SAPIENZA
UNIVERSITÀ DI ROMA

Contributions on Evolutionary Computation for Statistical Inference

Dottorato di Ricerca in Scuola di Scienze Statistiche (curriculum Statistica Metodologica) – XXX Ciclo

Candidate

Manuel Rizzo

Thesis Advisor

Prof. Francesco Battaglia

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Methodological Statistics

February 2018

Thesis defended on 26 February 2018
in front of a Board of Examiners composed by:

Prof. Cira Perna (chairman)
Prof. Paolo Giudici
Prof. Nicola Sartori

The thesis has been reviewed by:

Prof. Sandra Paterlini
Prof. Irene Poli

Contributions on Evolutionary Computation for Statistical Inference
Ph.D. thesis. Sapienza – University of Rome

Author's email: manuel.rizzo@uniroma1.it

Contents

List of Figures

List of Tables

Introduction

In many scientific fields the researchers, as well as the end-users, may face and analyze complex problems, in which difficulties may be due to computational constraints or may be intrinsic. There are, for example, many intractable optimization problems not having an analytical solution or being computationally prohibitive. Evolutionary Computation (EC) techniques have been introduced in the 1960s for dealing with such questions. They are based on metaphors of Darwin's principles, biology, genetics, and propose heuristic solutions to approach intricate problems, leading to methods named Evolutionary Algorithms (EAs). The easiness of implementation and the adaptability of such algorithms made EC a generally effective tool in a large variety of application fields.

In statistics there are many situations where complex problems arise, in particular concerning optimization. A general example is when the statistician needs to select, inside a prohibitively large discrete set, just one element, which could be a model, a partition, an experiment, or such: this would be the case of model selection, cluster analysis or the design of experiment. In other situations there could be an intractable function of data, such as a likelihood, which needs to be maximized, as it happens in model parameters estimation. These kind of problems are naturally well suited for EAs, and in the last 20 years a large number of papers has been concerned with applications of EAs in tackling statistical issues.

The present dissertation is set in this part of literature, as it reports several implementations of EAs for statistics, although being mainly focused on statistical inference problems. Original results are proposed, as well as overviews and surveys on several topics. EAs are employed and analyzed considering various statistical points of view, showing and confirming their efficiency and flexibility.

An outline of the thesis will follow, which includes citations of papers and publications, concerned also with conference presentations.

In Chapter 1 a general overview of EC is provided. Starting from an historical background of the field, structure of generic EAs is then discussed. The methods studied in the dissertation, Genetic Algorithms (GAs) above all, will be described more in depth. Chapter ends with a wide review of statistical applications of EAs, giving an idea of state-of-art.

In Chapter 2 EAs are applied to parametric estimation problems. When they are employed in such analysis a novel form of variability, related to their stochastic elements, is introduced. We shall analyze both variability due to sampling, associated with the selected estimator, and variability due to the EA. So in this chapter the EA is studied from a frequentist inference point of view, and its behaviour is asymptotically analyzed as the number of iterations increase. This analysis is set in a framework of statistical and computational tradeoff question, crucial in nowadays problems, by introducing cost functions related to both data acquisition and EA iterations. The proposed method will be illustrated by means of some model building problem examples. The topics of this chapter can be also found in following manuscripts:

2018 Statistical and Computational Tradeoff in Genetic Algorithm-Based Estimation. Under review (arXiv:1703.08676) (with F. Battaglia)

2017 On Variability Analysis of Evolutionary Algorithm-Based Estimation. In F. Greselin, F. Mola, M.A. Zenga (eds) *Cladag 2017 Book of Short Papers.* Universitas Studiorum. ISBN 978-88-99459-71-0

2016 Statistical and computational tradeoff in econometric models building by genetic algorithms. In A. Blanco-Fernandez, G. Gonzalez-Rodriguez (eds) *CFE-CMStatistics 2016 Book of Abstracts.* University of Seville. ISBN 978-9963-2227-1-1 (with F. Battaglia)

Chapter 3 is concerned with EAs employed in Markov Chain Monte Carlo (MCMC) sampling. When sampling from multimodal or highly correlated distribution is concerned, a possible strategy suggests to run several chains in parallel, in order to improve their mixing. If these chains are allowed to interact with each other then many analogies with EC techniques can be observed, and this has led to research in many fields. The chapter aims at reviewing various methods found in literature which conjugates EC techniques and MCMC sampling, in order to identify the specific and common procedures, and unifying them in a framework of EC. Although MCMC is a general topic, and this is confirmed by the diversity of research papers analyzed in the overview, it is generally employed in Bayesian inference procedures as far as statistical problems are concerned. The strength of EAs in this case is the capability of exploring the support of target distributions, which can be a posterior for example, by use of its operators and strategies. This work has been presented at conference:

2017 Evolutionary Computation and multiple chains MCMC sampling: an overview. In G. Gonzalez-Rodriguez, M. Hofmann (eds) *CFE-CMStatistics 2017 Book of Abstracts*. University of London. ISBN 978-9963-2227-4-2.

In Chapter 4 the GA is employed for building a complex statistical model. Here the focus is on a specific field, that is time series analysis, and a model for dealing with seasonality and structural changes is introduced. First issue is accounted by use of Periodic AutoRegressive (PAR) models, characterized by a large number of parameters; as far as structural changes can occur at each time instant, in our model we allow several PAR models linked at different changepoints. GAs are employed for identifying this model, as a complex combinatorial optimization problem is concerned. Effectiveness of the procedure is shown on both simulated data and real examples; these latter refer to river flow data in hydrology, for which also forecasting accuracy of fitted model is evaluated. The topic of this chapter is included in following papers:

2018 Periodic autoregressive models with multiple structural changes by genetic algorithms. To appear. Mathematical and Statistical Methods for Actuarial Sciences and Finance 2018 conference (with F. Battaglia and D. Cucina)

2018 Multiple changepoint detection in periodic autoregressive models with applications to river flow analysis. Under review (arXiv:1801.01697) (with D. Cucina and E. Ursu)

Chapter 5 contains some concluding remarks, concerning also future work, and a summary of the thesis.

Chapter 1

Evolutionary Computation and Statistics

1.1 Origins of Evolutionary Computation

Methods which are known today under the comprehensive name of *Evolutionary Computation* (EC) originated in the second half of 20th century. There was no single precursor, but rather several independent groups working on different lines of research, having in common the problem of dealing with complex situations, that would have converged during subsequent decades to a common EC framework.

One essential discussion originated in relation with the research that was creating Artificial Intelligence paradigms. Researchers in this field, in fact, had to specify concepts such as *intelligence* and *learning* in order to successfully build "thinking" machines. In a fundamental work, Lawrence Jerome Fogel and his group (Fogel et al., 1966), basing on previous discussions by Alan Turing, Leonard Ornstein and Walter B. Cannon among others, defined intelligence as "the capability of a system to adapt its behaviour to meet its goal in a range of environments", which suggested that both intelligence and learning concepts could have been set in a kind of evolutionary flow process. In the same work they also developed a correspondence between natural evolution, in the sense of Charles Darwin's theories, and the scientific method. This latter discussion supported the idea that an evolutionary process could be mechanized and programmed on a computing machine in algorithmic form. Starting from these ideas, Fogel and his group introduced *Evolutionary Programming*, the earliest EC method, in which a number of agents, called *finite state machines*, are assigned to predict some outputs starting from certain inputs, through a process which improves prediction at each iteration. This method has been refined through the years and has also been applied to other fields of science.

Along with Fogel's, two more research groups are universally recognized as essential for the development of EC paradigms: Ingo Rechenberg and Hans-Paul Schwefel worked on an algorithm called *Evolution Strategies*, designed to solve complex real-valued optimization problems by use of evolutionary methaphors, which still represents an established technique (Schwefel, 1975; Beyer & Schwefel, 2002); John Henry Holland, instead, employed the concept of evolutionary process for analyzing complex adaptive systems, capable of dealing with an uncertain and changing environment, using metaphors of biological populations evolution and genetics (Holland, 1967). The result of subsequent years of research is the *Genetic Algorithm*, the most successful EC technique, for simplicity and variety of applications. This latter algorithm has been widely studied in this thesis, mostly for optimization purposes, and it will be deepened in the next sections.

Across the decades EC has been deeply refined, leading to a huge number of algorithms, named Evolutionary Algorithms (EAs), proposed for many different problems and fields of science. A detailed review of these methods is beyond the scope of this dissertation, which will rather consider a small selection of EAs employed in statistical applications. The reader interested in a global overview of EC can refer to authoritative book references by, for example, Fogel (1995, 1998), Bäck (1996), Eiben & Smith (2003), De Jong (2006).

1.2 Evolutionary Algorithms

Although no universally accepted formal definition of EA is available in literature, there are some necessary key elements to contemplate when illustrating such algorithm. De Jong (2006) proposes to consider Charles Darwin evolutionary system as starting point, whose basic elements summarized in Table 1.1. These ingredients are adopted as metaphor to approach computational problem at hand: the population of *individuals* explores and exploits problem environment; birth/death process and variational inheritance regulate dynamics of population through algorithm iterations; the *fitness* is an attribute of each individual, and it might be linked to its goodness.

A simple EA structure is illustrated by the pseudocode in Table 1.2. This kind of template is quite general and little informative from the practical point of view. In this thesis, unless otherwise specified, we shall refer to EAs as optimization method, because it is one of the most prominent fields of application (including the subject of this dissertation), even if this point of view has stimulated some discussion

Individuals
One or more populations of individuals competing for limited resources

Reproduction
The notion of dynamically changing populations due to the birth and death of individuals

Fitness
A concept of fitness which reflects the ability of an individual to survive and reproduce

Inheritance
A concept of variational inheritance: offspring closely resemble their parents,
but are not identical

Table 1.1: Elements of an evolutionary system

in literature (see, for example, De Jong, 1993). In that case individuals represent candidate problem solutions, the fitness is related to objective function of the problem, birth/death process and variational inheritance drive the population through promising areas of search space. Before going any further, it is crucial to specify that EAs are characterized by stochastic moving rules, meaning that a probability distribution is built on possible solutions to be reached in subsequent steps; this also allows to allocate EAs in the category of *stochastic optimization* methods.

That being said, we shall introduce some notation and describe the dynamics of a simple EA: let f denote the fitness function, to be maximized (this can be easily generalized by considering minimization of the additive inverse of f), taking values on set Ω, which can be either discrete or continuous, and possibly multidimensional. Each individual ψ represents a possible solution $\underline{\theta}$ by convenient coding, and $\underline{\theta}^*$ denotes global optimum point of f. At each EA iteration (hereinafter referred to as *generation*) the population of individuals is subject to random *operators*, which allow to build an intermediate population: main operations are *selection*, based on fitness, which discriminate solutions that will contribute to subsequent steps, and *reproduction*, which effectively build new individuals (the *offspring*). The intermediate population is then handled in order to decide which and how many novel solutions will replace old ones, possibly resulting in a general improvement. The stopping criterion can be decided a priori, for example the reaching of a prefixed number of generations, or it may depend on the behaviour of algorithm, that is the case when no significant improvement is observed within a certain number of steps. A useful strategy, named *elitism*, has been proposed, in particular for optimization purposes, in order to maintain in population the best individual found up to current generation, irrespective of the effect of operators. User interested in optimization

```
1) Randomly generate an initial population
Do until some stopping criteria is met
2) Select individuals to be parents (biased by fitness)
3) Produce offsprings
4) Select individuals to die (biased by fitness)
End Do
```

Table 1.2: Basic EA pseudocode

may consider just the flow of these solutions, which is a monotonic non-decreasing sequence with respect to fitness.

Once selected the specific EA to tackle problem at hand, many choices on structure and configurations of algorithm are possible. These latter are linked also to the choice of probability rates of stochastic operators. This is a wide subject in the field of EC: there has been research focused on analyzing configurations before running the EA, an issue named *parameter tuning* (Eiben & Smit, 2011), and also studies on eventuality of online modifications of configuration, and that is the case of *parameter control* (Eiben et al., 1999; Lobo et al., 2007). In the present thesis we shall generally consider basic EAs with, for example, fixed length solution coding, fixed population size, basic operators with fixed probability. Also number of parents and offspring size coincide, so that final intermediate population replaces previous population. These choices have been made for the matter of simplicity and because they have been found effective in literature of statistical applications, including this dissertation.

1.3 Genetic Algorithms

The Genetic Algorithm (GA), is the most successful EA, for simplicity and variety of applications, including statistics. Introduced by Holland (1975) it has been deepened during decades, so that it is recognized as the main combinatorial optimization technique among EAs, as many authoritative books on the subject can confirm (Goldberg, 1989; Davis, 1991; Michalewicz, 1994; Mitchell, 1998; Vose, 1999; Reeves & Rowe, 2003).

Standard binary GA relies on direct biological and genetic inspiration: in fact solutions are coded in strings named *chromosomes*, composed by elements (*genes*) representing the genetic heritage of individual. While information carried by genes is called *genotype*, the practical meaning of solution, who is explicitly passed as argument to fitness function, is named as *phenotype*. Possible values of genes in this algorithm are only 0 or 1, called *bits* like in computer science theory.

There are at least three basic *genetic operators* employed at each generation:

- *Selection*, which randomly chooses solutions for subsequent steps. Among main type of selection we report: *roulette wheel selection*, for which individuals are selected with repetition proportionally to their fitness value; *rank selection*, similar to previous, but in this case selection probabilities are built on fitness ranks rather than absolute values, in order to avoid premature convergence of algorithm; *tournament selection*, for which an individual is compared with a group or with a single solution: if it wins, namely it has a better fitness than competitors, it is selected with probability p, and rejected with complementary probability.

- *Crossover*, the pure reproduction operator. It allows pairs of solutions to combine together, with a fixed rate pC, exchanging part of their genes and creating two new individuals. Original proposal by Holland (1975), called *single point crossover*, considers a common randomly chosen cutting point in parents, and two children are built by taking the left part from the first parent and the right part from the other, and vice versa. Other possible choices of crossover are the *k point*, that generalizes previous method, or *uniform*, which allows each individual gene of parents to be swapped, with probability 0.5 (also a generic rate p can be adopted, leading to *parametrized uniform crossover*).

- *Mutation* operator allows every bit to flip its value from 0 to 1, or vice versa, with a fixed probability pM, simulating random mutations in nature.

These operators are designed to balance two fundamental search strategies: *exploitation*, for which promising areas of search space are deepened, and it is assigned to selection and crossover, and *exploration*, designed to avoid premature convergence of algorithm, accomplished by allowing evaluation of random solutions (independently from fitness), possibly reaching unexplored areas of search space (task assigned to mutation).

Considerable success of GA has encouraged researchers to apply its philosophy also to non-discrete optimization problems. Wright (1991) and Goldberg (1991) proposed a *floating point GA* in order to solve continuous optimization problems (see also Herrera et al., 1998, for a comprehensive review). This new formulation employs direct real coding, so that genotype and phenotype coincide and computation time needed for decoding is saved. Whilst selection operator is unaffected by change of coding, mutation and crossover, as far as they operate on genotype, needs to be reformulated.

Among crossover between two parents some proposals rely on generating offsprings taking values, for each gene, from the real interval (*flat crossover*) or the discrete set (*discrete crossover*) composed by parents corresponding parameters; other authors introduced operators generating new genes basing on combinations between parents values (*arithmetic* and *linear crossover*); also a single-point crossover analogous to binary case (*simple crossover*) has been studied.

Mutation strategies range from simple operations, like random sampling within genes boundaries (*random mutation*), to more complex techniques, taking advantage of informations on local optima (*real number creep*) or considering sophisticated probability distributions (*ebein's mutation* or *modal mutation*). Also the equivalent of mutation operation in Evolution Strategies can be adopted.

A significant extension of standard GA proposes parallelization in order to save computational time, leading to the *Parallel GA* (for a survey, see Cantù-Paz, 1998). One special case is the Distributed GA (Tanese, 1989), for which the whole population is divided into a set of subpopulations and algorithm runs on each subpopulation. Information exchange between subpopulations is performed at selected steps by allowing individuals called *migrants* to shift to a different subpopulation, in order to prevent premature convergence. This strategy has shown good performances on several scenarios compared with standard GA.

1.4 Other Evolutionary Algorithms

We shall now shortly describe Differential Evolution and Estimation of Distribution Algorithm, two EAs which have been studied in this thesis along with GAs.

1.4.1 Differential Evolution

Rainer Storn and Kenneth Price introduced Differential Evolution (DE) in the 1990s as a simple and powerful tool for continuous global optimization (Storn & Price, 1997; Price et al., 2006). In this algorithm solutions are directly coded as real vectors, and the evolution consists of geometrical updating based on other vectors in the population. *Differential mutation* operator, in fact, for each vector $\underline{x}_i$ in the population builds a *mutant* $\underline{v}_i$ as follows:

$$\underline{v}_i = \underline{x}_{R0} + F(\underline{x}_{R1} - \underline{x}_{R2}),$$

where $\underline{x}_{R0}, \underline{x}_{R1}$ and $\underline{x}_{R2}$ are solutions selected in such a way that $i \neq R0 \neq R1 \neq R2$, and F is a positive scale factor. A *trial vector* is then built by parametrized uniform crossover, for which each gene can be inherited by either original vector $\underline{x_i}$ or mutant $\underline{v}_i$, with fixed probability CR. Generation terminates with a selection step: if the trial vector has a better fitness then original solution $\underline{x}_i$ it is retained, otherwise it is rejected and the original solution is maintained. This kind of selection mechanism ensures elitist property in DE.

Like the majority of EAs, many modifications of standard DE algorithm have been proposed in literature: the informed choice of vectors in differential mutation, for example including the best individual of previous generation; adoption of a randomized scale factor F, leading to so-called *dither* and *jitter* strategies, depending on whether randomization is done with respect to individuals or parameters, can make DE theoretically tractable (Zaharie, 2002).

1.4.2 Estimation of Distribution Algorithm

Estimation of Distribution Algorithm (EDA), or Probabilistic Model-Building Genetic Algorithm, although being a standard EA is very different from methods described previously. It has been introduced in a basic form by Mühlenbein & Paass (1996), and since then many sophisticated methods have been introduced (for a comprehensive account see Larrañaga & Lozano, 2001; Pelikan et al., 2002; Lozano et al., 2006). In EDA philosophy new solutions are generated at each generation g by a probability distribution $P^{(g)}$, estimated on the basis of population at generation g as follows: a subset $\mathbf{x} = \{\underline{x}_1, ..., \underline{x}_K\}$ of population at time g is drawn according to some selection operator; $\mathbf{x}$ is treated as a random sample from a multivariate probability distribution $P^{(g)}$, and it is used to estimate its parameters. In such a way features of selected individuals are used to "inform" the probability distribution of population, so that new generated individuals according to $P^{(g)}$ will be likely to preserve them.

Choices on type of distribution $P^{(g)}$ discriminates the type of EDA: a simple example is the Univariate Marginal Distribution Algorithm (Mühlenbein & Paas, 1996), in which $P^{(g)}$ is a multivariate normal with independent components; in Factorized Distribution Algorithm (Mühlenbein et al., 1999) fitness function is assumed to be additively decomposed in terms depending each on a subset of population, and $P^{(g)}$ is factorized as a consequence by including marginal and conditional distributions depending on these subsets; the Bayesian Optimization Algorithm (Pelikan, 2005) employs Bayesian networks in order to predict value of new solutions.

1.5 Convergence of Evolutionary Algorithms

Convergence of EAs is a difficult task to analyze, because probability distributions of moving rules does not usually have a known form, except for simple test cases.

The main reference on the subject is Rudolph (1997), in which convergence properties of many classes of EAs under several simplificative hypothesis are analyzed. Markov Chain theory is often employed for modeling algorithm dynamics, because the behaviour of many EAs at a certain generation can be described by considering only the population of solutions at previous step. In the same book Rudolph states a fundamental theorem:

Theorem 1. *Let us consider an EA with mutation probability* $pM \in (0,1)$*, arbitrary crossover operator and an elitist selection rule. The sequence* $D^{(g)} = f(X^{(g)}) - f^*$*, where* $f(X^{(g)})$ *is the fitness of best solution found up to generation* g *and* f^* *is the global optimum of* f*, is a nonnegative supermartingale that converges almost surely and in mean to zero.*

This latter theorem includes a wide class of EAs because, informally, it states that the convergence to global optimum is ensured if an elitist strategy is employed and if there is a nonzero probability of reaching any point of search space. In GAs, for example, it is trivial to satisfy these two properties. Hu et al. (2013), recently, stated global convergence of a modified DE algorithm (see also Knobloch et al., 2017) basing on Rudolph's philosophy: as far as the standard DE is naturally elitist, a mutation operator which consists in the random regeneration of solutions is periodically included before selection step, allowing each point of search space to be reached with nonzero probability. Studies concerning convergence of EDA can be found in Mühlenbein & Mahnig (1999) and Zhang & Mühlenbein (2004).

As far as GAs are concerned, generalizations have been proposed for extending Theorem 1 to time varying mutation or crossover rates (or both) by modeling the algorithm as a non homogeneous Markov Chain (Rojas Cruz et al., 2013; Pereira et al., 2015; Pereira et al., 2016). The latter reference includes also a review of other methods of studying GA convergence by Markov Chain modeling.

1.6 Evolutionary Algorithms in statistical applications

There are many situations in the statistical field where complex optimization problems arise, for multiple possible reasons: the objective function is non differentiable or has many discontinuities, and an analytical optimal solution could not be available; search space is prohibitively large or irregular (or both); sometimes the number of variables in statistical models or sample size may lead to dramatically time consuming procedures.

These kind of reported situations are sometimes beyond the means of standard procedures, so EC methods, naturally suited for such issues, have been introduced in statistical methodologies in the last decades (see book reference Baragona et al., 2011). This is also reflected by the number of R packages (R Core Team, 2013) introduced for dealing with EAs, as they include GA (*GA*, Scrucca, 2013), DE (*DEoptim*, Mullen et al., 2011), EDA (*copulaEDA*, Gonzalez-Fernandez & Soto, 2012), Covariance Matrix Estimation-Evolution Strategies (*cmaes*, Trautmann et al., 2011), Artificial Bee Colony Optimization (*ABCoptim*, Vega Yon & Muñoz, 2016), Self-Organising Migrating Algorithm (*soma*, Clayden, 2014) and other nature inspired or hybrid algorithms such as Particle Swarm Optimization (*pso*, Bendtsen, 2012, and *hydroPSO*, Zambrano-Bigiarini & Rojas, 2014) or Memetic Algorithm (*Rmalschains*, Bergmeier et al., 2016).

A non-exhaustive survey of statistical applications of EAs will follow, with the scope of illustrating various possibilities of implementation (most of which involving GAs) and giving an idea of state-of-art.

Parametric estimation

Paper by Chatterjee et al. (1996), employing GAs for model parametric estimation, is generally considered the first proposal to employ EAs in pure statistical applications. This kind of problem justifies EAs implementation when the objective function, such as a likelihood, is difficult to analyze by standard methods. GAs are favored researchers pick in this framework, although most of the problems considered refer to continuous supports. These works generally use a rule for representing a parameter defined on a real interval by binary coding (see, for example, Wright, 1991), and the standard binary GA is then employed, usually adopting fitness as a transformation of objective function. Here we report contributions on estimation of nonlinear regression (Kapanoglu et al., 2007), Johnson distribution family (Niermann, 2006), logistic regression (Chatterjee et al., 1996; Pasia et al., 2005), switching regression (Karavas & Moffitt, 2004), robust regression (Nunkesser & Morell, 2010), least absolute regression with censored data (Zhou & Whang, 2005), support vector regression (Santamarìa-Bonfil et al., 2015), ARMA models (Abo-Hammour et al., 2011), GARCH (Rizzo & Battaglia, 2016). Parametric estimation via EAs will be the main topic of Chapter 2.

Model identification

A natural application of GAs in statistics is model selection (or identification), in a both independent and dependent observations framework. The generic solution is a possible model: in an independent framework it is generally encoded to denote the presence or absence of variables; in time series also indications on model order must be provided. Fitness function is usually linked to penalized likelihood criteria, like AIC or BIC, or goodness of fit measures like the R^2 coefficient. Proposals in literature include identification of models such as linear regression (Minerva & Paterlini, 2002; Kapetanios, 2007), logistic regression (Aly, 2016), graphical models (Roverato & Poli, 1998). In time series analysis some contributions have been made in order to identify ARIMA models (Gaetan, 2000; Ong et al., 2005), periodic models (Ursu & Turkman, 2012; Ursu & Pereau, 2017), bilinear time series (Chen et al., 2001) and also complex nonlinear and nonstationary models. A review of such applications is included in Chapter 4, as its main subject is nonstationary time series models identification by GAs.

