Synthese (2008) 163:385–396
DOI 10.1007/s11229-007-9297-z

How experimental algorithmics can benefit from Mayo's extensions to Neyman–Pearson theory of testing

Thomas Bartz-Beielstein

Received: 9 November 2007 / Accepted: 9 November 2007 / Published online: 19 February 2008

Abstract Although theoretical results for several algorithms in many application domains were presented during the last decades, not all algorithms can be analyzed fully theoretically. Experimentation is necessary. The analysis of algorithms should follow the same principles and standards of other empirical sciences. This article focuses on stochastic search algorithms, such as evolutionary algorithms or particle swarm optimization. Stochastic search algorithms tackle hard real-world optimization problems, e.g., problems from chemical engineering, airfoil optimization, or bioinformatics, where classical methods from mathematical optimization fail. Nowadays statistical tools that are able to cope with problems like small sample sizes, non-normal distributions, noisy results, etc. are developed for the analysis of algorithms. Although there are adequate tools to discuss the statistical significance of experimental data, statistical significance is not scientifically meaningful per se. It is necessary to bridge the gap between the statistical significance of an experimental result and its scientific meaning. We will propose some ideas on how to accomplish this task based on Mayo's *learning model* (NPT*).

Keywords New experimentalism · Experimental algorithmics · Optimization · Theory of testing · Mayo's learning model · Significance

1 Introduction

Optimization problems are of great importance in practice, particularly in engineering and technology, business and finance. Algorithms to tackle optimization problems

T. Bartz-Beielstein (✉)
Faculty of Computer Science and Engineering Science, Cologne University of Applied Sciences, 51643 Gummersbach, Germany
e-mail: bartz@gm.fh-koeln.de

include classical techniques such as dynamic programming or gradient-based methods, but also modern techniques such as stochastic search heuristics. Examples of stochastic search heuristics include simulated annealing, tabu search, evolutionary computation, iterated local search, particle swarm optimization, and ant colony optimization. Research on these techniques relies on experimentation, because the applicability of theoretical results is limited to very special (artificial or simplified) situations.

The term *experimental algorithmics* will be used as a synonym for approaches in computer science that require experimental studies. Experimental approaches have been modified over the years. During the first phase of experimental research (before 1980), which can be characterized as 'foundation and development,' a comparison of different stochastic search algorithms was mostly based on mean values—almost no further statistics were used. In the second phase (1980–2000), classical statistical methods such as analysis of variance or regression techniques were introduced. In the past few years, statistical approaches that consider specific features of the algorithms became popular. However—even if adequate statistical tools for research in experimental algorithmics are under development—they do not bridge the gap between the statistical significance of an experimental result and its scientific meaning.

One of the major goals in experimental algorithmics is to demonstrate that and understand why one algorithm, A, outperforms a related algorithm, B. Researchers suppose that A and B behave differently, because one algorithm has features the other one lacks, e.g., an improved variation operator.

In this paper, we propose a methodology to analyze the relationship between statistical significance and scientific import based on a standard situation in experimental research. Common to all experiments is the need to compare two algorithms, a task that can be modeled within the framework of hypothesis testing. To test the hypothesis that algorithm A performs better than B, we first assume that they perform equally, i.e., there is no difference in means. Therefore, we face a standard situation from statistics, the comparison of samples from two populations.[1] This comparison applies also to the question of whether different algorithms exploit any 'systematic information' in the data (test problems) equally efficiently and effectively or not.

Neyman–Pearson theory of testing (NPT) defines a well-known framework for performing this comparison. We will take Deborah Mayo's extension of NPT (Mayo 1983) into consideration. Mayo, an important representative of the new experimentalism which is an influential trend in recent philosophy of science, has proposed a detailed epistemology of how scientific claims are and can be validated by experiment. Her philosophy, which she calls 'error statistics' because central to it is the importance of NPT error probabilities, generalizes and provides statistical methods to set-up experiments, to test algorithms, and to learn from the resulting errors and successes based on her concept of *severity*.

