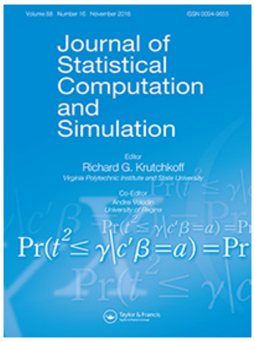

Taylor & Francis
Taylor & Francis Group

Journal of Statistical Computation and Simulation

ISSN: 0094-9655 (Print) 1563-5163 (Online) Journal homepage: http://www.tandfonline.com/loi/gscs20

Statistical and computational tradeoff in genetic algorithm-based estimation

Manuel Rizzo & Francesco Battaglia

To cite this article: Manuel Rizzo & Francesco Battaglia (2018) Statistical and computational tradeoff in genetic algorithm-based estimation, Journal of Statistical Computation and Simulation, 88:16, 3081-3097, DOI: 10.1080/00949655.2018.1499740

To link to this article: https://doi.org/10.1080/00949655.2018.1499740

Published online: 22 Jul 2018.

Submit your article to this journal

Article views: 63

View Crossmark data

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=gscs20

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION
2018, VOL. 88, NO. 16, 3081–3097
https://doi.org/10.1080/00949655.2018.1499740

Taylor & Francis
Taylor & Francis Group

Check for updates

Statistical and computational tradeoff in genetic algorithm-based estimation

Manuel Rizzo and Francesco Battaglia

Department of Statistical Sciences, Sapienza University of Rome, Rome, Italy

ABSTRACT
When a genetic algorithm (GA) is employed in a statistical problem, the result is affected by both variability due to sampling and the stochastic elements of algorithm. Both of these components should be controlled in order to obtain reliable results. In the present work we analyze parametric estimation problems tackled by GAs, and pursue two objectives: the first one is related to a formal variability analysis of final estimates, showing that it can be easily decomposed in the two sources of variability. In the second one we introduce a framework of GA estimation with fixed computational resources, which is a form of statistical and the computational tradeoff question, crucial in recent problems. In this situation the result should be optimal from both the statistical and computational point of view, considering the two sources of variability and the constraints on resources. Simulation studies will be presented for illustrating the proposed method and the statistical and computational tradeoff question.

ARTICLE HISTORY
Received 6 February 2018
Accepted 9 July 2018

KEYWORDS
Evolutionary algorithms; convergence rate; analysis of variability; least absolute deviation; autoregressive model; g-and-k distribution

1. Introduction

In recent years the increase in computing power and the huge growth in size of datasets have introduced many novel problems in statistics. Statisticians must now carefully consider computational complexity in efficiency and consistency analysis, in order to obtain both efficient and feasible results, subject to resources or time constraints. This kind of analysis, as it aims at balancing statistical efficiency and computational complexity, is generally named as *statistical and computational tradeoff* (or time-data tradeoff). For example, Dillon and Lebanon [1] studied consistency of a stochastic extension of composite likelihood estimators, whose formula depends also on parameters related to computational elements. Wang et al. [2], in a sparse principal component analysis framework, addressed the question of whether is possible to find an estimator which is computable in polynomial time, and then analyzed its minimax optimal rate of convergence. Several other proposals can be found in [3–10].

In a closely related contribution, due to Winker and Maringer [11], the limit behaviour of an estimator based on threshold accepting algorithm is analyzed in a GARCH model

CONTACT Manuel Rizzo ✉ manuel.rizzo@uniroma1.it 🖃 Department of Statistical Sciences, Sapienza University of Rome, Piazzale Aldo Moro 5, I-00185 Rome, Italy

© 2018 Informa UK Limited, trading as Taylor & Francis Group

estimation problem. They studied the joint convergence of the estimator, which depends on sample size, and the algorithm, as the number of iterations increases, and corresponding convergence rates were evaluated by simulation (see also [12,13]).

In the present paper we propose a statistical and computational tradeoff discussion employing a similar analysis as in [11], for general model building problems using genetic algorithms (GAs). At first we shall analyze variability of such methods when employed in parametric estimation problems, as they introduce an additional source of variability in the process. In this context we investigate the effect of both statistical and computational components on efficiency of results, focusing on their limit behaviour. Given this framework, the tradeoff is discussed by introducing cost functions in the analysis, related to both data acquisition and runtime of the algorithm, when an overall amount of resources is fixed. This scheme will give indications on how to optimally allocate a fixed amount of resources, and this is generally demanding when intractable or time-consuming problems are concerned. A selection of simulation examples will illustrate the proposed method, in which variability components will be empirically evaluated and several values for cost functions will be considered.

The paper is organized as follows: Section 2 describes standard GAs and their implementation in parametric estimation problems; in Section 3 the variability analysis and the tradeoff problem are specified; Section 4 displays the simulation examples selected for illustrating the proposed method; the last Section includes final comments and discusses future developments.

2. Genetic algorithms for models building

2.1. Overview of the algorithm

GAs are among the most important evolutionary computation techniques, because of their simplicity and versatility of applications. They were introduced by Holland [14] as a method for describing the adaptive processes of natural systems, adopting metaphors from biology and genetics. Main GAs application is linked to complex optimization problems [15–17], whose complexity might be due to the objective function, which might be non-differentiable, or to the search space, possibly very large or irregular.

In this framework the goal is to find the global optimum of a function, called *fitness*, which measures the goodness of solutions. In the metaphor the generic solution is represented by an individual, coded in a string called *chromosome*, whose elements represent the genetic heritage of the individual (*genes*). In the standard binary coding case, genes can only take values 0 or 1 (*bits*). At each iteration (or *generation*, in GA terminology) the algorithm considers a population of fixed size N of individuals evolving by means of genetic operators of *selection*, *crossover* and *mutation*. The *selection* randomly chooses solutions for subsequent steps, in general proportionally to their fitness value; by *crossover* two solutions are allowed to combine together, with a fixed rate pC, exchanging part of their genes and creating two new individuals; lastly, the *mutation* step allows every bit to flip its value from 0 to 1, or vice versa, with a fixed probability pM, providing a further exploration of search space. The resulting population replaces the previous one, and the generations flow stops when a certain condition is met, if for example a fixed number of generations is reached. It is also possible, adopting the *elitist* strategy, to maintain the best individual found up to current generation, in spite of modifications made by genetic operators. In that case, the user interested in optimization may analyze the succession of such solutions.