Clustering

Clustering observations sharing similar features has always been a fundamental topic in statistics and many other fields, because it implies considerable gain in simplicity and interpretability. In an era where sample sizes are growing exponentially it is evidently a highly demanding issue. These kind of problems are characterized by a very large discrete set of solutions, each of which generally refers to a possible partition of the considered dataset, often growing fast with problem dimension: this make clustering problems suitable for EAs applications, in particular GAs. After seminal papers by Raghavan & Birchand (1979), Bandyopadhyay et al. (1995) and Murthy & Chowdhury (1996) among others, there have been many contributions tackling clustering problem in different ways, although many of them do not generally refer to the statistical field. Since GAs are naturally suitable for non hierarchical methods and hard clustering, most of contributions have been made in this framework, even if also methods employing hierarchical strategies (Kuncheva, 1995; Tseng & Yang, 2001) and fuzzy logic (Choi & Moon, 2007; Maulik & Bandyopadhyay, 2003) have been introduced. In Baragona et al. (2011, sec. 7.2.2) a GA version of quick partition clustering was introduced; Falkenauer (1998) proposed the Grouping Genetic Algorithm, which directly encodes candidate partition, and has been found essential in subsequent research (Hruschka & Ebecken, 2003; Mutingi & Mbohwa, 2017, which provided a fuzzy version of method); model-based clustering by GAs has been studied in Baragona & Battaglia (2003) and Paterlini & Minerva (2003); some comparative accounts can be found in Baragona et al. (2006) and Paterlini & Minerva (2003); Paterlini & Krink (2006) proposed DE and Particle Swarm Optimization for partitional clustering; recently Vo-Van et al. (2017) introduced a modified GA for clustering probability density functions. Many contributions have been made by S. Bandyopadhyay, U. Maulik and S. Saha research group, proposing to evolve centroids and similar measures as in k-means algorithm (Maulik & Bandyopadhyay, 2000; Bandyopadhyay & Maulik, 2002), focusing on genetic multiobjective optimization (Bandyopadhyay et al., 2007; Saha & Bandyopadhyay, 2013; Pal et al., 2018) or basing on a novel distance measure based on symmetry as fitness function (Bandyopadhyay & Saha, 2007; Saha & Bandyopadhyay, 2009; Saha, 2017).

Concerning hybrid EAs for clustering, Jank (2006) provided a review of links between EAs and EM algorithm: an example is the case where EAs are employed for providing promising starting point for EM. Also GAs for clustering time series have been introduced (Baragona et al., 2001a; Bandyopadhyay et al., 2001).

Design of experiments

When designing experiments in areas such as biology or chemistry there is often a wide variety of possible factors to be combined in the analysis. For example the researcher must evaluate multiple combinations of factors (whose size may be not fixed), their levels, and also interactions between these factors. As far as high dimensional problems are concerned, standard experimentation may be economically infeasible, so novel methods have been proposed, including designs based on EAs. In these latter the evaluation of possible combinations of factor, levels and such is driven by the evolutionary paradigm. For example, Broudiscou et al. (1996) employed GAs for selecting D-optimal asymmetric designs; study in Angelis (2003) is concerned with finding A-optimal incomplete block designs by EAs; for a recent review of EAs for design of experiments see Lin et al. (2015). A generic framework of Evolutionary Design of Experiments has been proposed in many papers such as Poli (2006) or Forlin et al. (2007) (see Baragona et al., 2011, Chapter 5, for a summary of these contributions). Beside these proposals also an approach which aim at exploiting features of data obtained for each experiment in algorithm has been proposed: as far as a model is built to predict new candidate solutions, the method has been named Evolutionary Model-Based Experimental Design. Among the different models proposed in this framework we report Neural Networks (De March et al., 2009) and Bayesian Networks (Slanzi et al., 2009; Slanzi & Poli, 2014).

Bayesian analysis

A different kind of application, with a non optimization purpose, is the problem of sampling from complex distributions, mainly in a Bayesian inference framework. In this case researchers take advantage of exploratory features of EAs, and implementation is different compared to previous contributions. An overview of literature that proposes methods conjugating EC and Markov Chain Monte Carlo sampling is provided in Chapter 3.

Among other Bayesian problems, Jung & Marjoram (2011) implemented GAs for the choice of summary statistics weight in Approximate Bayesian Computation analysis; Franconi & Jennison (1997) employed them for finding maximum a posteriori estimates in Bayesian image analysis, while some contributions have also been made in the framework of Sequential Monte Carlo (or Particle Filtering): Higuchi (1997) proposed a new filter method based on GA; Kwok et al. (2005) introduced GA with purpose of mitigating the so-called *sample impoverishment* problem, very common in Particle Filtering.

Other applications

Lastly we shall report some miscellanea statistical applications of EAs: GAs have been proposed for optimal deletion of nodes in Bayesian networks (Larrañaga et al., 1997) and influence diagrams (Gómez & Bielza, 2004), outlier detection in both univariate (Baragona et al., 2001b) and multivariate time series (Cucina et al., 2014), for designing optimal statistical quality control procedures (Hatjimihail & Hatjimihail, 2002). Waagen et al. (1994) proposed hybrid Evolutionary Programming algorithms for nonparametric multivariate mixture density estimation, with classification purposes. In book by Palit & Popovic (2005, Chapter 5) a review of forecasting methods based on EAs is provided.

Chapter 2

Statistical and Computational Tradeoff in Evolutionary Algorithm-Based Estimation

2.1 Variability analysis

According to estimation theory a parameter estimate is naturally subject to sampling variability: in fact if we make inference using two different samples we obtain two possibly different results. This issue had to be deepened in all statistical inference approaches: here we refer to frequentist theory, for which sampling variability is closely related to the variability of selected estimators. When EAs are employed in the estimation process a new form of variability is introduced in the analysis, due to the stochastic nature of the algorithm. It refers to elements like the starting population, selection mechanism, stochastic reproduction rules: as a result of this, if we run an EA several times using the same data we may obtain different results. The total variability of an EA-based estimate can be easily decomposed in these two forms of variability, as shown in Baragona et al. (2011, p. 50) for the univariate case.

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

$$\theta^*(\underline{y}) - \theta = [\widehat{\theta}(\underline{y}) - \theta] + [\theta^*(\underline{y}) - \widehat{\theta}(\underline{y})]. \qquad (2.1)$$

As we will see in the following, the first term in square brackets depends on consistency of the estimates, while the second is related to EA convergence. Both of these quantities, referring to statistical and computational elements of the analysis, must be ensured to converge to zero in probability. A similar issue has been analyzed in Winker & Maringer (2009), where a Threshold Accepting algorithm is employed in a GARCH model estimation problem.

As long as we focus on models indexed by a vector $\underline{\theta} = (\theta_1, ..., \theta_k)$ then in practice we shall consider the corresponding multiparametric of (2.1). This means that we must define two random vectors $\widehat{\underline{\theta}}(\underline{y})$ and $\underline{\theta}^*(\underline{y})$, which are affected, respectively, by sampling variability and EA variability. Whilst $\widehat{\underline{\theta}}(\underline{y})$ is defined as the best statistical estimator, the EA component, for which the sample $\underline{y}$ is held fixed, needs to be defined.

If an elitist strategy is employed then we can define random vector $\underline{\theta}^{*(g)}(\underline{y})$ as the best estimate obtained up to generation g, which corresponds to the best individual of generation g. In our method we shall evaluate EA variability by studying the behaviour of this random vector among EA runs basing on Theorem 1 (Rudolph, 1997), which in our case it implies that sequence $\underline{\theta}^{*(g)}(\underline{y})$, $g = 1, ...$, will converge to $\widehat{\underline{\theta}}(\underline{y})$ when g goes to infinity. This means that when g increases then each EA run gets closer to convergence, so variability between runs tends to decrease as a consequence. So in our framework evaluating EA variability is closely related with studying convergence rate of the algorithm.

Having defined both random vectors $\widehat{\underline{\theta}}(\underline{y})$ and $\underline{\theta}^*(\underline{y})$, we shall also define their variance-covariance matrices, respectively Σ_S and Σ_{EA}, in order to relate to (2.1). Generic (i, j) elements of these matrices are:

$$\sigma_{ij}^S = \mathbb{E}_S[(\widehat{\theta}_i - \theta_i)(\widehat{\theta}_j - \theta_j)], \;\; i, j = 1, ...k,$$

$$\sigma_{ij}^* = \mathbb{E}_{EA}[(\theta_i^* - \widehat{\theta}_i)(\theta_j^* - \widehat{\theta}_j)], \;\; i, j = 1, ...k.$$

σ_{ij}^S and σ_{ij}^* measure the dependence between θ_i and θ_j induced, respectively, by sampling and EA. As long as we need 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_{ii}^S and σ_{ii}^* $(i = 1, ..., k)$ go to zero, with no practical interest on covariances. Therefore, if Σ_{TOT} is defined as the total variance-covariance matrix, then, using linearity of trace and under the same independence assumption of (2.1), we can write:

$$tr(\Sigma_{TOT}) = tr(\Sigma_S) + tr(\Sigma_{EA}). \tag{2.2}$$

In next section we shall employ and study this equation in a situation where both statistical observations recruiting and EA iterations have a certain and fixed cost.

2.2 Statistical and computational tradeoff

2.2.1 Problem specification

In recent years the huge growth in size of datasets and the increasing in computing power have introduced many novel problems in the statistical field. Computational elements, in fact, must now be carefully set in order to carry out successful statistical analysis. These elements may include the choice of computational methodology and must consider some resource or time constraints, which are crucial in real problems. Questions like these are known in literature as *statistical and computational tradeoff* (or time-data tradeoff) problems, which aim at balancing and optimizing statistical efficiency and computational complexity. This is a very general topic, so many different methodologies have been proposed in literature to deal with many different applications. Chandrasekaran & Jordan (2013) considered a class of parameters estimation problems for which they studied a theoretical relationship in the form of a convex relaxation between number of statistical observations, runtime of the selected algorithm and statistical risk. An algebraic hierarchy of these convex relaxations is built to successfully achieve the time-data tradeoff for different algorithms. Dillon & Lebanon (2010) studied consistency of intractable Stochastic Composite Likelihood estimators, whose formula depends also on parameters related to computational elements. Therefore they aimed at balancing statistical accuracy and computational complexity. Shender & Lafferty (2013) studied the tradeoff in Ridge Regression models introducing sparsity in the sample covariance matrix. Wang et al. (2016), in a Sparse Principal Component Analysis framework, addressed the question of whether is possible to find an estimator that is computable in polynomial time, and then analyzed its minimax optimal rate of convergence. Several other studies can be found in Yang et al. (2016), Jordan (2013), Berthet & Chandrasekaran (2016), Bruer et al. (2013), Chen & Xu (2016), Agarwal (2012).

In our framework, assuming that both statistical estimator and EA configurations are fixed, then we must figure out how to optimally balance statistical accuracy and

EA efficiency. If we consider consistent estimators 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. As far as EA efficiency is concerned, we refer to Theorem 1. Informally, an EA converges when g tends to infinity, but it is worth noting that in every EA generation each of the N chromosomes in population is evaluated on the basis of fitness function. Therefore, instead of considering the number of generations, we represent EA efficiency component by the number of fitness function evaluations V, also because it is usually the most computationally expensive step.

That being said, we shall study the behaviour of $tr(\Sigma_S)$ and $tr(\Sigma_{EA})$ 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 assumptions of Theorem 1 are fulfilled, then we can write $tr(\Sigma_S) = \mathcal{O}([f(n)]^{-1})$ and $tr(\Sigma_{EA}) = \mathcal{O}([h(V)]^{-1})$. In that case:

$$tr(\Sigma_{TOT}) = tr(W_S)\frac{1}{f(n)} + tr(W_{EA})\frac{1}{h(V)}, \tag{2.3}$$

where matrices W_S and W_{EA} are constant with respect to n and V, and depend, respectively, from the statistical model and from the EA. It is possible that sample size n may have an effect also on W_{EA}, because fitness function will change as a consequence. For this reason we shall include n in our fitness scaling procedure (details will be given in Section 2.3). In such a way we can strongly restrict the effect of n on the behaviour of algorithm and describe the total variability of an EA estimate by considering decomposition (2.3).

The statistical and computational tradeoff will now be analyzed by introducing some cost functions: $S(n)$ is related to the cost of recruiting 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 $\underline{\theta}^*(\underline{y})$ 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:

$$\left\{ \begin{array}{c} \underset{n,V}{min}\, tr(\Sigma_{TOT}) = tr(W_S)\frac{1}{f(n)} + tr(W_{EA})\frac{1}{h(V)} \\ s.t. \\ C = S(n) + VT(n) \end{array} \right\}$$

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 $tr(W_S)$, $tr(W_{GA})$, $f(\cdot)$ 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 $tr(\Sigma_{TOT})$, 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 fitness function includes a summation over the considered sample. In such a case $T(n) = nT$, $S(n) = nS$ and we can incorporate the cost constraint into the objective function obtaining:

$$\underset{n}{min}\, tr(\Sigma_{TOT}) = tr(W_S)\frac{1}{f(n)} + tr(W_{EA})\frac{1}{h([C - nS]/nT)}.$$

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

$$\tilde{V} = \frac{C - \tilde{n}S}{\tilde{n}T}. \tag{2.4}$$

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

$$\underline{\tilde{n}} = \frac{-SCtr(W_S) \pm C\sqrt{CTtr(W_S)tr(W_{EA})}}{CTtr(W_{EA}) - S^2 tr(W_S)}. \tag{2.5}$$

As far as n is a sample size, then we are interested only in the positive solution $\tilde{n}$ of (2.5).

2.2.2 Consistency and convergence rates

Functions $f(n)$ and $h(V)$ introduced in the previous subsection specify, respectively, consistency rate of statistical part and convergence rate of algorithmic part in equation (2.2). The assumption of linearity is a particular case that simplifies the tradeoff analysis. It is satisfied for $f(n)$ if we consider asymptotic efficient estimators: in that case, under some regularity conditions, $f(n) = n$.

On the other side, the behaviour of $h(V)$ is related to EA convergence rate. This is an essential issue for any optimization algorithm, and in the field of EC it has been analyzed in several ways. A part of literature focuses on comparing EAs with different configurations, identifying the algorithm optimizing convergence time (Eiben & Smit, 2011; Derrac et al., 2014); other researchers have developed more rigorous approaches, focusing on the convergence rate of single chromosome bits, limited to standard test problems like *OneMax* (Oliveto & Witt, 2014; Auger & Doerr, 2011); 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 cumulants (Prügel-Bennet et al., 2001; Shapiro, 2001; Reeves & Rowe, 2003). In such a way GA convergence can be evaluated by considering the limiting cumulants.

Recently, Clerc (2015) has proposed a theoretical framework for analyzing optimization performances. For a general stochastic algorithm (deterministic algorithms are considered as 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 several 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 the 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:

$$\sigma^2(\psi) = \mu(\psi) \int_{\tilde{R}} (r - \bar{r}(\psi))^2 \, p(\psi, r) dr, \tag{2.6}$$

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 (2.6) can be evaluated empirically: conditioning on J observed results $r(1), r(2), ..., r(J)$, obtained with effort ψ, the estimated variance is given by:

$$\hat{\sigma}^2(\psi) = \frac{1}{J-1} \sum_{j=1}^{J} [r(j) - \bar{r}_J(\psi)]^2, \tag{2.7}$$

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

In our method we shall employ a very similar approach for evaluating EA variability. As far as we are interested in convergence of θ_i^* to the optimum $\widehat{\theta}_i$ ($i = 1, ..., k$), then in both (2.6) and (2.7) we plug $\widehat{\theta}_i$ in place of theoretical and empirical means, and θ_i^* in place of results. In that case (2.6) corresponds to variance $\sigma_{ii}^* = \mathbb{E}_{EA}[(\theta_i^* - \widehat{\theta}_i)^2]$ in matrix Σ_{EA}. If we run an EA J times, obtaining $\theta_{1,i}^*, \theta_{2,i}^*, ..., \theta_{J,i}^*$ ($i = 1, ..., k$), then we get the estimates by:

$$\hat{\sigma}^*_{ii} = \frac{1}{J}\sum_{j=1}^{J}[\theta^*_{j,i} - \widehat{\theta}_i]^2, \; i = 1, ..., k. \tag{2.8}$$

The latter gives information on the generic EA result $\underline{\theta}^*_j$. As long as we are studying the behaviour of algorithm when number of generations increases, then we shall specify an expression such as (2.8) for each generation g. That is, we obtain the sequence of k dimensional parameter $\underline{\theta}$ variances, given a fixed maximum number of generations G:

$$\underline{\hat{\sigma}}^{*(g)} = (\hat{\sigma}^{*(g)}_{11}, \hat{\sigma}^{*(g)}_{22}, ..., \hat{\sigma}^{*(g)}_{kk}), \quad g = 1, ..., G. \tag{2.9}$$

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

$$\hat{\sigma}^{*(g)}_{ii} = w_{EA,i}\frac{1}{[V^{(g)}]^a} + \epsilon_g, \quad g = 1, ..., G, \tag{2.10}$$

where $[V^{(g)}]^a$ is the a-th power of the number of fitness evaluations up to generation g and $w_{EA,i}$ is the regression parameter. Out goal is to search for an a for which $[V^{(g)}]^a$ can be considered a reasonable EA convergence rate $h(V)$ for all components $\theta_i, \; i = 1, ..., k$. In that case $w_{EA,i}$ will become part of matrix W_{EA} (2.3).

2.3 Applications

We shall now illustrate the proposed method with some examples: a Least Absolute Deviation Regression estimation (code *LAD*), an Autoregressive model building (code *AR*) and a g-and-k distribution maximum likelihood estimation (code *gk*) problem. These problems will be tackled by GAs and DE. In order to discuss the tradeoff question for each of these experiment, we shall now give details on methods employed for obtaining variability estimates, motivations on choices of estimators and issues on GA and DE implementation. Simulations and computations were implemented by use of software R for all applications, and also R package *gk* (Prangle, 2017) for the last application.

2.3.1 EA configurations issues

In all applications we adopted a scaled exponential fitness, with purpose of maximization, for both GA and DE:

$$f(\underline{\psi}) = \exp\{g(\underline{\theta}; \underline{y})/n\}\,, \tag{2.11}$$

where $\underline{\psi}$ is the chromosome and $g(\underline{\theta}; \underline{y})$ is a problem dependent measure of goodness for solution $\underline{\theta}$. This kind of scaling procedure may allow to modify the shape of fitness function without changing solutions ranking and restrict the effect of sample size on the behaviour of algorithm.

Concerning GA implementation, we employed the standard binary version of the algorithm, with roulette wheel selection, single-point crossover, bit-flip mutation and elitist strategy. We referred to the following rule for encoding a real parameter θ in the real interval $[a, b]$:

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

where H is the number of genes considered and x_j is the j-th bit. As long as our interest is focused on a vector $\underline{\theta} = (\theta_1, ..., \theta_k)$ then the chromosome of length $M = k \cdot H$ includes the coding of each component. Length H of each genes group is constant, while coding interval $[a, b]$ can vary for each parameter. Since we are considering a kind of discretization of a continuous search space, we aim at building a fine grid in such a way that fitness function is adequately smooth on that grid, so that related loss of information is negligible.

Also basic DE has been considered, with standard differential mutation operator and parametrized uniform crossover, but with the slight modification described in Section 1.5, introduced for guaranteeing global convergence of procedure. Therefore at each generation one individual in the population is regenerated uniformly at random within parameter boundaries before the selection step.

A small preliminary simulation study limited to LAD experiment has been conducted for analyzing the effect of choice of configurations on EA variability. In this case we conducted regression (2.10) by: $tr(\Sigma_{EA}^{(g)}) = tr(W_{EA})\frac{1}{[V^{(g)}]} + \epsilon_g$. We considered population sizes $N = 50, 70$ (with related maximum number of generations, respectively, $G = 2000, 1450$) and analyzed following parameter choices:

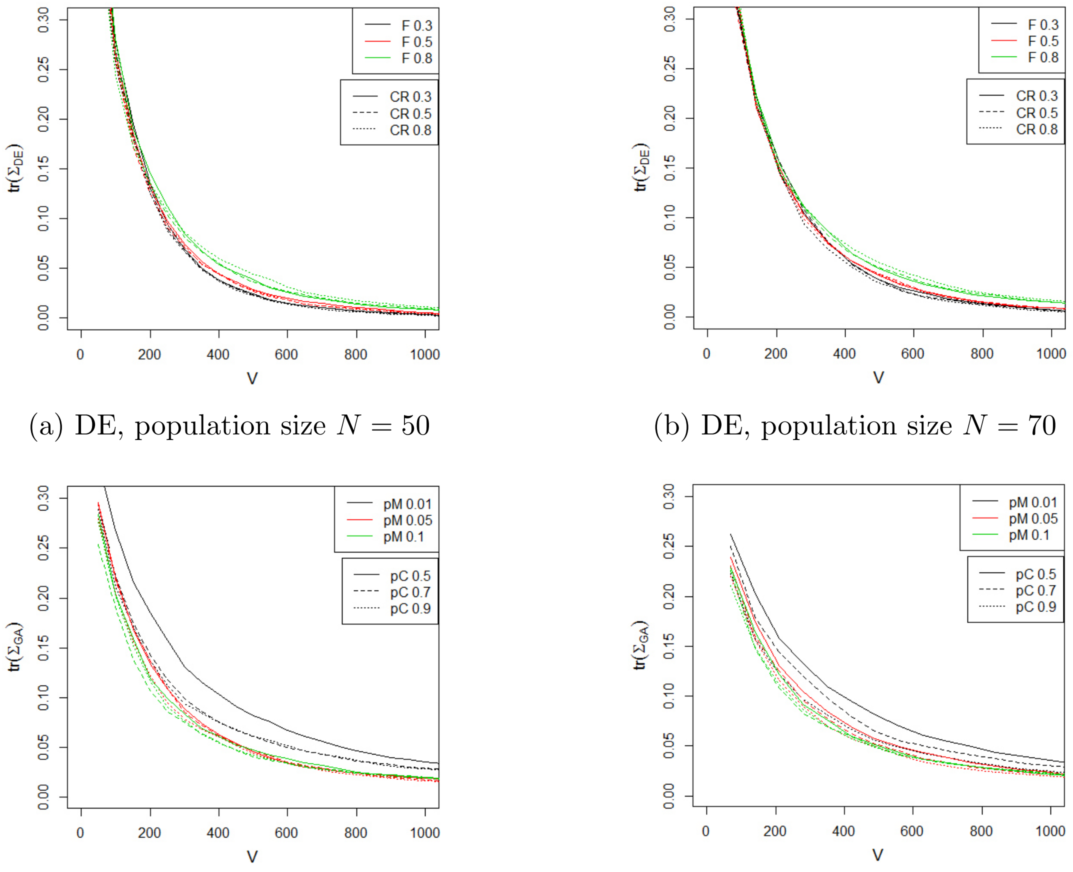
- GA: $pM = 0.01, 0.05, 0.10$; $pC = 0.5, 0.7, 0.9$
- DE: $F = 0.3, 0.5, 0.8$; $CR = 0.3, 0.5, 0.8$,

so that 18 configurations have been implemented for each EA.

Figure 2.1 shows the curves of $tr(\Sigma_{EA}^{(g)})$ estimates for all scenarios. DE experiments show a more homogeneous behaviour with respect to GAs (in particular CR

seems to have a very low effect), and in both algorithms as N increases the differences between experiments in each panel tend to reduce. However DE estimation seems to improve as F decreases, as the best behaviour is registered at 0.3. This is in contrast with general indications on choice of F given in literature devoted to standard DE, for which values of F lower than 0.4 are usually considered as not useful (see Price et al., 2006). Concerning GA the same happens for low mutation rate pM (with a worsening for low pC), possibly because if an elitist strategy is adopted then effect of exploration (task assigned to mutation operator) become dominant in the analysis.

In subsequent analysis a population of $N = 50$ individuals have been adopted in both GA and DE, and a maximum number of generations G has been fixed at 1400. Choices of specific configurations are $pM = 0.1$ and $pC = 0.7$ for GA and $F = 0.3$ and $CR = 0.5$ for DE. If not otherwise specified the initial population is generated uniformly at random.



(a) DE, population size $N = 50$

(b) DE, population size $N = 70$

(c) GA, population size $N = 50$

(d) GA, population size $N = 70$

Figure 2.1: Estimates of EA covariance matrix trace for LAD experiment

2.3.2 Simulation studies

As mentioned in subsection 2.2.2, if an estimator is asymptotically efficient then $f(n) = n$ in formula (2.3): we considered estimators which have this property. Afterwards we estimated sampling variability of estimators by simulating 10^4 samples and computing mean squared deviation of estimates obtained by software optimization routines from the true parameters, to get a quantification of W_S in (2.3).

On the other side, EA variability have been estimated by considering 10 equally-sized datasets. For each sample we computed variance estimates using $J = 500$ EA runs as shown in formulas (2.8) and (2.9); then we considered point-by-point average of these estimates for each g, obtaining final estimates to conduct regression analysis (2.10).

These regression analysis have been conducted for the three applications with $a = \frac{1}{3}, \frac{1}{2}, 1, 2$, and goodness of fit results (R^2 coefficient) are summarized in Table 2.1. In GA results a linear convergence rate is found dominant for experiments LAD and *gk*, while $a = 1/2$ rate is fittest for experiment AR; concerning DE the best rate is linear for all experiments. We adopted these convergence rates in tradeoff analysis of next section. As an example, Figure 2.2 shows the fitted convergence rate of parameter β_2 in LAD experiment using GA.

