Design of Evolutionary Algorithms—A Statistical Perspective

Olivier François and Christian Lavergne

***Abstract*—This paper describes a statistical method that helps to find good parameter settings for evolutionary algorithms. The method builds a functional relationship between the algorithm's performance and its parameter values. This relationship—a statistical model—can be identified thanks to simulation data. Estimation and test procedures are used to evaluate the effect of parameter variation. In addition, good parameter settings can be investigated with a reduced number of experiments. Problem labeling can also be considered as a model variable and the method enables identifying classes of problems for which the algorithm behaves similarly. Defining such classes increases the quality of estimations without increasing the computational cost.**

***Index Terms*—Evolutionary algorithm, experimental design, gamma distribution, generalized linear model.**

I. Introduction

EVOLUTIONARY algorithms (EAs) are defined by the user's choice of representations and operators such as selection, mutation, and recombination. These operators have many associated parameters, such as the mutation rate, the crossover rate, the mutation step size, etc., and there are other parameters of an EA that include the population size, the tournament size if applicable, and so forth. Different values of these parameters yield different performance, even on the same problems.

Determining parameter settings that yield efficient performance on a specific problem is a challenging issue. There are many ways to measure performance, but ultimately, the choice of which parameters to use depends largely on the practitioner's experience and trial-and-error experimentation. A straightforward means for comparing different parameter settings is to conduct numerous experiments with each setting and compare the resulting outcomes. Many users base their choice of parameters on such "hands on" experimentation. Unfortunately, such efforts are computationally intensive, particularly because to be truly effective, they must involve the simultaneous adjustment of multiple parameters.

An even more difficult challenge lies in attempting to extrapolate appropriate parameter settings for a new problem based on those obtained in other problems. As proved in the no free lunch theorem [41], any attempt to design a generally optimal set of parameters is useless. This result underscores the need for investigating and even first defining classes of problems for which EAs behave similarly. Some practitioners attempt to use parameter settings that have proved successful in other problems that appear similar. Such similarity, as perceived by the practitioner, may be unreliable because it is highly subjective.

Two fundamental questions arise when creating useful classes of problems: 1) by what criterion are the classes defined? and 2) by what method can we evaluate a problem to determine which class it belongs to? The answers involve a notion of similarity between problems or, more precisely, between the behavior that an EA exhibits on those problems. Furthermore, any answer must be statistical in nature because the behavior of an EA is stochastic.

This paper investigates statistical models of the performance of EAs and uses these models to create classes of problems. Similarity is defined as a statistical notion that is related to the level at which two models differ significantly. Furthermore, confidence intervals around optimal parameter settings can be given. The number of experiments required to find optimal parameter settings may be reduced drastically depending on whether statistical methods are used to process the random data generated by the EAs.

Our method involves two stages. A *problem-level* model is constructed at the first stage. This model describes a functional relationship between the performance of the EA and its parameters on a specific problem. A *global* or *group-level* model is constructed at the second stage to describe the behavior of the EA on a set of test problems. The ultimate goal is to assign reference labels to classes of test problems for the EA, where the EA exhibits similar within-class performance.

The paper is organized as follows. Section II provides background and describes the steps of the method. Section III describes the context of the application and introduces test problems and the EA. Section IV details models and argues for specific statistical hypotheses. Section V presents the results obtained on the test problems. An Appendix reviews generalized linear models and the associated statistical techniques that comprise of the main tool of this paper.

II. Method

A. Previous Works

Characterizing classes of problems for which an EA behaves similarly is a crucial issue in the evolutionary computation field. The relevance of this issue has been indicated recently by the no free lunch theorem [41]: two EAs that are evaluated on all classes of problems have equal expected performances.

Manuscript received February 9, 1999; revised July 26, 1999 and February 4, 2000.

O. François is with the École Nationale Supérieure d'Informatique et de Mathématiques Appliquées, 38041 Grenoble Cedex 9, France.

C. Lavergne is with the Institut National de Recherche en Informatique et Automatique, 38334 Saint-Ismier Cedex, France.

Publisher Item Identifier S 1089-778X(01)04140-6.

Evaluating the (*posteriori*) relevance of a set of problems as a test suite is a closely related issue [9], [10], [40]. EAs are expected to behave differently on each problem of the test suite. As a consequence, the optimal parameter settings obtained for a specific problem should not be reused for other problems without care.

In view of assessing EA performance, many indicators have been introduced when two instances of parameters are compared or when the EA is tested on two different problems. These indicators—epistatis variance [7], correlation length, fitness distributions [16], and fitness distance correlation [22]—attempt to relate the performances of an EA with some *a priori* statistical measures computed over the search space and the fitness landscape. However, the predictions made with some of these statistical indicators are not always so reliable [28].

Parameter design has been set up theoretically for some EAs, considering severe simplifications of algorithms or problems. Rechenberg's one-fifth rule [31] or Bäck's results [4] are famous examples. However, many counterexamples show that these rules led to algorithms that are neither convergent nor efficient in general [35].

There have been many experimental comparisons of EA operators or EA performances with different parameter settings, where elementary statistical tools have been used to interpret the results (see, e.g., [4], [8], [10], or [13]). Syswerda [38] compared uniform and one-point crossover in genetic algorithms. Genetic algorithms and evolution strategies have been compared on well-studied test problems [15], [36]. Other comparisons have been made concerning genetic algorithms and simulated annealing [20], [30] or point-based optimization and population-based optimization [29].

Fogel [13] stresses the need for well-constructed experimental designs, which are necessary for assessing the believability of null hypotheses. However, it is also frequently acknowledged that numerous experiments should be made to reduce the variability of the EA's final measures. This common belief has the following consequence: few instances of EA's parameter are usually compared, as each instance receives too much attention.

Using sophisticated statistical methods to assess or predict the performance of EAs is not a new idea. Statistical experimental design techniques have been considered many times. Reeves and Wright [32], [33] used linear models to treat epistasis in genetic algorithms. They viewed the genes as variables from which predictions can be drawn. Lis and Lis [26] used the Latin square experimental design to choose parameters for genetic algorithms. This has been done so that the best combination of values could be determined using the fewest number of experiments. Bergeret and Besse [5] compared genetic algorithms and simulated annealing using the standard analysis of variance technique on a problem related to quality control. Jian *et al.* [21] used a similar method to compare selection procedures in a niching context.

However, all these methods have the drawback of assuming Gaussian statistical distributions for the EA performance results. Such an assumption has been rejected many times in our experiments (see Section IV-B), outlining the need for other models.

This paper revisits the trial and error method, using statistical models to build a functional relationship between the expected final result of an EA and the parameter values. Attention is paid to the statistical hypotheses under which meaningful conclusions can be drawn. These hypotheses are strongly context dependent and may change whether different EAs or problems are considered.

In our work, the Gamma distribution seems more realistic than the Gaussian distribution with respect to experimental results. In this context, the methodology of standard statistical design (see e.g.,[25]) can be reused, within a generalized linear model framework [27]. Such an approach provides a new way of investigating the relationship between the parameters of an EA and its performance.

B. Method Description

In defining a problem-solving framework, specifying a representation and an objective (fitness) function is a first step. Then, the user chooses the EA's operators and attempts to tune the parameters. Parameter tuning based on trial and error is time consuming. To reduce the computational cost, one parameter is usually tuned at a time.

This section describes a method to treat experimental results obtained for a number of test problems when all parameters are adjusted simultaneously. Therefore, each parameter setting x_ψ corresponds to a cell in which a number of simulations are run according to some predefined experimental design (ψ symbolizes the parameters). The method involves two stages:

1) a problem level in which each problem receives separate attention;
2) a group level in which a set of problems is considered in a global way.

1) Problem-Level Model: At the problem level, a functional relationship between the expected performance of the EA and its parameter values can be considered

$$E[y] = h(\beta_0 + \varphi(x_\psi)) \tag{1}$$

where

y	performance measure;
x_ψ	parameter setting;
h	link function, whose choice is related closely to the distribution [and the variance $V(y)$] of the measures;
β_0	intercept coefficient;
φ	parametric *predictor*.

The performance measure y may be the fitness value at the end of the run (after a given number of fitness evaluations). In this case, the measure is usually a continuous value. However, the measure may also be an integer value, such as the number of fitness evaluations needed to reach a fitness level.

Equation (1) is called a statistical model. Many standard options are available to define such a model. For example (see [27]):

1) h is the identity function and $V(y) = \sigma^2$. This corresponds to the classical linear model (observations are assumed to be distributed according to Gaussian distributions);

2) h is the exponential function and $V(y) = E[y]$ (y is an integer value). This is a log-linear model corresponding to the Poisson distribution;
3) h is the exponential function and $V(y) \propto E[y]^2$. This is a generalized linear model associated with the Gamma distribution and the link is logarithmic.

Usually, φ is taken as a (multidimensional) polynomial and may include complex interactions between the parameters. The way the coefficients of this polynomial can be estimated depends on which family of distributions are the most relevant to the measures (Gaussian, Poisson, Gamma, or others). Deciding which family should be taken is therefore a critical step in building a model.

After the family has been fixed, preliminary models can be fitted. Going further, these models can usually be improved thanks to classical testing procedures. Starting with overfitting polynomials and pruning is a frequently used technique. In such a procedure, coefficients that correspond to small t-values can be removed, yielding a nested model. The significance of these coefficients is actually assessed thanks to the difference of residual deviance between the biggest model and the nested submodel (see the Appendix). At the end of the procedure, all predictor coefficients are significant. In addition, model accuracy can be computed thanks to the ratio

$$\rho = \frac{\text{Null Deviance} - \text{Residual Deviance}}{\text{Null Deviance}}.$$

Once the model has been determined, optimal parameter values ψ_{opt} can be estimated easily as well as confidence intervals around these values. Estimating optimal parameters should be done with the fewest number of experiments. Deciding whether the experimental design contains redundant data is a similar issue. This issue can be addressed by studying the robustness of estimations using reduced data sets. These designs can be created by removing a fraction of the data from the original data set. If the estimations obtained from reduced data sets are similar to those obtained from original data sets, then models can be considered as robust. Also, this provides a way of deciding which designs should be used in further experiments of the same nature.

2) Global Models: At a second level, global models should be considered. These models will correspond to measures recorded for several test problems. In this context, new relationships should be built, with problem labeling as a new variable

$$E[y] = h(\beta_0 + \varphi(x_\psi) + \alpha_F + \pi_F(x_\psi)). \tag{2}$$

In the statistical jargon, the problem label F corresponds to a factor. The coefficient α_F measures the influence of label F on the EA performance and π_F is a parametric functional that models the interactions between F and the parameters. Parameter settings may be optimal for two problems with different overall performances. In this case, the difference will be recorded into α_F. Up to this factor effect, defining a "class" of problems $\mathcal{F}$ for which the EA behaves similarly can be achieved by testing the null hypothesis

$$H_0\colon \pi_F \equiv 0, \qquad F \in \mathcal{F}.$$

Consider now a different perspective: the EA has been experimented on a number of test problems and a reference class has been identified successfully together with a reference model. A new problem is now under consideration. To test for the membership of the new problem to the reference class, a (small) number L of reference problems (e.g., $L \leq 4$) can be picked up randomly from the identified class and a statistical model is fitted using the L reference problems plus the new problem data sets. Therefore, this new model can be compared to the reference model. The membership is rejected if the new model differs from the reference model significantly. As far as possible, the smallest number of data should be used during the procedure.

Following this method, classes are identified through the repeated use of statistical tests. Choosing confidence levels associated with the tests is a key problem. In the forthcoming experimental analysis, significance levels will be kept very low, typically ranging from 0.01–0.001.[1] To identify classes, only rough features of EA's behavior should be captured. Low significance levels allow selecting the most statistically significant model coefficients. In contrast, high significance levels would be useful to detect fine behaviors, which may be specific to each problem.

III. Experimental Context

As this paper illustrates a method to deal with experimental results, specific sets of problems and algorithms will be focused on. Our statistical analysis will, therefore, be contextual and none of the results obtained thereafter could be applied to different problems or algorithms without experimenting anew.

Two sets of test problems and a test EA will be described. The first set of problems is drawn from a classical test suite. The second set consists of least square approximation problems using feedforward neural networks. The EA has two static parameters: the mutation step size (radius) and the fraction of offspring of mutation.

As underlined by Eiben *et al.* [11], parameter control may be a more efficient way to obtain good solutions for continuous minimization problems. However, parameter control schedules may themselves depend on internal parameters and some tuning may also be necessary for these parameters. A static EA has been chosen so that the discussion is more clear and the results easier to interpret, but the method can be useful for dynamic parameters as well.

A. Test Problems

1) Standard Test Problems: The first test set consists of four frequently studied minimization problems, arbitrarily picked up from a classical long list [10], [13], [37]. These four problems (see Table I) depend on a multidimensional variable a with component range in $I = (-5.12, +5.12)$. Problem f_1 corresponds to a quadratic function: the *sphere function*. Especially in low dimensions, the sphere problem is widely recognized as being easy for EAs. Three types of perturbation of the sphere function are also studied. Problem f_2 is a piecewise constant version of this problem. A random noise perturbation is considered in problem f_3 whereas the perturbation is deterministic in problem f_4.

[1] The significance level is the probability of rejecting the null hypothesis when it is true. We want this to happen very rarely.

TABLE I
FIRST SET OF TEST PROBLEMS $\lfloor a \rfloor$ DENOTES THE INTEGER PART OF THE SCALAR a. THE Ujs ARE i.i.d. SAMPLES FROM THE UNIFORM DISTRIBUTION OVER $(0, 1)$ (MODIFIED AT THE jth EVALUATION)

label	function
f_1	$\sum_{i=1}^{20} a_i^2$
f_2	$\sum_{i=1}^{20} \lfloor (20a_i + 0.5) \rfloor^2$
f_3	$\sum_{i=1}^{20} a_i^2 + 0.5U_j$
f_4	$\sum_{i=1}^{20} a_i^2 - 4.6\cos(\pi a_i/2) + 4.6$

As a test suite should emphasize different aspects of behavior, an EA is expected to produce various types of responses on this series. In Section V, the experimental analysis will actually show that the behavior of the EA looks significantly different for each of these problems.