Elitism is crucial as far as convergence is contemplated. In fact, most of convergence results have been obtained for elitist GAs, generally by use of Markov Chain theory. A fundamental theorem by Rudolph [18], that easily adapts to a wide class of evolutionary algorithms (EAs), considers an elitist GA with $pM > 0$ and models X_g, namely the best solution found up to generation g, by a Markov Chain. It states that, under the assumptions given above, the sequence $D_g = [f^* - f(X_g)]$, where f^* is the global optimum and $f(X_g)$ the fitness of X_g, is a non-negative supermartingale which converges almost surely to zero. Generalizations have been proposed in order to extend Rudolph's approach to either time-varying mutation or crossover rates (or both) by modeling GA as a non homogeneous Markov Chain [19–21]. Reference [21] includes also a review of other contributions analyzing GA convergence by Markov Chain theory. In our paper we employ a simple GA, so we shall refer to Rudolph theorem of convergence, which also allow to generalize the framework to other EAs. It is worth noting that this theorem just states the convergence of a GA, but it gives no information about its rate.

2.2. *Parametric estimation*

There are many complex statistical problems which are suitable for GAs application, like *outliers detection*, *cluster analysis* or *design of experiment* (for a comprehensive account see [22]). In this work we consider parametric model building problems, in which the function to be maximized, a likelihood for example, is hard to analyze, and standard methods may fail in finding good estimates (several applications can be found in [23–26]). In this situation a sample y is generated from a distribution known up to a parameters vector, and the inference is made by maximizing an objective function depending on both the parameter and y.

We shall now specify GA implementation, which consists in solutions coding and fitness function specification, for the problem at issue. Although floating-point GAs have been employed in literature to deal with real parameters optimization, we shall employ the simple binary coded GA described above. The standard rule [27] for binary encoding a parameter θ with values in the real interval $[a, b]$ is:

$$\theta = a + \frac{b-a}{2^H - 1} \sum_{j=1}^{H} 2^{j-1} x_j,$$

where x_j is the jth bit ($j = 1, \dots, H$). If the interest is focused on a vector $\theta = (\theta_1, \dots, \theta_k)$ then a chromosome of length $M = k \cdot H$ includes the coding of all components. Length H of each genes group is constant, but the coding interval $[a, b]$ can vary for each parameter. As far as we are considering a kind of discretization of a continuous search space, we aim at building a fine grid in such a way that the fitness function is adequately smooth on that grid, so that the related loss of information is negligible. This may be done by selecting a sufficiently large number of different values equispaced in the coding interval, or alternatively specifying the difference between two consecutive coding values (e.g. 10^{-k}).

The fitness f is proportional to the objective function, say $g(\theta; y)$. We shall consider a scaled exponential transformation of g:

$$f(\theta) = \exp\{g(\theta; y)/\tau\}, \tag{1}$$

where $\tau > 0$ is a problem dependent constant. This kind of scaling procedure allows to modify the shape of fitness function without changing the ranking of solutions, and influences mainly the selection operator. Fitness scaling is a widely discussed issue in GA theory, [15,28], [22, p.53], and it is generally recognized that the best choice depends on the problem nature. For the problems at hand we suggest to select $\tau = n$, as will be explained in Section 3.2.

As far as the choice of genetic operators is concerned, many options are possible and many related studies are available [15,17,29,30]. It is generally known that there is no universally dominant choice for any problem, so pilot studies are needed for a successful implementation. We decided to employ basic genetic operators: *roulette wheel selection*, in order to select chromosomes proportionally to their fitness value; *single point crossover*, for which a chromosome can exchange up to $k-1$ parameters in every recombination; standard *bit-flip mutation* strategy. Lastly, *elitism* is adopted for guaranteeing convergence of procedure.

3. Problem description

3.1. Variability decomposition

Following estimation theory, a parameter estimate is naturally subject to sampling variability: in fact if we make inference using two different samples we may obtain two possibly different results. When GAs are employed in the estimation process an additional form of variability is introduced in the analysis, due to the stochastic nature of the algorithm. It is related to the choice of starting population, selection mechanism, mutation and crossover rates, the random choice of cutting point in crossover. As a result of this, if we run a GA several times using the same sample we may obtain different results.

The total variability of a GA estimate can be easily decomposed in these two forms of variability, as shown in [22, p.50] for the univariate case.

We shall adopt the following notation: y is the sample of observations, θ the parameter, $\hat{\theta}(y)$ the best theoretical value (for example a maximum likelihood estimate), which can not be computed in practice, and $\theta^*(y)$ the result of optimization obtained via GA, that is an approximation to $\hat{\theta}(y)$ and depends on the observed sample as well. We assume stochastic independence between data and the algorithm, and decompose total error of a GA estimate as follows:

$$\theta^*(y) - \theta = [\hat{\theta}(y) - \theta] + [\theta^*(y) - \hat{\theta}(y)]. \tag{2}$$

The first term in square brackets depends on consistency of the estimates, while the second depends on the convergence of GA. The final convergence of the GA estimate is ensured if both of these components converge to zero in probability.

As long as in real applications GAs are generally employed in complex, often multiparametric problems, then we shall consider the multiparametric analogous to (2). In that case $\theta = (\theta_1, \dots, \theta_k)$ is the parameter vector of interest, $\hat{\theta}$ is the best theoretical value, while random vector $\theta^*(y)$, for which sample y is held fixed, is the result of GA.

If an elitist strategy is employed then we can define random vector $\theta^{*(g)}(y)$ as the best estimate obtained up to generation g, that corresponds to the best individual of generation g. In our method we shall evaluate GA variability by analyzing the behaviour of this random vector among GA runs, basing on Rudolph theorem that in our case implies that sequence $\theta^{*(g)}(y),\ g = 1, \ldots$ will converge to $\hat{\theta}(y)$ when g goes to infinity. This means that when g increases then each GA run gets closer to convergence, so variability between runs tends to decrease as a consequence. So, in our framework, evaluating the variability of the GA is closely related with studying the convergence rate of the algorithm.

Having defined both random vectors $\hat{\theta}(y)$ and $\theta^{*}(y)$ we shall define their variance–covariance matrices, respectively Σ_S and Σ_{GA}, in order to relate to (2). Generic (i,j) elements of these matrices are:

$$\sigma^S_{ij} = \mathbb{E}_S[(\hat{\theta}_i - \theta_i)(\hat{\theta}_j - \theta_j)], \quad i,j = 1, \ldots k,$$
$$\sigma^*_{ij} = \mathbb{E}_{GA}[(\theta^*_i - \hat{\theta}_i)(\theta^*_j - \hat{\theta}_j)], \quad i,j = 1, \ldots k.$$

σ^S_{ij} and σ^*_{ij} measure the dependence between the estimates of θ_i and θ_j induced, respectively, by sampling and GA. To get a scalar summary of these matrices, a possible choice is to consider the traces, a strategy often adopted in literature. This is reasonable in an optimization framework, because the optimum is reached when variances σ^S_{ii} and σ^*_{ii} $(i = 1, \ldots, k)$ go to zero, with no practical interest on covariances. Therefore, if Σ_{TOT} is defined as the total variance–covariance matrix, then, using the linearity of trace and under the same independence assumption of (2), we can write:

$$\mathrm{tr}(\Sigma_{TOT}) = \mathrm{tr}(\Sigma_S) + \mathrm{tr}(\Sigma_{GA}). \tag{3}$$

3.2. *Tradeoff problem*

Now we shall set the variability analysis of Section 3.1 in the framework of statistical and computational tradeoff. Assuming that both statistical estimator and GA configurations are fixed, we try to optimally balance statistical accuracy and GA efficiency.