Results of estimates of $tr(W_S)$ and $tr(W_{EA})$ are summarized in Table 2.2: they show that results on LAD and *gk* are similar in two algorithms, so we also expect similar results in tradeoff analysis. For computing these estimates we used simulated data of length $n = 200$ in all experiments.

The tradeoff will be discussed for the three applications by evaluating optimal $\tilde{n}$ on a common grid of values for linear cost functions S and T, assuming a fixed total effort $C = 10^5$. Comments on optimal V can be derived by complement. We shall make some remarks also for the case when computational cost T is estimated by 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.

Least Absolute Deviation **Estimation**

LAD regression is an alternative to Ordinary Least Squares regression, proven to be more robust to outliers (Bloomfield & Steiger, 1983, p.52). In this framework the estimator, which is asymptotically efficient (Bloomfield & Steiger, 1983, 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

Table 2.1: R^2 coefficient values related to four different regression analysis conducted on each parameters of each experiment, in order to estimate convergence rate of Σ_{EA}

GA

Exp	Param	$a = 1/3$	$a = 1/2$	$a = 1$	$a = 2$
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
AR	ϕ_1	0.7806	0.9338	0.8864	0.4655
	ϕ_2	0.9101	0.9896	0.7083	0.2622
	ϕ_3	0.9164	0.9835	0.6645	0.2200
	ϕ_4	0.8998	0.9767	0.6762	0.2228
	ϕ_5	0.8869	0.9726	0.6878	0.2306
	ϕ_6	0.8801	0.9698	0.6921	0.2325
	ϕ_7	0.8569	0.9597	0.7104	0.2453
	ϕ_8	0.8576	0.9635	0.7311	0.2641

DE

Experiment	Parameter	$a = 1/3$	$a = 1/2$	$a = 1$	$a = 2$
LAD	β_0	0.1069	0.3364	0.9282	0.7775
	β_1	0.1084	0.3322	0.9375	0.8133
	β_2	0.1067	0.3363	0.9356	0.7987
gk	A	0.1665	0.4472	0.8715	0.5361
	B	0.1543	0.4180	0.8014	0.4573
	g	0.1973	0.4837	0.7468	0.3631
	k	0.1137	0.3516	0.9541	0.8174
AR	ϕ_1	0.2292	0.4018	0.8847	0.9176
	ϕ_2	0.4782	0.6653	0.9336	0.6486
	ϕ_3	0.6619	0.8131	0.9003	0.5611
	ϕ_4	0.5079	0.6948	0.9330	0.6198
	ϕ_5	0.4562	0.6496	0.9330	0.6387
	ϕ_6	0.5049	0.6928	0.9339	0.6174
	ϕ_7	0.4504	0.6434	0.9285	0.6317
	ϕ_8	0.5515	0.7276	0.9141	0.5820

Table 2.2: Sampling and EA variability components estimates

Experiment	$tr(W_S)$	$tr(W_{GA})$	Conv rate	$tr(W_{DE})$	Conv rate
LAD	5.38	23.18	$1/V$	20.28	$1/V$
AR	12.26	17.74	$1/\sqrt{V}$	1315.46	$1/V$
gk	103.39	3897.25	$1/V$	3972.70	$1/V$

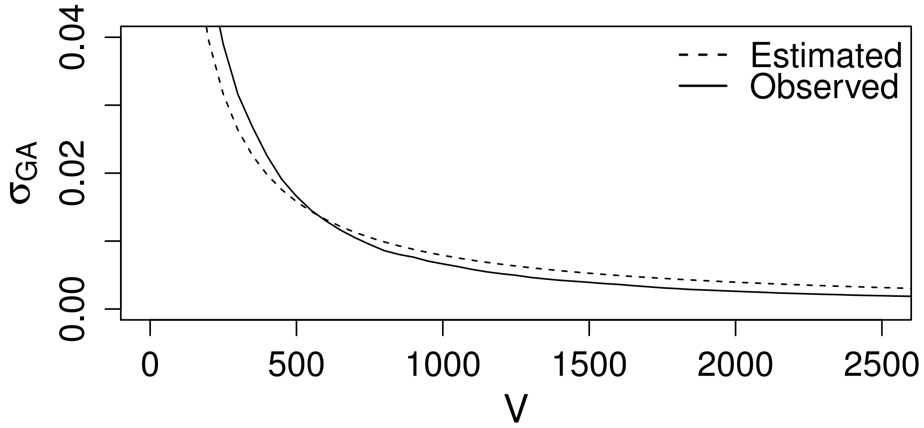


Figure 2.2: Observed (thick line) and estimated (dashed line) GA variability for parameter β_1 of LAD experiment ($w_{GA} = 7.9$, $R^2 = 0.97$)

optimal solution. Zhou & Wang (2005) have already employed a real valued GA to estimate the parameters of a LAD regression with censored data. Here we consider a standard linear regression model:

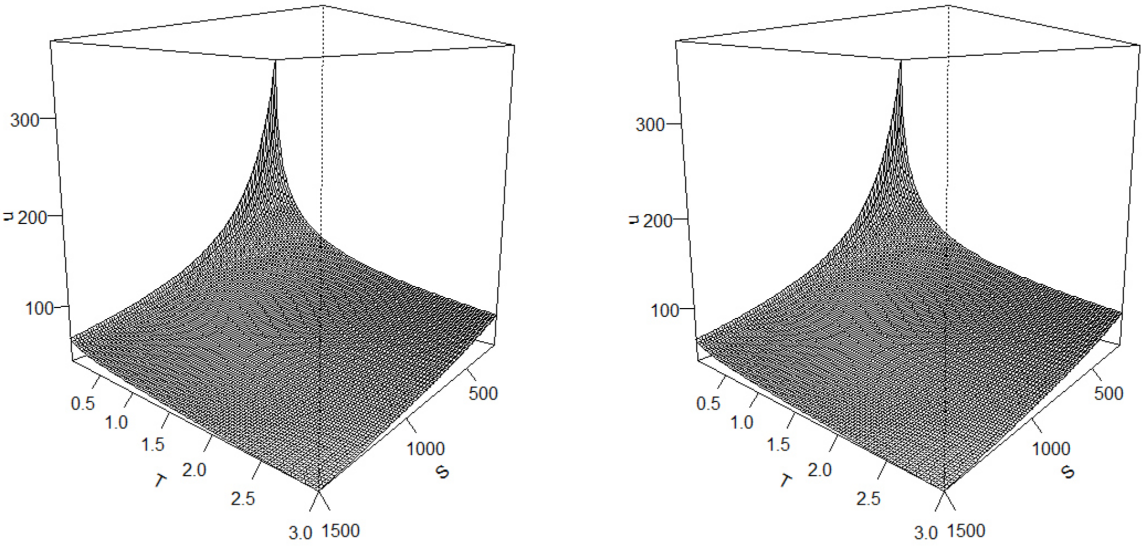
$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \epsilon_i, \qquad i = 1, ..., n,$$

where $(\underline{y}, \underline{x})$ is the observed dataset. The errors are not Gaussian, but distributed according to a heavy-tailed Student's t distribution with 5 degrees of freedom.

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

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

True parameters vector will be $\underline{\beta} = (0.5, 0.5, -0.5)$, coding interval boundaries will be $[-2, 2]$ for all parameters and each chromosome length in GA shall be $M = 24$.



(a) GA

(b) DE

Figure 2.3: Behaviour of optimal n for experiment *LAD*

Figure 2.3 shows the behaviour of optimal n (on z axis) with respect to a grid of values for cost functions S and T. Results are identical in two algorithms, as they show that $\tilde{n}$ obviously increases to large values as costs S and T decrease, and rapidly decreases as they increase.

Autoregressive Models Building

GAs have been widely applied for time series models identification (see Section 1.6). 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:

$$Y_t = \phi_1 Y_{t-1} + ... + \phi_p Y_{t-p} + \epsilon_t, \tag{2.12}$$

where Y_t is a zero mean random process, ϵ_t a Gaussian white noise and $\underline{\phi} = (\phi_1, ..., \phi_p)$ the parameters vector, for which some components may be constrained to zero.

Model (2.12) is usually identified by minimizing penalized likelihood criteria like AIC or BIC, to be minimized. In this work we shall consider BIC, because of its property of consistency (Hannan, 1980):

$$BIC(\underline{\phi}; \underline{y}) = n \, log \hat{\sigma}^2(p) + k \, log n, \tag{2.13}$$

where $\underline{y}$ is the observed time series, $\hat{\sigma}^2(p) = \sum_{i=1}^{n}(y_t - \phi_1 y_{t-1} - ... - \phi_p y_{t-p})^2/n$ and $k \leq p$ is the number of free parameters in the model. Sampling variability will be estimated on the basis of asymptotic efficiency property of AR models maximum likelihood estimator (Brockwell & Davis, 1991, p.386).

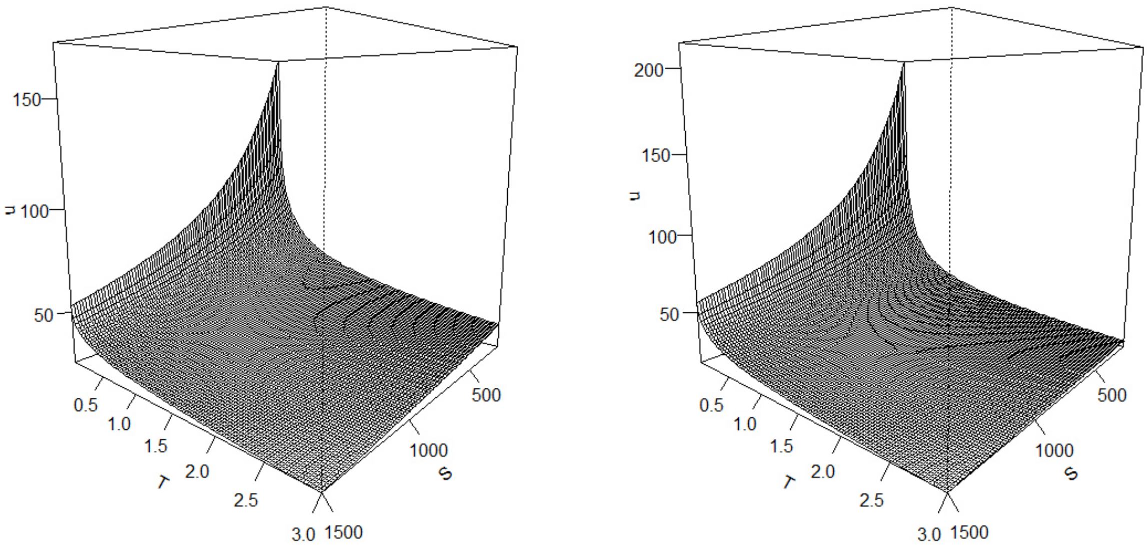
As true model we will consider an $AR(1)$ with $\phi_1 = 0.8$ and a maximum possible order $p = 8$. In GA the chromosome length shall be $M = 64$. In order to facilitate the identification of subset models we shall force the starting population of both GA and DE to include a chromosome that corresponds to a white noise (all parameters are zero), and also 8 chromosomes for which one of the parameters is zero, so that all $\phi_i = 0$ $(i = 1, ..., 8)$ 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 total lack of knowledge.

Fitness function shall be:

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

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

Figure 2.4 shows the analogous plot to Figure 2.3. Even in this case the two perspective plots of optimal n are very similar: some differences arise for small values of sampling cost S. Figure 2.5 highlights magnitude of these differences.



(a) GA

(b) DE

Figure 2.4: Behaviour of optimal n for experiment AR

Generally speaking, this experiment has lower values of $\tilde{n}$ with respect to LAD, possibly because fitness account also for model identification (e.g. estimating a ϕ_i value slightly different from zero implies may implies a slight decrease of the residual sum of squares, but a term k one unit larger in the penalization part of BIC). This may have implied slower GA convergence rate and large DE variability.

g-and-k Distribution Estimation

The *g-and-k* distribution was introduced in Haynes et al. (1997) 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 (Haynes et al., 2008) and non-life insurance modeling (Peters et al., 2016). For a univariate random sample $\underline{x} = (x_1, ..., x_n)$ the quantile function is:

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

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 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.

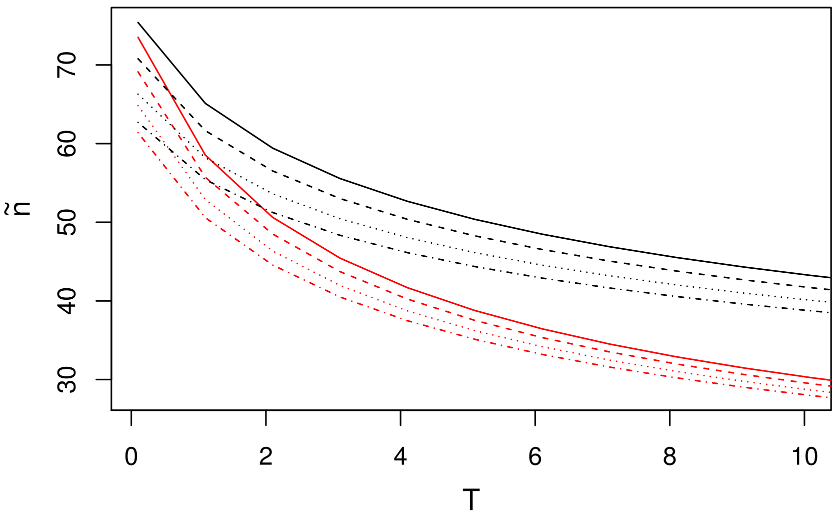


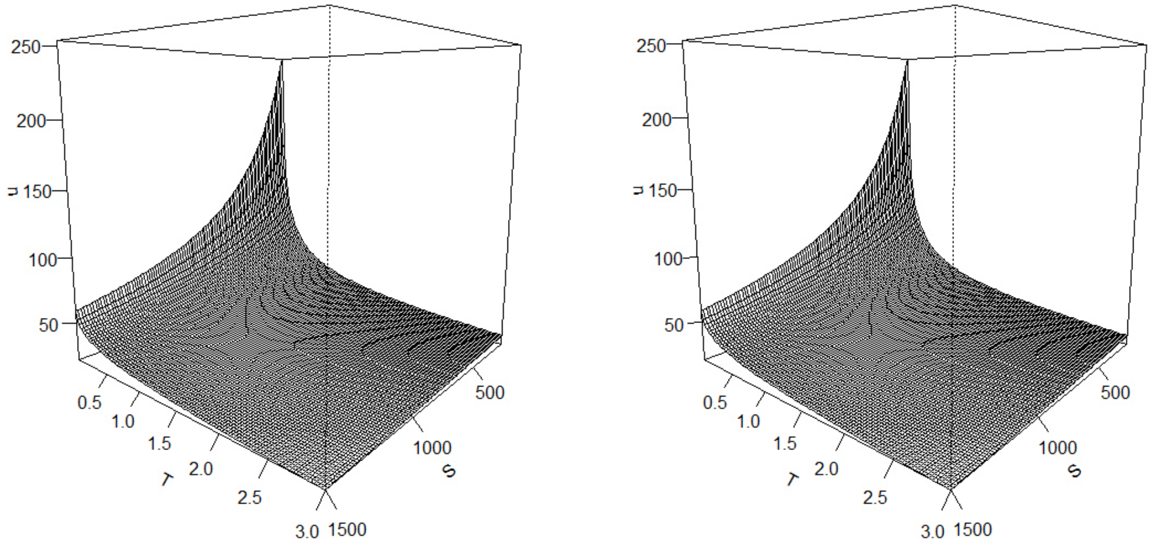
Figure 2.5: Optimal sample size with respect of T for both GA (in black) and DE (in red), with sampling cost S fixed at 50 (solid lines), 100 (dashed lines), 500 (dotted lines), 1000 (dashed and dotted lines)

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

$$L(\underline{\theta}|\underline{x}) = \left(\prod_{i=1}^{n} Q'_X(Q_X^{-1}(x_i|\underline{\theta})|\underline{\theta}) \right)^{-1}, \tag{2.14}$$

where $\underline{x}$ is the observed sample, $\underline{\theta} = (A, B, g, k)$ and $Q'_X(u|\underline{\theta}) = \partial Q_X / \partial u$. The main difficulty in computing (2.14) is the lack of a closed form expression for $Q_X^{-1}(x_i|\underline{\theta})$, 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 (Haynes & Mengersen, 2005) or indirect inference methods like Approximate Bayesian Computation (Allingham et al., 2009; Grazian & Liseo, 2015). We shall follow the pure likelihood approach proposed in Rayner & MacGillivray (2002). In this situation a numerical procedure has to be selected to maximize (2.14). 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.



(a) GA

(b) DE

Figure 2.6: Behaviour of optimal n for experiment gk

In our approach we shall consider the fitness:

$$f(\underline{\psi}) = \exp\{\, log L(\underline{\theta}|\underline{x})/n\, \}.$$

We will simulate data using the typical parameters generator vector $\underline{\theta} = (A, B, g, k) = (3, 1, 2, 0.5)$, with $c = 0.8$, which leads to an 'interesting far-from-normal distribution' (Allingham et al., 2009).

Each chromosome in GA implementation will have length $M = 28$, and coding interval boundaries shall be: $A \in [-10, 10]$, $B \in [0, 10]$, $g \in [-10, 10]$ and $k \in [-0.5, 10]$. If a decoded chromosome provides unacceptable values $B = 0$ or $k = -0.5$ then it is rejected and regenerated.

Concerning sampling variability, Rayner & MacGillivray (2002) 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.

Perspective plot for this experiment (Figure 2.6) shows a similar behaviour of optimal n to AR, even if general lower values of $\tilde{n}$ are observed, because also in this case experiment is very complex ($tr(W_{EA})/tr(W_S)$ ratio is large).

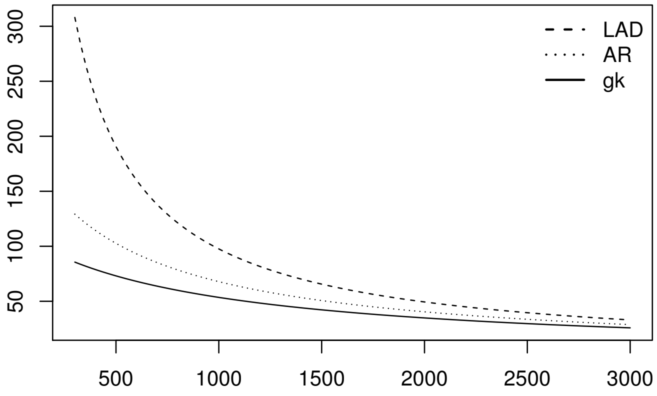
Lastly we shall make some comments on the behaviour of $\tilde{n}$ when sampling cost S varies and fitness evaluation cost T is estimated in each experiment by elapsed execution time (in seconds) of our computer for a single fitness evaluation, taking

gk as corner point (being the most expensive one). Results are: $T_{LAD}/T_{gk} = 0.007$ and $T_{AR}/T_{gk} = 0.101$. Figure 2.7 shows the behaviour of $\tilde{n}$ in this more realistic scenario, for which each computational cost ratio has been multiplied by a constant to highlight the behaviour of experiments. As GA and DE behaviour is identical, in both graphs the three curves are ranked with respect of computational cost and experiment complexity, that is related on both EA convergence rate and the magnitude of variability ratio $tr(W_{EA})/tr(W_S)$. 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.

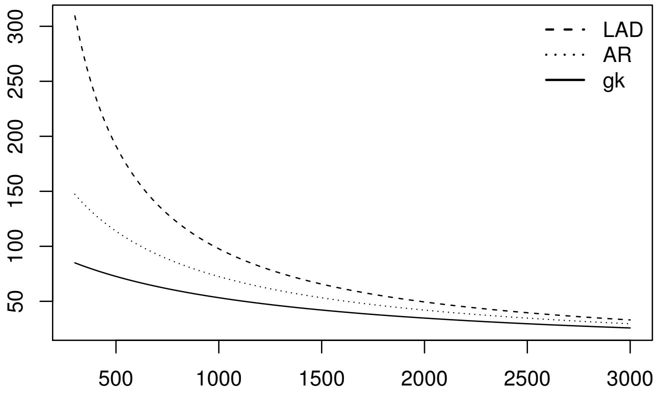
2.4 Concluding remarks

This chapter proposed a method for evaluating variability of EAs when employed in parametric estimation problems, valid for consistent estimators and convergent EAs. A statistical and computational tradeoff analysis involving the above specified variability analysis has been performed for three selected applications, in which GAs and DE have been employed. Results showed how the behaviour of optimal sample size changes with complexity of experiment and among two selected EAs. A comparative analysis of the three experiments, in which computational cost is estimated, also suggested that large sampling cost could influence optimal values more than complexity of the model, represented by statistical and computational variability. This is an interesting consideration, especially for real applications, where often large costs can decisively restrict the analysis.

The present method could be improved by considering other scalar summaries of statistical and computational variability. For example the determinant of Σ_S and Σ_{EA} could be more appropriate than trace. An other direction for further research is to generalize this framework to other statistical problems in which EAs 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 EAs among standard statistical methods. Lastly, the discussion on statistical and computational tradeoff can be naturally extended to other stochastic algorithms, like Particle Swarm Optimization, which could imply different conclusions on variability analysis and tradeoff.



(a) GA



(b) DE

Figure 2.7: Optimal sample size with fixed estimated computational cost

Chapter 3

Evolutionary Computation and Multiple Chains MCMC Sampling: an Overview

3.1 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) methods have received a huge attention in Bayesian statistics literature of the last decades because of the increasing availability of computing power. In fact, the occurrence of obtaining posterior distributions summaries is crucial in Bayesian inference, and most of times it implies to numerically compute multiple integrals and sampling from multivariate distributions not having an analytical form. In this framework MCMC represents the most established method, and research on this topic has led to the development of many variants (Robert & Casella, 2004).

The basic MCMC method can be summarized as follows: let us suppose we are interested in sampling from a target distribution $\pi(\underline{x}) \in \mathbb{R}^d$, analytically intractable. The MCMC consists in building a sequence of vectors $\underline{x}^t \in \mathbb{R}^d$, $t = 1, \dots$, that is a realization of a Markov Chain having $\pi(\cdot)$ as equilibrium distribution. Usually a certain number of iterations during first phases of algorithm is removed, in order to get rid of the dependence on starting points (*burn-in*). The method does not allow to sample directly from $\pi(\cdot)$, but it takes advantage of a *proposal* distribution $q(\cdot) \in \mathbb{R}^d$, from which it is easier to sample. Two main MCMC algorithms have been developed in literature: the *Gibbs sampling* and the *Metropolis Hastings* (MH). In the Gibbs sampling algorithm the proposal coincides with the univariate distribution of each component of π, given the other components, and it is called *full conditional*.

So at each iteration t a new value $\underline{x}^{t+1}$ is generated using the d full conditionals. The MH, on the other hand, does not have a standard specification of q. At each iteration t a *MH step* is performed, for which a pseudo-realization y is generated from $q(\cdot|\underline{x}^t)$ and it is accepted as a new chain state with probability:

$$\alpha(\underline{y}, \underline{x}^t) = min\{1, \frac{\pi(\underline{y})\, q(\underline{x}^t|\underline{y})}{\pi(\underline{x}^t)\, q(\underline{y}|\underline{x}^t)}\}.$$

If the proposed value $\underline{y}$ is not accepted then $\underline{x}^{t+1} = \underline{x}^t$. Possible variants of the algorithm are related to particular forms of the proposal: *symmetrical* proposals, for which $q(\underline{y}|\underline{x}) = q(\underline{x}|\underline{y})$, lead to the *Metropolis* algorithm, whose acceptance probability $\alpha(\underline{y}, \underline{x}^t)$ is equal to $min\{1, \frac{\pi(\underline{y})}{\pi(\underline{x}^t)}\}$; the *independence sampler* is obtained by choosing $q(\underline{y}|\underline{x}) = q(\underline{y})$; if $q(\underline{y}|\underline{x}) = q(\underline{y} - \underline{x})$ then MH turns into the *random walk Metropolis algorithm.*

The procedure can guarantee a sequence of pseudo-random values from $\pi(\cdot)$, namely the Markov Chain has $\pi(\cdot)$ as equilibrium distribution, if the resulting mechanism is aperiodic, irreducible and reversible. A sufficient, but not necessary condition, that ensures reversibility is that the mechanism satisfies the *detailed balance condition*: $\pi(\underline{y}|\underline{x}^t) \cdot \pi(\underline{x}^t) = \pi(\underline{x}^t|\underline{y}) \cdot \pi(\underline{y})$.