2) Approximation with Neural Networks: The second test set used to illustrate the method consists of least square approximation problems: one-dimensional (1-D) curves are approximated by artificial neural networks (ANNs). ANNs have well-known features. Local minima and plateaus make the local numerical methods (back-propagation and variants) converge slowly. EAs have often been applied as alternative methods to solve the least square fitting of ANNs [24], [42]. Nevertheless, the rules used for adjusting EA parameters are seldom mentioned. The output of the ANN is defined as follows. For all input u in $[-1, +1]$, one has

$$g(u) = w_{0o} + \sum_{j=1}^{4} w_{jo} s(w_{0j} u + w_{jj})$$

where s is the logistic function

$$s(u) = \frac{1}{1 + \exp(-u)}$$

and the w_{ij}s are unknown weights that take their values in a fixed interval $I = [-5.0, +5.0]$.

Given a sample $(u_k)_{1 \le k \le n}$ of size $n = 100$, the least square error is defined as

$$\mathrm{Err}(w) = \frac{1}{n} \sum_{k=1}^{n} (g(u_k) - f(u_k))^2.$$

The approximation problem consists of finding the w_{ij}s that minimize $\mathrm{Err}(w)$.

Four curves $f = f_i, i = 5, \ldots, 8$, are considered. Their plots are displayed in Fig. 1. Here, the minimization problems are in 13 dimensions. In what follows, approximation errors Err will be labeled $E_i, i = 5, \ldots, 8$.

These equations define a three-layered perceptron with four hidden units, a single input, and a single linear output. A three-layered perceptron can implement a universal approximating function (see [34]). However, since the w_{ij}s are of bounded range and the number of hidden units is fixed, the previous claim may be false. Hence, the difficulty of evaluating EA performance is increased by the fact that the best approximators are unknown.

Least square approximation of 1-D curves is, however, intuitive. Fig. 1 suggests that the approximations of $f_5, \ldots, f_8$ (especially with a model of fixed size and structure) are nearly of the same complexity. Any optimization algorithm is, therefore, expected to behave similarly for each approximation problem. Problems $E_5, \ldots, E_8$ are thus likely to constitute a reference "class" for further approximation problems of the same nature. The statistical analysis will confirm the intuition and yield unique optimal parameter settings for all problems.

B. Algorithms

The studied EA is a mutation/selection algorithm called MOSES [17]–[19] based on a two-component parameter

$$\psi = (r, p).$$

The two components are called the *radius* (mutation step size) and the *mutation probability*. The population consists of m solutions ($m \ge 2$).

The algorithm works as follows. The population at generation t is represented by the matrix $A = (a_{ij})$, where a_{ij} denotes the jth coordinate of the ith solution a_i, $1 \le i \le m$, $1 \le j \le D$. The population is initialized randomly. To construct the population at generation $t + 1$, a random number N is drawn according to the binomial distribution $\mathcal{B}(m - 1, p)$. Then, the N first rows of A are transformed by adding random variable U to a random coordinate j of a_i

$$a'_{ij} = a_{ij} + U, \qquad i \in [1, \ldots, N].$$

The variable U is uniformly distributed over the interval $[-r, r]$, where r is the mutation radius and the new solution a'_i is conditioned to stay in the definition subset. The $m - N$ remaining rows (or individuals) are replaced by the best solution obtained at generation t. This procedure is iterated until depletion of the computing resource (a fixed number of fitness evaluations).

This version of the algorithm is actually elitist. Among the m individuals, only $m-1$ are likely to undergo a mutation. The last individual is a witness of the best solution found so far. For sake of clarity, the population size m will be kept fixed ($m = 11$). This can be motivated by the following arguments.

1) The mean number of mutations at each generation would be easier to interpret than the parameter p and should have been considered instead of p as a variable. Nevertheless, this variable equals $p(m - 1)$. Fixing m and varying p is a convenient way of measuring its impact.
2) Parameters m and p strongly interact and the way they do so is clear. Keeping the population size fixed will make model choice easier, as p and r are more likely to be independent parameters.

Furthermore, it has been checked that increasing m by a factor of three has no significant effect on the performance (for the eight problems considered in this paper).

IV. MODELS

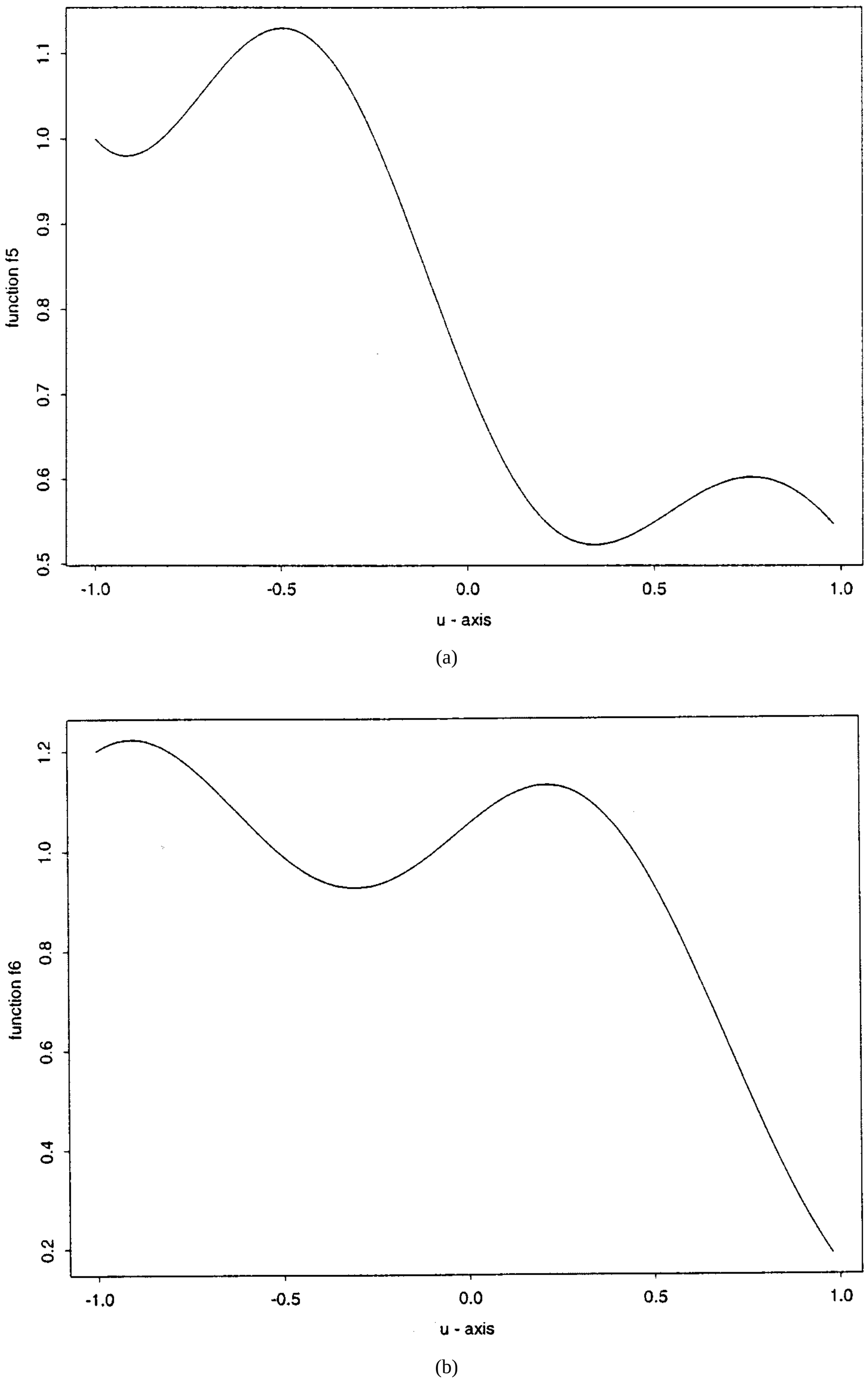
A. Basic Experimental Design

In order to identify models, the following experimental designs have been used. The EA was stopped after 2000 fitness evaluations for the first family of test problems, and after 5000 evaluations for the second one. The parameter r was discretized as

$$x_{r,i} = 0.2i, \qquad i = 1, \ldots, 20$$

(i.e., ranged between 0.2–4.0). The second parameter p was discretized as

$$x_{p,i} = 0.01 + 0.05i, \qquad i = 0, \ldots, 9$$



(a)

(b)

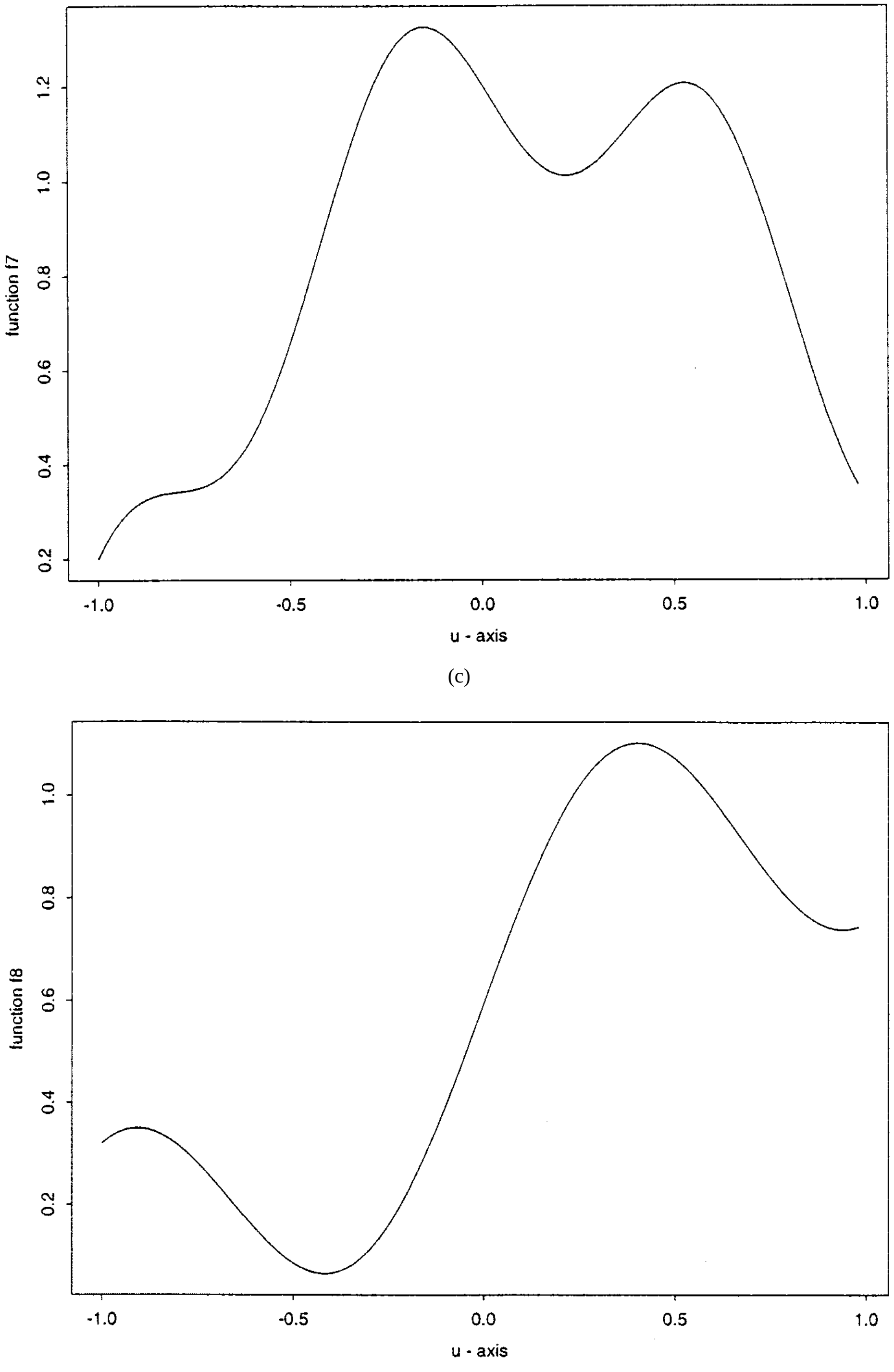
Fig. 1. (a), (b), (c), and (d) Plots of curves f_5, f_6, f_7, f_8. $f_5(u) = e^{-v}(1 + 0.5\ \sin^2(5v))$, $f_6(u) = \sin(\pi v) + 0.2\ \cos(5\pi v) + 0.3v + 0.23v^2$, $f_7(u) = 1 - v^2 + 0.2\ \cos(3\pi v) + ve^{-v}$ and $f_8(u) = 0.23\ \cos(3\pi v) - 0.23v^2 + 1.3v$, where $v = (1 + u)/2$, $u \in [-1, +1]$.

(i.e., ranged between 0.01–0.46) for the first family of problems, and

$$x_{p,i} = 0.1i, \qquad i = 1, \ldots, 10$$

for approximation problems. These experimental designs consist of 200 cells, arranged on a regular lattice. For the first set of problems, each cell has been given 10 repetitions and the full data set contains 2000 data. For approximation problems, each cell has been given five repetitions and the full data set contains 1000 data. Obtaining these data sets required a few minutes of computation (IBM workstation) for the classical test problems and two or three hours for approximation problems.

In all experiments, final fitness values have been taken as performance measures for the EA.