A scientific claim can only be said to be supported by experiment if that experiment provided a severe test of the claim and it passed. A severe test of a claim is one in which the claim would be unlikely to pass, if it were false. Mayo developed methods to

[1] There is no unique 'best' algorithm which performs better than any other algorithm on every test problem (Wolpert and Macready 1997). However, this theorem has only minor impact on practical problems (Bartz-Beielstein 2006).

set up experiments that enable the experimenter, who has a detailed knowledge of the effects at work, to learn from error. This paper is an attempt to transfer recent results from the error statistics to computer science, especially to experimental algorithmics.

Severity is introduced in Sect. 2. This introduction is based on concepts and methods presented in Mayo (1983) and Mayo and Spanos (2006). They are used to derive metastatistical rules to test whether statistically significant results are scientifically relevant. Section 3 summarizes the sequential parameter optimization which defines a standardized framework in experimental algorithmics. We discuss how severity can be integrated into this framework and how it bridges the gap between statistical results and their scientific meaning. This article concludes with a short summary and outlook.

The results presented in this paper are based on experimental analysis of stochastic search heuristics. They can be transferred easily to other kinds of algorithms.

2 Severity

2.1 NPT and Mayo's learning model

Neyman–Pearson theory can be interpreted as a means of deciding how to behave. A metastatistical problem is how to relate an empirical scientific inquiry to the statistical models of NPT. To contrast her reformulation of NPT with this behavioristic model, Mayo (1983) introduces the term *learning model*, or simply NPT*, for the former. NPT* goes beyond NPT, it uses the distribution of the test statistic to control error probabilities. Statistical tests are seen as 'means of learning about variable phenomena on the basis of limited empirical data.'

Consider a statistical model with some unknown parameter θ. Mayo (1983) claims that NPT* provides tools for specifying tests that 'will very infrequently classify an observed difference as significant (and hence reject H) when no discrepancy of scientific importance is detected, and very infrequently fail to do so (and so accept H) when θ is importantly very discrepant from θ_0.'

2.2 Neyman–Pearson tests and the severity function

The major goal introduced in Sect. 1 can be formulated in the context of Mayo's learning model. To stay consistent with Mayo's seminal text, we use the same notation found in Mayo and Spanos (2006). We consider a sample $X = (X_1, \ldots, X_n)$, where each X_i is assumed to be normal, i.e., $X_i \sim \mathcal{N}(\mu, \sigma^2)$, independent and identically distributed (NIID). Furthermore, we assume a known standard deviation σ.

In error-statistical testing, the primary hypothesis is posed as a question related to the data-generating mechanism. This question is formulated as a statistical hypothesis H_0 (null hypothesis), which assigns a probability to each possible outcome x. Since it gives the 'probability of outcome x under H_0', this probability is denoted as $P(x; H_0)$. To incorporate alternatives, a second hypothesis H_1 (alternative hypothesis) is formulated, so that the parameter space of the statistical model is entirely partitioned. Null and

alternative hypotheses concerning the mean μ are formulated as:

$$H_0\colon \mu \leq \mu_0 \quad \text{vs.} \quad H_1\colon \mu > \mu_0.$$

The experimental *test statistic* reads

$$\mathrm{d}(X) = \frac{(\overline{X} - \mu_0)}{\sigma/\sqrt{n}},$$

where $\overline{X}$ denotes the sample mean. Under the null hypothesis, $\mathrm{d}(X)$ is distributed as standard normal, i.e., $\mathrm{d}(X) \sim \mathcal{N}(\theta, \sigma^2)$. For a given probability α, the type I error is defined as

$$\text{Type I (or } \alpha\text{) error probability} = P(\mathrm{d}(X) > c_\alpha; H_0) \leq \alpha,$$

where the *cut-off point* c_α defines the rejection region

$$C_1(\alpha) = \{x : \mathrm{d}(x) > c_\alpha\}.$$

Consider a value μ_1 in the rejection region: $\mu_1 > \mu_0$. The type II error probability is defined as

$$\text{Type II (or } \beta\text{) error probability at } \mu_1 = P(\mathrm{d}(X) \leq c_\alpha;\ H_1) = \beta(\mu_1).$$