3.2 Multiple chains MCMC

When target distribution $\pi(\cdot)$ is multimodal or the components are strongly correlated then the values generated by a MCMC algorithm may tend to approach each other or getting trapped in local optima. In that case the chain is said not to be *mixing* well, and the resulting sampling would not adequately represent the support of target distribution. A possible approach proposes to let several Markov Chains run in parallel, mimicking the multi-start strategies of optimization algorithms to escape local optima. Each chain $\underline{x}_i$ in the resulting population $\mathbf{X} = \{\underline{x}_1, ..., \underline{x}_M\}$, $\mathbf{X} \in \mathbb{R}^{d \times M}$, is equipped with a possibly different equilibrium distribution $\pi_i(\underline{x}_i)$, and also a population distribution $\pi^*(\mathbf{X})$ may be specified. At each iteration the new chain states can be generated and accepted according to either the individual π or the population π^* distribution (or both). Detailed reviews of such methods can be found in Jasra et al. (2007) and Liang et al. (2011)

This way of proceeding has inspired many researchers to study analogies with EC. In fact, if the chains in the population are allowed to interact with each other then it could be reasonable to take advantage of EC peculiarities, whose strength is properly the interaction and combination between solutions. Although EC is mostly used for optimization, it can be easily introduced, at least in a basic form, in the framework of MCMC sampling.

We shall now describe few approaches found essential in subsequent research related with EAs, before surveying specific contributions on the framework.

Parallel Tempering

Parallel Tempering (PT), pioneered by Geyer (1991) and Hukushima & Nemoto (1996), could be considered a generalization to multiple chains of popular *Simulated Tempering* algorithm (ST; Marinari & Parisi, 1992, Geyer & Thompson, 1995). In this latter proposal, inspired by Simulated Annealing (Kirkpatrick et al., 1983), the target distribution law is given by $\pi(\underline{x}) \propto \exp\{-H(\underline{x})\}$, but sampling refers to a different distribution $\pi(\underline{x}) \propto \exp\{-H(\underline{x})/T\}$, known as *Boltzmann distribution*, where T is an auxiliary variable called *temperature*, taking values from a finite set named *ladder*. This so-called *cooling* strategy, for which T is updated at each iteration along with $\underline{x}$, may allow to facilitate the exploration of parameter space and speed up convergence of MCMC in multimodal problems. PT generalizes this approach by considering a population of M Markov Chains, each with its own Boltzmann invariant distribution $\pi_i(\underline{x}) \propto \exp\{-H(\underline{x})/T_i\}$, $i = 1, ..., M$, where ladder $\underline{T}$ is built as $T_1 > T_2 > ... > T_M = 1$, so that $\pi_M(\underline{x})$ is the distribution of interest $\pi(\underline{x}) \propto \exp\{-H(\underline{x})\}$. At the generic iteration of this algorithm a MH step is performed for each chain; then a *swap* step between two chains state, without involving temperatures, is proposed and accepted by a further MH step. This kind of mechanism may allow to speed up mixing of chains.

Snooker Algorithm

Snooker algorithm has been proposed in Gilks et al. (1994), along with a more general method named Adaptive Direction Sampling, in order to improve convergence of Gibbs sampling in many situations, for example multimodal problems. In the generic updating procedure of Snooker two chains in the population are randomly selected without replacement: first chain $\underline{x}_c$ (*current point*) is designated to be updated, while second chain $\underline{x}_a$, called *anchor point*, determines direction of updating. Difference $(\underline{x}_a - \underline{x}_c)$ specifies sampling direction so that the new chain $\underline{y}$ is built as follows:

$$\underline{y} = \underline{x}_c + r(\underline{x}_a - \underline{x}_c),$$

where r is sampled from density: $f(r) \propto |1-r|^{d-1}\,\pi(\underline{x}_c + r(\underline{x}_a - \underline{x}_c))$, chosen in order to guarantee convergence of each chain to the target distribution π (proof can be found in Roberts & Gilks, 1994).

3.3 GA based approaches

Holmes & Mallick

The first proposal to explicitly introduce EC for improving MCMC is due to Holmes & Mallick (1998). In their approach, called Parallel Adaptive Metropolis Sampling, they suggest to take advantage of GAs features for MCMC sampling in presence of high dimensionality and strong correlation between variables.

Here $\pi_i(\cdot) = \pi(\cdot),\ i = 1, ..., M$, and only a single chain in the population is modified at each iteration (as happens in Steady State GAs; Syswerda, 1989). This chain $\underline{x}_a$ is selected uniformly at random, and it is subdued to mutation with probability pM, or to crossover with complementary probability. The mutation operator is analogous as Evolution Strategies method, so that the new solution is built as: $\underline{x}_{a*} = \underline{x}_a + \underline{q}$, with $\underline{q} \sim N_d(0, \Sigma)$ and Σ is chosen to provide a moderate acceptance probability. This move, as far as it is symmetrical, is then evaluated by a Metropolis step, so it is accepted with probability: $\alpha(\underline{x}_a, \underline{x}_{a*}) = min\{1, \frac{\pi(\underline{x}_{a*})}{\pi(\underline{x}_a)}\}$. The s4elected crossover mechanism is in two step: at first, a standard uniform crossover is performed on two chains $\underline{x}_i$ and $\underline{x}_j$, randomly selected in such a way that $i \neq j \neq a$, obtaining a new solution $\underline{x}_u$; then $\underline{x}_u$ can be crossed with $\underline{x}_a$ by either moving along direction $(\underline{x}_u - \underline{x}_a)$ or by performing the reflection of $\underline{x}_a$ on $\underline{x}_u$ (with probability pC). The resulting solution is accepted by a Metropolis step. The above scheme turns out to be irreducible, aperiodic and reversible. Several features of this algorithm have been set considering computational complexity of method, for example the choice of symmetrical proposals, the exclusive contrast between mutation and crossover operators, the update involving a single chain at each generation.

The applications considered are a Bayesian estimation of neural networks (based on real data), characterized by multimodality, and a problem of inferring the number and location of knot points in Bayesian spline models, with strongly correlated variables: results are compared with a standard MH algorithm. The results showed that the proposed algorithm can traverse the state space much more widely than MH, and it moves around high posterior regions with good acceptance rates and reasonably sized updated proposals.

Liang & Wong

Evolutionary Monte Carlo (EMC; Liang & Wong, 2000, 2001a, 2001b) is one of the most important algorithm in the framework, and it is generally considered the original proposal that conjugates EC and MCMC. Here we shall review the real coded algorithm proposed in Liang & Wong (2001a); the other papers include analogous binary or integer versions of the procedure.

The authors proposed a method that conjugates features of GAs and Simulated Annealing, resulting in an algorithm that generalizes PT. In fact they adopt Boltzmann distribution $\pi(\underline{x}) \propto \exp\{-H(\underline{x})/\tau\}$ as distribution of interest, and refer to function $H(\cdot)$ as fitness. Each chain has its own equilibrium distribution $\pi_i(\underline{x}) \propto \exp\{-H(\underline{x})/T_i\}$, $i = 1, ..., M$, with ladder $\underline{T} = (T_1, ..., T_M)$, for which $T_1 > T_2 > ... > T_M = \tau$. Operators of *mutation*, *crossover* (having more options) and *exchange* are sequentially performed at each generation, and each intermediate population including new proposed values is accepted via MH step involving the population distribution π^*. Mutation operator, employed with probability pM, is structured as in Holmes & Mallick (1998), except for the MH step involving π^*. In crossover operations two chains $\underline{x_i}$ and $\underline{x_j}$ are selected uniformly at random or by roulette wheel. Two choices of crossover operator are then considered: standard GA crossovers, like k-points and uniform, or a novel *snooker crossover* (similar to the one introduced by Holmes & Mallick, 1998 and inspired by Snooker algorithm). In the latter case new chromosome $\underline{y}_i$ is obtained by: $\underline{y}_i = \underline{x}_j + r\underline{e}$, where $\underline{e} = (\underline{x}_j - \underline{x}_i/)||\underline{x}_j - \underline{x}_i||$ and r is sampled from density: $f(r) \propto |r|^{d-1}\pi(\underline{x}_j + r\underline{e})$. This snooker crossover move has been proven to leave distribution π^* invariant. Afterwards the *exchange* operation takes part, in which M individuals are selected to be swapped with neighbor chains (in term of temperature), as in PT. Setting $pM = 1$ leads to PT algorithm, while fixing both $pM = 1$ and $M = 1$ EMC reduces to a single-chain Metropolis Hasting algorithm.

Two kind of applications have been considered: Bayesian estimation of finite mixture of normal distributions (various examples, with both simulated and real data), that exhibit multimodality; Bayesian estimation of neural networks (with both simulated and real data, including Box-Jenkins gas furnace data), as done in Holmes & Mallick (1998), whose posterior distribution is both nonlinear and multimodal. Results showed that the EMC, compared to methods like PT, conjugate gradient Monte Carlo and Box-Jenkins approach, is a very good tool for sampling from complex distributions: simulation at high temperatures facilitates exploration of the search space and exchange operator can be viewed as a selection mechanism for localizing possible modal zones, so it may support exploitation.

EMC has been found successful in literature and has stimulated some research. Goswami & Liu (2007) provided an extension of Liang and Wong's algorithm called Target Oriented Evolutionary Monte Carlo (TOEMC). They studied several new exchange moves, related to fitness $H(\cdot)$ and ladder $\underline{T}$, in order to make the acceptance probability more stable. Furthermore they analyzed methods to optimally construct the ladder $\underline{T}$, basing on preliminary EMC runs, in order to localize promising modal regions. One of the authors also developed an R package providing EMC procedure (Goswami, 2011). An adaptive version of EMC has been introduced in Ren et al. (2008); Goswami et al. (2007) proposed some new operators for EMC in order to perform clustering; Gupta (2014) employed EMC for purpose of biclustering in Bayesian framework.

Battaglia

Another approach, proposed in parallel and independently from Liang and Wong's, is due to Battaglia (2001). The aim of this work was to develop a multiple chains MCMC sampling procedure in a complete GA framework, using the early proposal by Holmes & Mallick (1998) as a starting point. Also here $\pi_i(\cdot) = \pi(\cdot)$, but differences arise when genetic operators are concerned.

In fact a selection mechanism is introduced, subdued to a notion of fitness as a measure of adaptation of chains population at time t, considered as a candidate sample, to the target distribution $\pi(\cdot)$. In order to accomplish this, a finite partition $\{P_j, j = 1, ..., J\}$ of $\pi(\cdot)$ is built so that multivariate distribution of interest is summarized in the form of discrete univariate distributions, assigning probability $\pi_j = \int_{P_j} \pi(\underline{x}) d\underline{x}$ to values $j = 1, ..., J$. As a result of this, if $\underline{s} = (s_1, ..., s_J)$ represents frequencies of the discretized values in the population, a dissimilarity measure between $\underline{s}$ and the theoretical distribution $M\underline{\pi} = (M\pi_1, ..., M\pi_J)$ can be computed for evaluating the global adaption of current sample to the target distribution. In order to characterize the specific contribution of each chain $\underline{x}_i$ to global adaptation, which is analogous to define a fitness function in GAs, a score related to the induction of each partition is assigned as follows: it is equal to 1 if $\underline{x}_i \in P_j$, and to zero otherwise. Sampling M chains leads to the following equality that characterizes individual fitness: $f(\underline{x}_i) = M\pi_j/s_j$, $\underline{x}_i \in P_j$, meaning that the goodness of each element of P_j is uniformly shared between chromosomes belonging to that partition. This means that the probability $P(\underline{x}_i)$ of selecting chain $\underline{x}_i$ is equal to $\frac{\pi_j}{s_j}[\sum_{k \in J} \pi_k]^{-1}$, $\underline{x}_i \in P_j$, where $J = \{j : s_j > 0\}$.

The selected chromosomes undergo mutation and crossover, in order to guarantee the possibility of covering different areas of the support of $\pi(\cdot)$. Mutation operator could be either a generic MH or a Gibbs step, and it could be executed on whole chromosomes or on some genes. Author also observed that this kind of mutation strategy allows many new solutions to be generated, compared to the usual role of mutation in GAs. For this reason also a variant of algorithm is proposed, for which selection operator as it has been described is absent and replaced by MCMC procedure itself (each chain runs independently), so that only the reproduction is performed. Single point crossover between two parents is introduced, to be accepted with a Metropolis step (since the move is reversible) involving individual distribution π. So this approach tries to exploit promising modal zones of $\pi(\cdot)$ by building a partition and selecting chromosomes that mostly induce each partition; exploration role is assigned to mutation and crossover. One drawback of this approach is that π does not always allow for a natural partitioning, so it often needs to be estimated (author proposed an exponential smoothing).

Applications consisted in comparisons of different algorithm configurations, for example presence or absence of selection operators, in literature problems where π has not a closed form or has highly correlated components. Results showed a positive effect of crossover; also mutation was effective, but only when partition P_j was provided exactly, and not estimated.

Hu, Tsui

Hu & Tsui (2010) proposed to employ a Distributed GA in the multiple chains MCMC with multimodal or high dimensional target distributions, because it is known to be less likely to converge prematurely then standard GA. The resulting algorithm has been called Distributed Evolutionary Monte Carlo (DGMC).

Here the population $\mathbf{X}$ of chains is divided in J subpopulations $\{\mathbf{x}_1, ..., \mathbf{x}_J\}$ and $\pi_i(\cdot) = \pi(\cdot)$ (but a PT style cooling scheme could be also introduced in each subpopulation). At the beginning of each iteration the *migration* operator is employed with probability p_m: k subpopulations $i_1, ..., i_k$ are uniformly selected, and the so called *migration cycle* $O_k = (i_1 \to ... \to i_k \to i_1)$ is built. Then, in each subpopulation $i_j \in O_k$, an *emigrant* $\underline{x}_e^{i_j}$ is randomly chosen, so that $\underline{y}_e^{i_{j+1}} = \underline{x}_e^{i_j} + \underline{\delta}$ is the proposed value for each subpopulation, where $\underline{\delta}$ is called emigration noise. So the new subpopulation $\mathbf{y}_{i_{j+1}}$ is built as: $(\mathbf{x}_{i_{j+1}} \setminus \{\underline{x}_e^{i_{j+1}}\}) \cup \{\underline{y}_e^{i_{j+1}}\}$, and new population $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_J\}$ is accepted via MH step involving population distribution π^*, factorized with respect of subpopulations. After this step, in each subpopulation an exclusive mutation/crossover operator is proposed with probability q_m: standard floating point GAs mutation (as in previous contributions), or a snooker style crossover. In the first case new solution is accepted via individual MH step; concerning crossover authors analyzed snooker crossover proposals by Liang & Wong (2001) and ter Braak (2006) (to be described in next section), and employed a modification of this latter, in which the proposal is always accepted. A rate of q_c individuals is crossed to get the final subpopulation.

Applications included sampling from bimodal and multimodal mixture of normal distribuions (same as in ter Braak, 2006) and Bayesian estimation of a generalized logistic function (real data), performing also comparisons with other algorithms (EMC, PT and simple MH). Results showed faster and better mixing of DGMC with respect to other analyzed methods, because it could move more efficiently between far-separated modes.

Holloman, Lee, Higdon

An approach, proposed by Holloman et al. (2006), aimed to extend Simulated Sintering procedure (Liu & Sabatti, 1999) to multiple chains. This latter method, that generalizes ST and Gibbs Sampling, considers data whose continuous domain is discretized and modeled on multiple scales (or resolutions). The procedure incorporates elements from other Monte Carlo and MCMC techniques, like multigrid Monte Carlo (Goodman & Sokal, 1989), reversible jump MCMC (Green, 1995) and dynamic weighting (Wong & Liang, 1997).

Authors motivated their multiple chains implementation by observing the effect of data modeling at fine scale (high information but also many parameters and slow fitting procedures) versus coarser scales (less information but also more parsimony and less computational cost) for continuous phenomena measured on a discretized grid. Moreover they observed that coarser scales could facilitate the exploration of multimodal functions (for example a likelihood). Therefore they proposed a method that simultaneously evaluates chains at different resolutions, taking advantage of both fine and coarser scales benefits, analogously to what happens in multiple chains algorithms with cooling schemes like PT or EMC.

A model involving I scales is introduced, so that data at scale i are denoted by $\underline{z}^{(i)}$, with parameters vector of interest $\underline{\psi}^{(i)}$, $i = 1, ..., I$. This latter quantity is written in terms of two variables $\underline{\theta}^{(i)}$ and $\underline{\lambda}^{(i)}$ related, respectively, to the sharing information process between scales and to the remaining parameters, linked to $\underline{\psi}^{(i)}$ by a generally deterministic function $g(\cdot)$. As far as a Bayesian framework is concerned, model posterior distribution of interest $\pi(\cdot|Z)$ of the model, where Z denotes all available data, is built as the product of posteriors of each scale, defined as: $\pi^{(i)}(\underline{\psi}^{(i)}|\underline{z}^{(i)}) = \pi^{(i)}(\underline{\theta}^{(i)}, \underline{\lambda}^{(i)}|\underline{z}^{(i)}) \propto L(\underline{\psi}^{(i)}|\underline{z}^{(i)})\pi^{(i)}(\underline{\theta}^{(i)}, \underline{\lambda}^{(i)})$, where $L(\cdot|\underline{z}^{(i)})$ and

$\pi^{(i)}(\cdot)$ are, respectively, likelihood and prior distribution at scale i. Both of these quantities are problem dependent.

Authors described at first a novel Multiresolution GA, with purpose of likelihood maximization; afterwards, the Multiresolution GA-Style MCMC, which allows to sample from full posterior $\pi(\cdot|Z)$, is introduced. This algorithm considers M parallel chains, with $M \geq I$, encoding parameters $\underline{\theta}^{(i)}$, $\underline{\lambda}^{(i)}$ and resolution i. For a fixed number of iterations all chains are independently mutated by generic MH or Gibbs steps; after that, a swap is attempted between two selected individuals (either uniformly at random or proportionally with respect to posterior densities, assuring scales stratification if needed). These two paired chains undergo a standard uniform crossover step, involving elements of vector $\underline{\theta}^{(i)}$ only; a proposal distribution ζ, possibly different for each scale, is needed to generate also new values of $\underline{\lambda}^{(i)}$ given data and proposed $\underline{\theta}^{(i)}$. In some cases it could be useful to swap all elements of $\underline{\theta}^{(i)}$ in crossover (*full* swap). To ensure detailed balance of the swap a MH step involving posterior densities $\pi^{(i)}(\cdot|\underline{z}_i)$ of two selected scales, distributions $\zeta^{(i)}$ and selection probabilities of two chains are performed.

Application considered refers to single photon emission computed tomography (SPECT), for which authors focused on reconstructing two-dimensional images given data from various cameras, and an inverse problem in groundwater hydrology, in which inference is done on flow data. They compared the proposed Multiresolution GA-Style MCMC algorithm, considering both crossover and full swaps, with standard fine scale MCMC. Results showed superiority of proposed method over standard MCMC and also a positive effect of using full swaps only.

3.4 DE based approaches

Strens

One of the most important proposal, in term of citations and applications, is directly based on DE algorithm.

First studies are due to Strens (Strens et al., 2002; Strens, 2003), who introduced Direct Search Optimization methods, which do not require information about objective function gradient, in the framework of sampling from complex distributions. Procedure named Differential Evolution Sampler (DES), introduced in Strens et al. (2002) for continuous distributions, considers $\pi_i(\cdot) = \pi(\cdot)$ as an improvement with respect to algorithms with cooling scheme like PT or EMC, because in that case only one chain is actually used for providing samples. the differential mutation operator is employed as a kind of geometrical proposal for each chain $\underline{x}_i$, which produces a new solution $\underline{y}_i$ as follows: $\underline{y}_i = \underline{x}_i + \gamma(\underline{x}_{R1} - \underline{x}_{R2})$, $i \neq R1 \neq R2$, where scaling factor γ is realization of a random variable (as happens in *dither* and *jitter* strategies of DE). This differential mutation move, similar to snooker crossover, is proposed for all chains, and it guides the exploration of parameters space toward modal zones. It differs from standard differential mutation operator in DE algorithm, which includes a further randomly chosen vector $\underline{x}_{R0}$ in place of $\underline{x}_i$, in order to ensure reversibility of proposal. Vector difference $(\underline{x}_{R1} - \underline{x}_{R2})$ is optionally subject to a crossover operation with $\underline{0}$ vector. The move from $\underline{x}_i$ to $\underline{y}_i$ is accepted via Metropolis step. Sampling performance is assessed at each generation by use of Kullback-Leibler divergence between true and estimated density. This procedure is expected to generate useful proposals because chains population is likely to be adapted to the shape of π. Authors also suggested that including subpopulations in the algorithm, as proposed afterwards by Hu & Tsui (2010) in GA framework, could be beneficial, because local geometry of π could be better exploited.

DES has been compared with algorithms like Metropolis, PT and EMC in a mixture of normal distributions sampling with unequal variances, using Kullback-Leibler divergence to measure distance between true density $\pi(\cdot)$ and empirical density estimated by MCMC: results showed good performances of DES. A generally analogous procedure has been studied in Strens (2003) for discrete distributions sampling.

ter Braak, Vrugt

Meaningful extensions have been made by ter Braak and Vrugt group (ter Braak, 2006; ter Braak & Vrugt, 2008; Vrugt et al., 2009). In ter Braak (2006) an algorithm named Differential Evolution Markov Chain (DE-MC) is introduced for high dimensional target distributions sampling, motivated by simplicity, because the adopted mutation operator automatically provides information on scale and orientation of the proposal distribution. It is generally analogous to contribution in Strens et al. (2002), except for differential mutation operator, which has form: $\underline{y}_i = \underline{x}_i + \gamma(\underline{x}_{R1} - \underline{x}_{R2}) + \underline{e}$, $i \neq R1 \neq R2$, where γ is a scaling constant and $\underline{e}$ is random vector drawn from a symmetric distribution with small variance, for example a zero mean normal. A standard DE crossover operator can be included before the proposed solution is compared with $\underline{x}_i$: in that case every gene of $\underline{y}_i$ can be replaced by the equivalent gene of $\underline{x}_i$ with probability $(1 - pC)$. Author also suggested that applying crossover on blocks of genes, which may refer to correlated variables, could improve the effect of operator. Also a cooling scheme could be adopted, and initial population could be generated from a prior distribution, if a Bayesian problem is taken into account. The convergence of algorithm is monitored by use of $\hat{R}$ statistics (Gelman & Rubin, 1992).

An extension of the algorithm, called DE-MC$_Z$, has been provided in ter Braak & Vrugt (2008). Extensions have been made in order to lower the computational effort of the algorithm by decreasing the number of chains in population. In order to accomplish this a large matrix Z is built in order to include all generated chromosomes in generations: chains $\underline{x}_{R1}$ and $\underline{x}_{R2}$ for the mutation step will be selected from such matrix. The latter feature turns the method into an adaptive Metropolis sampler (Haario et al., 2001), as past chains state are involved. Furthermore a snooker style crossover, called *DE snooker* update, is introduced in the proposal mechanism, as it alternates with parallel direction updates, in order to diversify jumping possibilities. In these two papers authors considered applications on both known multivariate distributions sampling, like Student's t or mixtures of normal (as done in Liang & Wong, 2001a), and Bayesian problems like one-way random-effects model and nonlinear mixed-effect model. Effectiveness of proposed methods is shown to be comparable with respect to random walk Metropolis sampler. Furthermore DE-MC$_Z$ is shown to improve convergence time (namely lower the burn-in period) compared to standard DE-MC, and it is also parallelizable.

A further successful development, resulting in the most cited paper in this framework, has been called DiffeRential Evolution Adaptive Metropolis (DREAM; Vrugt et al., 2009). In this sophisticated algorithm the differential mutation step allows to generate proposals using higher-order (say number δ) pairs of chains, for increasing diversity, and also crossover of variable blocks (size d'), with probability CR, is proposed. Besides this, the burn-in period is crucial, because in such iterations the so-called *outlier chains*, which are solutions that still not have converged to modal zones, are handled; this issue, that can deteriorate quality of MCMC sampling, is managed by use of Inter-Quartile-Range (IRQ) statistics. During burn-in also a distribution of crossover probabilities CR is estimated for the algorithm in order to favor large jumps over smaller ones and decrease autocorrelation between two subsequent samples in each chain. Mutant is built as: $\underline{y}_i = \underline{x}_i + (1_d + \underline{e})\gamma(\delta, d')[\sum_{j=1}^{\delta} \underline{x}_{r_1(j)} - \sum_{k=1}^{\delta} \underline{x}_{r_2(k)}] + \underline{\epsilon}$, where $\underline{\epsilon}$ is drawn from a Uniform distribution and it is related to the scaling factor $\gamma(\delta, d')$.

Selected applications include sampling from high dimensional multivariate normal distributions, twisted Gaussian and bimodal distributions, and also a squared deviations likelihood function for dealing with a real dataset: DREAM algorithm showed the best overall performances in all selected applications. This method has received huge success in literature, especially in hydrological applications (see, for example, Laloy & Vrugt, 2013 and Brigode et al., 2013). Also an R package providing DREAM has been implemented by Guillaume et al. (2012).

3.5 EDA based approaches

Zhang, Cho

Zhang & Cho (2001) proposed an algorithm that conjugates efficiency of EAs and robustness of MCMC methods in order to identify systems architecture. As far as its main scope is maximization we will not dwell much on it.

The method, named evolutionary Markov Chain Monte Carlo (eMCMC), is set in an explicit Bayesian framework to find the architecture minimizing a fitness function. Starting from an initial population generated from a prior distribution, in fact, the problem dependent likelihood and then the posterior are computed for all individuals. New solutions are generated basing on the resulting posterior distribution, employing a kind of mutation and recombination operators, and a selection of best individuals is retained in subsequent generation. Drugan & Thierens (2004) observed that this method shares a number of features with EDA algorithm.