If we consider estimators having the property of consistency, then statistical accuracy can be naturally represented by sample size n, because if n increases then also estimator precision increases (and, in contrast, variability decreases), under some regularity conditions (see [31, p.470], in the case of Maximum Likelihood Estimators).

As far as GA efficiency is concerned, we refer to Rudolph theorem of convergence. Informally, a GA converges when g tends to infinity, but it is worth noting that in every GA generation each of the N chromosomes in the population is evaluated on the basis of fitness function. Therefore, instead of considering the number of generations, we represent GA efficiency by the number of fitness function evaluations V, also because it is usually the most computationally expensive step.

We shall study the behaviour of $\mathrm{tr}(\Sigma_S)$ and $\mathrm{tr}(\Sigma_{GA})$ when, respectively, $n \to \infty$ and $V \to \infty$. Let us introduce two functions $f(n)$ and $h(V)$ for which, respectively, $f(n) \to \infty$ when $n \to \infty$ and $h(V) \to \infty$ when $V \to \infty$. If we employ a consistent estimator and the assumptions of Rudolph theorem are fulfilled, then we can write $\mathrm{tr}(\Sigma_S) = \mathcal{O}([f(n)]^{-1})$

and $\mathrm{tr}(\Sigma_{GA}) = \mathcal{O}([h(V)]^{-1})$. In that case:

$$\mathrm{tr}(\Sigma_{TOT}) = \mathrm{tr}(W_S)\frac{1}{f(n)} + \mathrm{tr}(W_{GA})\frac{1}{h(V)}, \tag{4}$$

where matrices W_S and W_{GA} are constant with respect to n and V, and depend, respectively, from the statistical model and from the GA. However it is plausible that the sample size n has an effect on W_{GA}, because e.g. if n is large and the estimator is consistent, then the estimating function (e.g. the likelihood) is less variable around the optimum, so the related fitness function is easier to optimize. For this reason we shall let τ depend on sample size in the fitness scaling procedure (1), a choice which could also be adopted in other model building problems. Simulation studies showed us that adopting the choice $\tau = n$ allows to decisively restrict the effect of n on behaviour of the algorithm. Thus we shall employ this strategy so that we describe the total variability of a GA estimate by considering decomposition (4).

The statistical and computational tradeoff question will now be analyzed by introducing some cost functions: $S(n)$ is related to the cost of collecting a sample of n observations, $T(n)$ indicates the computational cost of one fitness function evaluation, which depends on the number of observations as well, because a solution is evaluated by analyzing the full sample. Hence the total cost C of obtaining an estimate $\theta^*(y)$ using n statistical observations and V fitness function evaluations is given by: $C = S(n) + VT(n)$. If total cost C is fixed and functions $S(\cdot)$ and $T(\cdot)$ are specified, we can write the tradeoff question as an optimization problem:

$$\left\{\begin{matrix} \min\limits_{n,V} \mathrm{tr}(\Sigma_{TOT}) = \mathrm{tr}(W_S)\frac{1}{f(n)} + \mathrm{tr}(W_{GA})\frac{1}{h(V)} \\ \text{s.t.} \\ S(n) + VT(n) = C \end{matrix}\right\}.$$

Therefore, in this framework we aim at minimizing the total variance–covariance matrix, which depends on intrinsic statistical and computational components. These latter, represented by $\mathrm{tr}(W_S)$, $\mathrm{tr}(W_{GA})$, $f(\cdot)$ and $h(\cdot)$, can be estimated if a known form is not available (details will be given in the following sections). Afterwards we search for optimal n and V minimizing $\mathrm{tr}(\Sigma_{TOT})$, given the constraint on total cost.

A particular case that simplifies the analysis is the assumption of linearity in n for cost functions T and S. This is reasonable because statistical observations are usually collected in sequence and if GA fitness function includes a summation over the observations. In such a case $T(n) = nT$, $S(n) = nS$ and we can incorporate the cost constraint into the objective function obtaining:

$$\min_n \mathrm{tr}(\Sigma_{TOT}) = \mathrm{tr}(W_S)\frac{1}{f(n)} + \mathrm{tr}(W_{GA})\frac{1}{h([C - nS]/nT)}. \tag{5}$$

The optimal solution $\tilde{n}$ can be found by minimizing numerically (5) conditionally on the form of consistency and convergence rates $f(\cdot)$ and $h(\cdot)$. $\tilde{V}$ is obtained by constraint:

$$\tilde{V} = \frac{C - \tilde{n}S}{\tilde{n}T}. \tag{6}$$

A particular case which allows to obtain a simple closed form expression for optimal n is obtained when $f(n) = n$ and $h(V) = V$. In that case, computing the derivative of objective function with respect to n, we obtain solutions:

$$\tilde{n} = \frac{-SC\,\mathrm{tr}(W_S) \pm C\sqrt{CT\,\mathrm{tr}(W_S)\mathrm{tr}(W_{GA})}}{CT\,\mathrm{tr}(W_{GA}) - S^2\,\mathrm{tr}(W_S)}. \tag{7}$$

Since n is natural we are interested in the positive solution of (7).

3.3. Consistency and convergence rates

Functions $f(n)$ and $h(V)$ introduced in the previous subsection specify, respectively, consistency rate of the statistical part and the convergence rate of algorithmic part in Equation (4). The assumption of linearity is a particular case that simplifies the tradeoff analysis. It is satisfied if we consider asymptotically efficient estimators: in that case, under some regularity conditions, $f(n) = n$ (see [31, p.472], in the case of Maximum Likelihood Estimators).