We have defined a *Neyman–Pearson test* (N–P test) $T(\alpha)$ with significance level α which rejects H_0 with data x_0 if and only if $\mathrm{d}(x_0)$ is greater than c_α (Mayo and Spanos 2006). The power of this test can be determined as

$$\mathrm{POW}(T(\alpha); \mu_1) = P(\mathrm{d}(X) > c_\alpha; \mu_1),$$

for $\mu_1 > \mu_0$. Note, that power is always calculated with respect to the cut-off point c_α. Power does not depend on the outcome x_0. Mayo and Spanos (2006) propose two conditions for a severe test:

A statistical hypothesis H passes a *severe* test T with data x_0 if

(S-1) x_0 agrees with H, and
(S-2) with very high probability, test T would have produced a result that accords *less* well with H than x_0 does, if H were false.

They define a severity function which has three arguments: a test, an outcome, and an inference or claim:

$$\mathrm{SEV}(\text{Test } T,\ \text{outcome } x,\ \text{claim } H).$$

For reasons of simplicity, we will use $\mathrm{SEV}(\mu; \mu_1)$ throughout the remainder of this article.

2.2.1 Severity interpretation of acceptance

Mayo and Spanos (2006) introduce the following formula for evaluating severity of test $T(\alpha)$ in the case of a statistically insignificant result (H_0 is accepted):

$$\begin{aligned} \text{SEV}(\mu \leq \mu_1) &= P(\text{d}(X) > \text{d}(x_0); \\ \mu \leq \mu_1 \text{ false}) &= P(\text{d}(X) > \text{d}(x_0); \mu > \mu_1). \end{aligned}$$

Setting severity and power in contrast with each other may be useful to clarify similarities and differences:

$$\begin{aligned} \text{POW}(T(\alpha); \mu_1) &= P(\text{d}(X) > c_\alpha; \mu = \mu_1), \\ \text{SEV}(T(\alpha); \mu_1) &= P(\text{d}(X) > \text{d}(x_0); \mu = \mu_1). \end{aligned}$$

To discuss severe testing as a concept for post-data inference, we will present a numerical example from experimental algorithmics.

Example 1 (Population size) Analyzing stochastic search algorithms, e.g., an evolution strategy (ES), we are interested in testing whether or not the population size has a significant influence on the performance of the algorithm. A minimization task was chosen as a test problem.[2] The population size was set to $s_A = 20$ and $s_B = 40$. The corresponding parameterizations of the ES will be referred to as A and B, respectively. The question is whether an increased population size improves the performance of the ES. Our inquiry can be formulated as a scientific claim:

Scientific Claim 1 (C) The algorithm's performance is not affected by population size.

Let $\overline{X}_A$ and $\overline{X}_B$ denote the average performance of A and B, respectively. Each algorithm is run $n = 100$ times. The sample difference between $\overline{X}_A$ and $\overline{X}_B$ is denoted as $\overline{X}$ and $\text{d}(X) = (\overline{X} - \mu_0)\sqrt{n}/\sigma$ is the experimental test statistic. For simplicity, we assume a known standard deviation $\sigma = 2$. With $\alpha = 0.025$, we can formulate the following test:

$$\begin{aligned} &H_0\text{: } \mu \leq 0 \text{ vs. } H_1 > 0. \\ &\text{Reject } H_0 \text{ iff } \text{d}(x_0) > 1.96, \text{ i.e., iff } \overline{x} \geq 0.4. \end{aligned}$$

The optimization practitioner considers a difference in means $\gamma^* = 0.2$ *substantially important*.

A random sample is drawn from A and B, i.e., the ES is run with 20 and 40 individuals. The average performance $\overline{x}_A$ of $n = 100$ runs of A is 56, whereas the average performance $\overline{x}_B$ of $n = 100$ runs of B is 56.3. The sample difference between $\overline{x}_A$ and $\overline{x}_B$ is $\overline{x} = 0.3$, so we have obtained a non-significant result: $\text{d}(x_0) = 1.5$.