Laskey, Myers

Laskey & Myers (2003) introduced Population Markov Chain Monte Carlo algorithm (popMCMC), a variety of adaptive MCMC sampler in which chains use information from other chains to adjust their proposal distributions. They appeal to EAs because of their natural information exchange features between solutions and their ability of avoiding to be trapped in local optima. Authors explicitly refer to a Bayesian network learning problem with missing observations and hidden variables, for which solution space is discrete.

The chains share a common target $\pi(\cdot)$ but have a different individual proposal distribution $q(\underline{x}_i^{t+1}; \underline{x}_i^t, \xi)$, where ξ is a novel parameter. This latter quantity is estimated by a proposal parameter function $\hat{\xi}(x_1^t, ..., x_M^t)$, which accounts for values of entire population. For example, in the selected Bayesian network application $\hat{\xi}$ includes information on frequencies of graph arcs and missing values. In general it can be chosen to fit interesting features of π (lower order marginal distributions of components are suggested). Each of the estimated models $q(\underline{x}_i^{t+1}; \underline{x}_i^t, \hat{\xi})$, $i = 1, ..., M$, generates a candidate, and the resulting population is evaluated via MH step. This procedure is adaptive at the level of individual, because each proposal distribution depends on global information; on the other side, at the level of population it is a Markov Chain with fixed transition probabilities. Heuristic comments are also provided in order to illustrate convergence of each chain to π, depending on the choice of $\hat{\xi}$. Convergence diagnostic is performed by use of $\hat{R}$ statistics.

popMCMC has been compared with a multiple chains MH algorithm with no information exchange and an EA with mutation and crossover for the Bayesian network learning problem. Results of application on literature data showed that incorporating information exchange increased the rate of improvement of solutions, and that MCMC algorithms had greater population diversity than EA, because of *post selection* features of MH step. Authors observed that superiority in performance of popMCMC with respect to MH could be due to the ability of incorporating statistical information from the entire population into the proposal distribution q. Also in this case similarities with EDA have been observed in Drugan & Thierens (2004).

3.6 Discussion

In our overview we have proposed a sort of categorization with respect to the specific EA inspiring authors. Following definition of evolutionary system, outlined in Section 1.2 and adopted in De Jong (2006) as basis for defining EAs, we shall now discuss methods with respect of their algorithmic features.

Population of individuals

The multiple chains MCMC framework provides a population of solutions, in our case running in parallel, for improving mixing and sampling from target distribution. In EC based MCMC goals are the same, and so are sampling methodologies: if a cooling scheme is adopted, as in EMC, only one chain will effectively provide samples from π; in other cases, if correct ergodic properties are satisfied, each chain is able to sample from the target. In this case the user may consider population states at a certain generation as a candidate random sample and evaluate its adherence to π (Battaglia, 2001; Strens et al., 2002). Concerning solutions coding, we mainly took into account continuous target sampling problems, for which these algorithms adopted direct encoding.

Fitness

In this chapter no optimization issue is concerned, so fitness has a naturally different purpose with respect to the rest of thesis. Up until now, in fact, it is defined as a goodness measure, to be as large as possible (in maximization problems); here there is a target distribution π to sample from, and it could be somehow related to fitness. Now, in generic iteration values generated by most of considered MCMC algorithms must be subdued to MH or Metropolis steps, in order to ensure them to sample from the correct invariant distribution. This step naturally biases search process towards high probability areas of invariant distribution, because they will be selected with high probability as a consequence; it is somehow analogous (even if less strong) to what happens with fitness function in optimization problems. In our problem, however, we consider sampling from generally multimodal targets, so other strategies must be adopted in order to let the algorithm efficiently sampling from all the support, avoiding to get trapped in local optima areas. If methods based on DE and EDA are naturally more capable of overcoming this drawback because their operators involve several chains (more insights will be provided in next subsection), GA based proposals, on the other hand, generally modify few solution at each generation, so other strategies have been employed, some of which operate directly on target distribution.

Liang and Wong's EMC adopted PT style cooling scheme, which allows each chain to have its own individual target distribution $\pi_i(\cdot) \propto \exp\{-H(\cdot)/T_i\}$, where $H(\cdot)$ is explicitly defined as fitness. By proceeding this way sampling at high temperatures facilitate broad exploration, and effective sampling from target distribution, which has the coolest temperature, is performed by means of exchange operation. There is an analogous reasoning behind Multiresolution GA by Holloman et al. (2006), because distribution of interest is taken as product of distributions at each scale. In this complex model, however, multiresolution scheme is applied also to observed data, in such a way that data modeled at coarse scale can support broader exploration of search space, while finer scales, on the other hand, allows to include as mushc details on target as possible. DGMC by Hu & Tsui (2010) employs subpopulations, which may separately explore and exploit possibly different portions of target support. In Battaglia (2001) a finite partition of π is built and a notion of fitness related to individuals contribution on inducing each partition is introduced. As long as reproduction probability is shared between individuals belonging to the same partition, several and possibly different zones of high probability can be detected in such a way.

Reproduction

Reproduction operators, which aim at building new solutions, play naturally the role of proposal distributions in MCMC, as long as they have stochastic features.

This topic highlights distinctions between approaches based on GA, DE and EDA, due to the number of individuals involved in process of building new solutions. In generic iteration of GA based methods, in fact, a small number of individuals is generally used to build new states: an Evolution Strategies style mutation operator, which involves a single chain, is ofter employed, together (or sometimes in substitution) with few snooker crossover updates, involving two of three individuals; wide exploration of the support is guaranteed by means of strategies described in previous subsection, like exchange (Liang & Wong, 2001a) or swapping (Holloman et al., 2006) between chains at different temperatures or scale, and migration (Hu & Tsui, 2010) between subpopulations. In DE based approaches, as in original algorithm, the new trial vector is proposed for each chain basing on values of other individuals (by use of differential mutation), performing also uniform crossover in order to account for correlation between variables. EDA methods build a proposal distribution basing on values of current population as a whole, so we can say that the magnitude of interaction is maximum in this case, with respect to other methods. A deep and unifying analysis of possible reproduction operators involving various number of chains in EC based MCMC has been provided in Drugan & Thierens (2005, 2010a, 2010b).

Turning to a computational point of view, it is interesting to mention the possibility of parallelizing these kind of MCMC methods (see Basse et al., 2016 for an account). It is clear that methods which involve few moves in reproduction are more suitable to be parallelized, because chains belonging to different cores need to have reached the same number of generations in order to be assembled for reproduction. This problem could be handled by employing some adaptive strategies, which allow to use samples from past generations, as in ter Braak & Vrugt (2008) and Vrugt et al. (2009).

Inheritance

Once that new individuals are generated by reproduction operators, it is necessary to discriminate the ones who will be included in subsequent generation. In generic MCMC this task is accomplished by strategies introduced to preserve ergodicity of chains, like MH or Metropolis step, which may be defined as post selection operators in EC terminology.

These steps, depending on strategies, may involve individual d-dimensional target distribution π_i (possibly constant with respect to i) or $M \times d$-dimensional population distribution π^*, as in population-based MCMC. In fact there are algorithms which evaluate acceptance of population as a whole after each reproduction (Liang & Wong, 2001) or some specific one (migration operator in Hu & Tsui, 2010), in order to preserve ergodicity of π^*. Most of methods, however, accept new proposed values evaluating just individual target distributions involved in reproduction.

In general, MH and Metropolis step are are crucial, especially in multiple chains algorithms, as long as computational complexity of procedures is taken into account. In Metropolis step, concerned when symmetrical proposal distributions are selected, acceptance probability does not include the proposal distribution (like mutation in GA based approaches or differential mutation in methods based on DE), meaning that some computational time is saved. These kind of issues have been studied, also in the form of tradeoffs, in Drugan & Thierens (2010a, 2010b).

3.7 Concluding remarks

Methods outlined in this chapter have been proposed by researchers from different fields of science, sometimes independently of each other. Therefore there have been different motivations and points of view behind these proposals, and giving a unifying framework to compare them is challenging.

M. Drugan and D. Thierens, both researchers in the field of EC, already cited in the course of chapter, produced a series of papers (Drugan & Thierens, 2004; 2006; 2010a; 2010b) in which most of algorithms discussed in this dissertation are reviewed. They provided general forms of proposal distributions, for example geometrical moves like rotation or translation, which may involve two or more chains in population. Moreover studies have been conducted for evaluating benefits of EAs features, like fitness proportionate selection, elitism, sophisticated offspring surviving rules on speed of convergence to invariant distribution. They also gave the following definition of *Evolutionary MCMC* (Drugan & Thierens, 2010a, 2010b):

Definition 1. *An evolutionary Markov chain Monte Carlo (EMCMC) algorithm is a population MCMC that exchanges information between individual states such that, at the population level, the EMCMC is an MCMC.*

Some of the algorithms in our survey fall into EMCMC category, but in general the condition on population level is rather strict for characterizing MCMC sampling, because many proposal moves can be evaluated individually for each chain.

We observe that EMCMC is a particular case of *Population-Based MCMC*, a category that includes methods in which multiple chains are allowed to run in a parallel manner. Mathematical description of method (Liang et al., 2001, p.123) states that if $\pi(\underline{x})$ is the target distribution then user shall sample from an augmented invariant distribution:

$$\pi^*(\mathbf{X}) = \prod_{i=1}^{M} \pi_i(\underline{x}_i), \tag{3.1}$$

where $\mathbf{X} = \{\underline{x}_1, ..., \underline{x}_M\}$ belongs to a M-dimensional space and $\pi_i = \pi$ for at least one i.

EMCMC is a Population-Based MCMC where chains are allowed to interact with each other, as happens with individuals in EAs, but as we said before the assumption that π^* is the invariant distribution of the population is somewhat strict for generalizing to all methods.

There is also no general agreement on how to evaluate method performance: in fact, as in MCMC literature, effort is generally spent to monitor convergence of chains to invariant distribution, while goodness of effective sampling is not deepened. In some papers authors analyze adherence of candidate sample at certain generation to target distribution (Battaglia, 2001; Strens et al., 2002; Drugan & Thierens, 2010a).

However, if a complex Bayesian problem is taken at hand a general indication would suggest to generate initial chains population by selected prior distribution; after that, methods based on DE, suggested to be simple and very effective in capturing multimodality and correlation between parameters, could be employed. The possibly large computational cost of these procedures, however, could deflect and make users prefer refined GA based approaches, which are less expensive but possibly competitive. However, the introduction of adaptive strategies can make parallelization feasible and computational complexity more tractable (as in ter Braak & Vrugt, 2008 and Vrugt et al., 2009). As far as this subject is concerned, we believe that adaptive strategies, which are among main topics in nowadays MCMC literature, will prove to be useful tools for improving EC based MCMC, from both efficiency and computational side (see, for example, Milgo et al., 2017, for up-to-date research, in which Covariance Matrix Adaptation-Evolution Strategies algorithm is set in MCMC framework).

Chapter 4

Multiple Changepoint Detection in Periodic Autoregressive Models by Means of Genetic Algorithms

4.1 Periodic models and regime changes

Many phenomena observed over time are subject to so-called seasonal effects, which are variations occurring at specific and regular time intervals every year. An intuitive example is the behaviour of a monthly business time series in the month of August, which is often closing month in companies (*August effect*). In general, *seasonality* needs to be conveniently accounted in a large variety of time series models in order to get realistic estimates and forecasting.

Among linear modeling a classical procedure aims at modifying the standard AutoreRegressive Integrated Moving Average (*ARIMA*) model employing the *seasonal differencing operator*: if the considered period magnitude is s, this operator subtracts from each observation the corresponding value at s previous time instants, obtaining Seasonal ARIMA (*SARIMA*). This way of proceeding, which involves relatively few parameters, has been proven useful when the mean for a given season is not stationary across years (Hipel & McLeod, 1994). It has also been observed, however, that it tends to perform less well when covariances and correlations within seasons are not stationary, because residuals could still disclose a seasonal behaviour.

For this reason different procedures of accounting for seasonality have been proposed in literature, leading to *periodic* models (general overviews can be found in Hipel & McLeod, 1994 and Franses & Paap, 2004). In this framework the simplest model is the Periodic AutoRegression (PAR; Gladyshev, 1961; Jones & Brelsford,

1967) which, as long as a seasonal time series of N years and period s is considered, has the following structure:

$$Y_{ns+k} = \sum_{i=1}^{p(k)} \phi_i(k) Y_{ns+k-i} + \epsilon_{ns+k},\ n = 0, ..., N-1,\ k = 1, ..., s, \tag{4.1}$$

where series in season k follows an AR($p(k)$), with parameters $\phi_i(k),\ i = 1, .., p(k)$. Franses (1994) introduced also an unusual multivariate representation of model (4.1), useful for analyzing stationarity properties of the model. Also periodic modifications of other linear models, as Periodic Moving Average (PMA; Cipra, 1985) or Periodic AutoRegressive Moving Average (PARMA; Vecchia, 1985), have been introduced in literature, even if it has been observed that they do not generally add significant benefits over PAR models (McLeod, 1994; Franses & Paap, 2004).

As far as PAR model building is concerned, the identification can generally be performed in several ways. As a first step, non-periodic models are estimated and seasonality evaluated in residuals. Similarly, also statistical tests in which null hypothesis is the lack of periodic variation in model can be performed. A more general approach is the selection of model order by conventional penalization criteria, like AIC, BIC or MDL. Ordinary maximum likelihood or least squares estimation of parameters can be then performed.

The diagnostic checking for PAR models has been proposed in McLeod (1994), in which results on distribution of residual autocorrelations are derived and a novel test statistics based on Ljung-Box portmanteau is introduced.

Let us now introduce a different source of deviation from basic linear models, due to the fact that a time series could switch its behaviour, implying the existence of several regimes. The change between one regime and an other could occur at every time instant or be due to the reaching of a certain value of series. In the first case we generally have a nonstationary but linear model (*structural change*; Bai & Perron, 1998), while the second falls in the field of *threshold models* (Tong, 2012), which is characterized by nonlinearity but stationarity. These are two different situations which require different modeling features: in this chapter we shall only focus on structural changes, set in a periodic modeling framework.

A structural change (or changepoint) can be defined as a modification in the structure of a time series occurring at a certain time instant. This kind of change could affect mean, variance or model structure as a whole, and more than one change could occur in the time series span. Real examples of structural change could be the effect of a modification in governmental policies on a financial time series, or a change in gauging location on climate and hydrological series. Ignoring the effect of these changes, possibly located at unknown times, can lead to misleading estimation and forecasting. Among approaches proposed in literature for dealing with structural changes we focus on methods which aim at selecting an approximate model by optimization of an appropriate objective function, like AIC (Kitagawa & Akaike, 1978; Ninomiya, 2015). In this framework there have also been proposals based on GAs, which will be reviewed in Section 4.3.

4.2 Model description

We shall now describe in depth our proposal of simultaneously modeling seasonality and regime changes in time series. Concerning the first point, we shall focus on pure PAR models, allowing also subset selection; multiple structural changes can segment the series into several PAR processes.

The period of time series is s and is assumed to be known. Observation in season k of the $n+1$ year is denoted by X_{ns+k}, with $n = 0, 1, \dots, N-1$ and $k = 1, \dots, s$. There are M different regimes, each of which contains an integer number of years, and τ_{j-1} denotes the first year of regime j. The first regime includes years from $\tau_0 = 1$ to $\tau_1 - 1$, second regime contains years from τ_1 to $\tau_2 - 1$, third regime contains years from τ_2 to $\tau_3 - 1$, and so on. The regime structure, specified by $m = M - 1$ changepoints, is summarized as follows:

$$1 \equiv \tau_0 < \tau_1 < \ldots < \tau_m < \tau_M \equiv N + 1.$$

In order to ensure reasonable estimates it is required that each regime contains at least a minimum number mrl of years, therefore $\tau_j \geq \tau_{j-1} + mrl$, $\forall j$. We let $R^j = \{\tau_{j-1}, \tau_{j-1} + 1, \dots, \tau_j - 1\}$, so that if year n belong to set R^j then the time $ns + k$ is in regime j. For the seek of simplicity we assume that total number of observations T is a multiple of s.

The model driving our work is given by:

$$X_{ns+k} = a^j + b^j(ns + k) + W_{ns+k},\ n \in R^j,\ j = 1, 2, \dots, M,\ 1 \leq k \leq s, \tag{4.2}$$

where $W_{ns+k} = Y_{ns+k} + \mu_k^j$ and process Y_{ns+k} is a PAR given by:

$$Y_{ns+k} = \sum_{i=1}^{p^j(k)} \phi_i^j(k) Y_{ns+k-i} + \epsilon_{ns+k}. \tag{4.3}$$

We assume that trend parameters a^j and b^j depend only on the regime, whereas means μ_k^j are allowed to change also with seasons. The autoregressive maximum model order at season k in the j-th segment is given by $p^j(k)$, so that $\phi_i^j(k)$, $i = 1, \ldots, p(k)$, represent the PAR coefficients of season k in the j-th segment; in our procedure these latter will be allowed to be constrained to zero. For simplicity, we assume that $p^j(k) = p$, $\forall j, k$. Error process ϵ_{ns+k} in equation (4.3) is a periodic white noise, with $E(\epsilon_{ns+k}) = 0$ and $Var(\epsilon_{ns+k}^j) = \sigma_{j,k}^2 > 0$. Unless otherwise stated we assume that each segment is periodic stationary with period s, in the sense that

$$Cov(Y_{n+s}, Y_{m+s}) = Cov(Y_n, Y_m),$$

for all integers n and m.

Summarizing, the proposed model is characterized by following parameters:

a) *External parameters*:

N	number of years
s	number of seasons
p	maximum autoregressive order
M	maximum number of regimes
mrl	minimum number of observations per regime

b) *Structural parameters :*

m	number of changepoints
$\tau_1, \tau_2, \ldots, \tau_m$	changepoints location
PAR subset indicators	denote constrained coefficients $\phi_i^j(k)$

c) *Regression parameters*

$a_1, a_2, \ldots, a_M$	constants
$b_1, b_2, \ldots, b_M$	slopes
μ_k^j	seasonal means; regime j, season k
$\phi_i^j(k)$	AR parameters; regime j, season k, lag i
$\sigma_j^2(k)$	residual variance; regime j, season k

In order to build our model, structural and regression parameters must be conveniently estimated. Conditionally on model structure, the regression parameters are analytically estimated. The selection of optimal structural parameters, on the other side, is a complex combinatorial problem for which no closed form solution is available. As far as it involves the evaluation of a very large number of possible combination, GAs are naturally suitable for this issue.

4.3 Model building

As outlined in Section 1.6, model identification is among the most important and natural applications of GAs to statistics. This issue is especially demanding in time series models exhibiting nonlinearity or nonstationarity (or both), because the search space is prohibitively large. GAs have been widely applied for identifying threshold models among last 15 years: Wu & Chang (2002) proposed them for two-regimes SETAR models; Yau et al. (2015) identified TAR models by GAs; many contributions have been made by R. Baragona research group, as they involve models such SETARMA (Baragona et al., 2004a), DTARCH (Baragona & Cucina, 2008), DTGARCH (Baragona & Battaglia, 2006), EXPAR (Baragona et al., 2002), PLTAR (Baragona et al., 2004b), multivariate SETAR (Baragona & Cucina, 2013). In the case of structural changes modeling, the time series exhibits a nonstationary behaviour, as it could switch regime at each time instant. Davis et al. (2006) employed a piecewise stationary AR process for modeling structural changes, and used GAs for model identification; Jeong & Kim (2013) set changepoint detection by GAs in a Bayesian modeling framework; recent paper by Doerr et al. (2017) provided hints for saving computational time when GAs are employed in this identification problem; Battaglia & Protopapas (2011, 2012) employed GAs for detecting regime changes in time series exhibiting also nonlinear behaviour.

4.3.1 Identification and estimation

In our model the GA must account for both changepoints detection and subset PAR selection. Work by Lund et al. (2007) and Lu et al. (2010) are concerned with changepoint detection in periodic and autocorrelated time series, when only change in mean are contemplated. Our results share a number of similarities with their finding allowing in the same time a generalization of results, because a change can cause model structure as whole to be modified. Details on our GA proposal, which employ a standard binary coding, will follow.

The model structure of a generic solution is encoded in a binary chromosome (genotype), which corresponds to a phenotype associated to the following vector:

$$m,\, \tau_1,\, \tau_2, \ldots, \tau_m, \underline{\delta}^1, ..., \underline{\delta}^M, \tag{4.4}$$

where $\underline{\delta}^1, ..., \underline{\delta}^M$ are binary sequences specifying parameters $\phi_i^j(k)$ constrained to zero for regime j, season k and lag i.

A candidate segmentation is encoded in a binary chromosome as follows: first two bits give number of changepoints m (limited to a maximum of 3 in our study, so that a number of regimes up to 4 is allowed); subsequent bit intervals, whose length is custom fixed, produce changepoint times $\tau_1, ..., \tau_m$. This part of encoding must ensure following constraints:

$$mrl{+}1 \leq \tau_1,\ mrl{+}\tau_1 \leq \tau_2,\ ...,\ mrl{+}\tau_{m-2} \leq \tau_{m-1},\ mrl{+}\tau_{m-1} \leq \tau_m \leq N{-}mrl{-}1,$$

due to the fact that a minimum number mrl of observations must be contained in each regime. In order to accomplish this the bit intervals encode m real numbers $th_i \in (0,1),\ i = 1, ..., m$, constructed to determine percentage of remaining values to place a changepoint. In fact, when placing a new changepoint there are some illegal positions, due to above specified constraints: this implies that mrl observations must be left out from both the beginning and the end of considered segment. This strategy depends on the candidate number of regimes, so that changepoints are uniquely identified in these four possible ways:

- If $m = 0$ (one regime) then $\tau_1 = N + 1$.
- If $m = 1$ (two regimes) then $\tau_1 = mrl + 1 + (N - 2mrl) \times th_1$
- If $m = 2$ (three regimes) then:
  - $\tau_1 = mrl + 1 + (N - 3mrl) \times th_1$
  - $\tau_2 = mrl + \tau_1 + (N - 2mrl - \tau_1 + 1) \times th_2$
- If $m = 3$ (four regimes) then:
  - $\tau_1 = mrl + 1 + (N - 4mrl) \times th_1$
  - $\tau_2 = mrl + \tau_1 + (N - 3mrl - \tau_1 + 1) \times th_2$
  - $\tau_3 = mrl + \tau_2 + (N - 2mrl - \tau_2 + 1) \times th_3$

Such an encoding procedure, introduced in Battaglia & Protopapas (2012), allows each possible chromosome to be legal, so there is no computational time wasted on evaluating infeasible solutions. Last bits in the chromosome directly produce vectors of subset PAR indicators $(\underline{\delta}^1, ..., \underline{\delta}^M)$.

Conditioning on a candidate model structure, regression parameters estimation is performed in the fitness evaluation step as follows:

- Trend parameters estimates $\underline{\hat{a}}$ and $\underline{\hat{b}}$ are obtained by Ordinary Least Squares (OLS) method:

$$\min_{\underline{a},\underline{b}} \sum_{j=1}^{M} \sum_{k=1}^{s} \sum_{n \in R^j} \left[X_{ns+k} - a^j - b^j(ns+k)\right]^2,$$

that leads to detrended data $\hat{W}_{ns+k} = X_{ns+k} - \hat{a}^j - \hat{b}^j(ns+k)$,
$n \in R^j,\ \ j = 1, ..., M,\ \ k = 1, ..., s$

- Seasonal means $\underline{\hat{\mu}}$ are computed as follows:

$$\hat{\mu}_k^j = \frac{1}{\tau_j - \tau_{j-1}} \sum_{n \in R^j} \hat{W}_{ns+k},\ \ j = 1, ..., M;\ k = 1, ..., s,$$

which implies: $\hat{Y}_{ns+k} = \hat{W}_{ns+k} - \hat{\mu}_k^j$

- Autoregressive parameters estimation is performed separately for each regime and season. Each of these series $\underline{z}$ is selected from $\underline{\hat{Y}}$, and it is incorporated in a design matrix Z of dimensions $(\tau_j - \tau_{j-1}) \times p$, which includes lagged observations. Parameter constraints are specified by a $(p - q) \times p$ matrix H, where q is the number of free parameters. These constraints are designated on the basis of PAR subset indicators $\underline{\delta}$ as follows:

  - For each lag i, the element $[p(k-1)+i]$ of $\underline{\delta}^j$ vector is evaluated
  - If value is equal to 1 then a row equal to the i-th row of I_p identity matrix is added to H.

  Final estimate $\underline{\hat{\phi}}$ of $\underline{\phi}$ is obtained by constrained optimization, with linear constraint given by $H\underline{\phi} = 0$. Explicitly, in matrix form:

$$\underline{\hat{\phi}} = \underline{\hat{\phi}}_{LS} - (Z'Z)^{-1}H'[H(Z'Z)^{-1}H']^{-1}H\underline{\hat{\phi}}_{LS},$$

  where $\underline{\hat{\phi}}_{LS} = (Z'Z)^{-1}Z'\underline{z}$.