(c)

(d)

Fig. 1. *(Continued.)* (a), (b), (c), and (d) Plots of curves $f_5,\ f_6,\ f_7,\ f_8$. $f_5(u) = e^{-v}(1+0.5\ \sin^2(5v))$, $f_6(u) = \sin(\pi v) + 0.2\ \cos(5\pi v) + 0.3v + 0.23v^2$, $f_7(u) = 1 - v^2 + 0.2\ \cos(3\pi v) + ve^{-v}$ and $f_8(u) = 0.23\ \cos(3\pi v) - 0.23v^2 + 1.3v$, where $v = (1+u)/2$, $u \in [-1, +1]$.

B. Modeling Assumptions

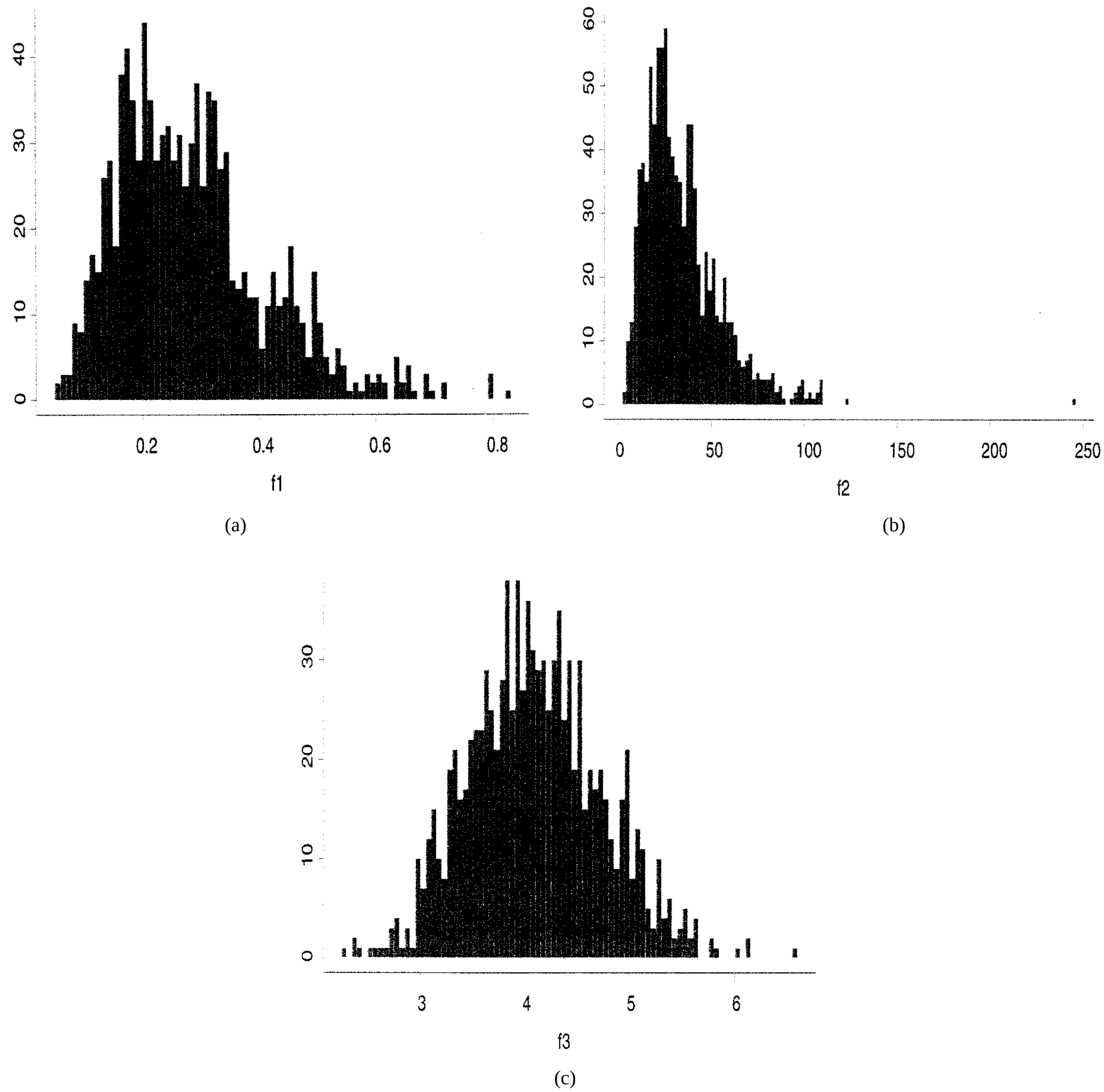
In building statistical models such as (1) or (2), choosing the link function h and the predictor φ are the critical steps. Such choices and the way the predictor can be fitted depend on which family of distributions is the most relevant to the data.

Statistical data analysis often relies upon the assumption that distributions are Gaussian [25]. In what follows, Gamma distributions are compared to Gaussian distributions and appear to be better suited. To validate the comparison, 1000 simulations were done with parameters

$$r = 1.2, \quad p = 0.3$$

for problems 1, 2, and 3, and

$$r = 1.5, 3.0, \quad p = 0.2$$



(a)

(b)

(c)

Fig. 2. Histograms of the performance measure for problems (a) f_1, (b) f_2, and (c) f_3 by MOSES. $r = 1.2, p = 0.3$ (1000 repetitions).

TABLE II
MEAN AND STANDARD ERROR FOR THE PERFORMANCE MEASURES ASSOCIATED WITH PARAMETERS $r = 1.2$, $p = 0.3$ (1000 REPETITIONS) AND PROBLEMS $f_1, \ldots, f_4$ (2000 FITNESS EVALUATIONS). FOR f_4, TWO DIFFERENT RADIUS SETTINGS ARE USED

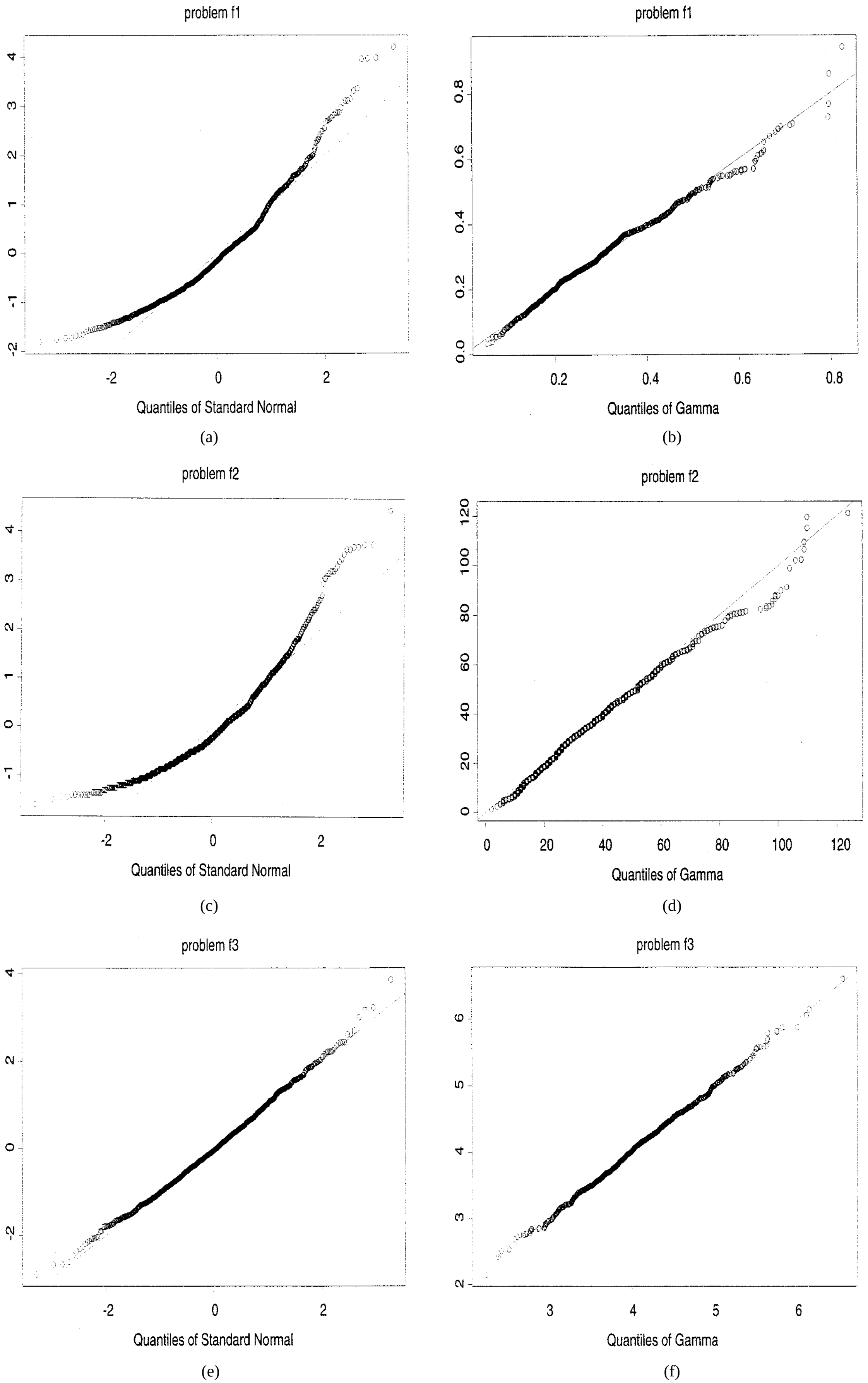
	f_1	f_2	f_3	f_4 $(r = 1.5)$	f_4 $(r = 3.0)$
mean	0.285	34.73	4.100	4.732	1.852
std. err.	0.129	21.22	0.637	6.021	0.871

for problem 4. A rapid investigation showed that these parameter settings were typical for running the EA. Although improvable, good results were obtained in short simulation time.

In contrast with Gaussian distributions, Gamma distributions are asymmetrical and have long tails. Except for problem f_3, the mean and standard deviation of the data are nearly of the same magnitude (Table II). The histograms (Fig. 2) show asymmetrical empirical distributions. Moreover, empirical distributions are actually "heavy-tailed": large deviations from mean values can be observed.

A graphical method has been used to compare the Gaussian and the Gamma hypotheses for all data sets. The method plots the quantiles of the performance sample against those of the reference distribution (Gaussian and Gamma), the fit being highest as the plots are aligned (Fig. 3). Gamma distributions have been validated for problems $f_1, \ldots, f_3$ (confirmed by the Kolmogorov–Smirnoff test).

For problem f_4, two radii have been experimented. When $r = 1.5$, the histogram has a peaked aspect (Fig. 4) and the average performance is poor. Better performances have been obtained when $r = 3.0$. The empirical distribution is then consistent with the others. The first parameter setting ($r = 1.5$) is obviously suboptimal: the performance is strongly correlated to the initialization of the EA, as the population gets trapped in the minimum closest to the initial population. Nevertheless, the convex hull of the histogram seems consistent with the other problems and conditionally to each peak, the shape of the empirical distribution is similar to those obtained in the former experiments. For f_4, the Gamma distribution seems



(a)

(b)

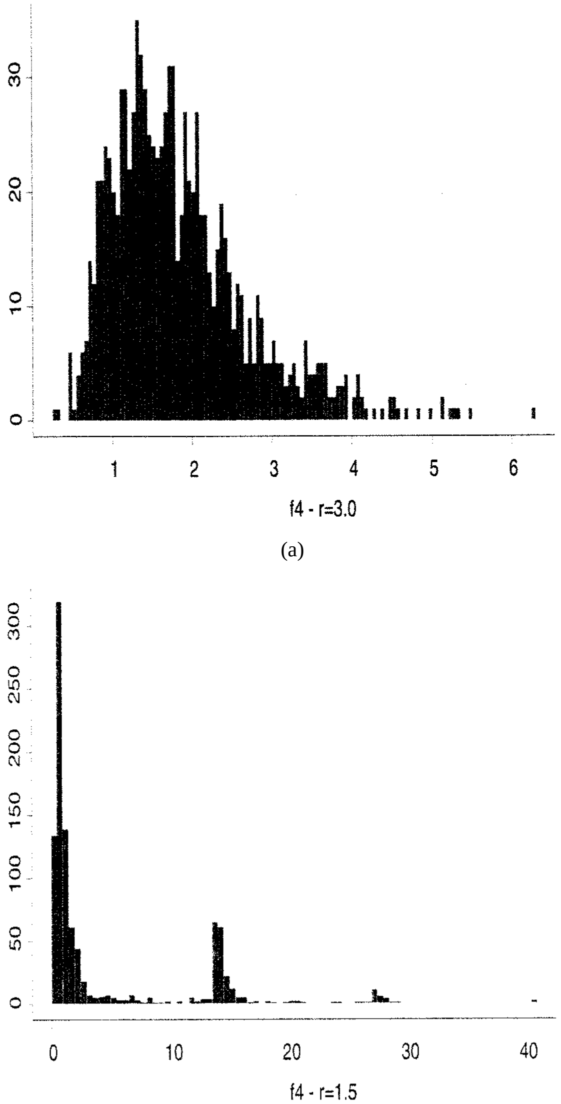
(c)

(d)

(e)

(f)

Fig. 3. (a), (c), and (e) Qqnorm and (b), (d), and (f) qqplot of performance measure for problems f_1, f_2, and f_3. [The qqnorm and qqplot methods plot the quantiles of the performance sample against those of the reference distribution (Gaussian and Gamma), the fit being highest as the plots are aligned.]



(a)

(b)

Fig. 4. Histograms of the performance measure for problem f_4 by MOSES. (a) $r = 3.0$ and (b) $r = 1.5$ with $p = 0.2$ (1000 repetitions).

more suited to the measures than the Gaussian distribution (Fig. 5).