On the other side, the behaviour of $h(V)$ is related to GA convergence rate. This is an essential issue for any optimization algorithm, and in the field of EAs it has been analyzed in several ways. A part of literature focuses on comparing EAs with different configurations and searching for the scenario which optimizes convergence time [29,32]; other researchers have developed more rigorous approaches, focusing on the convergence rate of single chromosome bits, limited to classic problems like *OneMax* [33,34]; a different proposal, inspired by *statistical mechanics*, studies GA behaviour by modeling it as a complex system and summarizing its probability distribution through generations by considering the cumulants [35–37]. In such a way GA convergence can be evaluated by considering the limiting cumulants.

Recently, Clerc [38, p.69] proposed a theoretical framework for analyzing optimization performances. For a general stochastic algorithm (deterministic algorithms are considered a particular case of this class) he introduced a bivariate probability density $p(\psi, r)$, called *Eff-Res*, that is function of both optimization *result* r and computational *effort* ψ, spent for obtaining r. By analyzing this function it is possible to deepen different useful questions: for a given result r, the probability of obtaining r with a generic effort ψ; for a given effort ψ, the probability of obtaining a generic result r. Our interest is focused on the latter question, because if we fix a computational effort related to number of fitness evaluations, we are interested in how the result r varies. The theoretical variance of results for fixed effort can be written as:

$$\sigma^2(\psi) = \mu(\psi)\int_{\tilde{R}} (r - \bar{r}(\psi))^2 p(\psi, r)\,\mathrm{d}r, \tag{8}$$

where $\tilde{R}$ is the set of possible results, $\bar{r}(\psi)$ the theoretical mean result for fixed effort and $\mu(\psi)$ the normalization coefficient of $p(\psi, r)$. Expression (8) can be evaluated empirically: conditioning on J observed results $r(1), r(2), \ldots, r(J)$, obtained with effort ψ, the estimated variance is given by:

$$\hat{\sigma}^2(\psi) = \frac{1}{J-1}\sum_{j=1}^{J}[r(j) - \bar{r}_J(\psi)]^2, \tag{9}$$

where $\bar{r}_J(\psi)$ is the empirical mean of results.

In our method we shall employ a similar approach for evaluating GA variability. As far as we are interested in convergence of θ_i^* to the optimum $\hat{\theta}_i$ ($i = 1, \dots, k$), then in both (8) and (9) we plug $\hat{\theta}_i$ in place of theoretical and empirical means, and θ_i^* in place of results. In that case (8) corresponds to variance $\sigma_{ii}^* = \mathbb{E}_{GA}[(\theta_i^* - \hat{\theta}_i)^2]$ in matrix Σ_{GA}. If we run a GA J times, obtaining $\theta_{1,i}^*, \theta_{2,i}^*, \dots, \theta_{J,i}^*$ ($i = 1, \dots, k$), then we get the estimates by:

$$\hat{\sigma}_{ii}^* = \frac{1}{J}\sum_{j=1}^{J}[\theta_{j,i}^* - \hat{\theta}_i]^2, \quad i = 1, \dots, k. \tag{10}$$

The latter gives information on generic GA result θ_i^*. As long as we are studying the behaviour of algorithm when the number of generations increases, we shall specify an expression such as (10) for each generation g. That is, we obtain the sequence of variances, given a fixed maximum number of generations G:

$$\hat{\sigma}^{*(g)} = (\hat{\sigma}_{11}^{*(g)}, \hat{\sigma}_{22}^{*(g)}, \dots, \hat{\sigma}_{kk}^{*(g)}), \quad g = 1, \dots, G. \tag{11}$$

In order to study GA convergence rate we shall conduct the following regression analysis for each parameter indexed by i:

$$\hat{\sigma}_{ii}^{*(g)} = w_{GA,i}\frac{1}{[V^{(g)}]^a} + \epsilon_g, \quad g = 1, \dots, G, \tag{12}$$

where $[V^{(g)}]^a$ is the a-th power of the number of fitness evaluations up to generation g and $w_{GA,i}$ is the regression parameter. In principle each parameter θ_i could have a different convergence rate, but the GA evolves each component in a similar way and with identical operators. Thus, we consider a uniform convergence rate as a reasonable approximation, and we shall refer to a unique a for which $[V^{(g)}]^a$ is assumed to be the GA convergence rate $h(V)$ for all components θ_i, $i = 1, \dots, k$. In that case $w_{GA,i}$ will become part of matrix W_{GA} in (4).

4. Applications

We will now illustrate the proposed method with some model building problem examples: we propose a Least Absolute Deviation Regression estimation (code *LAD*), a selection of Autoregressive model building problems (code *AR*) and a g-and-k distribution maximum likelihood estimation (code *gk*). In the following subsections we shall separately describe these problems, discussing also motivations on choice of estimators and use of GAs. Then we will present results related to variability estimates and the tradeoff analysis.

General indications on GA implementation are given in Section 2.2, and in the following we will outline the specific issues for each problem. As far as GA operators are concerned, we fixed crossover and mutation rates at, respectively, 0.7 and 0.1, maximum number of generations G at 1400 and population size N at 50. If not otherwise specified the initial population was generated uniformly at random. These configurations have been chosen on the basis of empirical studies for guaranteeing stability and convergence of the procedure. Software R [39] was used for all simulations and computations, along with package *gk* [40] for the last application.

4.1. LAD

LAD regression is an alternative to Ordinary Least Squares regression, proven to be more robust to outliers [41, p.52]. In this framework the estimator, which is asymptotically efficient [41, p.44], is the function that minimizes the sum of absolute values of errors. This function is neither differentiable nor convex, so numerical methods must be employed to find an optimal solution. Zhou and Wang [42] have already employed a real valued GA to estimate the parameters of a LAD regression with censored data. In this paper we consider a standard linear regression model:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \epsilon_i, \quad i = 1, \dots, n,$$

where (y, x) is the observed dataset. The errors are not Gaussian, but distributed according to a heavy-tailed Student's t distribution with five degrees of freedom.

As far as our goal is maximization, then fitness function shall be:

$$f(\beta) = \exp\left\{ -\sum_{i=1}^{n} \left| y_i - \beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} \right| / n \right\}.$$

Each chromosome length shall be $M = 24$ and coding interval boundaries will be $[-2, 2]$ for all parameters. The parameters are encoded into 256 equispaced values between -2 and 2, and the difference between two consecutive values is about 0.015, that we consider sufficiently small. The parameter vector used in the simulations is $\beta = (0.5, 0.5, -0.5)$.

4.2. AR

GAs have been widely applied in the field of time series analysis [22, p.85]. In fact related parameters estimation and model identification problems may not be straightforward due to the intractability of objective functions or to the size of search spaces. The latter question is common in model identification problems, and it has been studied also for standard ARMA models [43,44]. Here we address the problem of how to simultaneously identify and estimate subset AR models, given a fixed maximum order.