- Lastly, estimate of innovation variances $\hat{\sigma}_j^2(k)$ is performed for each regime and season on final residuals, considering that each regime has a possibly different sample size.

The fitness must include a term linked to the goodness of fit and a part related to a penalization on number of parameters. Many options are available: we shall consider a criterion inspired by NAIC, introduced by Tong (1990, p.379) for threshold models, and given by:

$$g = \{\sum_{j=1}^{M}\sum_{k=1}^{s} n_j(k)\log[\hat{\sigma}_j^2(k)] + IC\sum_{j=1}^{M}\sum_{k=1}^{s} P_j(k)\}/T, \tag{4.5}$$

where $\hat{\sigma}_j^2(k)$ is the model residual variance of series in regime j and season k, $n_j(k)$ is related sample size, $P_{j,k}$ is related number of parameters, IC is the penalization term. Final fitness function is a scaled exponential transformation of g, for a purpose of maximization: $f = \exp(-g/\beta)$, where β is a constant.

As far as the choice of genetic operators is concerned we propose standard roulette wheel selection, bit-flip mutation, and a modified single-point crossover: the only cutting points allowed to be selected are the ones which subdivide phenotype (4.4), instead of genotype as usual. In such a way parameter structures can be naturally inherited by offspring. Elitist strategy is also employed.

4.4 Applications

In this section the validity of proposed methodology is studied. In the first part we shall focus on simulated data, while in the second real hydrological data will be analyzed, employing a modified version of the GA and also evaluating forecasting accuracy of fitted models. Computations will be performed by use of Matlab.

4.4.1 Simulations

The estimation procedure outlined in subsection 4.3.1 will be implemented in a small simulation study. We shall focus on monthly data (period $s = 12$), observed in $N = 100$ years. Such time series will be generated according to five possible scenarios:

A) 3 regimes, with varying PAR parameters and trend

B) 2 regimes, with varying PAR parameters and trend

C) 2 regimes, where only trend varies with regime

D) 2 regimes, where only innovation variances vary with regime

E) 1 regime

Time series generated according to these scenarios are shown in Figure 4.1. Seasonal means μ_k^j are always fixed at zero and the trend is built to be piecewise linear continuous. Innovation variances are equal to 1 in all experiments, except for scenario D in which they are 1 in first regime and 2 in the second. PAR parameters vary between regimes only in scenarios A and B, meaning that changepoints detection will be more difficult in scenarios C and, in particular, D, where regime switch is due only to innovations variance. These parameters are defined in following matrices:

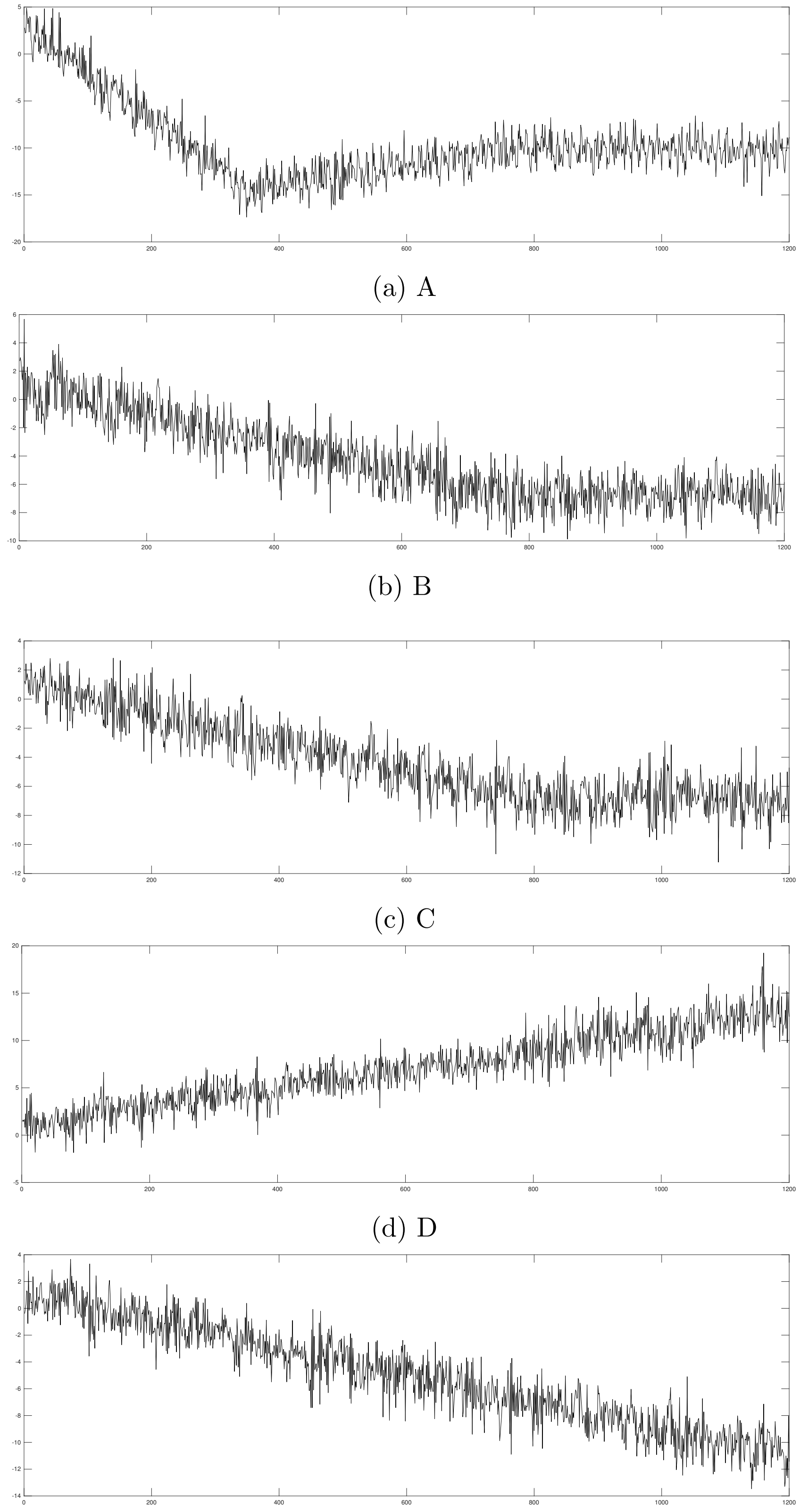
$$\underline{\phi}^1 = \begin{bmatrix} 0.3 & 0.3 & 0.3 & 0.42 & 0.42 & 0.42 & -0.8 & -0.8 & -0.8 & 0.42 & 0.42 & 0.42 \\ 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.35 & 0.35 & 0.35 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{\phi}^2 = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.22 & 0.22 & 0.22 & -0.24 & -0.24 & -0.24 & -0.5 & -0.5 & -0.5 \\ 0.3 & 0.3 & 0.3 & 0 & 0 & 0 & 0.23 & 0.23 & 0.23 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0.25 & 0 & 0 & 0 \end{bmatrix}$$

where number of columns is period s and rows number indicates maximum autoregressive order $p = 3$. Matrix $\underline{\phi}^1$ denotes PAR parameters of first regime in all experiments; $\underline{\phi}^2$ is associated to the second regime only in experiments A and B, while in C and D $\underline{\phi}^1$ denotes also parameters of second regime. Third regime in experiment A is generated according to a white noise.

Concerning GA configurations, we fixed minimum number of years per regime mrl at 10, maximum number M of regimes at 4 and a maximum autoregressive order p at 3. In the fitness function we fixed $IC = 2$ so that penalization structure of AIC criterion is resembled. We adopted GA operators and configurations outlined in subsection 4.3.1, with crossover, mutation rate and population size fixed at, respectively, 0.7, 0.2 and 50. Scaling constant β in fitness was equal to 10.

Table 4.1 shows results of computations, obtained with $G = 1000$ generations; it reports true and estimated changepoints, along with the absolute value of bias related to trend parameter estimates. Results are satisfactory in all models, particularly in tricky scenarios such as C and D. Plots of residual autocorrelations in Figure 4.2 confirm adequacy of fitted models.



(a) A

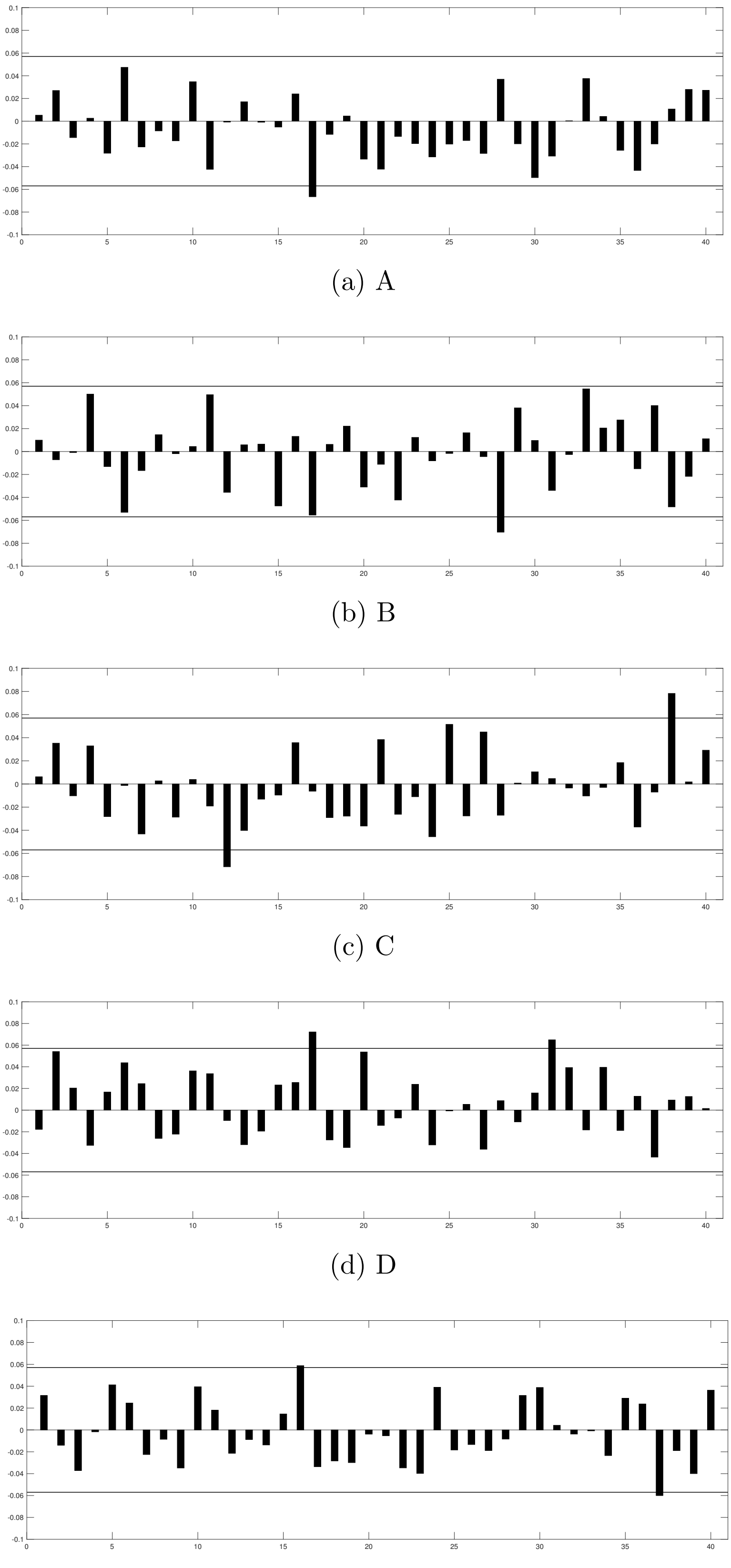
(b) B

(c) C

(d) D

(e) E

Figure 4.1: Simulated time series of five scenarios



(a) A

(b) B

(c) C

(d) D

(e) E

Figure 4.2: Residual autocorrelations of the fitted models in five scenarios

	$\tau_1, ..., \tau_m$	$\hat{\tau}_1, ..., \hat{\tau}_m$	Bias $[\hat{a}^1, \hat{b}^1]$	Bias $[\hat{a}^2, \hat{b}^2]$	Bias $[\hat{a}^3, \hat{b}^3]$
A)	$31, 70$	$30, 70$	[0.0069,0]	[0.0850,0.0002]	[0.0596,0]
B)	66	66	[0.0808,0]	[0.0356,0]	/
C)	66	68	[0.037,0]	[0.1014,0]	/
D)	66	64	[0.1710,0.0005]	0.3640,0.0005	/
E)	/	/	[0.0367,0.0001]	/	/

Table 4.1: Results for simulated data

4.4.2 Real data

We shall now study the effectiveness of proposed methodology in river flow analysis. Majority of hydrological time series, in fact, display seasonality and have been extensively analyzed with periodic models (Hipel & McLeod, 1994). Moreover, discontinuities are often introduced in this kind of series as a result of anthropogenic impacts or changes in instrumentation, location and climatic oscillations. Further plausible reasons are modifications in reservoir system management or new water pricing. In many cases, changepoints are located at known times (dam construction, measure instrument change) and it is easy to take into account their effects. When changepoints are located at unknown times and their features are ignored the time series estimation can be misleading (Lu & Lund, 2007; Lund et al., 2007). In view of all this, changepoint detection becomes a demanding job especially if its identification is required soon after occurrence (e.g. flood predictions). Many authors have considered the problem of detecting a single changepoint in hydrology (Cobb, 1978; Buishand, 1984; Hipel & Mcleod, 1994), but more realistic multiple changepoints situations should be considered.

We shall analyze monthly data related to two river flows, having different lengths, means of annual flows and located in different regions. They consist of:

- flows of Garonne river measured at Tonneins, France;
- flows of Saugeen river measured at Walkerton, Canada.

The GA employed in these two analysis includes a modification with respect of basic algorithm described in subsection 4.3.1. Its phenotype considers only $m, \tau_1, ..., \tau_m$ as candidate structural parameters, and in the fitness evaluation step it enumerates all 2^p possible subset AR(p) models in each regime and season: only result on the best one (in terms of fitness) is reported. This version allows to select the best possible subset for each segmentation, but it is computationally feasible only when

	Years of changepoint	Fitness	$RMSE$	MAE	$MAPE$
$\mathrm{PAR}_{(0;3)}$	/	1.203	0.247	0.221	2.356
$\mathrm{PAR}_{(1;3;10)}$	1989	1.211	0.273	0.213	2.266
$\mathrm{PAR}_{(2;3;12)}$	1977, 1989	1.214	0.272	0.213	2.264
$\mathrm{PAR}_{(3;3;10)}$	1970, 1988, 1998	1.224	0.314	0.251	2.610

Table 4.2: Results of evaluation criteria of the logarithmic forecast errors for Garonne

number p is small. In our case it is reasonable because the autoregressive procedure must capture short term dependence, while the underlying behaviour is mainly accounted by analysis of regime changes. Genetic operators and rates are chosen as in subsection 4.4.1.

Before running the GA time series are logarithmically transformed and last year is removed, as it is used to evaluate forecasting, which is performed by standard one-step-ahead procedure. Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) have been selected as forecasting accuracy indicators (an account on these measures is given in Hyndman & Koehler, 2006). Several experiments have been conducted considering various combinations of model external parameters p, mrl and M. Conditioning on four possible values of M, which include stationary model (no changepoints) and situations with possible structural changes up to, respectively, 1, 2 and 3, we selected four models for which the best value of fitness function has been observed. Forecasting accuracy of these models, labelled as $\mathrm{PAR}_{(M;p;mrl)}$ $(M = 0, 1, 2, 3)$, will be then evaluated.

Garonne river

The Garonne river, which flows through Spain and France, is the third largest river in France in terms of flow. Its total length is about 647 km with a catchment area of 51500 km^2 at Tonneins. It is the main contributor to the Gironde Estuary which is the major European fluvial-estuarine system. Flow measures are recorded at the Tonneins gauging station, where there is no tidal effect. Data are obtained from daily discharge measurements in cubic meter per second (m^3/s) from January 1959 to December 2010 (DIREN-Banque Hydro, French water monitoring). Daily data flows are then transformed in monthly data consisting in flows averaged for one month. The final time series of mean monthly flows of Garonne, from January 1959 to December 2010, including 624 observation (52 years), has been analyzed also in Ursu & Pereau (2016). It is shown, along with log-transformed data, in Figure 4.3.

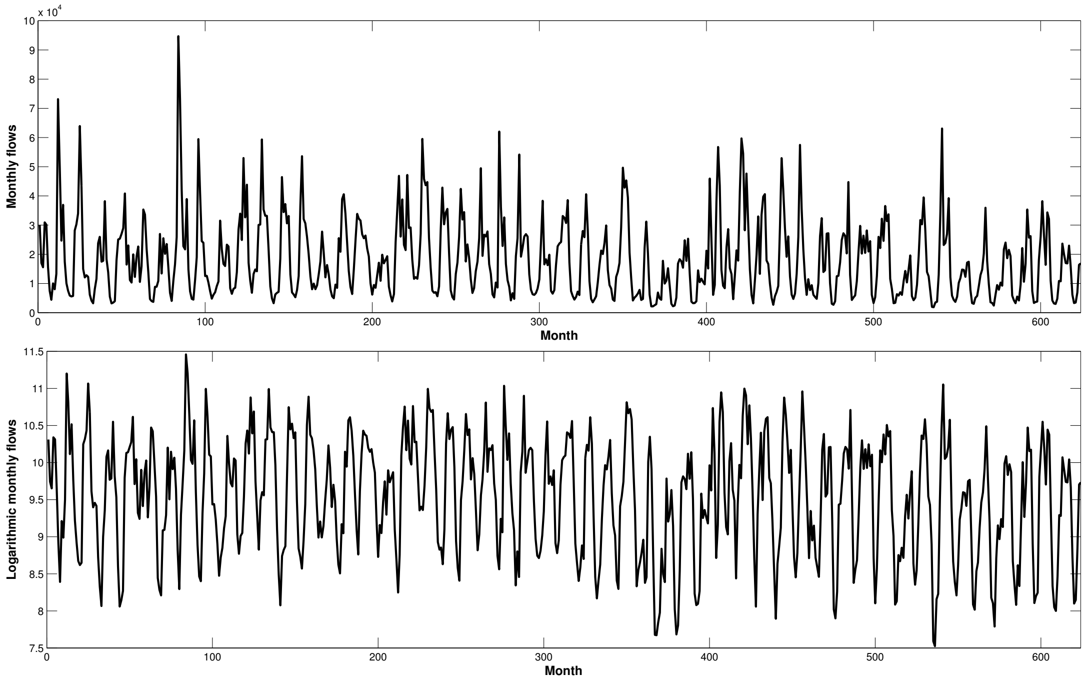


Figure 4.3: Monthly flows (up) and logarithmic monthly flows (down) for the Garonne river.

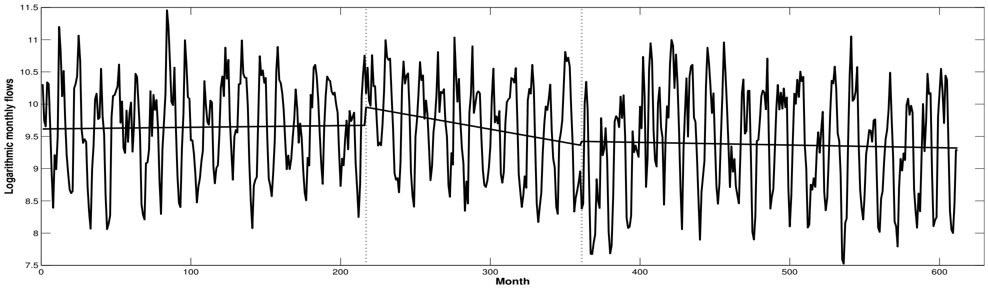


Figure 4.4: Changepoints detected on years 1977 and 1989 for Garonne river

Table 4.2 shows results on changepoint detection, goodness of fit and forecasting accuracy. We observe that years 1988 or 1989 are detected as possible changepoints in all configurations. According to Caballero et al. (2007), years 1988-1989 seems to be the driest in decade 1980-1990. Moreover, the air temperature over Western Europe showed an abrupt shift at the end of 1980s. For a better understanding of climatic changes and their impact on water resources, Brulebois et al. (2015) studied a subset of 119 temperatures, 122 rainfall and 30 hydrometric stations over the entire France. They detected a shift in annual mean air temperature in 1987-1988 for more than 75% of the 119 temperature stations. They also detect a shift between 1985 and 1990 for 18 hydrometric stations.

As far as goodness of fit is concerned, we observe that fitness values are increasing

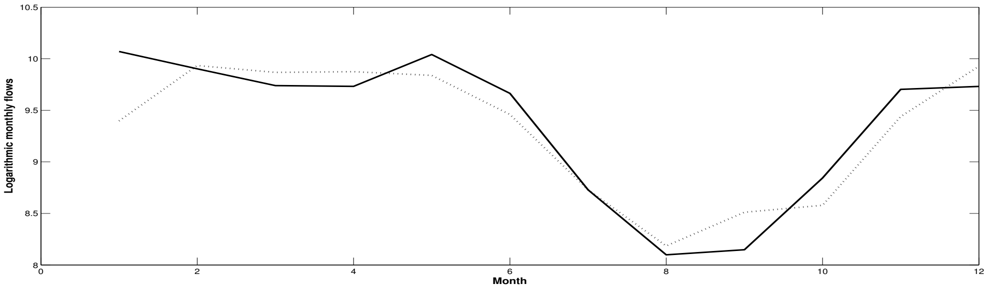


Figure 4.5: Logarithmic flows of Garonne (full line) and one-step PAR forecasts (dashed line).

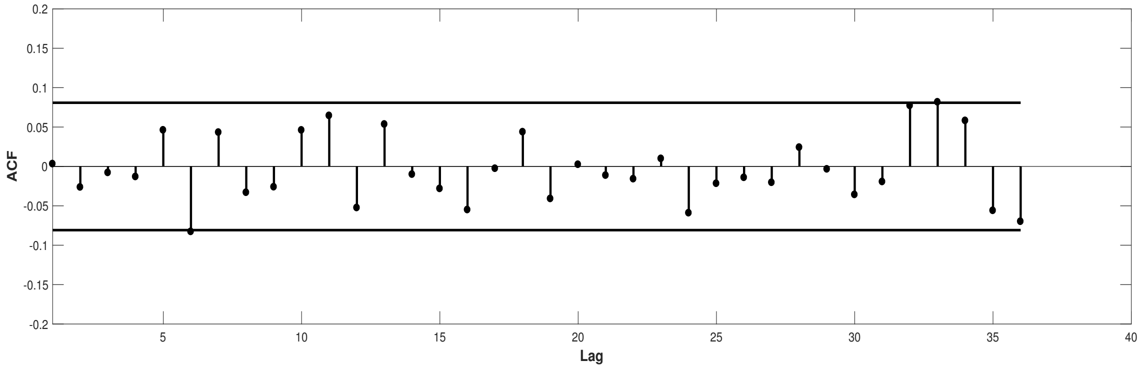


Figure 4.6: Autocorrelation function (ACF) of the residuals of the fitted PAR model with two changepoints to the Garonne flow.

with the number of regimes. Results on forecasting accuracy show that model with no changepoints forecasts better in terms of RMSE with respect of other models, while best values for MAE and MAPE are observed for three regimes model. In this comparison we select this latter model considering both performances on estimation and forecasting. Figure 4.4 shows the segmentation selected in this model, while in Figure 4.5 the true and predicted logarithmic values of Garonne flows are reported. As a diagnostic check, the residual autocorrelations for three regimes model up to lag 36 have been computed. They are reported in Figure 4.6 and provide evidence on adequacy of the proposed model.

Saugeen river

The Saugeen River is located in southern Ontario, Canada; it begins in the Osprey Wetland Conservation Lands and flows generally north-west about 160 kilometres (99 miles) before exiting into Lake Huron. Starting from 1950 it is served by Saugeen Valley Conservation Authority (SVCA), a corporate body founded for managing and preserving water and other natural resources in river watershed. Data analyzed are

	Years of changepoint	Fitness	$RMSE$	MAE	$MAPE$
$\text{PAR}_{(0;1)}$	/	1.187	0.485	0.371	11.184
$\text{PAR}_{(1;1;7)}$	1970	1.191	0.352	0.286	9.017
$\text{PAR}_{(2;3;5)}$	1965, 1970	1.207	0.375	0.296	9.338
$\text{PAR}_{(3;2;7)}$	1950, 1958, 1970	1.201	0.376	0.296	9.264

Table 4.3: Results of evaluation criteria of the logarithmic forecast errors for Saugeen river

average monthly riverflow from January 1915 until December 1976, measured at Walkerton, Ontario, and are showed in Figure 4.7. This series, among many other river flow data, is discussed in Noakes et al. (1985).