Turn now to the choice of a predictor. Because r and p are likely to have independent effect on the performance, additive functionals are the natural assumption

$$\varphi(x_{(r,p)}) = \varphi_r(x_r) + \varphi_p(x_p)$$

with φ_r and φ_p taken as 1-D polynomials (of unknown order)

$$\varphi_r(x_r) = \beta_{1,r}x_r + \beta_{2,r}x_r^2 + \cdots$$

and

$$\varphi_p(x_p) = \beta_{1,p}x_p + \beta_{2,p}x_p^2 + \cdots.$$

To validate the independence hypothesis, graphical techniques have also been utilized. These techniques merely consist of plotting the average performance as a function of r for fixed p. In doing so, the experimental design described in Section IV-A has been used. The results are displayed in Fig. 6. The ten performance curves do not intersect, giving evidence that interactions between p and r are weak. Similar plots have been obtained for all problems.

In summary, the Gamma distribution appears to be a relevant choice in modeling the performance measure for problems $f_1, \ldots, f_4$. The Gamma hypothesis has also been assessed for approximation problems leading to similar results (Fig. 7). Denoting y as the measure corresponding to x_ψ, the model attempts a relationship between the mean performance of the EA and the predictor through a logarithmic link (see Section II-B1)

$$E[y] = \exp(\beta_0 + \varphi_r(x_r) + \varphi_p(x_p)). \tag{3}$$

The logarithmic link warrants that the mean is nonnegative (all test problems take nonnegative values). Moreover, estimation procedures have to work with small and large values at the same time. The logarithmic link ensures a better numerical behavior of these numerical procedures (see the Appendix).

As concern group level models, the new predictor can be given by

$$\eta = \beta_0 + \varphi(x_\psi) + \alpha_F + \pi_F(x_\psi). \tag{4}$$

Recall that α_F measures the influence of the problem label F on the expected score and

$$\begin{aligned}\pi_F\left(x_{(r,p)}\right) &= \beta_{1,rF}x_r + \beta_{1,pF}x_p + \beta_{2,rF}x_r^2 + \beta_{2,pF}x_p^2 + \cdots\\ &= \pi_{rF}(x_r) + \pi_{pF}(x_p)\end{aligned}$$

can be taken as a sum of independent 1-D polynomials.

V. Results

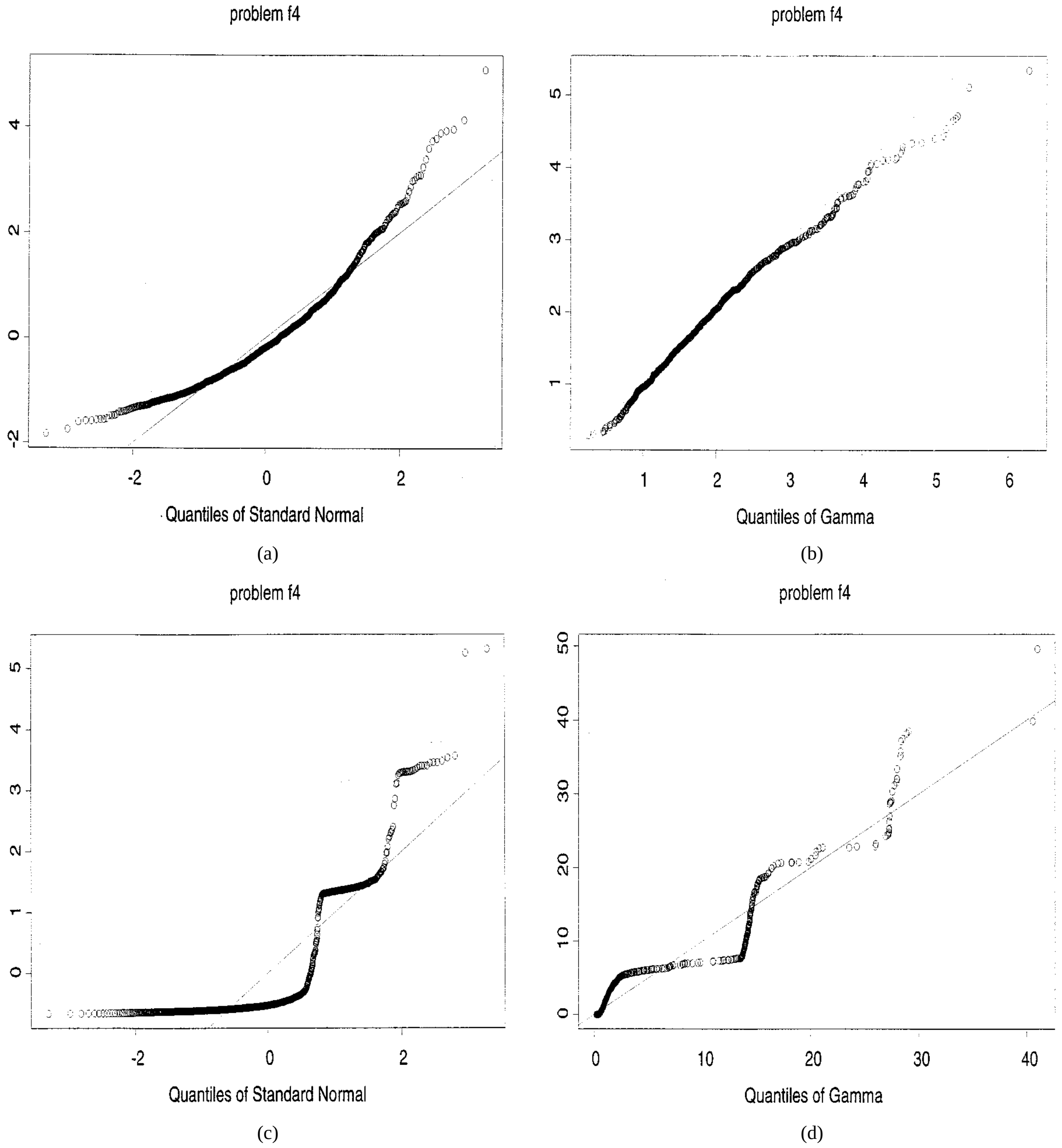
In dealing with statistical models, a specific syntax is used. This syntax is shared by many statistical softwares and also by users of ANOVA [25]. The symbol $+$ corresponds to independent additive effects and the symbol $*$ corresponds to interactions. Moreover, $\beta_0 + \varphi_r(x_r)$ is denoted by $\mathrm{poly}(r, d)$ ($1 + d$ degrees of freedom), $F + \mathrm{poly}(r, d)$ corresponds to $\alpha_F + \beta_0 + \varphi_r(x_r)$, and $F * \mathrm{poly}(r, d)$ corresponds to $\beta_0 + \varphi_r(x_r) + \alpha_F + \pi_{rF}(x_r)$. For estimation and test procedures, the software *S-Plus* has been used [6], [39]. This software gives the null and residual deviance, the fitted coefficients $\hat{\beta}$, and their t-values (see the Appendix). Graphical outputs and confidence intervals are also displayed by *S-Plus*.

A. Classical Test Problems

In this section, problem-level models are investigated for the first family of problems. The steps leading to the estimation of optimal parameters are detailed for problem f_1. A reduced experimental design is also described, which will be validated for all problems in the test suite.

1) Test Problem f_1: First, the order of the predictors φ_r and φ_p must be determined. The method starts with an overfitting model

$$M_0\colon \mathrm{poly}(r, 10) + \mathrm{poly}(p, 8).$$



(a) (b) (c) (d)

Fig. 5. Qqnorm and qqplot of performance measure for problem f_4 with two radii. (a) and (b) $r = 3.0$. (c) and (d) $r = 1.5$.

The residual deviance (Resid. Dev) is equal to 1070.545 and the number of degrees of freedom (Resid. Df) is 1981. The ration-of-likelihood test (χ^2 test) shows that less complex models can be used. For instance, consider the nested model

$$M_1\colon \text{poly}(r, 8) + \text{poly}(p, 4).$$

The χ^2 test rejects M_0 (the p-value is equal to 0.0785, and is larger than 0.001). Consider now

$$M_2\colon \text{poly}(r, 8) + \text{poly}(p, 3).$$

The analysis-of-deviance table (Table III) gives the p-value equal to 0.0228 and M_0 is again rejected. Model M_2 is, therefore, selected. It has a good accuracy: 87% of the null deviance can be explained.

Coefficient estimation is the next step. Estimations (value) are given in Table IV with their standard errors (Std. Error) and t-values. The dispersion parameter for the Gamma family can be estimated by $\phi = 0.5392$. All coefficients are significant (at level 0.001). Decreasing the order of model M_2 would, therefore, be irrelevant.

The predictors φ_r and φ_p are plotted in Fig. 8 with their confidence intervals. These plots allow computing numerical values for the optimal parameters. For problem f_1, these values are

$$r^1_{\text{opt}} \approx 1.0 \quad p^1_{\text{opt}} \approx 0.01.$$

Confidence intervals around the predictors are tight. This indicates that the number of data per cell may be reduced. To assess the value of this claim, estimation and testing procedures have been performed again, keeping a single data in each cell.

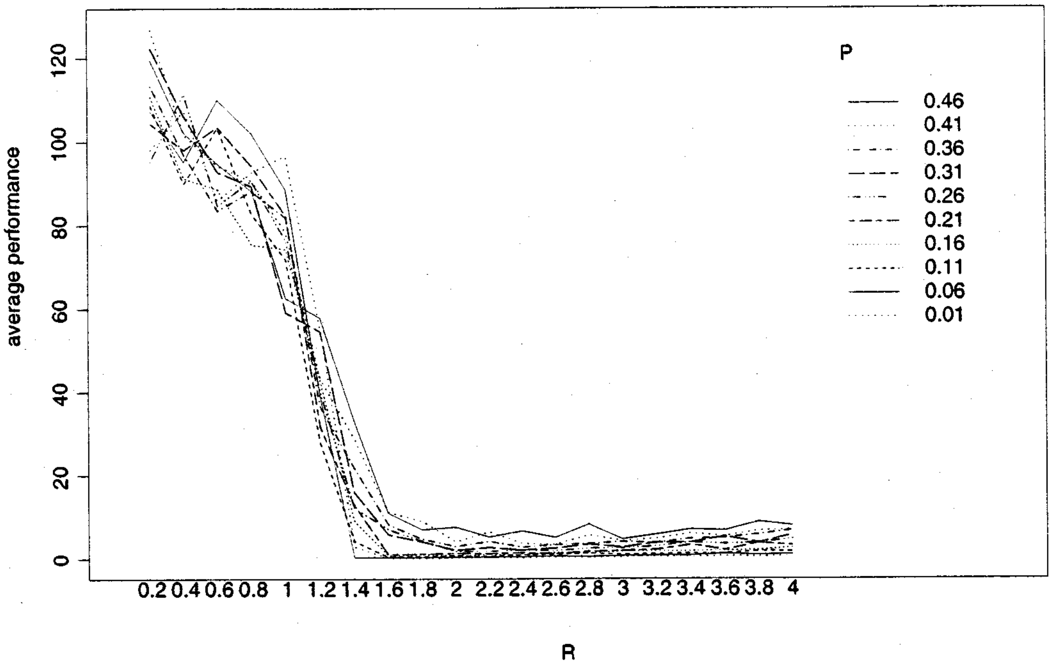


Fig. 6. Interaction plot for problem f_4 (2000 evaluations).

Henceforth, the data set contains 200 data. In order to obtain reliable conclusions, the procedure has been restarted from five reduced subsets. In each case, the resulting model is compared to the overfitting model $\text{poly}(r, 9)+\text{poly}(p, 4)$ (Table V). The resulting models actually differ from the model built with all data. However, Figs. 8 and 9 exhibit similar predictor curves and that optimal parameter settings are unchanged.

2) Problems $f_2, \ldots, f_4$: The same steps as before have been carried out for problems $f_2, \ldots, f_4$. For problem f_2, the selected model is $\text{poly}(r, 6) + \text{poly}(p, 4)$ (2000 data). With 200 data, the selected model writes $\text{poly}(r, 5) + \text{poly}(p, 2)$. For f_3, the models are $\text{poly}(r, 7) + \text{poly}(p, 3)$ (2000 data) and $\text{poly}(r, 6) + \text{poly}(p, 2)$ (200 data). Figs. 10 and 12 give the fitted predictors. Plots resulting from reduced data sets are displayed in Figs. 11, 13 and 14. Optimal parameter settings for problems $f_1, \ldots, f_4$ are given in Table VI.

3) Group Models: This section investigates reference classes of problems among $\{f_1, \ldots, f_4\}$. To identify all classes, all pairs must be inspected.

To fix ideas, consider the pair $\{f_1, f_2\}$. According to Section V-A1 and Section V-A2, f_1 and f_2 seem to exhibit similar features, such as similar performance plots or optimal parameters. To decide whether these problems are indeed in the same class, new models must be identified using a paired data set (4000 data). For doing so, a two-level factor F can be introduced. At the end of the selection procedure, φ_r and φ_p are of respective order ten and four. Table VII displays the residual deviances for four different models. The three first models include at least one interaction between (r, p) and F. The last model (model 4) contains no interaction. Model 1 is the complete model. The χ^2-statistic (model 2 versus model 1) equals 46.73. However, the probability that a χ^2-statistic with ten degrees of freedom exceeds 46.73 is equal to 1.06e-006. Interactions between F and r are, thus, significant (at level 0.001). For the other models (models 3, 4), the values of the test statistics are even worse and the null hypothesis is again rejected. The "class" hypothesis can therefore be rejected.