How can we determine if it was not a rash decision to take this non-significant result as reasonable evidence that an important difference in means is absent? And,

[2] Note, we are considering only one specific *instance* of this minimization task here. A more general approach is discussed in Sect. 3.

what if the outcome is much smaller than $d(x_0) = 1.5$, say, $d(x_0) = -1$? Power is a well-known and commonly used tool to determine the 'quality' of statistical tests. However, it is identical for both outcomes, e.g., considering a difference $\gamma^* = 0.2$ as meaningful, power can be calculated as:

$$\text{POW}(T(\alpha, \gamma^* = 0.2) = 0.169.$$

Mayo and Spanos (2006) argue that the relevant threshold, post-data, is not the cut-off point c_α, but the standardized outcome $d(x_0)$.

$$\text{SEV}(T(\alpha), d(x_0), \mu \leq \mu_1) = P(d(X) > d(x_0); \mu = \mu_1).$$

This formula enables us to answer the question: 'How severely does $\mu \leq \mu_1$ pass with $\overline{x} = 0.3$ ($d(x_0) = 1.5$)'? Figure 1 illustrates the situation in case of a statistically insignificant result, i.e., 'Accept H_0'.

The assertion:

'We may infer that any discrepancy from 0.0 is absent or no greater than 1.0.'

can be calculated as $\text{SEV}(\mu \leq 1) = 0.9997$.

However, if we are too demanding, severity warns us about too extreme conclusions. For $d(x_0) = 1.5$, ($\overline{x} = 0.3$) the severity of the assertion:

'We may infer that any discrepancy from 0.0 is absent or no greater than 0.1.'

can be calculated as $\text{SEV}(\mu \leq 0.1) = 0.16$. Even if a discrepancy of 0.1 from H_0 exists, an insignificant result would occur 84% of the time.

Note, that $\text{SEV}(\mu \leq 0.2) = 0.309$ and $\text{SEV}(\mu \leq 0.3) = 0.5$. □

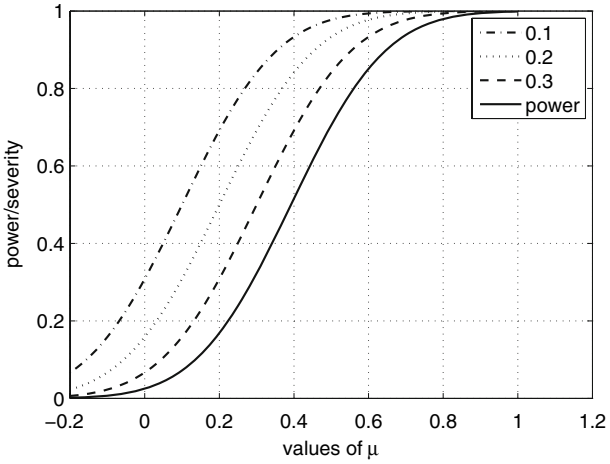

Fig. 1 Case 'Accept H_0'—power versus severity. The severity for $\mu \leq 0$ with different outcomes x_0. Power curve is the solid line. Similar figure as in Mayo and Spanos (2006)

A small probability of detecting a given discrepancy γ^* from μ_0 provides poor evidence that so small a discrepancy is absent.

In practice, there is no unique γ^*. Severity enables experimentalists to determine as small a discrepancy from the null hypothesis as possible. Plots like the one shown in Fig. 1 can be useful in establishing the smallest discrepancy from the null via post-data analysis. For a certain level of severity, e.g., $\gamma = 0.95$, the experimenter can determine the discrepancy γ that is related to this level. Figure 2 in Mayo and Spanos (2006) perspicuously illustrates this procedure.

