1

Research in Evolutionary Computation

This work tries to lay the groundwork for experimental research in evolutionary computation. We claim that experiments are necessary—a purely theoretical approach cannot be seen as a reasonable alternative. Our approach is related to the discipline of *experimental algorithmics*, which provides methods to improve the quality of experimental research. However, many approaches from experimental algorithmics are based on Popperian paradigms:

1. No experiment without theory.
2. Theories should be falsifiable.

Following Hacking (1983) and Mayo (1996), we argue that:

1*. An experiment can have a life of its own.
2*. Falsifiability should be complemented with verifiability.

This concept, known as the *new experimentalism*, is an influential discipline in the modern philosophy of science. It provides a statistical methodology to learn from experiments. For a correct interpretation of experimental results, it is crucial to distinguish the statistical significance of an experimental result from its scientific meaning. This work attempts to introduce the concept of the new experimentalism in evolutionary computation.

1.1 Research Problems

At present, it is intensely discussed which type of experimental research methodologies should be used to improve the acceptance and quality of *evolutionary algorithms* (EA). A broad spectrum of presentation techniques makes new results in *evolutionary computation* (EC) almost incomparable. Sentences like "This experiment was repeated ten times to obtain significant results" or "We have proven that algorithm A is better than algorithm B" can still be found in current EC publications. Eiben & Jelasity (2002) explicitly list four problems:

Problem 1.1. The lack of standardized test-functions, or benchmark problems.

Problem 1.2. The usage of different performance measures.

Problem 1.3. The impreciseness of results, and therefore no clearly specified conclusions.

Problem 1.4. The lack of reproducibility of experiments.

These problems provide guidelines for our analysis and will be reconsidered in Chap. 9. In fact, there is a gap between theory and experiment in evolutionary computation. How to promote good standards and quality of research in the field of evolutionary computation was discussed during the *Genetic and Evolutionary Computation Conference* (GECCO) in 2002. Bentley noted:

> Computer science is dominated by the need to publish, publish, publish, but sometimes this can happen at the expense of research. All too often poor papers, clumsy presentations, bad reviews or even bad science can clutter a conference, causing distractions from the more carefully prepared work (Bentley 2002).

There is a great demand for these topics, as one can see from the interest in tutorials devoted to these questions during two major conferences in evolutionary computation, the *Congress on Evolutionary Computation* (CEC) and GECCO (Bartz-Beielstein et al. 2003d; Wineberg & Christensen 2004; Bartz-Beielstein & Preuß 2004, 2005a, b).

1.2 Background

Evolutionary computation shares these problems with other scientific disciplines such as simulation, artificial intelligence, numerical analysis, or industrial optimization (Dolan & More 2002). Cohen's survey of 150 publications from the proceedings of the Eighth National Conference on Artificial Intelligence, which was organized by the *American Association for Artificial Intelligence*, "gave no evidence that the work they described has been tried out on more than a single example problem" (Cohen et al. 2000). He clearly demonstrated that there is no essential synergy between experiment and theory in these papers.

Cohen (1995) not only reported these negative results, he also provided valuable examples for how empirical research can be related to theory. Solutions from other disciplines that have been applied successfully for many years might be transferable to evolutionary computation. We have chosen four criteria to classify existing experimental research methodologies that have a lot in common with our approach. First, we can mention effective approaches.

They find a solution but are not very efficient and are not focused on understanding. Greedy, or brute-force approaches belong to this group. Second, meta-algorithms can be mentioned. They might locate good parameter sets, though without providing much insight into how sensitive performance is to parameter changes. Third, approaches that model problems of mostly academic interest can be listed. These approaches consider artificial test functions or infinite population sizes. Finally, the fourth category comprehends approaches that might be applicable to our problems although they have been developed with a different goal. Methods for deterministic computer experiments can be mentioned here. We will give a brief overview of literature on experimental approaches from these four domains.

1.2.1 Effective Approaches

The methodology presented in this book has its origins in statistical *design of experiments* (DOE). But classical DOE techniques as used in agricultural or industrial optimization must be adapted if applied to optimization models since stochastic optimization uses pseudorandom numbers (Fisher 1935). Randomness is replaced by pseudorandomness. For example, blocking and randomization, which are important techniques to reduce the systematic influence of different experimental conditions, are unnecessary in computer-based optimization. The random number seed is the only random element during the optimization run.

Classical DOE techniques are commonly used in simulation studies—a whole chapter in a broadly cited textbook on simulation describes experimental designs (Law & Kelton 2000). Kleijnen (1987, 1997) demonstrated how to apply DOE in simulation. As simulation is related to optimization (simulation models equipped with an objective function define a related optimization problem), we suggest the use of DOE for the analysis of optimization problems and algorithms (Kelton 2000).