The same steps have been carried out for all pairs $\{f_i, f_j\}$. Each time, two nested models have been considered ($M_1 \subset M_0$). Model M_0 writes

$$M_0\colon F * (\text{poly}(r, 10) + \text{poly}(p, 4))$$

(3984 degrees of freedom). Model M_1 is defined as

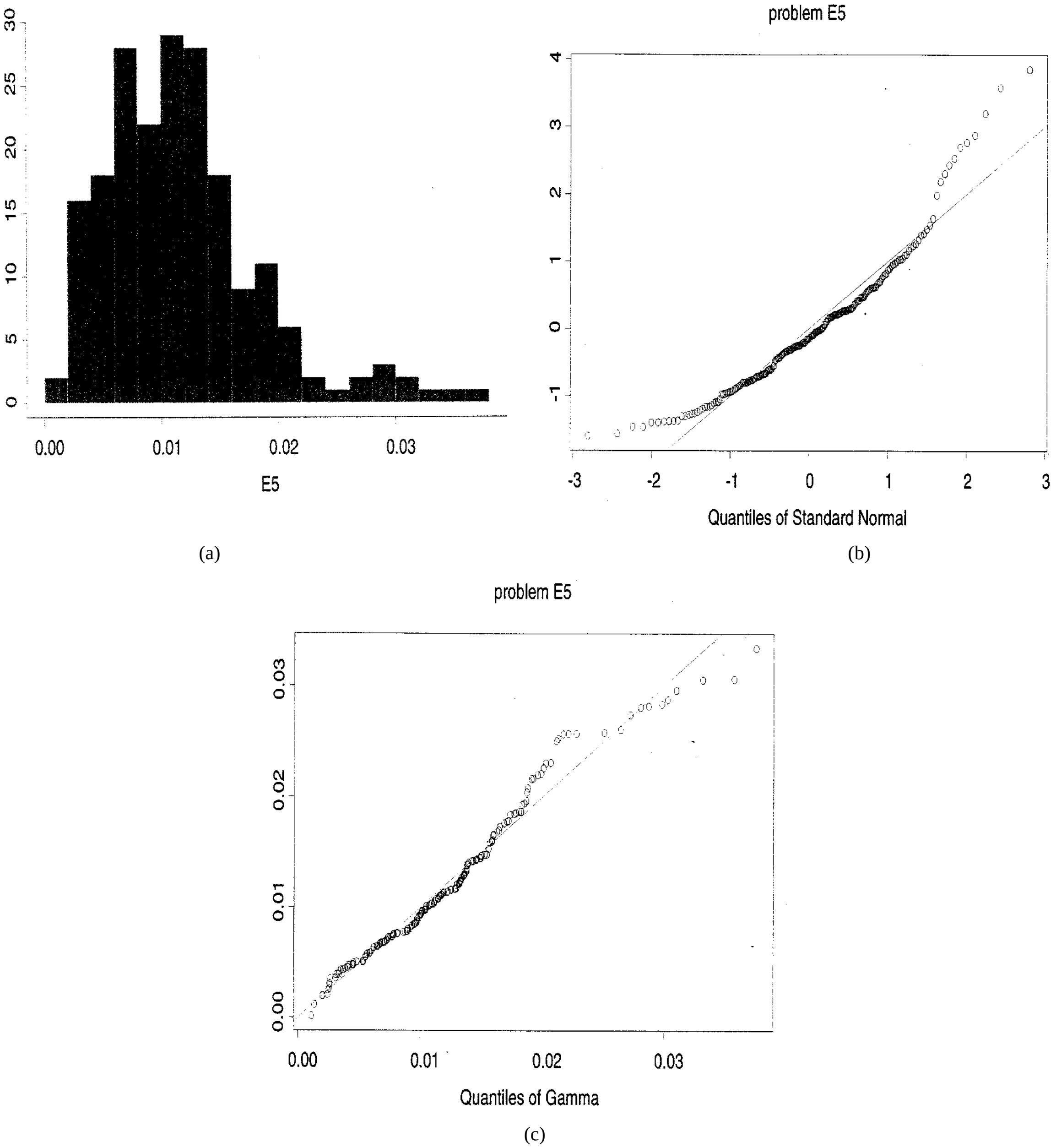
$$M_1\colon F + \text{poly}(r, 10) + \text{poly}(p, 4)$$

(3970 degrees of freedom). Table VIII allows building the test for interactions. The smallest difference of deviance is equal to 57 and corresponds to the pair $\{f_1, f_2\}$, which has been already rejected as a class.

B. Approximation Problems

The last section failed in identifying reference classes among problems $\{f_1, \ldots, f_4\}$ due to significant differences in EA's behavior for these four problems. With regard to approximation problems, the situation is intuitively different since EAs are expected to behave similarly. This section will demonstrate that $E_5, \ldots, E_8$ can be indeed identified as a reference class. Also, individual models will not be described in this Section. This step is useless as far as global models can be directly introduced.

First, problems E_5 and E_6 will be considered simultaneously. Then, problems E_5, E_6, and E_7 will be grouped. For these problems, the effect of the problem level F will be significant, but no interaction between F and the parameters will be detected. Finally, the significance of $E_5, \ldots, E_8$ as a class will be discussed and the membership of new problems will be tested for.



(a) (b)

(c)

Fig. 7. (a) Histograms, (b) qqnorm, and (c) qqplot of performance measure for problem E_5 by MOSES. $r = 1.9, p = 0.2$, (200 repetitions with 5000 evaluations).

TABLE III
ANALYSIS OF DEVIANCE FOR f_1 WITH 2000 DATA

Terms	Resid. Df	Resid. Dev	Pr(Chi)
poly(r, 8) + poly(p, 3)	1988	1079.287	
poly(r, 10) + poly(p, 8)	1981	1070.545	0.0228022

1) Problems E_5 and E_6: This paragraph studies the set $\{E_5, E_6\}$ as a possible class. To do so, a two-level factor F can be introduced. The process starts from

$$M_0\colon F * (\text{poly}(r, 8) + \text{poly}(p, 6))$$

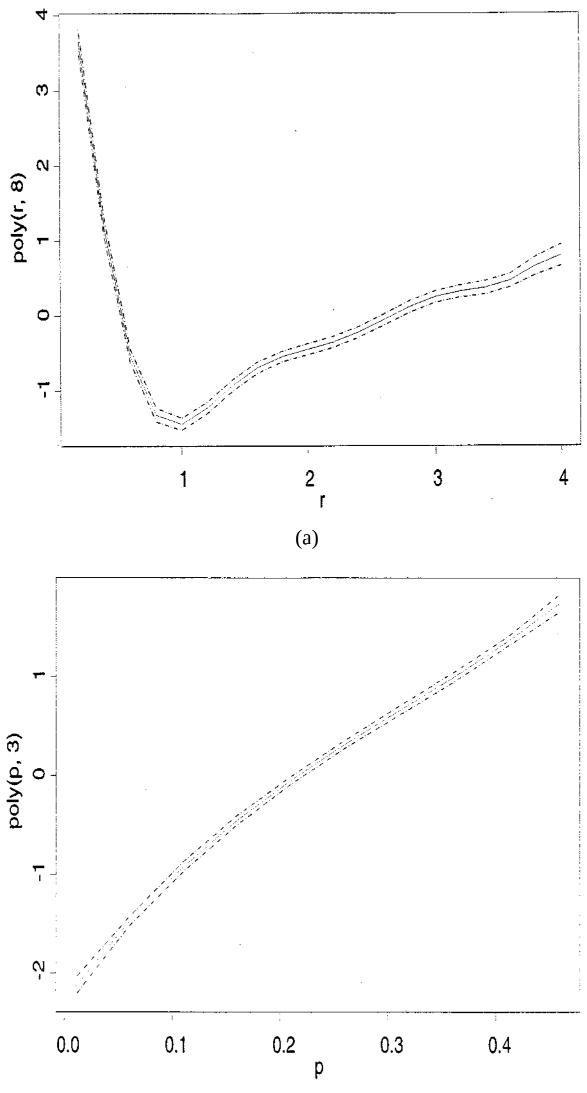
which includes all interactions between F and the parameters. When interactions are removed, the nested model writes

$$M_1\colon F + \text{poly}(r, 8) + \text{poly}(p, 6).$$

TABLE IV
ESTIMATED VALUES IN MODEL $\text{poly}(r, 8) + \text{poly}(p, 3)$ FOR PROBLEM f_1 OBTAINED FROM 2000 DATA

Coefficients:	Value	Std. Error	t value
(Intercept)	2.28192210	0.56472783	4.040747
r	-7.73218611	4.73795554	-1.631967
r^2	-31.71714555	14.08981586	-2.251069
r^3	74.61389521	20.41209940	3.655376
r^4	-67.05538019	16.34733188	-4.101916
r^5	31.70382178	7.60089218	4.171066
r^6	-8.33806859	2.04039894	-4.086489
r^7	1.15479860	0.29303819	3.940779
r^8	-0.06573634	0.01741675	-3.774318
p	15.02748255	1.06016925	14.174607
p^2	-26.56179841	5.34642148	-4.968145
p^3	28.16702601	7.47440633	3.768463

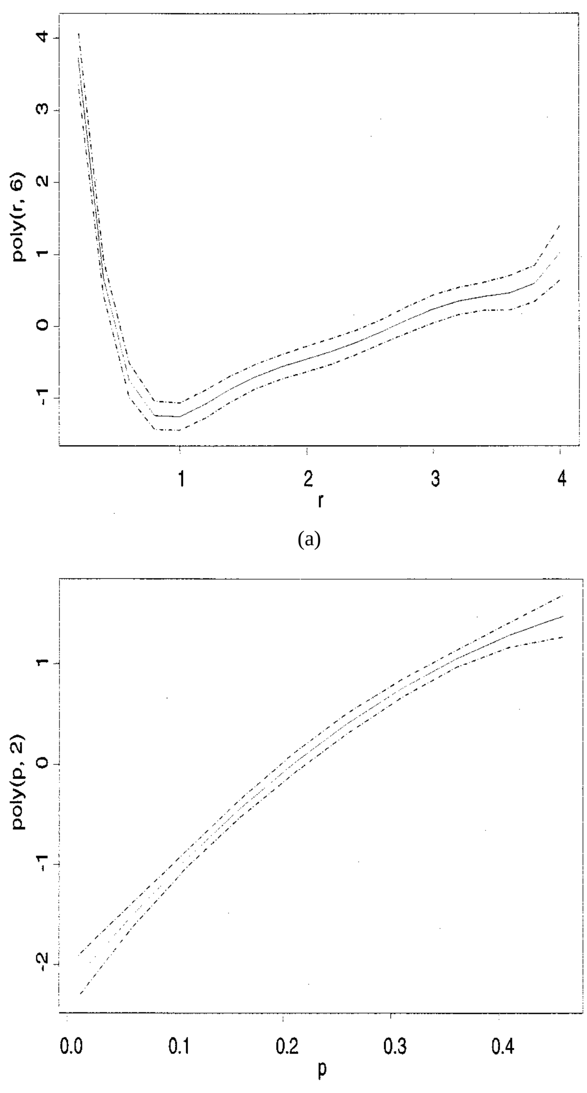
The χ^2 test actually rejects M_0. This indicates that problems E_5 and E_6 may be treated as being in the same class. According to



(a)

(b)

Fig. 8. Plots of predictors (a) φ_r and (b) φ_p for problem f_1 obtained with 2000 data.



(a)

(b)

Fig. 9. Plots of predictors (a) φ_r and (b) φ_p for problem f_1 obtained with 200 data.

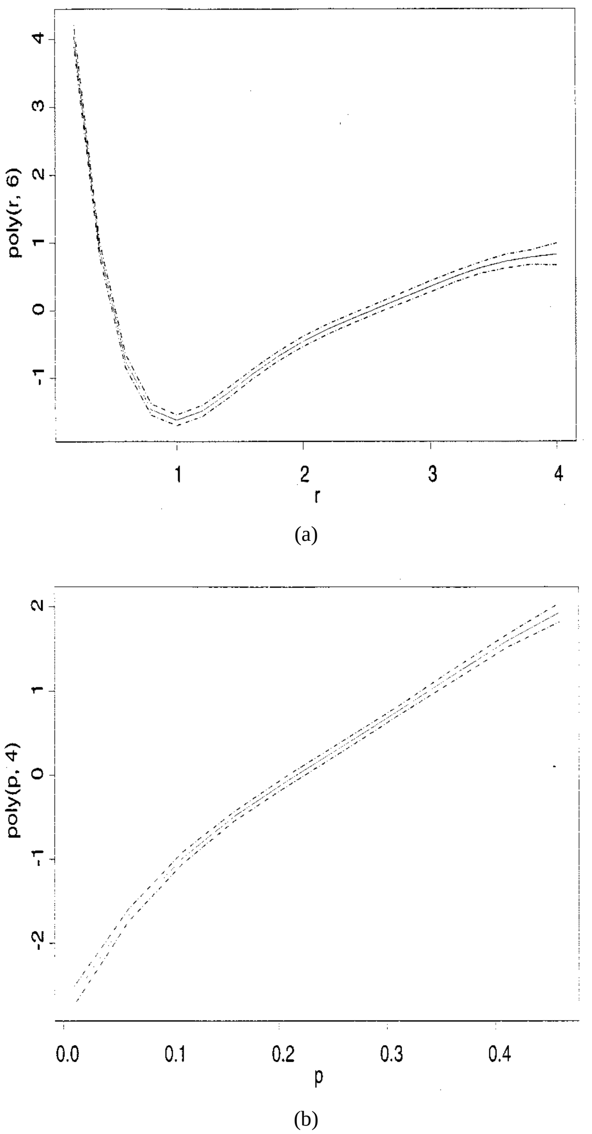
TABLE V
ANALYSIS OF DEVIANCE TABLE FOR THE PROBLEM f_1 OBTAINED FOR FIVE 200 DATA SETS

Terms	Resid. Df	Resid. Dev	Pr(Chi)
data set 1			
poly(r, 5) + poly(p, 1)	193	110.3767	
poly(r, 9) + poly(p, 4)	186	105.3896	0.2708398
data set 2			
poly(r, 6) + poly(p, 1)	192	104.9225	
poly(r, 9) + poly(p, 4)	186	100.8522	0.1929086
data set 3			
poly(r, 5) + poly(p, 1)	193	128.0421	
poly(r, 9) + poly(p, 4)	186	122.2777	0.0921381
data set 4			
poly(r, 6) + poly(p, 2)	191	97.50055	
poly(r, 9) + poly(p, 4)	186	94.18561	0.1180324
data set 5			
poly(r, 5) + poly(p, 2)	192	114.3157	
poly(r, 9) + poly(p, 4)	186	109.2602	0.2245523

Table IX, the reference model for the class $\{E_5, E_6\}$ is M_2 : $\mathrm{poly}(r, 3) + \mathrm{poly}(p, 4)$. This means that F is nonsignificant ($\alpha_F = 0$). As a consequence, the algorithm behaves identically for both problems.