The general equation of an AR model of order p is:

$$Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \epsilon_t, \tag{13}$$

where Y_t is a zero mean random process, ϵ_t a Gaussian white noise and $\phi = (\phi_1, \dots, \phi_p)$ the parameter vector, where some components may be constrained to zero.

Model (13) is usually identified by minimizing penalized likelihood criteria like AIC or BIC. In this work we shall consider BIC, because of its property of consistency [45]:

$$\mathrm{BIC}(\phi; y) = n \log \hat{\sigma}^2(p) + k \log n, \tag{14}$$

where y is the observed time series, $\hat{\sigma}^2(p) = \sum_{i=1}^{n} (y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p})^2 / n$ and $k \leq p$ is the number of unconstrained parameters in the model. Sampling variability will be estimated on the basis of asymptotic efficiency property of AR models maximum likelihood estimator [46, p.386].

We will consider a eight-dimensional parameter solution space, so that AR models up to order $p = 8$ can be identified and estimated by the procedure. A selection of four generating processes will be analyzed:

- AR_A: $\phi_1 = 0.8, \phi_2 = \phi_3 = \cdots = \phi_8 = 0$
- AR_B: $\phi_1 = 0.7, \phi_2 = -0.1, \phi_3 = \phi_4 = \cdots = \phi_8 = 0$
- AR_C: $\phi_1 = 0.8, \phi_2 = 0, \phi_3 = -0.1, \phi_4 = \phi_5 = \cdots = \phi_8 = 0$
- AR_D: $\phi_1 = 0.6, \phi_2 = -0.1, \phi_3 = \phi_4 = \cdots = \phi_8 = 0$

Chromosome length shall be $M = 64$ (8 genes for each parameter), and coding will necessarily include the case of generic parameter $\phi_i = 0$ equal to zero, having a crucial impact on penalization term of (14). The chosen grid is similar to the LAD case. In order to facilitate the identification of subset models we shall force the starting population to include a chromosome representing a white noise (all parameters are zero), and also eight chromosomes for which one of the parameters is zero, so that all $\phi_i = 0$ $(i = 1, \ldots, 8)$ cases are represented. The remaining chromosomes will be generated uniformly at random, coherently with other applications. This may be a reasonable strategy in a situation of lack of prior knowledge.

Fitness function shall be:

$$f(\phi) = \exp\{-\text{BIC}(\phi; y)/n\},$$

and coding interval will be $[-2, 2]$ for each ϕ_i.

4.3. *gk*

The *g-and-k* distribution was introduced by Haynes et al. [47], as a family of distributions specified by a quantile function. It is a very flexible tool which has been applied to statistical control charts techniques [48] and non-life insurance modelling [49]. For a univariate random sample $x = (x_1, \ldots, x_n)$ the quantile function is:

$$Q_X(u_i \mid A, B, g, k) = A + Bz_{u_i}\left(1 + c\frac{1 - e^{-gz_{u_i}}}{1 + e^{-gz_{u_i}}}\right)(1 + z_{u_i}^2)^k, \quad i = 1, \ldots, n,$$

where z_{u_i} is the u_i-th quantile of standard normal distribution, A and $B > 0$ are location and scale parameters, g measures skewness in the distribution, $k > -0.5$ is a measure of kurtosis and c is a constant introduced to make the distribution proper. By combining values of the four parameters several essential distributions like Normal, Student's t or Chi-square can be derived.

Maximum Likelihood estimation of this distribution is a kind of so-called *intractable likelihood* problem. The expression of likelihood is given by:

$$L(\theta \mid x) = \left(\prod_{i=1}^{n} Q'_X(Q_X^{-1}(x_i \mid \theta) \mid \theta)\right)^{-1}, \tag{15}$$

where x is the observed sample, $\theta = (A, B, g, k)$ and $Q'_X(u \mid \theta) = \partial Q_X/\partial u$.

The main difficulty in computing (15) is the lack of a closed form for expression $Q_X^{-1}(x_i \mid \theta)$, that must be obtained numerically, for example with Brent's method.

A lot of research on g-and-k distributions estimation has been made in a Bayesian framework, using Markov Chain Monte Carlo [50] or indirect inference methods like Approximate Bayesian Computation [51,52].

In this paper we shall follow the pure likelihood approach proposed by Rayner and MacGillivray [53]. In this situation a numerical procedure must be selected to maximize (15). They proposed a Nelder–Mead simplex algorithm, reporting some limitations, related also to the need of using several starting points. In the final discussion they also observed that metaheuristic methods like GAs could be more successful in this optimization problem.

In our GA approach we shall consider the fitness:

$$f(\theta) = \exp\{\log\ L(\theta \mid x)/\}.$$

We will simulate data using the typical parameters generator vector $\theta = (A, B, g, k) = (3, 1, 2, 0.5)$, with $c = 0.8$, that leads to an 'interesting far-from-normal distribution' [51, p.192].

Each chromosome will have length $M = 28$, and coding interval boundaries shall be: $A \in [-10, 10]$, $B \in [0, 10]$, $g \in [-10, 10]$ and $k \in [-0.5, 10]$. In this case the grid includes 128 equispaced values for each parameter. If a decoded chromosome provides unacceptable values $B = 0$ or $k = -0.5$ then it is rejected and regenerated.

Concerning sampling variability, Rayner and MacGillivray [53] investigated the approximation of maximum likelihood estimator variability by Cramer–Rao variance bound, which is of order $\mathcal{O}(n^{-1})$. In estimating sampling variability we shall allow for this asymptotic approximation of Σ_S.

4.4. Results

4.4.1. Variability analysis

Sampling and algorithmic variabilities were estimated by simulation for each experiment, using data of length $n = 200$ generated according to parameters specified in previous subsections.

We considered asymptotically efficient estimators, for which $f(n) = n$ in formula (4). We then estimated variability of estimators using 10,000 samples; the mean squared deviations of estimates obtained by software optimization routines from the true parameters allowed us to get a quantification of W_S in (4).

On the other side, GA variability has been estimated using 10 datasets. For each sample we computed variance estimates using $J = 500$ GA runs as shown in formulas (10) and (11); then we considered point by point average of these estimates with respect to g, obtaining final estimates for regression analysis (12).

This latter has been conducted for the experiments with $a = 1/3, 1/2, 1, 2$: goodness of fit results (R^2 coefficients) are summarized in Table 1 for experiments *LAD* and *gk*, and in Table 2 for experiments concerning *AR*. In the latter case we did not include the results for $a = 2$, because they showed a uniformly worse behaviour with respect to the other cases.