Since severity and power correspond if $\mathrm{d}(x_0)$ is close to the critical point c_α, severity can be seen as a refinement of power calculations. Severity also sheds some light on the so-called *large n problem*:

> In the context of a statistical test, does a given p-value convey stronger evidence about the null hypothesis in a larger trial than in a smaller trial, or vice versa? (Gregoire 2001)

Severity solves this problem directly: An α-significant differences with larger sample size n passes $\mu > \mu_1$ less severely than with a smaller n (cf. also Fig. 4 in Mayo and Spanos (2006)).

A similar discussion of the severity interpretation of rejection can be found in Mayo and Spanos (2006). They demonstrate how error-statistical tools can extend

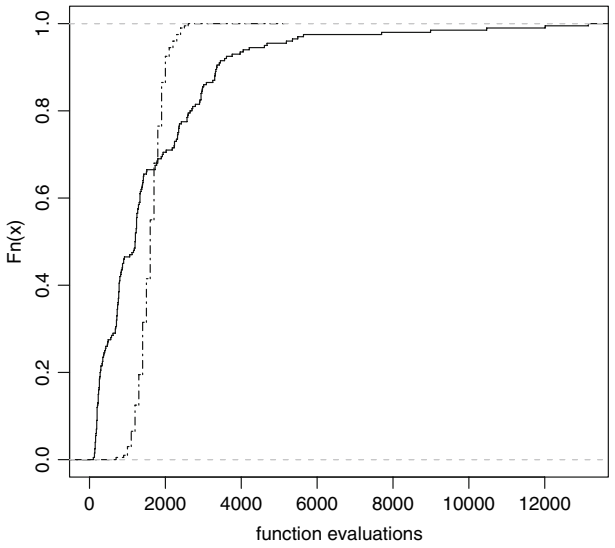

Fig. 2 Run length distribution to determine an adequate number of function evaluations. The algorithm to be analyzed is run n times with different seeds on one problem instance. For each successful run, the number of function evaluations m is recorded. If the run fails, m is set to infinity. Both curves illustrate results of one algorithm on one problem instance but with different algorithm parameters. The empirical cumulative distribution shows that 2000 function evaluations are adequate, because approximatively 50% of both configurations were able to reach the goal. Both parametrizations of the algorithm are able to reach the goal in nearly every run, if 10,000 function evaluations are chosen. Therefore, 10,000 function evaluations may result in ceiling effects

commonly used pre-data error probabilities (significance level and power) by using post-data interpretations of the resulting rejection or acceptance.

3 Sequential parameter optimization

Example 1 describes a typical situation from experimental algorithmics: a comparison of two algorithms. This is a simple one-shot scenario where NPT* can be profitably applied. We claim that combining well-established techniques from experimental planning with NPT* will lead to new knowledge and scientific progress. *Sequential parameter optimization* (SPO) is one possible step in this direction (Bartz-Beielstein 2006).

Mayo's seminal discussion of Brownian motion (Mayo 1996) can be used as a guideline on how to set up a hierarchy of models and strategies for arriving at severe tests. In contrast to the experimental analysis of Brownian motion, computer scientists are able to control every cause of an algorithm's behavior. Unfortunately, there are many parameters influencing the algorithm's behavior. The first step is to make these parameters explicit. By designing experiments and systematically varying these parameters, we can detect important factors and interactions.

Before experiments can be performed, a scientific question or goal has to be formulated. As an example, we consider particle swarm optimization (PSO). PSO uses a communication structure or social network, which assigns neighbors for each individual to interact with. It is of great interest to determine in which situations fully informed swarms, i.e., swarms with global social networks, are superior to swarms that use local information only. There is no general answer to this question. However, a comparison can be performed in certain environments. Mathematically speaking, we consider a special type of mathematical optimization problem, e.g., quadratic programming (QP) problems. Quadratic programming problems can be formulated as: minimize (with respect to x) $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x}+\mathbf{c}^T\mathbf{x}$, where Q denotes a symmetric $n \times n$ matrix, and $\mathbf{c}$ is any $n \times 1$ vector. The whole class of QP problems cannot be analyzed in one step. Therefore, we consider a simple QP instance first, e.g., $f(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$. Even this simple optimization problem requires the specification of several factors before experiments can be started. The term *problem design* subsumes these factors. In addition to the objective function f, the starting point, the problem dimension, the available time (budget) for running the algorithm and the stopping criterion belong to the problem design.