2) Problems $E_5, \ldots, E_7$: In this paragraph, the set $E_5, \ldots, E_7$ is studied. The size of the corresponding data set is equal to 3000. A two-level factor F can be used again. Problems E_5 and E_6 are regrouped at the first level and the second level corresponds to E_7. The selected model is $F + \mathrm{poly}(r, 3) + \mathrm{poly}(p, 5)$. Its residual deviance is equal to 775.6885 (2990 degrees of freedom). Table X shows that the test for interactions is nonsignificant. This indicates that problem E_7 belongs to the class $\{E_5, E_6\}$. Nevertheless, the factor F has a significant effect on the performance measure.

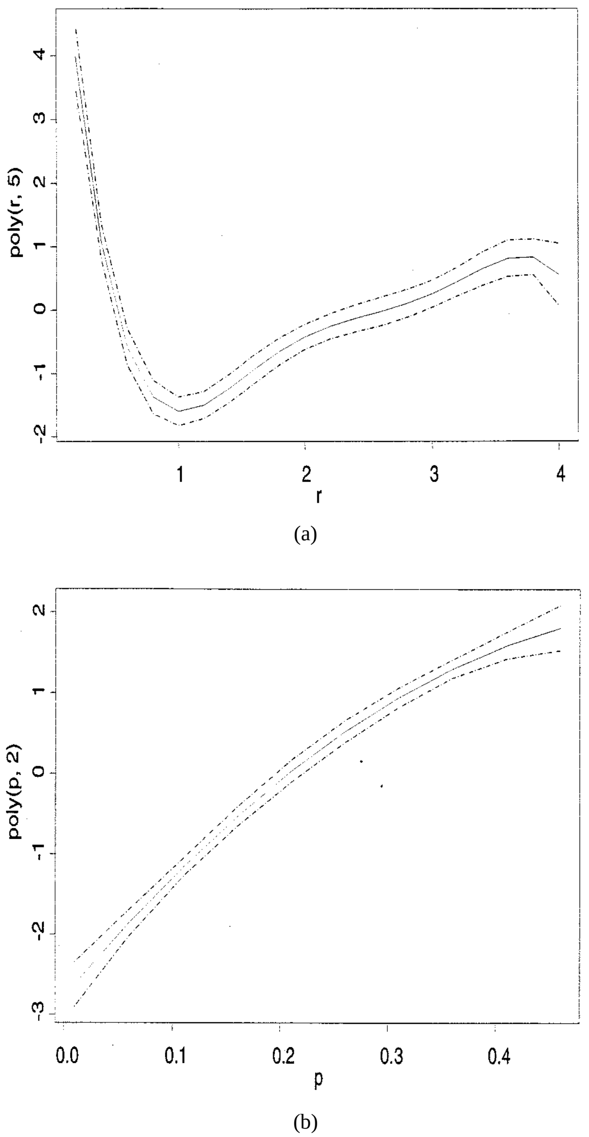
3) Group Model for $E_5, \ldots, E_8$: The full set $E_5, \ldots, E_8$ is now studied. Here, a three-level factor F can be introduced. Table XI shows that interactions between F and p are nonsignificant. When interactions between F and r are removed, small p-values are reported (Table XII). The null hypothesis is nevertheless acceptable ($0.0019 > 0.001$). A set of problems for which the algorithm behaves similarly has been thus identified. To reach this conclusion, a low significance level (0.001) must



(a)

(b)

Fig. 10. Plots of predictors (a) φ_r and (b) φ_p for problem f_2 obtained with 2000 data.



(a)

(b)

Fig. 11. Plots of predictors (a) φ_r and (b) φ_p for problem f_2 obtained with 200 data.

be used. (To compare, the significance level below which f_2 and f_3 can be regrouped is 1.0e-006.) For the reference class $E_5, \ldots, E_8$, the model writes $F + \text{poly}(r, 3) + \text{poly}(p, 4)$. The residual deviance is equal to 1099.394 with 3990 degrees of freedom and model accuracy equals $\rho = 47\%$.

4) Smaller Designs: As in Section V-A1, smaller experimental design can be created in order to investigate the robustness of the class $E_5, \ldots, E_8$. With a single datum per cell, models built from distinct data sets can differ significantly. On the other hand, when two data are used, conclusions become robust. Tables XIII and XIV display test and estimation results for model $F{+}\text{poly}(r, 3){+}\text{poly}(p, 4)$ (data set of size $4\times 2\times 200 = 1600$). The predictor plots are displayed in Fig. 15, which shows that parameter p_{opt} is between 0.1 and 0.4 and that parameter r_{opt} is between 1.3 and 2.2. Confidence intervals around these values are actually tighter than those obtained from separate models.

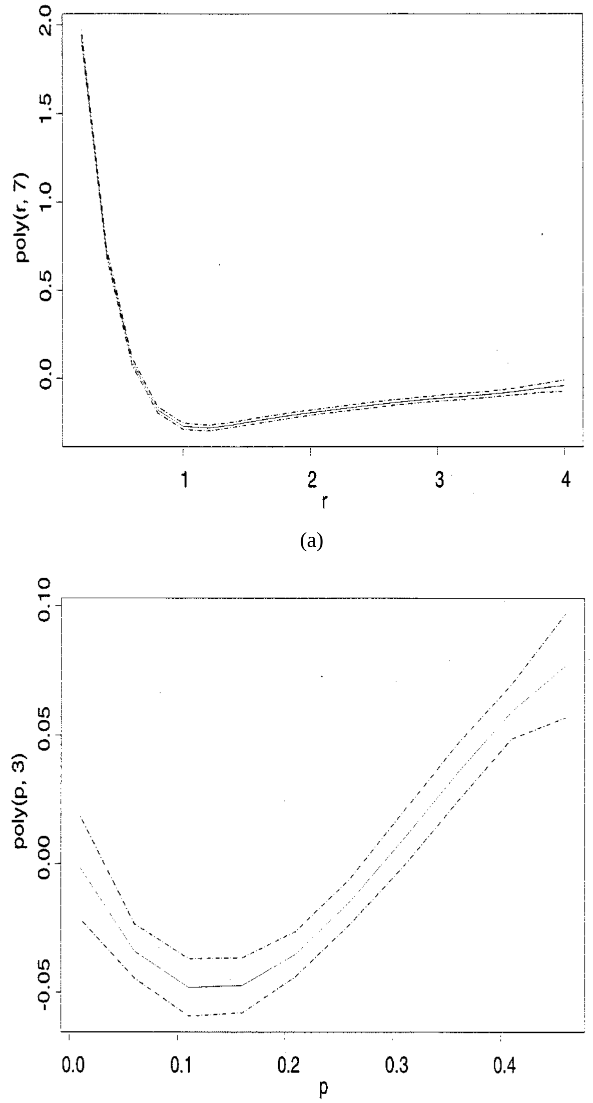
5) Test for Membership: The test has been applied to each problem $f_1, \ldots, f_4$. After the experiment using a reduced experimental design, 400 new data can be added to the previous data set ($1600 + 400$ data in all). A factor C (*class*) with two levels can then be introduced. The first level is associated to the class $E_5, \ldots, E_8$. The second level is associated to the new data. A fourth level is added to the factor F. A problem belongs to the class $E_5, \ldots, E_8$ if all interactions between C and $\psi = (r, p)$ are nonsignificant. To build the test, models

$$M_0\colon F + C * (\text{poly}(r, 8) + \text{poly}(p, 6))$$

and

$$M_1\colon F + \text{poly}(r, 8) + \text{poly}(p, 6)$$

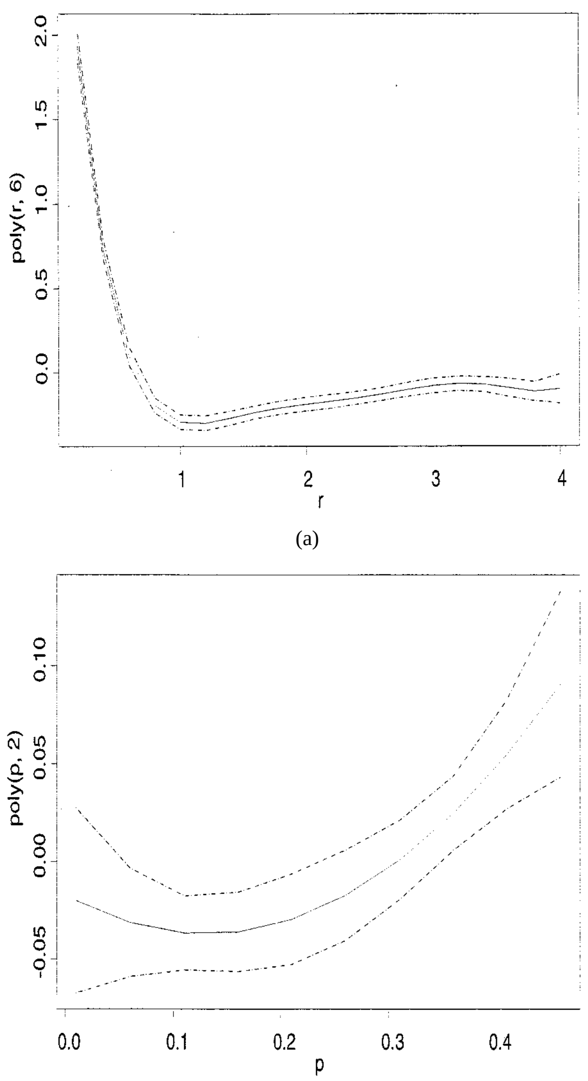
can be used. Table XV gives the residual deviance of M_0 (1968 degrees of freedom) and M_1 (1982 degrees of freedom) for all f_is. The lowest difference has been obtained for problem f_3. In this case, the test rejects M_1. As a result, none of the f_is belong to class $E_5, \ldots, E_8$.



(a)

(b)

Fig. 12. Plots of predictors (a) φ_r and (b) φ_p for problem f_3 obtained with 2000 data.



(a)

(b)

Fig. 13. Plots of predictors (a) φ_r and (b) φ_p for problem f_3 obtained with 200 data.

VI. Discussion

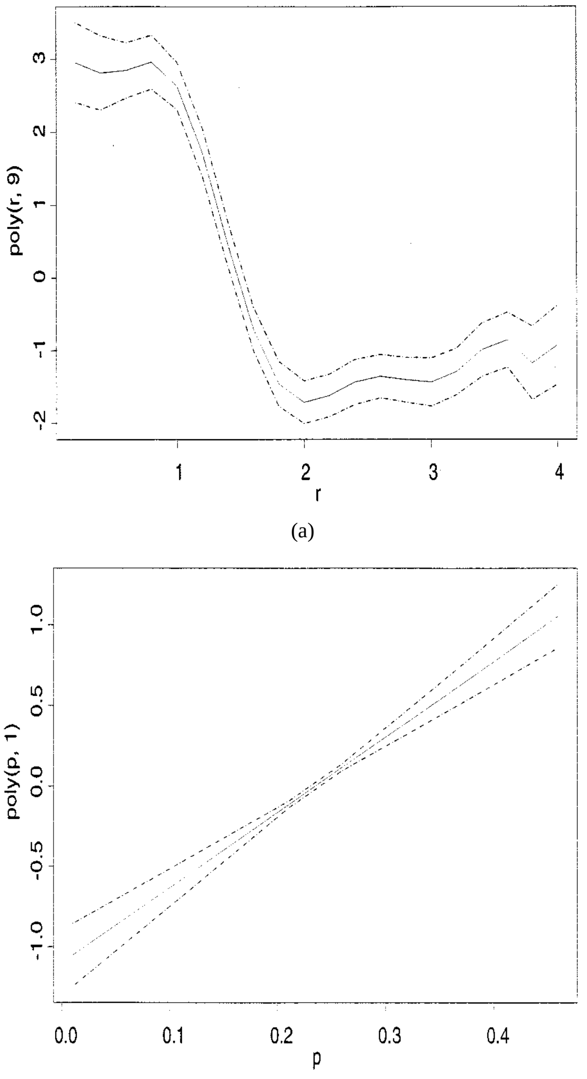
The method introduced in this paper has two merits. First, it provides useful knowledge about EA's behavior. Second, it allows reducing the computational efforts needed to study further problems, reusing and validating the knowledge obtained for a set of problems in a statistical way. In what follows, each of these two points is discussed in turn.

The two families of problems discussed in this paper emphasize two different lines of EA's behavior. In the first test set, all problems are built upon the same structure: a quadratic shape with perturbations. In problem f_4, the local minima are regularly distributed, and the attraction basins are approximately of the same volume. The experimental analysis shows that the best performances are obtained with very small mutation probabilities (around 0.01). On the other hand, local minima are not regularly located in the search space of approximation problems. In this case, the optimal mutation probabilities are not so small (about 0.2–0.3).

As often happens with static parameters, a user is faced with a tradeoff between speed and accuracy. Large mutation steps allow rapid scanning of the search space, but may lead to inaccurate solutions. On the other hand, small steps can lead to accurate solutions, but the computational cost may be prohibitive. This view corresponds to a single individual who improves through large steps to avoid local minima and small steps to gain in accuracy. For MOSES, this situation corresponds to small mutation probabilities, so that the EA resembles the $(1+1)$ evolution strategy [3]

$$X_{t+1} = \begin{cases} X_t + U, & \text{if } f(X_t + U) < f(X_t) \\ X_t, & \text{otherwise} \end{cases}$$

for all $t \geq 0$. Reaching the best performances with very small mutation probabilities indicates that using a population-based search may be useless. On the other hand, increasing the value of p is actually an efficient way for MOSES to exit from local minima. In such a case, many individuals must be involved in the dynamics, as the number of offspring by mutation can be



(a)

(b)

Fig. 14. Plots of predictors (a) φ_r and (b) φ_p for problems f_4 obtained with 200 data.