Table 1. R^2 coefficient values related to four different regression analysis conducted on the parameters of experiments *LAD* and *gk*, in order to estimate the convergence rate of Σ_{GA}.

Experiment	Parameter	$a = 1/3$	$a = 1/2$	$a = 1$	$a = 2$
LAD	β_0	0.1883	0.4781	0.9775	0.7247
	β_1	0.1943	0.4835	0.9792	0.7298
	β_2	0.1910	0.4790	0.9763	0.7250
gk	A	0.3538	0.6635	0.9525	0.6370
	B	0.2060	0.4949	0.9179	0.5984
	g	0.2722	0.5883	0.7585	0.3511
	k	0.1268	0.3563	0.9548	0.9071

Table 2. Σ_{GA} convergence rate estimates for experiments *AR*.

Parameter	AR_A	AR_B	AR_C	AR_D
ϕ_1	(0.84,0.95,0.91)	(0.92,0.99,0.83)	(0.91,0.98,0.85)	(0.90,0.97,0.86)
ϕ_2	(0.91,0.98,0.83)	(0.99,0.99,0.67)	(0.98,0.99,0.68)	(0.99,0.95,0.55)
ϕ_3	(0.91,0.98,0.82)	(0.98,0.98,0.66)	(0.99,0.92,0.51)	(0.97,0.97,0.64)
ϕ_4	(0.89,0.97,0.85)	(0.96,0.99,0.73)	(0.99,0.96,0.57)	(0.95,0.98,0.71)
ϕ_5	(0.88,0.97,0.85)	(0.94,0.99,0.74)	(0.96,0.98,0.69)	(0.99,0.98,0.63)
ϕ_6	(0.87,0.97,0.86)	(0.89,0.97,0.80)	(0.93,0.98,0.75)	(0.91,0.97,0.77)
ϕ_7	(0.85,0.95,0.88)	(0.93,0.98,0.75)	(0.93,0.98,0.77)	(0.95,0.99,0.72)
ϕ_8	(0.85,0.95,0.89)	(0.98,0.99,0.68)	(0.91,0.98,0.80)	(0.99,0.97,0.62)

Note: R^2 coefficient values are reported in parenthesis with respect to the convergence rate as follows: ($a = 1/3$, $a = 1/2, a = 1$).

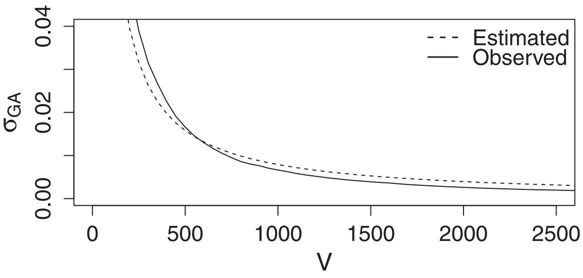

Figure 1. Observed (thick line) and estimated (dashed line) GA variability for parameter β_1 of *LAD* experiment ($w_{GA} = 7.9, R^2 = 0.97$).

Concerning *LAD* and *gk* the best fits are observed for $a = 1$ (as an example, Figure 1 shows the fit for parameter β_2 of *LAD* experiment), while $a = 1/2$ rate is generally dominant for the *AR* experiments. In these latter the values of $tr(W_S)$ and $tr(W_{GA})$ have been estimated at, respectively, 12.26 and 17.74 for AR_A, 11.39 and 7.87 for AR_B, 12.34 and 17.89 for AR_C, 10.59 and 5.46 for AR_D. This can suggest that the complexity may be closely related to the value of the largest parameter, because variability values decrease with ϕ_1, and because AR_A and AR_C show a similar behaviour. We shall perform the tradeoff analysis in the next subsection considering only experiment AR_A (hereinafter referred to as *AR*).

Table 3 reports the results on estimates of $tr(W_S)$ and $tr(W_{GA})$ for the three applications, obtained with a linear convergence rate for *LAD* and *gk*, and with a square root convergence rate for *AR*.

Table 3. Sampling and GA variability components estimates.

Experiment	tr(W_S)	tr(W_{GA})
LAD	5.38	23.18
AR	12.26	17.74
gk	103.39	3897.25

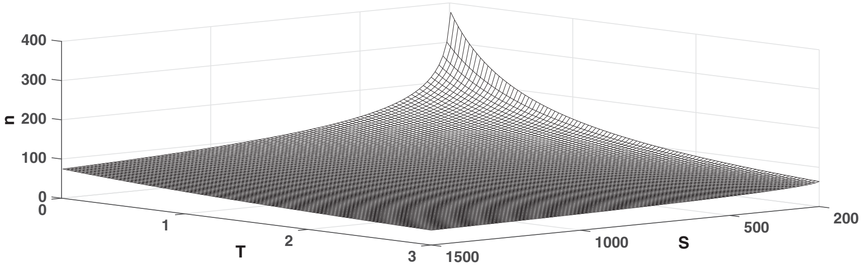

Figure 2. Behaviour of optimal *n* for experiment *LAD*.

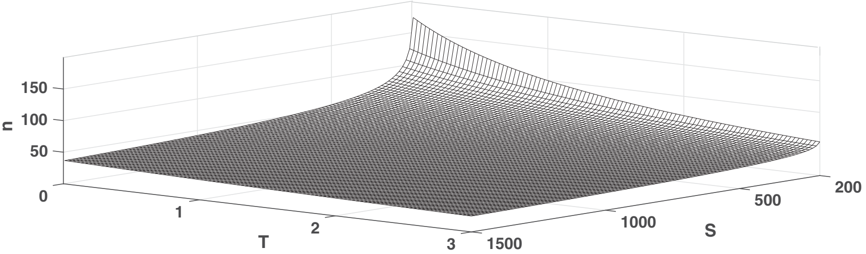

Figure 3. Behaviour of optimal *n* for experiment *AR*.

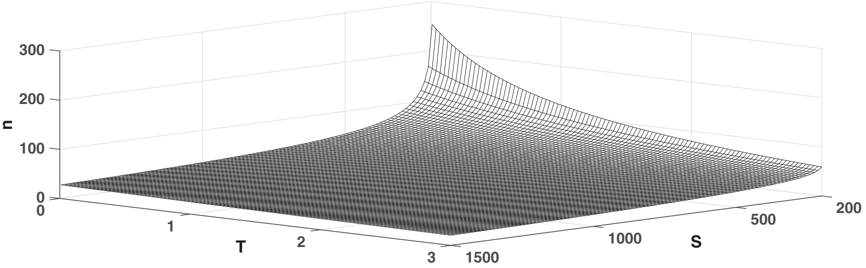

Figure 4. Behaviour of optimal *n* for experiment *gk*.