The selection of an adequate problem design is complicated. Consider the following pitfalls in choosing among available algorithms, which occur in several studies from experimental algorithmics.

Example 2 (Floor and ceiling effects) Choosing a problem instance which cannot be solved by any algorithm produces the statistically meaningful result that 'there is no difference in performance' for every algorithm. This is a floor effect. Alternatively, problem instances which are far too easy produce similar problems (ceiling effects). NPT* and severity provide adequate tools to avoid floor and ceiling effects. Mayo and Spanos (2006) state: 'If a test has a very low probability to detect the

existence of a given discrepancy from μ_0, then such negative result is poor evidence that so small a difference is absent.' □

How can floor or ceiling effects be avoided? Run length distributions (Fig. 2) are useful tools in this context.

In addition to factors from the problem design, there are also factors related to the *algorithm design* that need to be considered. In addition to the communication structure mentioned above, we have to consider the swarm size and two parameters called 'cognitive coefficients' (Clerc and Kennedy 2006). All in all there are nine factors: five factors related to the problem design and four factors related to the algorithm design.

Systematic variation of these nine factors may shed some light on the question of required information exchange between particles. But how can we determine whether the fully informed swarm is superior to the local variant? To judge the performance of different algorithms, a performance measure is needed. The choice of an adequate performance measure is not trivial.[3] Based on the run length distributions from Fig. 2, the best function value from 2000 function evaluations was chosen as a performance measure for the experiments. Because PSO is a stochastic algorithm, i.e., similar starting conditions may lead to different results, we consider the average function value from 100 repeats. Note that this choice is not generic; other measures such as the median, the minimum or the standard deviation (or combinations of these) are possible.

The experimental goal is formulated as a scientific claim $\mathcal{C}$, e.g., 'For the objective function $f(x) = x^2$, PSO with global information outperforms PSO which uses local information.' This scientific claim is then broken into several statistical hypotheses as illustrated in Fig. 3.

To give an example, we present one typical hypothesis.

Example 3 (Statistical hypothesis) A linear regression model can be used to determine the effects of the algorithm factors on performance. Since there are four factors, the model can be formulated as: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon$, where y represents the performance, x_1 represents the number of particles informed, x_2 represents the swarm size, x_3 and x_4 represent the cognitive coefficients (ϵ denotes the error term). The hypotheses we wish to test are H_0: $\beta_1 = 0$ against H_1: $\beta_1 \neq 0$. □

Each statistical hypothesis is tested experimentally. Results from the *design and analysis of* (*computer*) *experiments* guide the decision how to vary the factors systematically (Kleijnen 1987; Montgomery 2001). Important factors are screened out, e.g., by stepwise regression. Now it is important to see whether the information exchange between particles has a significant effect on the performance.

Finally, the algorithm with global information exchange is compared to the variant that processes only local information. This comparison uses techniques described in Example 1 and can be supported by graphical tools such as box-plots.

SPO comprehends a discussion of the experimental result: although statistically significant, it can be scientifically meaningless. An objective interpretation of rejecting or

[3] While discussing Brownian motion, Mayo (1996) presents an example of 'measuring the wrong thing.'