This work is not the first attempt to use classical DOE methods in EC. However, our approach takes the underlying problem instance into account. Therefore, we do not try to draw any problem-independent conclusions such as: "The optimal mutation rate in genetic algorithms is 0.1." In addition, we propose an approach that is applicable if a small amount of function evaluations are available only. Schaffer et al. (1989) proposed a complete factorial design experiment that required 8400 run configurations; each configuration was run to 10,000 fitness function evaluations. Feldt & Nordin (2000) use statistical techniques for designing and analyzing experiments to evaluate the individual and combined effects of genetic programming parameters. Three binary classification problems are investigated in a total of 7 experiments consisting of 1108 runs of a machine code genetic programming system. Myers & Hancock (2001) present an empirical modeling of genetic algorithms. This approach requires 129,600 program runs. François & Lavergne (2001) demonstrate the applicability of *generalized linear models* (GLMs) to design

evolutionary algorithms. Again, data sets of size 1000 or even more are necessary, although a simplified evolutionary algorithm with 2 parameters only is designed.

As we include methods from computational statistics, our approach can be seen as an extension of these classical approaches. Furthermore, classical DOE approaches rely strongly on hypothesis testing. The reconsideration of the framework of statistical hypothesis testing is an important aspect in our approach.

1.2.2 Meta-Algorithms

The search for useful parameter settings of algorithms itself is an optimization problem. Optimization algorithms, so called meta-algorithms, can be defined to accomplish this task. Meta-algorithms for evolutionary algorithms have been proposed by many authors (Bäck 1996; Kursawe 1999). But this approach does not solve the original problem completely, because it requires the determination of a parameter setting of the meta-algorithm.

Additionally, we argue that the experimenter's skill plays an important role in this analysis. It cannot be replaced by automatic rules. The difference between automatic rules and learning tools is an important topic discussed in the remainder of this book.

1.2.3 Academic Approaches

Experimental algorithmics offer methodologies for the design, implementation, and performance analysis of computer programs for solving algorithmic problems (Demetrescu & Italiano 2000; Moret 2002). McGeoch (1986) examined the application of experimental, statistical, and data analysis tools to problems in algorithm analysis. Barr & Hickman (1993) and Hooker (1996) tackled the question how to design computational experiments and how to test heuristics. Aho et al. (1997) tried "to achieve a greater synergy between theory and practice."

Most of these studies were focused on *algorithms*, and not on *programs*. Algorithms can be analyzed on a sheet of paper, whereas the analysis of programs requires real hardware. The latter analysis includes the influence of rounding errors or limited memory capacities. We will use both terms simultaneously, because whether we refer to the algorithm or the program will be clear from the context.

Compared to these goals, our aim is to provide methods for very complex real-world problems, when only a few optimization runs are possible, i.e., optimization via simulation. The elevator supervisory group controller study discussed in Beielstein et al. (2003a) required more than a full week of round-the-clock computing in a batch job processing system to test 80 configurations.

Our methods are applied to real computer programs and not to abstract algorithms. A central topic in complexity theory is to answer the question NP

$\neq$ P. It is assumed that the class of problems that can be solved *nondeterministically in polynomial time* (NP) is different from the class of problems that can be solved in *polynomial time* (P). Problems in NP are—in contrast to problems in P—considered difficult and not efficiently solvable. However, analyses from complexity theory are not sufficient for some problems (Weihe et al. 1999). Many simple problems belong to NP. Niedermeier (2003) develops a recent approach to overcome this "dilemma of NP-hardness." Furthermore, there is an interesting link between programs (experimental approach) and algorithms (complexity theory) as discussed in Example 1.1.

Example 1.1 (Hooker 1994). Consider a small subset of very special *traveling salesperson problems* (TSP) T. This subset is NP-complete, and any class of problems in NP that contains T is ipso facto NP-complete. Consider the class P' that consists of all problems in P and T. As P' contains all easy problems in the world, it seems odd to say that problems in P' are hard. But P' is no less NP-complete than TSP. Why do we state that TSP is hard? Hooker (1994) suggests that "we regard TSP as a hard class because *we in fact find problems in TSP to be hard in practice.*" We acknowledge that TSP contains many easy problems, but we are able to generate larger and larger problems that become more and more difficult. Hooker suggests that it is this empirical fact that justifies our saying that TSP contains characteristically hard problems. And, in contrast to P', TSP is a natural problem class, or as philosophers of science would say, a natural kind. ■

1.2.4 Approaches with Different Goals

Although our methodology has its origin in DOE, classical DOE techniques used in agricultural and industrial simulation and optimization tackle different problems and have different goals.

Parameter control deals with parameter values (*endogenous strategy parameters*) that are changed during the optimization run (Eiben et al. 1999). This differs from our approach, which is based on parameter values that are specified before the run is performed (*exogenous strategy parameters*). The assumption that specific problems require specific EA parameter settings is common to both approaches.