4.4.2. Tradeoff

The tradeoff will be discussed for the three applications by evaluating optimal sample size $\tilde{n}$, minimizing $\mathrm{tr}(\Sigma_{TOT})$ under the costs constraint. We will assume a fixed total cost $C = 10^5$ and a grid of values for linear cost functions S (sampling) and T (computational), in order to study the effect of costs on optimal allocation. Comments on $\tilde{V}$ can be derived by complement. We shall make some remarks also for the case in which computational cost T

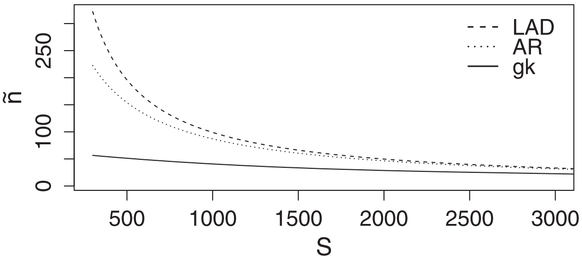

Figure 5. Optimal sample size with fixed estimated computational cost.

is estimated with time (in seconds) needed in our computer to evaluate fitness in the three experiment, using *gk* as corner point. In this way we can make more realistic comparative comments.

Figure 2 shows the behaviour of optimal *n* (on vertical axis) for *LAD* with respect to a grid of values for cost functions *S* and *T*. It obviously increases to large values as costs *S* and *T* decrease, and rapidly decreases as they increase. Figure 3 shows the analogous plot for *AR*. This experiment has a slower GA convergence rate with respect to *LAD* and *gk*, possibly because of the effect of model identification in the fitness (e.g. estimating a ϕ_i value slightly different from zero may imply a slight decrease of the residual sum of squares but *k* is one unit larger in (14)). For this reason values of optimal *n* are generally lower than *LAD*. Perspective plot for *gk* (Figure 4) shows a similar behaviour of optimal *n* with respect to *AR*, because even if in this case there is a linear GA convergence rate, the experiment is more complex ($\mathrm{tr}(W_{GA})/\mathrm{tr}(W_S)$ ratio is much larger).

Lastly we shall make some comments on the behaviour of $\tilde{n}$ when sampling cost *S* varies and fitness evaluation cost *T* is estimated in each experiment by elapsed execution time (in seconds) of our computer for single fitness evaluation, taking *gk* as corner point. Results are: $T_{LAD}/T_{gk} = 0.007$ and $T_{AR}/T_{gk} = 0.101$. Figure 5 shows the behaviour of $\tilde{n}$ in this more realistic scenario, where each computational cost ratio has been multiplied by a constant to highlight the behaviour of each experiment. In this case the three curves are ranked with respect to computational cost and experiment complexity, that is related on both GA convergence rate and variability ratio $\mathrm{tr}(W_{GA})/\mathrm{tr}(W_S)$ magnitude. *gk* experiment shows lowest values of $\tilde{n}$, but when *S* increases the three experiments tend to conform to common values, suggesting that a large sampling cost could have a larger influence in the tradeoff than model complexity.

5. Discussion

In this paper we considered parametric estimation problems involving GAs, for which we proposed a theoretical framework for analyzing variability of results. In this context we studied the effect of statistical and computational elements of the estimation process on variability. Then we introduced some cost functions related to both data acquisition and algorithm performance in order to specify a statistical and computational tradeoff analysis. Lastly we illustrated the proposed framework on three model building examples, for gaining some insights on optimal allocation of resources. Results of applications showed how the behaviour of optimal sample size changes with complexity of experiment. A comparative analysis of the three experiments in which computational cost was estimated also suggested that large sampling cost could have a greater influence on optimal values than model complexity.

The present study could be improved by considering other scalar summaries of statistical and computational variability. For example the determinant of Σ_S and Σ_{GA} could be more appropriate than trace. An other direction for further research is to generalize this framework to other statistical problems in which GAs are involved. In fact there are many complex optimization problems in the statistical field, and understanding variability and tradeoff more in deep could facilitate the integration of GAs among standard statistical methods. Lastly, the discussion of statistical and computational tradeoff could be extended also to estimation problems in which other nature-inspired algorithms for continuous optimization are employed, like *differential evolution* (DE) [54] or *particle swarm optimization* (PSO) [55], for which there is direct real coding. In fact the specific stochastic elements in these algorithms, for example the differential mutation in DE or the parameter regulating particle velocity in PSO, could provide different convergence rates of algorithmic variability.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Manuel Rizzo http://orcid.org/0000-0003-1927-969X
Francesco Battaglia http://orcid.org/0000-0002-9791-9771

References

[1] Dillon J, Lebanon G. Stochastic composite likelihood. J Mach Learn Res. 2010;11:2597–2633.
[2] Wang T, Berthet Q, Samworth RJ. Statistical and computational trade-offs in estimation of sparse principal components. Ann Stat. 2016;44(5):1896–1930.
[3] Yang Y, Wainwright MJ, Jordan MI. On the computational complexity of high-dimensional Bayesian variable selection. Ann Stat. 2016;44(6):2497–2532.
[4] Chandrasekaran V, Jordan MI. Computational and statistical tradeoffs via convex relaxation. Proc Natl Acad Sci. 2013;110(13):E1181–E1190.
[5] Shender D, Lafferty J. Computation-risk tradeoffs for covariance-thresholded regression. In: Dasgupta S, McAllester D, editors. Proceedings of the 30th international conference on machine learning Vol. 28. Atlanta (GA); 2013. p. 756–764.
[6] Jordan MI. On statistics, computation and scalability. Bernoulli. 2013;19(4):1378–1390.
[7] Berthet Q, Chandrasekaran V. Resource allocation for statistical estimation. Proc IEEE. 2016;104(1):111–125.
[8] Bruer JJ, Tropp JA, Cevher V, et al. Designing statistical estimators that balance sample size, risk, and computational cost. IEEE J Sel Top Signal Process. 2015;9(4):612–624.
[9] Chen Y, Xu J. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. J Mach Learn Res. 2016;17(27):1–57.
[10] Agarwal A. Computational trade-offs in statistical learning [dissertation]. Berkeley (CA): University of California; 2012.
[11] Winker P, Maringer D. The convergence of estimators based on heuristics: theory and application to a GARCH model. Comput Stat. 2009;24(3):533–550.

[12] Fitzenberger B, Winker P. Improving the computation of censored quantile regressions. Comput Stat Data Anal. 2007;52(1):88–108.