TABLE VI
OPTIMAL PARAMETER SETTINGS FOR $f_1, \ldots, f_4$ OBTAINED WITH 200 (AND 2000) DATA

problem	r_{opt}	p_{opt}
f_1	1.0	0.01
f_2	1.0	0.01
f_3	1.2	0.15
f_4	2.0	0.01

TABLE VII
ANALYSIS OF DEVIANCE TABLE FOR PROBLEMS f_1, f_2 (4000 DATA)

	Terms	Resid. Df	Resid. Dev
model 1:	F * (poly(r, 10) + poly(p, 4))	3970	2278.892
model 2:	F * poly(p, 4) + poly(r, 10)	3980	2308.677
model 3:	F * poly(r, 10) + poly(p, 4)	3974	2312.136
model 4:	F + poly(r, 10) + poly(p, 4)	3984	2335.744

greater than one. Large optimal mutation probabilities indicate that MOSES may optimally escape from minima through a series of individual mutations and not through a single optimal jump.

TABLE VIII
RESIDUAL DEVIANCES FOR MODELS M_0 : $F + \text{poly}(r, 10) + \text{poly}(r, 4)$ AND M_1 : $F * (\text{poly}(r, 10) + \text{poly}(p, 4))$. MODEL M_0 IS REJECTED IN EACH CASE

	(f_1, f_2)	(f_1, f_3)	(f_1, f_4)	(f_2, f_3)	(f_2, f_4)	(f_3, f_4)
model M_0	2336	2778	5637	3379	2860	3495
model M_1	2279	1124	2294	1259	2429	1274

TABLE IX
ANALYSIS OF DEVIANCE TABLE FOR PROBLEMS E_5, E_6 (2000 DATA)

Terms	Resid. Df	Resid. Dev	Pr(Chi)
poly(r,3)+poly(p,4)	1992	578.0422	
F * (poly(r,8)+poly(p,6))	1970	568.4354	0.196943

TABLE X
ANALYSIS OF DEVIANCE TABLE FOR PROBLEMS $E_5, \ldots, E_7$ (3000 DATA)

Terms	Resid. Df	Resid. Dev	Pr(Chi)
F + poly(r,8) + poly(p,6)	2984	772.8264	
F * (poly(r,8) + poly(p,6))	2970	767.3329	0.23860

TABLE XI
ANALYSIS OF DEVIANCE TABLE FOR PROBLEMS $E_5, \ldots, E_8$ (4000 DATA). TEST FOR INTERACTIONS BETWEEN F AND p IS NONSIGNIFICANT

Terms	Resid. Df	Resid. Dev	Pr(Chi)
F * poly(r,8) + poly(p,6)	3967	1080.759	
F * (poly(r,8) + poly(p,6))	3955	1076.708	0.3566172

TABLE XII
ANALYSIS OF DEVIANCE TABLE FOR PROBLEMS $E_5, \ldots, E_8$ (4000 DATA). TEST FOR INTERACTIONS BETWEEN F AND r IS NONSIGNIFICANT

Terms	Resid. Df	Resid. Dev	Pr(Chi)
F + poly(r,8) + poly(p,6)	3983	1092.291	
F * poly(r,8) + poly(p,6)	3967	1080.759	0.0019618

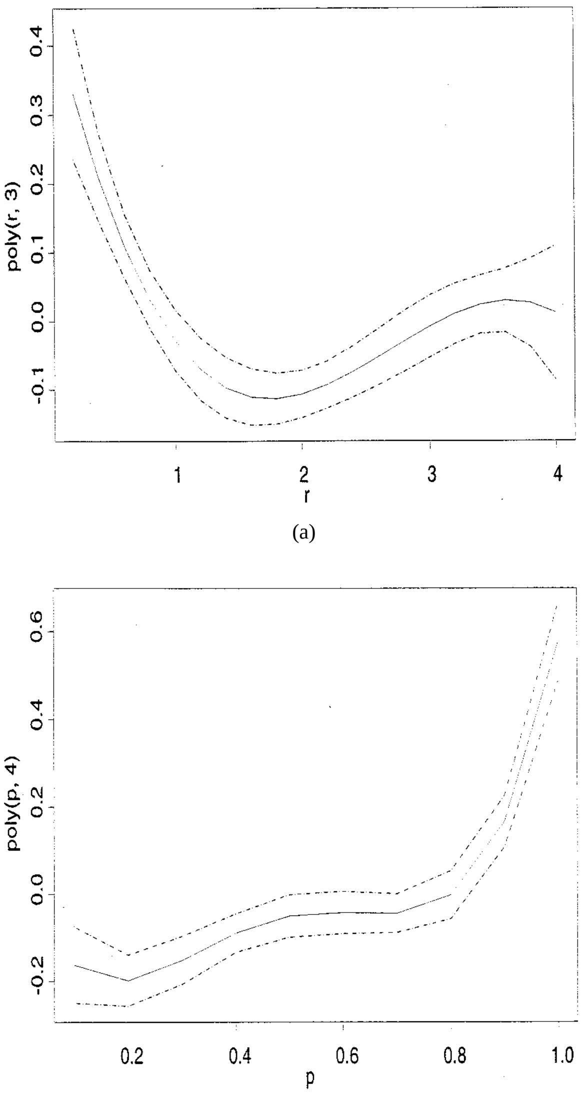
TABLE XIII
ANALYSIS OF DEVIANCE TABLE FOR PROBLEMS $E_5, \ldots, E_8$ OBTAINED FROM THE REDUCED DESIGN (1600 DATA)

Terms	Resid. Df	Resid. Dev	Pr(Chi)
F + poly(r,3) + poly(p,4)	1590	441.1227	
F * (poly(r,8) + poly(p,6))	1555	425.9549	0.1033434

TABLE XIV
COEFFICIENT VALUES IN MODEL $F + \text{poly}(r, 3) + \text{poly}(p, 4)$ FOR $E_5, \ldots, E_8$ (REDUCED DESIGN)

	Value	Std. Error	t value
(Intercept)	-3.79652211	0.1755074	-21.631694
$F2(E7)$	0.95154617	0.0372731	25.529033
$F3(E8)$	0.77552739	0.0372731	20.806625
r	-0.81133955	0.1495469	-5.425317
r^2	0.34370327	0.0816882	4.207502
r^3	-0.04256202	0.0128052	-3.323806
p	-3.50869706	1.7919751	-1.958005
p^2	15.46275968	6.2063203	2.491454
p^3	-24.05242920	8.2933123	-2.900220
p^4	12.61449758	3.7490879	3.364684

Reducing the computational effort needed when parameter settings are experimented by hand is the second merit of the method. In general, correct assessment of EA performance can



(a)

(b)

Fig. 15. Plots of predictors (a) φ_r and (b) φ_p for the class of problems $E_5 - E_8$ obtained with 4^*400 data.

TABLE XV
RESIDUAL DEVIANCES FOR MODELS $M_0 : F + C * (\mathrm{poly}(r,\,8) + \mathrm{poly}(p,\,6))$ AND $M_1 : F + \mathrm{poly}(r,\,8) + \mathrm{poly}(p,\,6)$. SMALLEST DIFFERENCE IS OBTAINED FOR f_3. NONE OF THE f_iS BELONG TO THE CLASS $E_5, \ldots, E_8$

	f_1	f_2	f_3	f_4
model M_0	702.354	638.966	461.5186	823.806
model M_1	1754.222	1460.780	588.2489	1295.513

be achieved with reduced experimental designs, which are worth being investigated systematically. Compared to trial and error, the method is more reliable, as the uncertainty on optimal parameters can be assessed precisely by using confidence intervals associated to predictors.

In practical situations, all computational effort is, however, put into finding a single solution. In this paper, efficient parameter settings have been determined by using short runs. Extrapolating the optimal parameter settings to longer runs is generally difficult. Nevertheless, a useful set of rules can be drawn from this work.

1) The fact that MOSES may explore local minima on a single space scale and misses interesting parts of the landscape requiring smaller or larger scales is a danger related to short simulation times. This phenomenon may happen when mutation probabilities are too small. (Indeed, small mutation probabilities indicate that the search space is explored repeatedly from the current best solution.) Keeping small mutation probabilities seems a worth strategy for unimodal problems or when local minima are regularly distributed. In general, it may be beneficial to overestimate the mutation probability so that different scales can also be explored.
2) As a classical paradigm for static EAs, optimal step sizes obtained in short simulation times are obviously suboptimal in longer runs: The optimal radius necessarily shifts toward zero in longer runs. Nevertheless, this study indicates that the performance curve has a typical shape. There is a sharp increase for small values and the curve is rather flat elsewhere. This outlines the danger in underestimating the optimal radius. On the other hand, performances remain stable above the optimal value (to a large extent). Consequently, the greatest care should be taken in decreasing the radius: keeping optimal values obtained in short runs seems a reasonable strategy.

To validate the previous claims, MOSES has been run on approximation problems with two different parameter settings. The first configuration is

$$p = 0.2, \quad r = 0.6.$$

In this case, the number of fitness evaluations has been taken equal to 80 000. The second configuration is

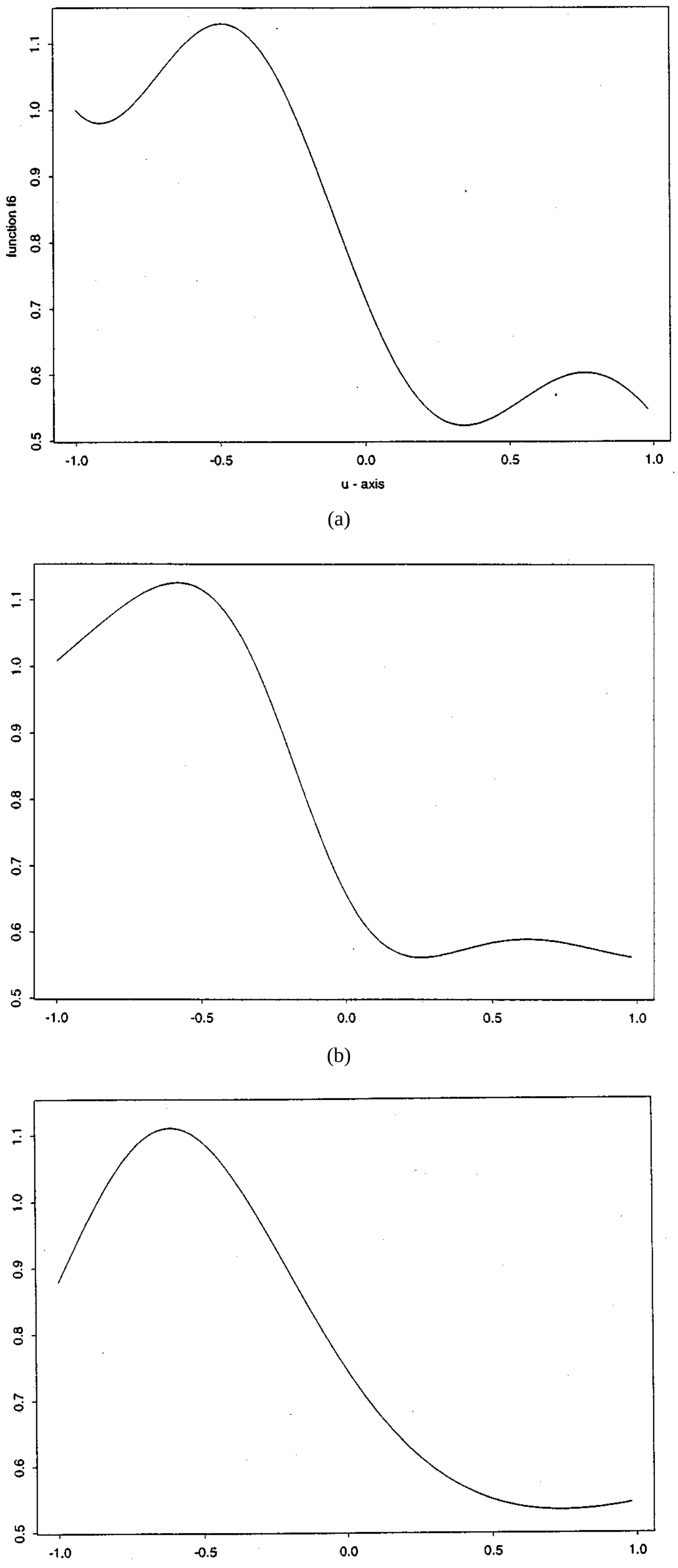
$$p = 0.2, \quad r = 1.6$$

and the number of evaluations has been fixed to 50 000. In all cases, simulations show that the best performance is given by the second setting which corresponds to optimal values obtained in short runs (Fig. 16). With a suboptimal configuration, the approximation is coarser although the number of evaluations is higher. In this example, extrapolating estimated values to longer runs seems correct.

VII. CONCLUSION

In this paper, a statistical methodology has been introduced to help a user of EAs to choose efficient parameter settings. Statistical tools allow managing the simulation data recorded during the phase of readjustment. This work shows how to manage a reduced number of simulations for each parameter setting. In addition, the accuracy of estimations can be improved thanks to the identification of problem classes.

The statistical model used in this work is a black box model, which does not take into account any geometrical properties of the minimization problem. In the real world, analytical properties are usually unavailable and black box models must be considered. This work is nevertheless contextual: neither immediate rules of thumb nor useful tricks nor directly replicable conclusions can be given. However, the method seems to be dedicated to specific application contexts where problem type does not vary



(a)

(b)

(c)

Fig. 16. Plots of the curve (a) f_5, (b) solution found with optimal parameters (50 000 evaluations), and (c) a solution found with suboptimal parameters (80 000 evaluations).

too much (from a problem to the others). In our opinion, approximation or classification tasks reflect this context accurately.