Design and analysis of computer experiments (DACE) as introduced in Sacks et al. (1989) models the deterministic output of a computer experiment as the realization of a stochastic process. The DACE approach focuses entirely on the correlation structure of the errors and makes simplistic assumptions about the regressors. It describes "how the function behaves," whereas regression as used in classical DOE describes "what the function is" (Jones et al. 1998, p. 14). DACE requires other experimental designs than classical DOE, e.g., Latin hypercube designs (McKay et al. 1979). We will discuss differences and similarities of these designs and present a methodology for how DACE can be applied to stochastic optimization algorithms.

Despite the differences mentioned above, it might be beneficial to adapt some of these well-established ideas from other fields of research to improve the acceptance and quality of evolutionary algorithms.

1.3 Common Grounds: Optimization Runs Treated as Experiments

Gregory et al. (1996) performed an interesting study of dynamic scheduling that demonstrates how synergetic effects between experiment and theory can evolve. Johnson et al. (1989, 1991) are seminal studies of simulated annealing. Rardin & Uzsoy (2001) presented a tutorial that discusses the experimental evaluation of heuristic search algorithms when the complexities of the problem do not allow exact solutions. Their tutorial described how to design test instances, how to measure performance, and how to analyze and present the experimental results. They demonstrated pitfalls of commonly used measures such as the algorithm-to-optimal ratio, that measures how close an algorithm comes to producing an optimal solution.

Birattari et al. (2002) developed a "racing algorithm" for configuring metaheuristics that combines blocking designs, nonparametric hypothesis testing, and Monte Carlo methods. The aim of their work was "to define an automatic hands-off procedure for finding a good configuration through statistical guided experimental evaluations." This is unlike the approach presented here, which provides means for understanding algorithms' performance (we will use datascopes similar to microscopes in biology and telescopes in astronomy). However, Chiarandini et al. (2003) demonstrate that racing can be used interactively and not only as a monolithic block. These studies—although based on classical DOE techniques only—are closely related to our approach.

Optimization runs will be treated as experiments. In our approach, an experiment consists of a problem, an environment, an objective function, an algorithm, a quality criterion, and an initial experimental design. We will use methods from computational statistics to improve, compare, and understand algorithms' performances. The focus in this work lies on natural problem classes: Its elements are problems that are based on real-world optimization problems in contrast to artificial problem classes (Eiben & Jelasity 2002). Hence, the approach presented here might be interesting for optimization practitioners who are confronted with a complex real-world optimization problem in a situation where only few preliminary investigations are possible to find good parameter settings.

Furthermore, the methodology presented in this book is applicable a priori to tune different parameter settings of two algorithms to provide a fair comparison. Additionally, these methods can be used in other contexts to improve the optimization runs. They are applicable to generate systematically feasible starting points that are better than randomly generated initial points, or to guide the optimization process to promising regions of the search space.

Meta-model assisted search strategies as proposed in Emmerich et al. (2002) can be mentioned in this context. Jin (2003) gives a survey of approximation methods in EC.

Before introducing our understanding of experimental research in EC, we may ask about the importance of experiments in other scientific disciplines. For example, the role of experiments in economics changed radically during recent decades.

1.3.1 Wind Tunnels

The path-breaking work of Vernon L. Smith (2002 Nobel Prize in Economics together with Daniel Kahneman) in experimental economics provided criteria to find out whether economic theories hold up in reality. Smith demonstrated that a few relatively uninformed people can create an efficient market. This result did not square with theory. Economic theory claimed that one needed a horde of "perfectly informed economic agents." He reasoned that economic theories could be tested in an experimental setting: an economic wind tunnel. Smith had a difficult time getting the corresponding article published (Smith 1962). Nowadays this article is regarded as the landmark publication in experimental economics.

Today, many cases of economic engineering are of this sort. Guala (2003) reports that before "being exported to the real world" the auctions for mobile phones were designed and tested in the economic laboratory at Caltech. This course of action suggests that experiments in economics serve the same function that a wind tunnel does in aeronautical engineering. But, the relationship between the object of experimentation and the experimental tool is of importance: How much reductionism is necessary to use a tool for an object? Table 1.1 lists some combinations. Obviously some combinations fit very well, whereas others make no sense at all.

Table 1.1. Relationship between experimental objects and experimental tools. Some combinations, for example, reality–computer, require some kind of reductionism. Others, for example, algorithm–wind tunnel, are useless

Object of experimentation	Experimental tool
Reality	Computer
Reality	Thought experiment
Reality	Wind tunnel
Airplane	Computer
Airplane	Thought experiment
Airplane	Wind tunnel
Algorithm	Computer
Algorithm	Thought experiment
Algorithm	Wind tunnel

We propose an experimental approach to analyze algorithms that is suitable to discover important parameters and to detect superfluous features. But before we can draw conclusions from experiments, we have to take care that the experimental results are correct. We have to provide means to control the error, because we cannot ensure that our results are always sound. Therefore the concept of the new experimentalism is regarded next.