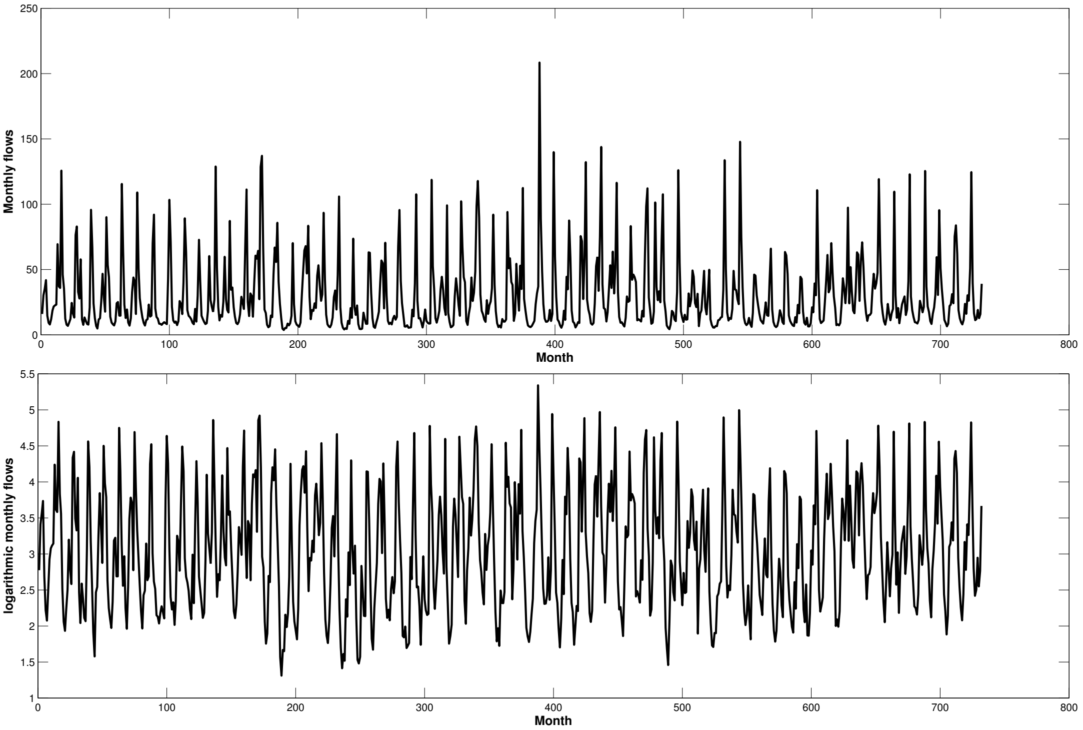


Figure 4.7: Monthly flows (up) and logarithmic monthly flows for Saugeen river

Table 4.3 shows results of optimal models: year 1970 is always detected as possible changepoint. One reason would be related to works aimed at reconstructing Denny's Dam, in which a popular conservation area for fishing is located. In fact, between the end of 1960s and the beginning of 1970s, Great Lakes Fishery Commission managed to rebuild Denny's Dam in order to provide an effective bloackage against parasites such as sea lamprey, preventing them from infiltrating in Saugeen river. Being Denny's Dam among the biggest dykes of river course this could have had a non ignorable effect on its flow. There have also been important human work on

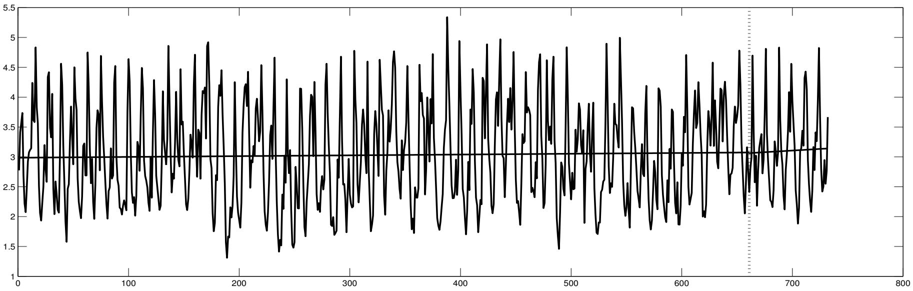


Figure 4.8: Changepoint detected on year 1970 for Saugeen

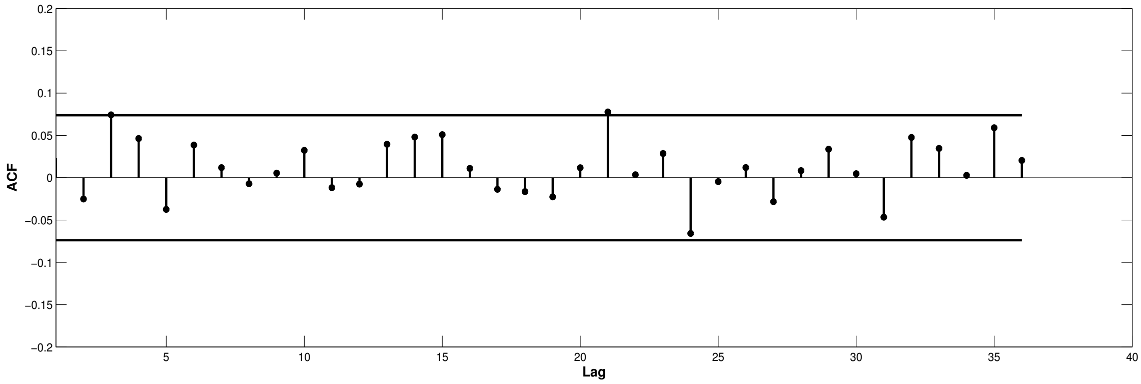


Figure 4.9: Autocorrelation function (ACF) of the residuals of fitted PAR model with one changepoint to the Saugeen flow.

Saugeen in the 1950s: main reason of SVCA creation in 1950 was, indeed, flood control management. Walkerton business district, which is the gauging station, has been subject to major floods in early and mid 1900. This has led to the construction, starting from 1956, of 2.4 km of dykes and floodwalls to protect the central business district as well as residential neighborhoods from potential floods.

Concerning estimation and forecasting, the best fitness is obtained for model with two changepoints, and forecasting accuracy is found best for model with single changepoint considering all measures. Figure 4.8 plots time series with this changepoint. We shall also perform some comparison with results of literature. Wong et al. (2007) proposed a functional-coefficient autoregression (NFCAR) model in order to estimate and forecast monthly flows of Saugeen. Forecasting performance, measured on natural data, have been compared to PAR(1) model results by Noakes et al. (1985), resulting in an improvement in terms of MAE from 10.8986 to 10.3689. Corresponding value of our model $PAR_{(1;1;7)}$ (with one changepoint) computed on natural data is 9.4827, which further improves performance of both standard PAR(1) model and NFCAR. Residual analysis, shown in Figure 4.9, confirms adequacy of our model.

4.5 Concluding remarks

This chapter proposed a method to account for seasonality and structural changes in time series by employing PAR models linked at different changepoints. GA based identification showed promising results on both simulations and real data. Application of procedure on river flows data of Garonne (France) and Saugeen (Canada), for which changepoints could be possibly due to both human activities and climatic oscillations, proved also good performances in terms of forecasting, a highly demanding issue in hydrology.

In our study we examined monthly data with changepoints allowed only at the end of the year (that is, a multiple of number of seasons). Modifications of the method proposed in the present paper are under study: techniques for monthly, weekly or daily time series with periodic structure allowing changepoints at any season are worth pursuing. In fact, detecting a changepoint in the middle of a year will prevent dispersing its effects over adjacent seasons. Moreover, as far as PAR models are based on a large number of parameters, one could question on whether it is necessary to consider a separate AR model for each season: we allowed to build subset PAR models in order to conveniently decrease number of parameters, but a considerable gain in parsimony would be achieved by reducing number of seasons in PAR model (Hipel & McLeod, 1994 and Franses & Paap, 2004 proposed several hypothesis tests). Lastly, it is known that a stationary autoregressive process has a short memory (Brockwell & Davis, 1991; Robinson, 2003). Time series which exhibit long range dependence are characterized by autocorrelations which decays very slowly, while a stationary autoregressive process have rapidly decaying autocorrelations. Focusing on our case study, hydrological data generally exhibit structural changes and long range dependence (Song & Bondon, 2013). Therefore long memory process with periodic structure could be appropriate for hydrological data.

Conclusions

In this thesis we analyzed a selection of statistical inference problems employing Evolutionary Algorithms (EAs) as computational tool. In this field they are considered a non-standard procedure, so their behaviour is not generally well understood and there is lack of an established theoretical background. In the course of dissertation we studied EAs from different statistical points of view, making our contributions on the state-of-art many-sided.

Chapter 2 was concerned with model parametric estimation by EAs, from a classical inference point of view. In fact we analyzed the behaviour of EA-based estimators by evaluating their variability and asymptotic efficiency, as usually done in classical inference theory. The non-standard element is that we consider the EA as a random variable in the analysis, which introduces a further source of variability. The statistical and computational tradeoff question allows to set the analysis in realistic situations, which have become crucial as long as size of datasets is dramatically increasing. Our analysis is not restricted to EAs but is valid also for any stochastic algorithm having property of global convergence, so natural future contributions would be devoted to generalize this procedure to other algorithms, maybe also related to an evolutionary behaviour (such as Particle Swarm Optimization). In addiction, our method could be improved by summarizing the covariance matrices in other possible ways (we considered trace of covariance matrix, but other choices, like the determinant for example, are plausible).

In Chapter 3 an overview on algorithms that conjugates EC philosophy and Markov Chain Monte Carlo (MCMC) methodology has been given. Although MCMC is a general procedure, as statisticians we can set the problem in a Bayesian inference framework, where problems of sampling from complex distributions are crucial. Contributions reviewed in the course of chapter have introduced many EAs with many different strategies for sampling from complex target distributions is on the agenda. They have been proposed in different fields of science, sometimes independently on each other: we analyzed them from an EC prospective, trying to unify them in a common framework and highlighting the strengths and weaknesses. Future work is related to adaptive MCMC strategies, which have already been proven to be effective by some authors of our review, and could decisively improve EC based MCMC methods on both the computational and efficiency side.

Chapter 4 focused on time series analysis. GAs have been employed for building a complex model, which account for both seasonality, by use of PAR models, and regime changes. Proposed methodology has been proven to be effective in capturing both of these features in data, as shown in simulations and river flow data. As the procedure seems promising it can be naturally improved: we assumed that structural changes could fall only at the end of the year, but it would be worth pursuing to let it occur at any season of the year, as it would be also prevent dispersing its effects over adjacent seasons. Also a considerable gain in parsimony would be achieved by reducing the number of seasons in PAR models, because they are possibly not all essential. Lastly, beside hydrology, this kind of model could be successfully applied in many other fields, like climatology (there are already some papers dealing with periodic modeling and structural breaks detection) or also finance.

In conclusion, we truly hope that the topics proposed and analyzed in this work, including discussions of literature, may stimulate new ideas of research.

Acknowledgments

Prima di far calare il sipario, qualche pensiero sparso qua e là.

Il primo non può andare ad altri che a Francesco Battaglia. Fin dai tempi della scelta del relatore per la tesi di laurea Francesco ha saputo guidarmi e consigliarmi con saggezza, empatia e inesauribile pazienza (fidatevi, ce ne voleva tanta). Se in questi anni sono riuscito a intraprendere e portare a termine un percorso come quello di un dottorato di ricerca, pieno di scelte e momenti difficili, ma che mi ha portato tante soddisfazioni, lo devo soprattutto a lui. Ringrazio poi tutti gli amici e colleghi di ben sette cicli di dottorato, la cui compagnia ha reso bellissimo questo percorso, in particolare Luca, Dox, Marco e Alessia. Allo stesso modo ringrazio Domenico, anche per il suo supporto morale, e i tanti assegnisti, ricercatori e professori che ho incontrato durante la mia permanenza al Dipartimento di Scienze Statistiche da studente e dottorando, per essere stati dei punti di riferimento sia come insegnanti che come colleghi. I also want to thank Sandra Paterlini to welcome me at EBS University in Wiesbaden for my Ph.D visiting period, making my first stay abroad a comfortable one (it was not trivial for me). For the same reason I thank and say hi to my EBS office colleagues Margherita, Philipp, Wenwei, Nicola, Louis and Max. In conclusione ringrazio la mia famiglia per il supporto che mi ha sempre dato nelle mie scelte e che mi ha permesso di andare avanti anche in questa sfida.

Bibliography

Abo-Hammour, Z.E.S., Alsmadi, O.M., Al-Smadi, A.M., Zaqout, M.I., Saraireh, M.S. (2012). ARMA model order and parameter estimation using genetic algorithms. *Mathematical and Computer Modelling of Dynamical Systems*, 18(2), 201-221.

Agarwal, A. (2012). *Computational Trade-offs in Statistical Learning.* Ph.D Thesis, University of California, Berkeley.

Allingham, D., King, R.A.R., Mengersen, K.L. (2009). Bayesian estimation of quantile distributions. *Statistics and Computing*, 19(2), 189-201.

Aly, W.M. (2016). A new approach for classifier model selection and tuning using logistic regression and genetic algorithms. *Arabian Journal for Science and Engineering*, 41(12), 5195-5204.

Angelis, L. (2003). An evolutionary algorithm for A-optimal incomplete block designs. *Journal of Statistical Computation and Simulation*, 73(10), 753-771.

Auger, A., Doerr, B. (eds) (2011). *Theory of Randomized Search Heuristics - Foundations and Recent Developments.* World Scientific.

Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms.* Oxford: Oxford University Press.

Bai, J., Perron, P. (1998). Estimating and testing linear models with multiple structural changes. *Econometrica*, 47-78.

Bandyopadhyay, S., Maulik, U. (2002). Genetic clustering for automatic evolution of clusters and application to image classification. *Pattern recognition*, 35(6), 1197-1208.

Bandyopadhyay, S., Maulik, U., Baragona, R. (2010). Clustering multivariate time series by genetic multiobjective optimization. *Metron*, 68(2), 161-183.

Bandyopadhyay, S., Maulik, U., Mukhopadhyay, A. (2007). Multiobjective genetic clustering for pixel classification in remote sensing imagery. *IEEE transactions on Geoscience and Remote Sensing*, 45(5), 1506-1511.

Bandyopadhyay, S., Murthy, C.A., Pal, S.K. (1995). Pattern classification with genetic algorithms. *Pattern recognition letters*, 16(8), 801-808.

Bandyopadhyay, S., Saha, S. (2007). GAPS: A clustering method using a new point symmetry-based distance measure. *Pattern recognition*, 40(12), 3430-3451.

Baragona, R., Battaglia, F. (2003). Multivariate mixture models estimation: a genetic algorithm approach. In M. Schader, W. Gaul, M. Vichi (eds) *Between Data Science and Applied Data Analysis*, 133-142, Springer.

Baragona, R., Battaglia, F. (2006). Genetic algorithms for building double threshold generalized autoregressive conditional heteroscedastic models of time series. In: A. Rizzi, M. Vichi (eds) *Compstat 2006 - Proceedings in Computational Statistics*, 441-452. Physica-Verlag HD.

Baragona, R., Battaglia, F., Calzini, C. (2001a). Clustering of time series with genetic algorithms. *Metron*, 59(1), 113-130.

Baragona, R., Battaglia, F., Calzini, C. (2001b). Genetic algorithms for the identification of additive and innovation outliers in time series. *Computational Statistics & Data Analysis*, 37(1), 1-12.

Baragona, R., Battaglia, F., Cucina, D. (2002). A note on estimating autoregressive exponential models. *Quaderni di Statistica*, 4(1), 71-88.

Baragona, R., Battaglia, F., Cucina, D. (2004a). Estimating threshold subset autoregressive moving-average models by genetic algorithms. Metron, 62(1), 39-61.

Baragona, R., Battaglia, F., Cucina, D. (2004b). Fitting piecewise linear threshold autoregressive models by means of genetic algorithms. *Computational Statistics & Data Analysis*, 47(2), 277-295.

Baragona, R., Battaglia, F., Poli, I. (2011). *Evolutionary Statistical Procedures - An Evolutionary Computation Approach to Statistical Procedures Design and Applications*. Berlin: Springer-Verlag.

Baragona, R., Bocci, L., Medaglia, C.M. (2006). Genetic clustering algorithms: A comparison simulation study. *International Journal of Modelling and Simulation*, 26(3), 190-200.

Baragona, R., Cucina, D. (2008). Double threshold autoregressive conditionally heteroscedastic model building by genetic algorithms. *Journal of Statistical Computation and Simulation*, 78(6), 541-558.

Baragona, R., Cucina, D. (2013). Multivariate Self-Exciting Threshold Autoregressive Modeling by Genetic Algorithms. *Jahrbücher für Nationalökonomie und Statistik*, 233(1), 3-21.

Basse, G., Smith, A., Pillai, N. (2016). Parallel Markov Chain Monte Carlo via Spectral Clustering. *Artificial Intelligence and Statistics*, 1318-1327.

Battaglia, F. (2001). Genetic algorithms, pseudo-random numbers generators, and Markov chain Monte Carlo methods. *Metron*, 59(1-2), 131-155

Battaglia, F., Protopapas, M.K. (2011). Time-varying multi-regime models fitting by genetic algorithms. *Journal of Time Series Analysis*, 32(3), 237-252.

Battaglia, F., Protopapas, M.K. (2012). Multi-regime models for nonlinear nonstationary time series. *Computational Statistics*, 27(2), 319-341.

Bendtsen, C. (2012). pso: Particle Swarm Optimization. R package version 1.0.3. URL: http://CRAN.R-project.org/package=pso.

Bergmeir, C., Molina, D., Benitez, J.M. (2016). Memetic Algorithms with Local Search Chains in R: The Rmalschains Package. *Journal of Statistical Software*, 75(4), 1-33. doi:10.18637/jss.v075.i04.

Berthet, Q., Chandrasekaran, V. (2016). Resource Allocation for Statistical Estimation. In *Proceedings of the IEEE*, 104(1), 111-125.

Beyer, H.G., Schwefel, H.P. (2002). Evolution strategies - A comprehensive introduction. *Natural computing*, 1(1), 3-52.

Bloomfield, P., Steiger, W.L. (1983). *Least absolute deviations: Theory, applications and algorithms.* Boston: Birkhäuser.

Brigode, P., Oudin, L., Perrin, C. (2013). Hydrological model parameter instability: A source of additional uncertainty in estimating the hydrological impacts of climate change?. *Journal of Hydrology*, 476, 410-425.

Brockwell, P.J., Davis, R.A. (1991). *Time series: theory and methods.* New York: Springer.

Broudiscou, A., Leardi, R., Phan-Tan-Luu, R. (1996). Genetic algorithm as a tool for selection of D-optimal design. *Chemometrics and intelligent laboratory systems*, 35(1), 105-116.

Bruer, J.J., Tropp, J.A., Cevher, V., Becker, S.R. (2013). Designing Statistical Estimators That Balance Sample Size, Risk, and Computational Cost. *IEEE Journal of Selected Topics in Signal Processing*, 9(4), 612-624.

Brulebois, E., Castel, T., Richard, Y., Chateau-Smith, C., Amiotte-Suchet, P. (2015). Hydrological response to an abrupt shift in surface air temperature over France in 1987/88. *Journal of Hydrology*, 531, 892-901.

Buishand, T.A., 1984. Tests for detecting a shift in the mean of hydro- logical time series. *Journal of Hydrology* 75, 51-69.

Caballero, Y., Voirin-Morel, S., Habets, F., Noilhan, J., LeMoigne, P., Lehenaff, A., Boone, A. (2007). Hydrological sensitivity of the Adour-Garonne river basin to climate change. *Water Resources Research*, 43(7).

Cantú-Paz, E. (1998). A survey of parallel genetic algorithms. *Calculateurs paralleles, reseaux et systems repartis*, 10(2), 141-171.

Chandrasekaran, V., Jordan, M.I. (2013). Computational and statistical tradeoffs via convex relaxation. In *Proceedings of the National Academy of Sciences*, 110(13), E1181-E1190.

Chatterjee, S., Laudato, M., Lynch, L. A. (1996). Genetic algorithms and their statistical applications: an introduction. *Computational Statistics & Data Analysis*, 22(6), 633-651.

Chen, C.W., Cherng, T.H., Wu, B. (2001). On the selection of subset bilinear time series models: a genetic algorithm approach. *Computational Statistics*, 16(4), 505-517.

Chen, Y., Xu, J. (2016). Statistical-Computational Tradeoffs in Planted Problems and Submatrix Localization with a Growing Number of Clusters and Submatrices. *Journal of Machine Learning Research*, 17(27), 1-57.

Choi, Y.S., Moon, B.R. (2007). Feature selection in genetic fuzzy discretization for the pattern classification problems. *IEICE transactions on information and systems*, 90(7), 1047-1054.

Cipra, T. (1985). Periodic moving average process. *Aplikace matematiky*, 30(3), 218-229.

Clayden, J. (2014). soma: General-Purpose Optimisation With the Self-Organising Migrating Algorithm. R package version 1.1.1. URL: http://CRAN.R-project.org/package=soma.

Clerc, M. (2015). *Guided Randomness in Optimization.* Wiley.

Cobb, G.W. (1978). The problem of the Nile: conditional solution to a change-point problem. *Biometrika*, 65(2), 243-251.

Cucina, D., Di Salvatore, A., Protopapas, M.K. (2014). Outliers detection in multivariate time series using genetic algorithms. *Chemometrics and Intelligent Laboratory Systems*, 132, 103-110.

Davis, L. (1991). *The Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold.

Davis, R.A., Lee, T.C.M., Rodriguez-Yam, G.A. (2006). Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 101(473), 223-239.

De Jong, K.A. (1993). Genetic algorithms are NOT function optimizers. *Foundations of genetic algorithms*, 2, 5-17.

De Jong, K.A. (2006). *Evolutionary computation: a unified approach.* Cambridge: MIT press.

De March, D., Forlin, M., Slanzi, D., Poli, I. (2009). An evolutionary predictive approach to design high dimensional experiments. In R. Serra, I. Poli, M. Villani (eds) *Artificial life and evolutionary computation: proceedings of WIVACE 2008*, 81-88. World Scientific Publishing Company, Singapore

Derrac, J., Garcìa, S., Hui, S., Suganthan, P.N., Herrera, F. (2014). Analyzing convergence performance of evolutionary algorithms: A statistical approach. *Information Sciences*, 289, 41-58.

Dillon, J.V., Lebanon, G (2010). Stochastic Composite Likelihood. *Journal of Machine Learning Research*, 11, 2597-2633.

Doerr, B., Fischer, P., Hilbert, A., Witt, C. (2017). Detecting structural breaks in time series via genetic algorithms. *Soft Computing*, 21(16), 4707-4720.

Drugan, M.M., Thierens, D. (2004). Evolutionary markov chain monte carlo. In P. Collet, E. Lutton, M. Schoenauer, P. Liardet, C. Fonlupt (eds) *International Conference on Artificial Evolution (Evolution Artificielle)*, 63-76, Springer Berlin Heidelberg.

Drugan, M.M., Thierens, D. (2005). Recombinative EMCMC algorithms. In *Proceedings of IEEE Congress on Evolutionary Computation, CEC'05*, 2024-2031, IEEE Press, Piscataway.

Drugan, M.M., Thierens, D. (2010a). Geometrical recombination operators for real-coded evolutionary mcmcs. *Evolutionary computation*, 18(2), 157-198.

Drugan, M.M., Thierens, D. (2010b). Recombination operators and selection strategies for evolutionary Markov Chain Monte Carlo algorithms. *Evolutionary intelligence*, 3(2), 79-101.

Eiben, A.E., Hinterding, R., Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on evolutionary computation*, 3(2), 124-141.

Eiben, A.E., Smit, S.K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1(1), 19-31.

Eiben, A.E., Smith, J.E. (2003). *Introduction to evolutionary computing.* Heidelberg: Springer.

Falkenauer, E. (1998). *Genetic algorithms and grouping problems.* New York: Wiley.

Fogel, D.B. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence.* Piscataway, NJ: IEEE Press.

Fogel, D.B. (1998). *Evolutionary Computation: The Fossil Record.* Piscataway, NJ: IEEE Press.

Fogel, L.J., Owens, A.J., Walsh, M.J. (1966). *Artificial intelligence through simulated evolution.* New York: Wiley.

Forlin, M., De March, D., Poli, I. (2007). The model-based genetic algorithms for designing mixture experiments. Working paper 18, European centre for living technology, Venice.

Franconi, L., Jennison, C. (1997). Comparison of a genetic algorithm and simulated annealing in an application to statistical image reconstruction. *Statistics and Computing*, 7(3), 193-207.

Franses, P.H. (1994). A multivariate approach to modeling univariate seasonal time series. *Journal of Econometrics*, 63(1), 133-151.

Franses, P.H., Paap, R. (2004). *Periodic time series models.* Oxford: Oxford University Press.

Gaetan, C. (2000). Subset ARMA model identification using genetic algorithms. *Journal of Time Series Analysis*, 21(5), 559-570.

Gelman, A., Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 457-472.

Geyer, C.J. (1991). Markov Chain Monte Carlo maximum likelihood. In E.M. Keramidas (ed) *Computing Science and Statistics, Proceedings of the 23rd Symposium on the Interface*, 156-163, Interface Foundation of North America.

Geyer, C.J., Thompson, E.A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference, *Journal of American Statistical Association*, 21, 303-311.