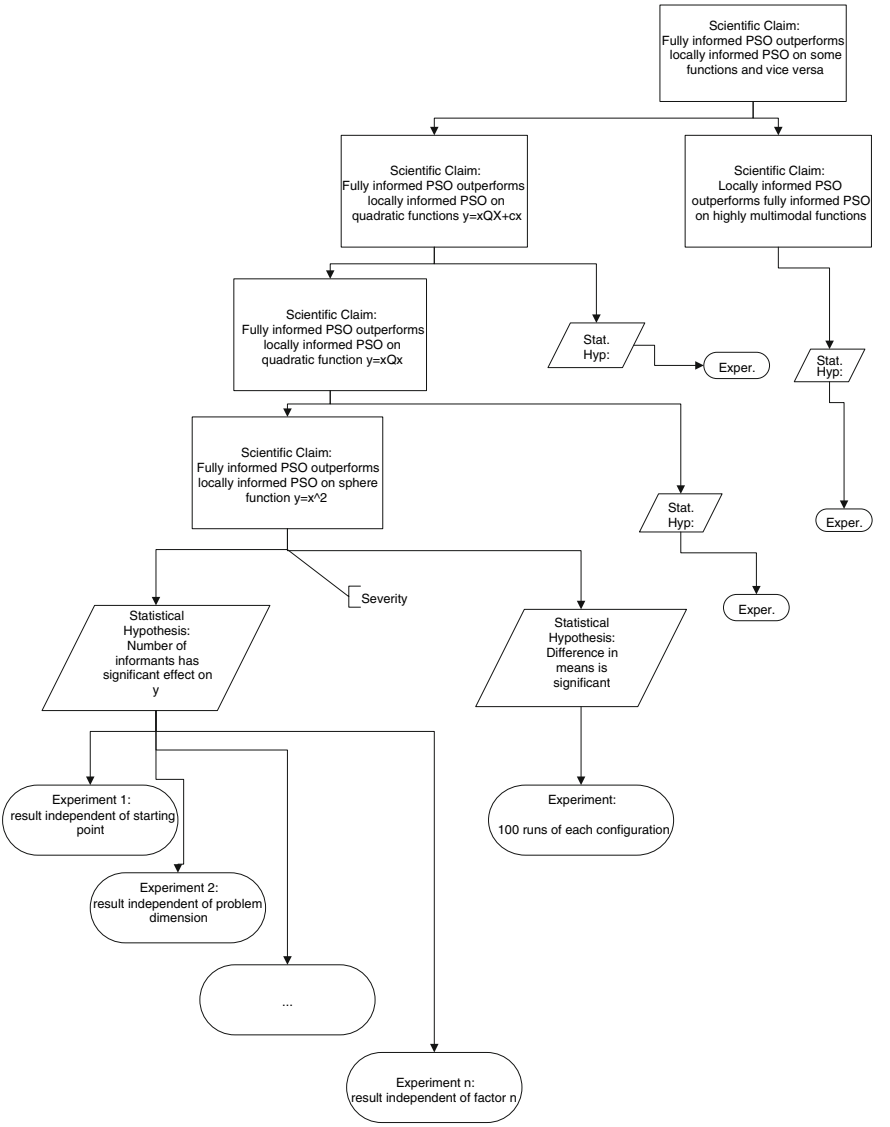

Fig. 3 Dividing a scientific claim into multiple statistical hypotheses. General scientific claims are formulated as specific scientific claims. These are formulated as statistical hypotheses, which can be tested experimentally. The figure illustrates a few elements of this procedure

accepting statistical hypotheses should be presented. Here NPT* comes into play. Consequences that arise from this decision to accept or reject should be discussed as well. The experimenter's skill plays an important role for this decision. The experimental setup should be reconsidered at this stage and questions like 'Have suitable test functions or performance measures been chosen?' must be answered.

SPO has been applied in several domains such as machine engineering, aerospace industry, elevator group control, graph drawing, technical thermodynamics, vehicle

routing and bio-informatics. It is also a useful tool to determine improved algorithm designs (tuning). A sequential technique is utilized to perform this tuning very efficiently. This is important for complex real-world optimization tasks, which allow only a few function evaluations. SPO illustrates how experimental algorithmics can benefit from concepts developed in philosophy of science.

4 Summary and outlook

We described the current situation of experimental research in computer science, especially in experimental algorithmics. Several statistical tools that reflect the requirements of today's optimization practitioners are being developed nowadays. However, almost no tools exist that enable an interpretation of and learning from scientific results. Mayo's models of statistical testing bridge this gap. We demonstrated how approaches introduced in Mayo (1983) and Mayo and Spanos (2006) can be transferred to experimental algorithmics. An example was presented to illustrate this approach. SPO was presented as an integrated approach for tuning, analyzing, and understanding computer algorithms.