APPENDIX
GENERALIZED LINEAR MODELS

This section is devoted to mathematical definitions. Three hypotheses characterize generalized linear models (GLMs): 1) the distribution of the variable to explain; 2) the linear predictor; and 3) the function that links the predictor to the mean of the variable (link function).

Denote by y the N-dimensional vector of observations. It corresponds to samples of a variable Y. The components Y_i ($i = 1, \ldots, N$) of Y are assumed to be independent and distributed according to a distribution taken from the exponential family [27]. More precisely, the probability density function of Y_i is

$$f(y_i, \theta_i, \phi) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right)$$

where θ_i is a *canonical* parameter and ϕ a *dispersion* parameter. Functions b and c are specific to each distribution. The exponential family includes numerous classical laws: Binomial, Poisson, Gaussian, Gamma, etc. In particular, the Gamma distribution is defined as

$$\forall\, y_i \in R_+, \qquad G(\alpha, \lambda_i)(y_i) = \frac{\lambda_i^\alpha}{\Gamma(\alpha)}\, y_i^{\alpha-1} e^{-\lambda_i y_i}.$$

$\lambda_i > 0$, $\alpha > 0$. Its expectation is equal to α/λ_i and its variance equals α/λ_i^2. The parameters θ_i, ϕ, and a, b are given by

$$\theta_i = \frac{\lambda_i}{\alpha}, \quad b(\theta_i) = \log(\theta_i), \quad \phi = \frac{1}{\alpha}, \quad a(\phi) = -\phi.$$

Like in linear models, a linear predictor can be defined from the explanatory variables as

$$\eta = X\beta$$

where β is a d-dimensional vector of unknown parameters and X a $N \times d$ matrix, fixed by the experimental design. The relationship between β and θ is described through the link function h

$$X\beta = h(b'(\theta)).$$

Coefficients (β_j) can be estimated thanks to the maximum likelihood method. The implementation of this method is described hereafter. According to the independence of coordinates, the log likelihood is given by

$$\log L(\theta_i; y) = \sum_{i=1}^{N}\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right]$$
$$= \sum_{i=1}^{N} \log L_i(\theta_i; y_i).$$

The function $\log L$ can be differentiated with respect to each β_j. Denote

$$\mu_i = E[Y_i] = b'(\theta_i).$$

For all $i \in 1, \ldots, N$, $j \in 1, \ldots, d$, the partial derivatives with respect to β_j are equal to

$$\begin{aligned}\frac{\partial \log L_i}{\partial \beta_j} &= \frac{\partial \eta_i}{\partial \beta_j}\,\frac{\partial \mu_i}{\partial \eta_i}\,\frac{\partial \theta_i}{\partial \mu_i}\,\frac{\partial \log L_i}{\partial \theta_i}\\ &= X_{ij}\,\frac{1}{g'(\mu_i)}\,\frac{1}{b''(\theta_i)}\,\frac{y_i-\mu_i}{a(\phi)},\end{aligned}$$

$$\frac{\partial \log L}{\partial \beta_j} = \sum_{i=1}^{N} X_{ij}\,\frac{1}{g'(\mu_i)^2 Var(Y_i)}\,g'(\mu_i)(y_i-\mu_i).$$

Thus, by introducing the diagonal matrix

$$W_\beta = (Var(Y_i)g'(\mu_i)^2\delta_{ij})_{i,\,j=1,\ldots,N}$$

the maximum likelihood equations write as

$$X'W_\beta^{-1}\,\frac{d\eta}{d\mu}(y-\mu) = 0 \qquad (5)$$

where

$$\frac{d\eta}{d\mu} = \left(\frac{d\eta_i}{d\mu_i}\,\delta_{ij}\right)_{i,\,j=1,\ldots,N} = (g'(\mu_i)\delta_{ij})_{i,\,j=1,\ldots,N}.$$

The iterative procedure used to solve these equations is the Fisher's scores procedure. It proceeds with the following steps ($t \geq 0$)

$$\begin{aligned}\beta^{[t+1]} &= \beta^{[t]} - \left(E\left[\left\{\frac{\partial^2 \log L}{\partial\beta\partial\beta'}\right\}\right]^{[t]}\right)^{-1}\frac{\partial \log L^{[t]}}{\partial \beta}\\ &= \beta^{[t]} + \left(X'W_{\beta^{[t]}}^{-1}X\right)^{-1} X'W_{\beta^{[t]}}^{-1}\,\frac{d\eta}{d\mu}^{[t]}\left(y-\mu^{[t]}\right).\end{aligned}$$

The *deviance* is defined as

$$D(y;\hat{\beta}) = 2[\log L(y;y) - \log L(y;\hat{\beta})]$$

where $L(y;\hat{\beta})$ is the value of the loglikelihood obtained by plugging the estimated parameter $\hat{\beta}$ in L. The *residual deviance* is defined as

$$D^*(y;\hat{\beta}) = \phi D(y;\hat{\beta}).$$

The deviance is asymptotically distributed according to the χ^2 distribution with $N-d$ degrees of freedom and the parameter ϕ can be estimated by

$$\hat{\phi} = \frac{1}{N-d}\sum_{i=1}^{N}\frac{y_i-\hat{\mu}_i}{b''(\hat{\theta}_i)}$$

with $\hat{\theta}_i = b'^{-1}(\hat{\mu}_i)$.

Such a property allows for building a test for comparing the fit of two models. To test model M_0 against model M_1 with $M_0 \subset M_1$, denote by D_0^*, D_1^* and d_0, d_1 the residual deviances and degrees of freedom associated with M_0 and M_1. Under smooth assumptions, the test at level α is defined by the critical region

$$R_\alpha = \{\xi > \chi^2_{1-\alpha}(d_0-d_1)\}$$

with

$$\xi = \frac{D_0^* - D_1^*}{\phi}$$

and $\chi^2_{1-\alpha}(d_0-d_1)$ is the $(1-\alpha)$-quantile of a χ^2 distribution with (d_0-d_1) degrees of freedom. Model M_0 is rejected if $\xi > \chi^2_{1-\alpha}(d_0-d_1)$.

For more details on the estimation procedures for GLMs, the reader is referred to [27].

References

[1] E. H. L. Aarts and J. H. M. Korst, *Simulated Annealing and Boltzmann Machines*. New York: Wiley, 1988.

[2] T. Bäck, "Optimal mutation rates in genetic algorithms," in *Proc. 5th Int. Conf. Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 2–8.

[3] ——, *Evolutionary Algorithms in Theory and Practice*. Oxford, U.K.: Oxford Univ. Press, 1996.

[4] T. Bäck and H. P. Schwefel, "An overview of evolutionary algorithms for parameters optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 1–24, Spring 1993.

[5] F. Bergeret and P. Besse, "Simulated annealing, weighted simulated annealing and genetic algorithms at work," *Comput. Stat.*, vol. 12, no. 4, pp. 447–465, Oct. 1997.

[6] J. M. Chambers and T. J. Hastie, *Statistical Models in S*. London, U.K.: Chapman & Hall, 1991.

[7] Y. Davidor, "Epistasis variance: a view-point on GA-hardness," in *Foundations of Genetic Algorithms*, G. J. E. Rawlins, Ed. San Mateo, CA: Morgan Kaufmann, 1991, pp. 29–35.

[8] L. Davis, *Handbook of Genetic Algorithms*. New York: Van Nostrand, 1991.

[9] ——, "Bit climbing, representational bias, and test suite design," in *Proc. 4th Int. Conf. Genetic Algorithms*, R. K. Belew and L. B. Booker, Eds. San Mateo, CA, 1991, pp. 18–23.

[10] K. A. De Jong, "The analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Univ. Michigan, Ann Arbor, MI, 1975.

[11] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 124–141, July 1999.

[12] D. B. Fogel, *System Identification Through Simulated Evolution: A machine Learning Approach to Modeling*. Needham Heights, MA: Ginn Press, 1991.

[13] ——, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. New York, NY: IEEE Press, 1995.

[14] D. B. Fogel and J. W. Atmar, "Comparing genetic operators with Gaussian mutations in simulated evolutionary process using linear systems," *Biol. Cybern.*, vol. 63, no. 2, pp. 111–114, Feb. 1990.

[15] D. B. Fogel and L. C. Stayton, "On the effectiveness of crossover in simulated evolutionary optimization," *Biosyst.*, vol. 32, no. 3, pp. 171–182, May 1994.

[16] D. B. Fogel and A. Ghozeil, "Using fitness distributions to design more efficient evolutionary computations," in *Proc. 3rd IEEE Int. Conf. Evolutionary Computation*, T. Fukuda, Ed. Piscataway, NJ: IEEE Press, 1996, pp. 11–19.

[17] O. François, "An evolutionary strategy for global minimization and its Markov chain analysis," *IEEE Trans. Evol. Comput.*, vol. 2, pp. 77–90, Sept. 1998.

[18] ——, "Convergence in simulated evolution algorithms," *Complex Syst.*, vol. 10, no. 4, pp. 311–325, July 1996.

[19] ——, "Controlling mutation/selection algorithms with stochastic approximation," in *Proc. Congr. Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1999, pp. 1487–1493.

[20] L. Ingber and B. Rosen, "Genetic algorithms and very fast simulated reannealing: A comparison," *Math. Comput. Model.*, vol. 16, no. 11, pp. 87–100, Nov. 1992.

[21] Z. Jian, X. Yuan, Z. Zeng, B. P. Buckles, and C. Koutsougeras, "S. Amer. Niching in an ES/EP context," in *Proc. Congr. Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1999, pp. 1426–1433.

[22] T. Jones and S. Forrest, "Fitness distance correlation as a measure of problem difficulty for genetic algorithms," in *Proc. 6th Int. Conf. Genetic Algorithms*, L. J. Eshelman, Ed. San Mateo, CA: Morgan Kaufmann, 1995, pp. 184–192.

[23] A. J. Keane, "A brief comparison of some evolutionary optimization methods," in *Proc. Conf. Applied Decision Technologies*, 1995, pp. 28–35.

[24] L. C. Kwong Hui, K. Y. Lam, and C. W. Chea, "Global optimization in neural network training," *Neural Comput. Applicat.*, vol. 5, pp. 58–64, 1997.

[25] H. R. Lindman, *Analysis of Variance in Experimental Design*. New York: Springer-Verlag, 1992.

[26] J. Lis and M. Lis, "Self-adapting parallel genetic algorithm with the dynamic mutation probability, crossover rate and population size," in *Proc. 1st Polish Nat. Conf. Evolutionary Computation*, J. Arabas, Ed., 1996, pp. 324–329.

[27] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed. London, U.K.: Chapman & Hall, 1989.

[28] B. Naudts and L. Kallel, "Comparison of summary statistics of fitness landscapes," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 1–15, Apr. 2000.

[29] V. Nissen and J. Propach, "On the robustness of population-based versus point-based optimization in the presence of noise," *IEEE Trans. Evol. Comput.*, vol. 2, pp. 107–119, November 1998.

[30] K. Park, "A comparative study of genetic search," in *Proc. 6th Int. Conf. Genetic Algorithms*, J. L. Eshelman, Ed. San Mateo, CA: Morgan Kaufmann, 1995, pp. 512–519.

[31] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution* (in German). Stuttgart, Germany: Fommann-Holzboog Verlag, 1973.

[32] C. R. Reeves and C. C. Wright, "An experimental design perspective on genetic algorithms," in *Foundations of Genetic Algorithms 3*, D. Whitley and M. Vose, Eds. San Mateo, CA: Morgan Kaufmann, 1995, pp. 7–22.

[33] ——, "Epistasis in genetic algorithms: An experimental design perspective," in *Proc. 6th Int. Conf. Genetic Algorithms*, J. L. Eshelman, Ed. San Mateo, CA: Morgan Kaufmann, 1995, pp. 7–22.

[34] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[35] G. Rudolph, "Self-adaptation and global convergence: A counter-example," in *Proc. Congr. Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1999, pp. 646–651.

[36] N. N. Schraudolph and R. K. Belew, "Dynamic parameter encoding for genetic algorithms," *Machine Learning*, vol. 9, no. 1, pp. 9–21, June 1992.

[37] H.-P. Schwefel, *Evolution and Optimum Seeking*. New York: Wiley, 1995.

[38] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proc. 3rd Int. Conf. Genetic Algorithms*, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 2–9.

[39] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with Splus*, 2nd ed. New York: Springer-Verlag, 1997.

[40] D. Whitley, K. Mathias, S. Rana, and J. Dzubera, "Building better test functions," in *Proc. 6th Int. Conf. Genetic Algorithms*, J. L. Eshelman, Ed. San Mateo, CA: Morgan Kaufmann, 1995, pp. 512–519.

[41] D. H. Wolpert and W. G. MacReady, "No free lunch theorem for optimization," *IEEE Trans. Evol. Comput.*, vol. l, pp. 67–82, Apr. 1997.

[42] X. Yao, "Evolutionary artificial neural networks," *Int. J. Neural Syst.*, vol. 4, no. 3, pp. 203–222, July 1993.

Olivier François received the Magistère de Mathématiques et Applications and the Ph.D. degrees in mathematics from the University of Grenoble, Grenoble, France, in 1989 and 1992, respectively.

Since 1994, he has been an Associate Professor with the École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble, Grenoble, France. His current research interests include applied probability, Markov chains, neural networks, evolutionary algorithms, and complex systems.

Christian Lavergne received the Ph.D. degree in applied mathematics from the Paul Sabatier University, Toulouse, France, in 1984.

He is currently a Professor of Statistics at the University of Paul Valéry, Montpellier, France. He has been a Member of Institut National de Recherche en Informatique et Automatique since 1996. From 1988 to 1996, he was an Associate Professor with the Institut National Polytechnique de Grenoble, Grenoble, France, where he received his Habilitation in 1995. His current research interests include statistic modeling, generalized linear models, and mixed and heteroscedastic models.