[13] Mandes A, Gatu C, Winker P. Convergence of heuristic-based estimators of the GARCH model. In: Borgelt C, Gil MA, Sousa JMC, Verleysen M, editors. Towards advanced data analysis by combining soft computing and statistics. Berlin, Heidelberg: Springer; 2013. p. 151–163.

[14] Holland JH. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press; 1975.

[15] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading: Addison-Wesley; 1989.

[16] Michalewicz Z. Genetic algorithms + data structures = evolution programs. Berlin: Springer; 1996.

[17] Mitchell M. An introduction to genetic algorithms. Cambridge: MIT Press; 1998.

[18] Rudolph G. Convergence properties of evolutionary algorithms. Hamburg: Verlag Dr. Kovac; 1997.

[19] Rojas Cruz JA, Pereira AGC. The elitist non-homogeneous genetic algorithm: almost sure convergence. Stat Probab Lett. 2013;83(10):2179–2185.

[20] Pereira AGC, de Andrade BB. On the genetic algorithm with adaptive mutation rate and selected statistical applications. Comput Stat. 2015;30(1):131–150.

[21] Pereira AGC, Campos VSM. Multistage non homogeneous Markov chain modeling of the non homogeneous genetic algorithm and convergence results. Commun Stat-Theory Methods. 2016;45(6):1794–1804.

[22] Baragona R, Battaglia F, Poli I. Evolutionary statistical procedures: an evolutionary computation approach to statistical procedures, design and applications. Berlin: Springer-Verlag; 2011.

[23] Chatterjee S, Laudato M, Lynch LA. Genetic algorithms and their statistical applications: an introduction. Comput Stat Data Anal. 1996;22(6):633–651.

[24] Karavas VN, Moffitt LJ. Evolutionary computation of a deterministic switching regressions estimator. Comput Stat. 2004;19(2):211–225.

[25] Kapanoglu M, Ozan Koc I, Erdogmus S. Genetic algorithms in parameter estimation for nonlinear regression models: an experimental approach. J Stat Comput Simul. 2007;77(10):851–867.

[26] Rizzo M, Battaglia F. On the choice of a genetic algorithm for estimating GARCH models. Comput Econ. 2016;48(3):473–485.

[27] Wright AH. Genetic algorithms for real parameter optimization. In: Rawlins GJE, editor. Foundation of genetic algorithms. San Mateo (CA): Morgan Kaufmann; 1991. p. 205–218.

[28] Kreinovich V, Quintana C, Fuentes O. Genetic algorithms: what fitness scaling is optimal?. Cybern Syst. 1993;24(1):9–26.

[29] Eiben AE, Smit SK. Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm Evol Comput. 2011;1(1):19–31.

[30] Grefenstette JJ. Optimization of control parameters for genetic algorithms. IEEE Trans Syst Man Cybern. 1986;16(1):122–128.

[31] Casella G, Berger RL. Statistical inference. Pacific Grove (CA): Duxbury; 2002.

[32] Derrac J, Garcia S, Hui S, et al. Analyzing convergence performance of evolutionary algorithms: a statistical approach. Inf Sci. 2014;289:41–58.

[33] Oliveto PS, Witt C. On the runtime analysis of the simple genetic algorithm. Theor Comput Sci. 2014;545:2–19.

[34] Auger A, Doerr B, editors. Theory of randomized search heuristics: foundations and recent developments. Singapore: World Scientific; 2011.

[35] Prügel-Bennett A, Rogers A. Modelling genetic algorithm dynamics. In: Kallel L, Naudts B, Rogers A, editors. Theoretical aspects of evolutionary computing. Berlin: Springer-Verlag; 2001. p. 59–58.

[36] Shapiro JL. Statistical mechanics theory of genetic algorithms. In: Kallel L, Naudts B, Rogers A, editors. Theoretical aspects of evolutionary computing. Berlin: Springer-Verlag; 2001. p. 87–108.

[37] Reeves CR, Rowe JE. Genetic algorithms: principles and perspectives – a guide to GA theory. London: Kluwer Academic Publishers; 2003.

[38] Clerc M. Guided randomness in optimization. Hoboken: Wiley; 2015.

[39] R Core Team. R: A language and environment for statistical computing. Vienna: R Foundation for Statistical Computing; 2013. Available from: http://www.Rproject.org/.

[40] Prangle D. gk: g-and-k and g-and-h distribution functions. R package version 0.4.0; 2017. Available from: http://CRAN.R-project.org/package = gk.

[41] Bloomfield P, Steiger WL. Least absolute deviations: theory, applications and algorithms. Boston: Birkhäuser; 1983.

[42] Zhou X, Wang J. A genetic method of LAD estimation for models with censored data. Comput Stat Data Anal. 2005;48:451–466.

[43] Gaetan C. Subset ARMA model identification using genetic algorithms. J Time Ser Anal. 2000;21(5):559–570.

[44] Minerva T, Poli I. Building ARMA models with genetic algorithms. In: Boers EJW, et al., editors. Workshops on applications of evolutionary computation. Berlin, Heidelberg: Springer; 2001. p. 335–342.

[45] Hannan EJ. The estimation of the order of an ARMA process. Ann Stat. 1980;8(5):1071–1081.

[46] Brockwell PJ, Davis RA. Time series: theory and methods. New York: Springer; 1991.

[47] Haynes MA, Gatton ML, Mengersen KL. Generalized control charts for nonnormal data. Brisbane, Australia: School of Mathematical Sciences, Queensland University of Technology; 1997. (Technical Report No. 97/4).

[48] Haynes MA, Mengersen KL, Rippon P. Generalized control charts for non-normal data using g-and-k distributions. Commun Stat-Simul Comput. 2008;37(9):1881–1903.

[49] Peters G, Chen W, Gerlach RH. Estimating quantile families of loss distributions for non-life insurance modelling via L-moments. Risks. 2016;4(2):14.

[50] Haynes MA, Mengersen KL. Bayesian estimation of g-and-k distributions using MCMC. Comput Stat. 2005;20(1):7–30.

[51] Allingham D, King RAR, Mengersen KL. Bayesian estimation of quantile distributions. Stat Comput. 2009;19(2):189–201.

[52] Grazian C, Liseo B. Approximated integrated likelihood via ABC methods. Stat Interface. 2015;8(2):161–171.

[53] Rayner GD, MacGillivray HL. Numerical maximum likelihood estimation for the g-and-k and generalized g-and-h distributions. Stat Comput. 2002;12(1):57–75.

[54] Price K, Storn RM, Lampinen JA. Differential evolution: a practical approach to global optimization. Berlin: Springer Science & Business Media; 2006.

[55] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE Conference on Neural Networks; 1995; Perth, Australia. Piscataway, NJ: IEEE Service Center; 1995. p. 1942–1948.