Commenting on Mayo and Spanos (2006), who, while discussing questions of the role of error probabilities, state, 'Not that practitioners are waiting for philosophers to sort things out', we conclude this article with a short outlook from the perspective of an experimenter in computer science. There is a growing interest in—and need for—sound experimental methodologies as recent workshops such as the 'Workshop On Empirical Methods for the Analysis of Algorithms' (Bartz-Beielstein and Preuss 2006a,b) and a series of tutorials given at GECCO and CEC, the leading conferences in the field of evolutionary optimization, demonstrate (Bartz-Beielstein and Preuss 2004, 2005).

We suppose that only a handful of researchers in computer science are aware of some of these fundamental methodological discussions in the philosophy of science. However, discussing questions like the 'significance test controversy' (Morrison and Henkel 1970) produces interest even in this application oriented research community. There is a increasing demand for NPT* tools that enable practitioners simply to plug in their data and support the interpretation of their results. Articles such as Mayo and Spanos (2004) present important examples of how ideas from philosophy of science can be made popular in other research communities.

References

Bartz-Beielstein, T. (2006). *Experimental research in evolutionary computation—The new experimentalism*. Berlin, Heidelberg, New York: Springer.

Bartz-Beielstein, T., & Preuss, M. (2004). CEC tutorial on experimental research in evolutionary computation. In *IEEE Congress on Evolutionary Computation, Tutorial Program*. Tutorials given at CEC in 2004 and 2005.

Bartz-Beielstein, T., & Preuss, M. (2005). GECCO tutorial on experimental research in evolutionary computation. In *2005 Genetic and Evolutionary Computation Conference, Tutorial Program*. Tutorials given at GECCO in 2005, 2006, and 2007.

Bartz-Beielstein, T., & Preuss, M. (2006a). Considerations of budget allocation for sequential parameter pptimization (SPO). In *Workshop on Empirical Methods for the Analysis of Algorithms (EMAA), Tutorial*

Program. Held in conjunction with the International Conference on Parallel Problem Solving From Nature (PPSN IX) (pp. 35–40).

Bartz-Beielstein, T., & Preuss, M. (2006b). Sequential parameter optimization (SPO) and the role of tuning in experimental analysis. In *Workshop on Empirical Methods for the Analysis of Algorithms (EMAA). Held in conjunction with the International Conference on Parallel Problem Solving From Nature (PPSN IX)* (pp. 5–6). Invited talk.

Clerc, M., & Kennedy, J. (2006). Standard PSO version 2006. http://www.particleswarm.info/Standard_PSO_2006.c. Cited 11 August 2007.

Gregoire, T. (2001). Biometry in the 21st Century: Whither statistical inference? (Invited Keynote). Proceedings of the Forest Biometry and Information Science Conference held at the University of Greenwich, June 2001. http://cms1.gre.ac.uk/conferences/iufro/proceedings/gregoire.pdf. Cited 19 May 2004.

Kleijnen, J. P. C. (1987). *Statistical tools for simulation practitioners*. New York, NY: Marcel Dekker.

Mayo, D. G. (1983). An objective theory of statistical testing. *Synthese, 57*, 297–340.

Mayo, D. G. (1996). *Error and the growth of experimental knowledge*. Chicago, IL: The University of Chicago Press.

Mayo, D. G., & Spanos, A. (2004). Methodology in practice: Statistical misspecification testing. *Philosophy of Science, 71*, 1007–1025.

Mayo, D. G., & Spanos, A. (2006). Severe testing as a basic concept in a Neyman–Pearson philosophy of induction. *British Journal for the Philosophy of Science, 57*, 323–357.

Montgomery, D. C. (2001). *Design and analysis of experiments* (5th ed.). New York, NY: Wiley.

Morrison, D. E., & Henkel, R. E. (Eds.) (1970). *The significance test controversy—A reader*. London, UK: Butterworths.

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82.