Gilks, W.R., Roberts, G.O. (1996). Strategies for improving MCMC. In W.R. Gilks, S. Richardson, D. Spiegelharter (eds) *Markov chain Monte Carlo in practice*, 6, 89-114

Gilks, W.R., Roberts, G.O., George, E.I. (1994). Adaptive direction sampling. *The statistician*, 179-189

Gladyshev, E.G. (1961). Periodically correlated random sequence. *Sovietic Mathematics*, 385-388.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* New York: Addison-Wesley.

Goldberg, D.E. (1991). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex systems*, 5(2), 139-167.

Gómez, M., Bielza, C. (2004). Node deletion sequences in influence diagrams using genetic algorithms. *Statistics and Computing*, 14(3), 181-198.

Gonzalez-Fernandez, Y., Soto, M. (2012). copulaedas: An r package for estimation of distribution algorithms based on copulas. *Journal of Statistical Software*, 58(9). URL: http://www.jstatsoft.org/v58/i09/.

Goodman, J., Sokal, A.D. (1989). Multigrid monte carlo method. conceptual foundations. *Physical Review D*, 40(6), 2035.

Goswami, G. (2011). EMC: Evolutionary Monte Carlo (EMC) algorithm. R package version. 1.3. http://CRAN.R-project.org/package=EMC

Goswami, G., Liu, J.S. (2007). On learning strategies for evolutionary Monte Carlo. *Statistics and Computing*, 17(1), 23-38.

Goswami, G., Liu, J.S., Wong, W.H. (2007). Evolutionary Monte Carlo methods for clustering, *Journal of Computational and Graphical Statistics*, 16(4), 855-876.

Govaerts, B., Sanchez, R.P. (1992). Construction of exact D-optimal designs for linear regression models using genetic algorithms. *Belgian Journal of Operations Research, Statistics and Computer Science*, 32(1), 153-174.

Grazian, C., Liseo, B. (2015). Approximated Integrated Likelihood via ABC methods. *Statistics and Its Interface*, 8(2), 161-171.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4), 711-732.

Guillaume, J., Andrews, F. (2012). dream: DiffeRential Evolution Adaptive Metropolis. R package version 0.4-2. URL http://CRAN.R-project.org/package=dream

Gupta, M. (2014). An evolutionary Monte Carlo algorithm for Bayesian block clustering of data matrices, *Computational Statistics & Data Analysis*, 71, 375-391.

Haario, H, Laine, M., Mira, A., Sakman, E. (2006). DRAM: Efficient adaptive MCMC. *Statistics and Computing*, 16, 339-354.

Hannan, E.J. (1980). The estimation of the order of an ARMA process. *Annals of Statistics*, 8(5), 1071-1080.

Hatjimihail, A.T., Hatjimihail, T.T. (2002). Design of statistical quality control procedures using genetic algorithms. arXiv: cs/0201024.

Haynes, M.A., Gatton, M.L., Mengersen, K.L. (1997). Generalized control charts for nonnormal data. Technical Report No. 97/4, School of Mathematical Sciences, Queensland University of Technology, Brisbane, Australia.

Haynes, M.A., Mengersen, K.L. (2005). Bayesian Estimation of g-and-k Distributions using MCMC. *Computational Statistics*, 20(1), 7-30.

Haynes, M.A., Mengersen, K.L., Rippon, P. (2008). Generalized Control Charts for Non-Normal Data Using g-and-k Distributions. *Communication in Statistics - Simulation and Computation*, 37(9), 1881-1903.

Herrera, F., Lozano, M., Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioral analysis. *Artificial intelligence review*, 12(4), 265-319.

Higuchi, T. (1997). Monte Carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation*, 59(1), 1-23.

Hipel, K.W., McLeod, A.I. (1994). *Time series modelling of water resources and environmental systems (Vol. 45).* Amsterdam: Elsevier.

Holland, J.H. (1967). Nonlinear environments permitting efficient adaptation. In *Computer and Information Sciences II*, Academic Press.

Holland, J.H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* Ann Arbor, MI: University of Michigan Press.

Holloman, C.H., Lee, H.K., & Higdon, D.M. (2006). Multiresolution genetic algorithms and Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 15(4), 861-879.

Holmes, C.C., Mallick, N.K. (1998). Parallel Markov chain Monte Carlo sampling: an evolutionary based approach. Technical Report, Imperial College, London.

Hyndman, R.J., Koehler, A.B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4), 679-688.

Hruschka, E.R., Ebecken, N.F. (2003). A genetic algorithm for cluster analysis. *Intelligent Data Analysis*, 7(1), 15-25.

Hu, B., Tsui, K.W. (2010). Distributed evolutionary Monte Carlo for bayesian computing. *Computational Statistics & Data Analysis*, 54(3), 688-697.

Hu, Z., Xiong, S., Su, Q., Zhang, X. (2013). Sufficient conditions for global convergence of differential evolution algorithm. *Journal of Applied Mathematics*, 2013.

Hukushima, K., Nemoto, K. (1996). Exchange Monte Carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan*, 65(6), 1604-1608.

Jank, W. (2006). The EM algorithm, its randomized implementation and global optimization: Some challenges and opportunities for operations research. In F.B. Alt, M.C. Fu, B.L. Golden (eds) *Perspectives in operations research*, 367-392.

Jasra, A., Stephens, D.A., Holmes, C.C. (2007). On population-based simulation for static inference. *Statistics and Computing*, 17(3), 263-279.

Jeong, C., Kim, J. (2013). Bayesian multiple structural change-points estimation in time series models with genetic algorithm. *Journal of the Korean Statistical Society*, 42(4), 459-468.

Jones, R.H., Brelsford, W.M. (1967). Time series with periodic structure. *Biometrika*, 54(3-4), 403-408.

Jordan, M.I. (2013). On statistics, computation and scalability. *Bernoulli*, 19(4), 1378-1390.

Jung, H., Marjoram, P. (2011). Choice of summary statistic weights in approximate Bayesian computation. *Statistical applications in genetics and molecular biology*, 10(1).

Kapanoglu, M., Ozan Koc, I., Erdogmus, S. (2007). Genetic algorithms in parameter estimation for nonlinear regression models: an experimental approach. *Journal of Statistical Computation and Simulation*, 77(10), 851-867.

Kapetanios, G. (2007). Variable selection in regression models using nonstandard optimisation of information criteria. *Computational Statistics & Data Analysis*, 52(1), 4-15.

Karavas, V.N., Moffitt, L.J. (2004). Evolutionary computation of a deterministic switching regressions estimator. *Computational Statistics*, 19(2), 211-225.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

Kitagawa, G., Akaike, H. (1978). A procedure for the modeling of non-stationary time series. *Annals of the Institute of Statistical Mathematics*, 30(1), 351-363.

Knobloch, R., Mlýnek, J., Srb, R. (2017). The classic differential evolution algorithm and its convergence properties. *Applications of Mathematics*, 62(2), 197-208.

Kuncheva, L.I. (1995). Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters*, 16(8), 809-814.

Kwok, N.M., Fang, G., Zhou, W. (2005). Evolutionary particle filter: re-sampling from the genetic algorithm perspective. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference of*, 2935-2940. IEEE.

Laloy, E., Vrugt, J.A. (2012). High-dimensional posterior exploration of hydrologic models using multiple-try DREAM (ZS) and high-performance computing. *Water Resources Research*, 48(1).

Larrañaga, P., Kuijpers, C.M., Poza, M., Murga, R.H. (1997). Decomposing Bayesian networks: triangulation of the moral graph with genetic algorithms. *Statistics and Computing*, 7(1), 19-34.

Larrañaga, P., Lozano, J.A. (eds) (2001). *Estimation of distribution algorithms: A new tool for evolutionary computation (Vol. 2).* Boston, MA: Kluwer Academic Publisher.

Laskey, K. B., Myers, J. W. (2003). Population markov chain monte carlo. *Machine Learning*, 50(1), 175-196.

Liang, F., Liu, C., Carroll, R. (2011). Advanced Markov chain Monte Carlo methods: learning from past samples. Vol.714. John Wiley & Sons.

Liang, F., Wong, W.H. (2000). Evolutionary Monte Carlo: Applications to C p model sampling and change point problem. *Statistica sinica*, 10(2) 317-342.

Liang, F., Wong, W.H. (2001a). Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models. *Journal of the American Statistical Association*, 96(454), 653-666

Liang, F., Wong, W.H. (2001b). Evolutionary Monte Carlo for protein folding simulations. *The Journal of Chemical Physics*, 115(7), 3374-3380

Lin, C.D., Anderson-Cook, C.M., Hamada, M.S., Moore, L.M., Sitter, R.R. (2015). Using genetic algorithms to design experiments: a review. *Quality and Reliability Engineering International*, 31(2), 155-167.

Liu, J.S., & Sabatti, C. (1999). Simulated sintering: Markov chain Monte Carlo with spaces of varying dimensions. In J.M. Bernardo, J.O. Berger, A.P. Dawid & A.F.M. Smith (eds) *Bayesian Statistics 6*, 389-413.

Lobo, F.J., Lima, C.F., Michalewicz, Z. (eds) (2007). *Parameter setting in evolutionary algorithms (Vol. 54).* Springer Science & Business Media.

Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds) (2006). *Towards a new evolutionary computation: advances on estimation of distribution algorithms (Vol. 192).* New York: Springer Science & Business Media.

Lu, Q., Lund, R. (2007). Simple linear regression with multiple level shifts. *Canadian Journal of Statistics*, 35(3), 447-458.

Lu, Q., Lund, R., Lee, T.C. (2010). An MDL approach to the climate segmentation problem. *The Annals of Applied Statistics*, 4(1), 299-319.

Lund, R., Wang, X.L., Lu, Q.Q., Reeves, J., Gallagher, C., Feng, Y. (2007). Changepoint detection in periodic and autocorrelated time series. *Journal of Climate*, 20(20), 5178-5190.

Marinari, E., Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme, *Europhysics Letters*, 19, 451.

Maulik, U., Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern recognition*, 33(9), 1455-1465.

Maulik, U., Bandyopadhyay, S. (2003). Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. *IEEE Transactions on geoscience and remote sensing*, 41(5), 1075-1081.

McLeod, A.I. (1994). Diagnostic checking of periodic autoregression. *Journal of Time Series Analysis*, 15(2), 221-223.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs.* New York: Springer-Verlag.

Milgo, E., Ronoh, N., Waiganjo, P., Manderick, B. (2017). Comparison of Adaptive MCMC Samplers. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*

Minerva, T., Paterlini, S. (2002). Evolutionary approaches for statistical modelling. In D.B. Fogel, M.A. El-Sharkam, G. Yao, H. Greenwood, P. Iba, P. Marrow, M. Shakleton (eds) *P Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, Vol.2, 2023-2028. Piscataway, NJ: IEEE Press.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms.* Cambridge, MA: MIT Press.

Mühlenbein, H., Mahnig, T. (1999). Convergence theory and applications of the factorized distribution algorithm. *CIT. Journal of computing and information technology*, 7(1), 19-32.

Mühlenbein, H., Paass, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters. In W. Ebeling, I. Rechenberg, H.O. Schwefel, H.M. Voigt (eds) *Parallel Problem Solving from Nature - PPSN IV*, 178-187.

Mühlenbein, H., Mahnig, T., Rodriguez, A.O. (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2), 215-247.

Mullen, K., Ardia, D., Gil, D., Windover, D., Cline, J. (2011). 'DEoptim': An R Package for Global Optimization by Differential Evolution. *Journal of Statistical Software*, 40(6), 1-26. URL: http://www.jstatsoft.org/v40/i06/.

Murthy, C.A., Chowdhury, N. (1996). In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters*, 17(8), 825-832.

Mutingi, M., Mbohwa, C. (2017). Fuzzy Grouping Genetic Algorithms: Advances for Real-World Grouping Problems. In *Grouping Genetic Algorithms. Studies in Computational Intelligence*, 666, 67-86. Springer International Publishing.

Niermann, S. (2006). Evolutionary estimation of parameters of Johnson distributions. *Journal of Statistical Computation and Simulation*, 76(3), 185-193.

Ninomiya, Y. (2015). Change-point model selection via AIC. *Annals of the Institute of Statistical Mathematics*, 67(5), 943-961.

Noakes, D.J., McLeod, A.I., Hipel, K.W. (1985). Forecasting monthly riverflow time series. *International Journal of Forecasting*, 1(2), 179-190.

Nunkesser, R., Morell, O. (2010). An evolutionary algorithm for robust regression. *Computational Statistics & Data Analysis*, 54(12), 3242-3248.

Oliveto, P.S., Witt, C. (2014). On the runtime analysis of the Simple Genetic Algorithm. *Theoretical Computer Science*, 545, 2-19.

Ong, C.S., Huang, J.J., & Tzeng, G.H. (2005). Model identification of ARIMA family using genetic algorithms. *Applied Mathematics and Computation*, 164(3), 885-912.

Pal, M., Saha, S., Bandyopadhyay, S. (2018). DECOR: Differential Evolution using Clustering based Objective Reduction for Many-Objective Optimization. *Information Sciences*, 423, 200-218.

Palit, A.K., Popovic, D. (2005). *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications*. London: Springer Science & Business Media.

Pasia, J.M., Hermosilla, A.Y., Ombao, H. (2005). A useful tool for statistical estimation: genetic algorithms. *Journal of Statistical Computation and Simulation*, 75(4), 237-251.

Paterlini, S., Krink, T. (2006). Differential evolution and particle swarm optimisation in partitional clustering. *Computational statistics & data analysis*, 50(5), 1220-1247.

Paterlini, S., Minerva, T. (2003). Evolutionary approaches for cluster analysis. In A. Bonarini, F. Masulli, G. Pasi (eds) *Soft Computing Applications. Advances in Soft Computing*, vol 18. Physica, Heidelberg.

Pelikan, M. (2005). *Hierarchical Bayesian optimization algorithm : toward a new generation of evolutionary algorithms*. Springer-Verlag.

Pelikan, M., Goldberg, D.E., Lobo, F.G. (2002). A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, 21(1), 5-20.

Pereira, A.G., Campos, V.S. (2016). Multistage non homogeneous Markov chain modeling of the non homogeneous genetic algorithm and convergence results. *Communications in Statistics - Theory and Methods*, 45(6), 1794-1804.

Pereira, A.G., de Andrade, B.B. (2015). On the genetic algorithm with adaptive mutation rate and selected statistical applications. *Computational Statistics*, 30(1), 131-150.

Peters, G.W., Chen, W., Gerlach, R.H. (2016). Estimating Quantile Families of Loss Distributions for Non-Life Insurance Modelling via L-Moments. *Risks*, 4(2), 14.

Poli, I. (2006). Evolutionary design of experiments. Working paper 18, European Centre for Living Ttechnology, Venice, PACE Report.

Prangle, D. (2017). gk: g-and-k and g-and-h Distribution Functions. R package version 0.5.0. http://CRAN.R-project.org/package=gk.

Price, K., Storn, R.M., Lampinen, J.A. (2006). *Differential evolution: a practical approach to global optimization.* Berlin: Springer Science & Business Media.

Prügel-Bennett, A., Rogers, A. (2001). Modelling genetic algorithm dynamics. In L. Kallel, B. Naudts, A. Rogers (eds) *Theoretical aspects of Evolutionary Computing* (pp. 59-58). Berlin: Springer-Verlag.

R Core Team (2013) R: A language and environment for statistical computing. Vienna: R Foundation for Statistical Computing. URL: http://www.R-project.org/

Raghavan, V.V., Birchard, K. (1979). A clustering strategy based on a formalism of the reproductive process in natural systems. In *Proceedings of the Second International Conference on Information Storage and Retrieval*, 10-22.

Rayner, G.D., MacGillivray, H.L. (2002). Numerical maximum likelihood estimation for the g-and-k and generalized g-and-h distributions. *Statistics and Computing*, 12(1), 57-75.

Reeves, C.R., Rowe, J.E. (2003). *Genetic algorithms - Principles and perspectives - A guide to GA theory.* London: Kluwer Academic Publishers.

Ren, Y., Ding, Y., Liang, F. (2008). Adaptive evolutionary Monte Carlo algorithm for optimization with applications to sensor placement problems, *Statistics and Computing*, 18(4), 375-390.

Rizzo, M., Battaglia, F. (2016). On the Choice of a Genetic Algorithm for Estimating GARCH Models. *Computational Economics*, 48(3), 473-485.

Robert, C., Casella, G. (2004). *Monte Carlo Statistical Methods.* New York: Springer-Verlag.

Roberts, G.O., Gilks, W.R. (1994). Convergence of adaptive direction sampling, *Journal of multivariate analysis*, 49(2), 287-298.

Robinson, P.M. (ed) (2003). *Time series with long memory.* Advanced Texts in Econometrics. Oxford University Press.

Rojas Cruz, J.A., Pereira, A.G.C. (2013). The elitist non-homogeneous genetic algorithm: Almost sure convergence. *Statistics & Probability Letters*, 83(10), 2179-2185.

Roverato, A., & Poli, I. (1998). A genetic algorithm for graphical model selection. *Journal of the Italian Statistical Society*, 7, 197-208.

Rudolph, G. (1997). *Convergence Properties of Evolutionary Algorithms.* Hamburg: Verlag Dr. Kovac.

Saha, S. (2017). Enhancing point symmetry-based distance for data clustering. *Soft Computing*, 1-28.

Saha, S., Bandyopadhyay, S. (2009). A new point symmetry based fuzzy genetic clustering technique for automatic evolution of clusters. *Information Sciences*, 179(19), 3230-3246.

Saha, S., Bandyopadhyay, S. (2013). A generalized automatic clustering algorithm in a multiobjective framework. *Applied Soft Computing*, 13(1), 89-108.

Santamaría-Bonfil, G., Frausto-Solís, J., Vázquez-Rodarte, I. (2015). Volatility forecasting using support vector regression and a hybrid genetic algorithm. *Computational Economics*, 45(1), 111-133.

Schwefel, H.P. (1975). *Evolutionsstrategie und numerische Optimierung.* Ph.D Thesis, Technische Universität Berlin.

Scrucca, L. (2013). GA: A Package for Genetic Algorithms in R. *Journal of Statistical Software*, 53(4), 1-37. URL: http://www.jstatsoft.org/v53/i04/.

Shapiro, J.L. (2001). Statistical mechanics theory of genetic algorithms. In L. Kallel, B. Naudts, A. Rogers (eds) *Theoretical aspects of Evolutionary Computing* (pp. 87-108). Berlin: Springer-Verlag.

Shender, D., Lafferty, J. (2013). Computation-Risk Tradeoffs for Covariance-thresholded Regression. In *Proceedings of The 30th International Conference on Machine Learning*, 756-764.

Slanzi, D., De March, D., Poli, I. (2009). Evolutionary probabilistic graphical models in high dimensional data analysis. In F. Mola, C. Conversano, V. Vinzi, N. Fisher (eds) *European regional meeting of the international society for business and industrial statistics*, 124-125. Cagliari, TAPILA editore.

Slanzi, D., Poli, I. (2014). Evolutionary Bayesian network design for high dimensional experiments. *Chemometrics and Intelligent Laboratory Systems*, 135, 172-182.

Song, L., Bondon, P., 2013. Structural changes estimation for strongly- dependent processes. *Journal of Statistical Computation and Simulation* 83, 1783?1806

Storn, R., Price, K. (1997). Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.

Strens, M.J. (2003). Evolutionary MCMC sampling and optimization in discrete spaces. In T. Fawcett, N. Mishra (eds) *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, 736-743, AAAI Press, Menlo Park.

Strens, M.J., Bernhardt, M., Everett, N. (2002). Markov chain Monte Carlo sampling using direct search optimization. In C. Sammut, A.G. Hoffmann (eds) *Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002)*, 602-609. Morgan Kaufmann, San Francisco.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In J.D. Schaffer (ed) *Proceedings of the third international conference on Genetic algorithms*, 2-9, Morgan Kaufmann Publishers Inc.

Tanese, R. (1989). Distributed genetic algorithms. In J.D. Schaffer (ed) *Proceedings of the third international conference on Genetic algorithms*, 434-439, Morgan Kaufmann Publishers Inc.

ter Braak, C.J.F. (2006). A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces. *Statistics and Computing*, 16(3), 239-249.

ter Braak, C.J.F., Vrugt, J.A. (2008). Differential evolution Markov chain with snooker updater and fewer chains. *Statistics and Computing*, 18(4), 435-446.

Tong, H. (1990). *Non-linear time series. A Dynamical System Approach.* Oxford: Oxford University Press.

Tong, H. (2012). *Threshold models in non-linear time series analysis (Vol. 21).* New York: Springer Science & Business Media.

Trautmann, H., Mersmann, O. Arnu, D. (2011). cmaes: Covariance Matrix Adapting Evolutionary Strategy. R package version 1.0-11. URL: http://CRAN.R-project.org/package=cmaes

Tseng, L.Y., Yang, S.B. (2001). A genetic approach to the automatic clustering problem. *Pattern Recognition*, 34(2), 415-424.

Ursu, E., Pereau, J.C. (2016). Application of periodic autoregressive process to the modeling of the Garonne river flows. *Stochastic environmental research and risk assessment*, 30(7), 1785-1795.

Ursu, E., Pereau, J.C. (2017). Estimation and identification of periodic autoregressive models with one exogenous variable. *Journal of the Korean Statistical Society.* doi: https://doi.org/10.1016/j.jkss.2017.07.001

Ursu, E., Turkman, K.F. (2012). Periodic autoregressive model identification using genetic algorithms. *Journal of Time Series Analysis*, 33(3), 398-405.

Vecchia, A. V. (1985). Periodic autoregressive-moving average (PARMA) modeling with applications to water resources. *Journal of the American Water Resources Association*, 21(5), 721-730.

Vega Yon, G., Muñoz, E. (2016). ABCoptim: An implementation of the Artificial Bee Colony (ABC) Algorithm. R package version 0.14.0, URL: https://github.com/gvegayon/ABCoptim.

Vo-Van, T., Nguyen-Thoi, T., Vo-Duy, T., Ho-Huu, V., Nguyen-Trang, T. (2017). Modified genetic algorithm-based clustering for probability density functions. *Journal of Statistical Computation and Simulation*, 87(10), 1964-1979.

Vose, M. (1999). *The Simple Genetic Algorithm.* Cambridge, MA: MIT Press.

Vrugt, J.A., ter Braak, C.J.F., Diks, C.G.H., Robinson, B.A., Hyman, J.M., Higdon, D. (2009). Accelerating Markov chain Monte Carlo simulation by differential evolution with self-e randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3), 273-290.

Waagen, D.E., Parsons, M.D., McDonnell, J.R., Argast, J.D. (1994). Evolving multivariate mixture density estimates for classification. In S. Chen (ed) *Proceedings SPIE 2304, Neural and Stochastic Methods in Image and Signal Processing III*, 2304, 175-187. International Society for Optics and Photonics.

Wang, T., Berthet, Q., Samworth, R.J. (2016). Statistical and computational trade-offs in estimation of sparse principal components. *Annals of Statistics*, 44(5), 1896-1930.

Winker, P., Maringer, D. (2009). The convergence of estimators based on heuristics: theory and application to a GARCH model. *Computational Statistics*, 24, 533-550.

Wong, H., Ip, W. C., Zhang, R., Xia, J. (2007). Non-parametric time series models for hydrological forecasting. *Journal of Hydrology*, 332(3), 337-347.

Wong, W.H., Liang, F. (1997). Dynamic weighting in Monte Carlo and optimization. *Proceedings of the National Academy of Sciences*, 94(26), 14220-14224.

Wright, A.H. (1991). Genetic algorithms for real parameter optimization. *Foundations of genetic algorithms*, 1, 205-218.

Wu, B., Chang, C.L. (2002). Using genetic algorithms to parameters (d, r) estimation for threshold autoregressive models. *Computational Statistics & Data Analysis*, 38(3), 315-330.

Yang, Y., Wainwright, M.J., Jordan, M.I. (2016). On the computational complexity of high-dimensional Bayesian variable selection. *The Annals of Statistics*, 44(6), 2497-2532.

Yau, C.Y., Tang, C.M., Lee, T.C. (2015). Estimation of multiple-regime threshold autoregressive models with structural breaks. *Journal of the American Statistical Association*, 110(511), 1175-1186.

Zaharie D. (2002). Critical values for the control parameters of differential evolution algorithms. In R. Matousek (ed) *Proceedings of MENDEL 2002, 8th international conference on soft computing*, 62-67. Brno.

Zambrano-Bigiarini, M., Rojas, R. (2014). hydroPSO: Particle Swarm Optimisation, with focus on Environmental Models. R package version 0.3-4.

Zhang, B.T., Cho, D.Y. (2001). System identification using evolutionary Markov chain Monte Carlo. *Journal of Systems Architecture*, 47(7), 587-599.

Zhang, Q., Mühlenbein, H. (2004). On the convergence of a class of estimation of distribution algorithms. *IEEE Transactions on evolutionary computation*, 8(2), 127-136.

Zhou, X., Wang, J. (2005). A genetic method of LAD estimation for models with censored data. *Computational statistics & data analysis*, 48(3), 451-466.