the beginning, who the involved people

are. Especially, the decision maker can be opaque until quite late in the project. It is then necessary to clarify the skill level, goals, and interests of all involved people, identify required trade-off decisions (e.g., cost minimization vs. performance maximization) and propose an initial approach that aligns with all those constraints. There is very little research on the required cognitive skills for understanding different optimization approaches, for example single vs multi- or even many-objective optimization. It can be helpful to build a mock-up of the final optimization output (like a Pareto curve or a parallel coordinate plot) and check with the involved people, if this could be a useful project outcome. Simplifying the interpretation of results by for example using desirabilities should be discussed.

Another important topic is to clarify how optimization results can be validated. Is the objective function the real benchmark or just an approximative simulation? Can the results be measured independently (for example in a wind tunnel)? Maybe there is an expert who can judge the solutions derived from the optimizer based on years of practical, real-world experience to generate additional trust. A big drawback in evolutionary optimization is that in practice there are few guarantees that help to assess the current state of the optimization. Finding the optimal solution can't be guaranteed and even it is found, it can't be proven. Methods like MILP (Mixed Integer Linear Programming) provide theoretical guarantees and upper bounds, that help deciding on next steps. In evolutionary optimization, future progress is hard to estimate and it is unclear how much better solutions might get if additional resources are invested. For a customer, getting additional guardrails is very helpful.

For the academic community our suggestion is to invest more into understanding the psychological and cognitive aspects of optimization projects, working more closely with experts from these fields to improve transparency, explainability, and trust-related aspects. It would also be advisable from our experience to work more on the practical usability of optimization methods rather than excelling in benchmarks. Methods that are intuitive, with few elements and hyperparameters, and robust (good performance over many different problems rather than top performance in some) are of high practical relevance.

Finally, a community effort to define a set of good first algorithms to try on any unknown problem would help to increase trust.

Acknowledgments. The authors would like to thank the anonymous reviewers for invaluable feedback. Summarizing a variety of observations and ideas into a short conference paper turned out to be much more difficult than (we) anticipated. Based on the reviews we tried to clarify our line of argumentation, but want to apologize for any remaining hardships when reading this article.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Büscher, C., Sumpf, P.: "Trust" and "confidence" as socio-technical problems in the transformation of energy systems. Energy, sustainability and Society **5**, 1–13 (2015)
2. Carver, C.S., Scheier, M.F.: On the structure of behavioral self-regulation. In: Handbook of self-regulation, pp. 41–84. Elsevier (2000)
3. Goel, V.: Comparison of Well-Structured & Ill-Structured Task Environments and Problem Spaces. In: Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society (1992)
4. Guerreiro, A.P., Fonseca, C.M., Paquete, L.: The hypervolume indicator: Computational problems and algorithms. ACM Computing Surveys (CSUR) **54**(6), 1–42 (2021)
5. Johnson, J.G., Busemeyer, J.R.: Decision making under risk and uncertainty. Wiley Interdisciplinary Reviews: Cognitive Science **1**(5), 736–749 (2010)
6. Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. In: Handbook of the Fundamentals of Financial Decision Making, chap. Chapter 6, pp. 99–127. World Scientific (2013). https://doi.org/10.1142/9789814417358_0006
7. Kreuzberger, D., Kühl, N., Hirschl, S.: Machine learning operations (MLOps): Overview, definition, and architecture. IEEE access **11**, 31866–31879 (2023)
8. Lehman, J., Stanley, K.O.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary computation **19**(2), 189–223 (2011)
9. Liu, Q., Lanfermann, F., Rodemann, T., Olhofer, M., Jin, Y.: Surrogate-assisted many-objective optimization of building energy management. IEEE Computational Intelligence Magazine **18**(4), 14–28 (November 2023)
10. Lynch, C., Ashley, K.D., Pinkwart, N., Aleven, V.: Concepts, Structures, and Goals: Redefining Ill-Definedness. International Journal of Artificial Intelligence in Education **19**, 253–266 (2009)
11. Menzel, S., Olhofer, M., Sendhoff, B.: Application of free form deformation techniques in evolutionary design optimisation. In: Proceedings of 6th World Congress on Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil (2005)
12. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological review **63**(2), 81 (1956)
13. Pan, L., He, C., Tian, Y., Wang, H., Zhang, X., Jin, Y.: A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. IEEE Transactions on Evolutionary Computation **23**(1), 74–88 (2019). https://doi.org/10.1109/TEVC.2018.2802784
14. Quesada, J., Kintsch, W., Gomez, E.: Complex problem-solving: A field in search of a definition? Theoretical Issues in Ergonomics Science **6**(1), 5–33 (Jan 2005). https://doi.org/10.1080/1463922052000331311553
15. Rodemann, T.: Industrial portfolio management for many-objective optimization algorithms. In: 2018 IEEE Congress on Evolutionary Computation (CEC) (2018). https://doi.org/10.1109/CEC.2018.8477693
16. Rodemann, T.: A many-objective configuration optimization for building energy management. In: IEEE (ed.) WCCI 2018 Conference. IEEE (July 2018)
17. Rodemann, T.: A comparison of different many-objective optimization algorithms for energy system optimization. In: Kaufmann, P., Castillo, P. (eds.) Applications of Evolutionary Computation. pp. 1–16. LNCS 11454, Springer (April 2019)
18. Rodemann, T., Attig, C.: How can digital twins help to accelerate the transition to a carbon-neutral energy system? In: Lecture Notes in Computer Science. Springer, LNCS, LNAI, LNBI (2024)

19. Samuelson, W., Zeckhauser, R.: Status quo bias in decision making. Journal of risk and uncertainty **1**, 7–59 (1988)
20. Slovic, P., Finucane, M.L., Peters, E., MacGregor, D.G.: Risk as analysis and risk as feelings: Some thoughts about affect, reason, risk and rationality. In: The feeling of risk, pp. 21–36. Routledge (2013)
21. Slovic, P., Peters, E., Finucane, M.L., MacGregor, D.G.: Affect, risk, and decision making. Health Psychology **24**(4, Suppl), S35–S40 (2005). https://doi.org/10.1037/0278-6133.24.4.S35
22. Sörensen, K.: Metaheuristics—the metaphor exposed. International Transactions in Operational Research **22**(1), 3–18 (2015)
23. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Computational Intelligence Magazine **12**(4), 73–87 (2017). https://doi.org/10.1109/MCI.2017.2742868
24. Wagner, T., Trautmann, H.: Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. IEEE Transactions on Evolutionary Computation **14**(5), 688–701 (2010)