1.3.2 The New Experimentalism

The new experimentalism is an influential trend in recent philosophy of science that provides statistical methods to set up experiments, to test algorithms, and to learn from the resulting errors and successes. The new experimentalists are seeking a relatively secure basis for science, not in theory or observation but in experiment. To get the apparatus working for simulation studies is an active task. Sometimes the recognition of an oddity leads to new knowledge. Important representatives of the new experimentalism are Hacking (1983), Galison (1987), Gooding et al. (1989), Mayo (1996), and Franklin (2003). Deborah Mayo, whose work is in the epistemology of science and the philosophy of statistical inference, proposes a detailed way in which scientific claims are validated by experiment. A scientific claim can only be said to be supported by experiment if it passes a severe test. A claim would be unlikely to pass a severe test if it were false. Mayo developed methods to set up experiments that enable the experimenter, who has a detailed knowledge of the effects at work, to learn from error.

1.4 Overview of the Remaining Chapters

The first part of this book (Chaps. 1 to 6) develops a solid statistical methodology, which we consider to be essential in performing computer experiments. The second part, which is entitled "Results and Perspectives" (Chaps. 7 and 8) describes applications of this methodology.

New concepts for an objective interpretation of experimental results are introduced. Each of the following seven chapters closes with a summary of the key points. The concept of the new experimentalism for computer experiments and central elements of an understanding of science are discussed in Chap. 2. It details the difference between demonstrating and understanding, and between significant and meaningful. To incorporate these differences, separate models are defined: models of hypotheses, models of experimental tests, and models of data. This leads to a reinterpretation of the *Neyman–Pearson theory of testing* (NPT). Since hypothesis testing can be interpreted objectively, tests can be considered as learning tools. Analyzing the frequency relation between the acceptance (and the rejection) of the null hypothesis and the difference in means enables the experimenter to learn from errors. This concept of learning

tools provides means to extend Popper's widely accepted claim that theories should be falsifiable.

Statistical definitions for Monte Carlo methods, classical design and analysis of experiments, tree-based regression methods, and modern design and analysis of computer experiments techniques are given in Chap. 3. A bootstrap approach that enables the application of learning tools if the sampling distribution is unknown is introduced. This chapter is rather technical, because it summarizes the relevant mathematical formulas.

Computer experiments are conducted to improve and to understand the algorithm's performance. Chapter 4 presents optimization problems from evolutionary computation that can be used to measure this performance. Before an elevator group control problem is introduced as a model of a typical real-world optimization problem, some commonly used test functions are presented. Problems related to test suites are discussed as well.

Different approaches to set up experiments are discussed in Chap. 5. Classical and modern designs for computer experiments are introduced. A sequential design based on DACE that maximizes the expected improvement is proposed.

Search algorithms are presented in Chap. 6. Classical search techniques, for example, the Nelder–Mead "simplex" algorithm, are presented as are stochastic search algorithms. The focus lies on particle swarm optimization algorithms, which build a special class of bioinspired algorithms.

The discussion of the concept of optimization provides the foundation to define performance measures for algorithms in Chap. 7. A suitable measure reflects requirements of the optimization scenario or the experimental environment. The measures are categorized with respect to effectivity and efficiency. Now, the necessary components according to the discussion in the previous chapters to perform computer experiments are available: a problem, an environment, an objective function, an algorithm, a quality criterion, and an experimental design. After summarizing a classical DOE approach of finding better suited exogenous parameters (tuning), a sequential approach that comprehends methods from computational statistics is presented. To demonstrate that our approach can be applied to any arbitrary optimization algorithm, several variants of optimization algorithms are tuned. Tools from error statistics are used to decide whether statistically significant results are scientifically meaningful.

Chapter 8 closes the circle opened in Chap. 2 on the discussion of testing as an automatic rule and as a learning tool. Provided with the background from Chap. 2, the aim of Chap. 8 is to propose a method to learn from computer experiments and to understand how algorithms work. Various schemes for selection under noise for direct search algorithms are presented. Threshold selection is related to hypothesis testing. It serves as an example to clarify the difference between tests as rules of inductive behavior and tests as learning tools. A summary and an outlook conclude this book in Chap. 9.

Introducing the new experimentalism in evolutionary computation provides tools for the experimenter to understand algorithms and their interactions with optimization problems. Experimentation is understood as a means for testing hypotheses, the experimenter can learn from error and control the consequences of his decisions. The methodology presented here is based on the statistical methods most widely used by today's practicing scientists. It might be able "to offer genuine hope for a recovery of some of the solid intuitions of the past about the objectivity of science"(Ackermann 1989).