Juillet 2002 **Numéro 29**

Cahiers de l'IMA

Optimisation of Mixture Models: Comparison of Different Strategies

André Berchtold

Institut de Mathématiques Appliquées
Faculté des S.S.P.
Université de Lausanne
BFSH 2
CH-1015 Lausanne

Optimisation of Mixture Models: Comparison of Different Strategies

André Berchtold[1]

Abstract

Mixture models are usually optimized using an Expectation-Maximization (EM) algorithm. However, it is well-known that this method can sometimes converge toward a critical point of the solution space which is not the global maximum. To minimize this problem, different strategies using different combinations of algorithms can be used. In this paper, we compare by the mean of numerical simulations different strategies using EM, CEM, SEM, and genetic algorithms for the optimization of mixture models. Our results indicate that two-stage procedures combining an exploration and an optimization phases provide the best results, especially when these methods are applied on a population of initial conditions rather than on a solely starting point.

Keywords: Mixture models, Mixture Transition Distribution model (MTD), Expectation-Maximization (EM) algorithm, Classification EM (CEM), Stochastic EM (SEM), Genetic Algorithm (GA).

[1]Email: Andre.Berchtold@imaa.unil.ch, Web: www.andreberchtold.com

Contents

List of Tables

List of Figures

1 Introduction

In most cases, the parameters of mixture models are estimated using an Expectation-Maximisation (EM) algorithm. This method presents the advantage to converge surely toward a critical point of the log-likelihood of the solution space. However, nothing insure in practice that this critical point is the global maximum rather than a local maximum, or even a saddlepoint. To solve this problem, several possibilities have been proposed. We can cite the use of several successive runs of an EM algorithm (Biernacki, Celeux & Govaert, 2001), the combination of an EM algorithm with a Classification EM (CEM) or a Stochastic EM (SEM) algorithm (Biernacki, Celeux & Govaert, 2001), the use of a Newton-type algorithm (Böhning, 2001), and the use of Genetic Algorithms (GA, Berchtold (2001)).

Since the solution space of mixture models can be very complex with many local maxima, an efficient optimization method must be able to explore this space entirely. There are mainly two types of strategies to ensure that. The first one consists of starting several times *independently* the optimization process, using different sets of initial values. Obviously, the greater the number of starts, the greater the probability to find the global maximum. The second possibility is to start the algorithm considering *simulteanously* several possible solutions, and to combine them together in order to improve the probability of reaching the global maximum. This second solution implies the use of a genetic algorithm and may require a smaller number of different starting values.

Note that we concentrate here on finding the combination of parameters leading to the global maximum of the log-likelihood. It has been pointed out that this particular extremum is not always the most interesting one. For instance, it can correspond to a very unusual combination of parameters, or it can be influenced by a very few number of data points. Nevertheless, we consider that this discussion is out of the scope of this paper. Moreover, the methods presented here can also be used when we are interested in identifiying all the critical points of a solution space.

The aim of this paper is to compare the performance of several optimization strategies based either on EM-type or on genetic algorithms. We also tried to determine whether the same algorithm must be used during the whole optimization process, or if the optimization has to be divided into two parts: an initialisation part during which optimal starting values are identified, and an optimization part during which the global optimum is really looked for. The paper is organized as follow: the next section recalls mixture models as well as EM and genetic algorithms; sections 3 and 4 present the results of two sets of numerical simulations; the last section summarizes our findings and provides general recommendations.

2 Mixture models and optimization algorithms

In this section, we present the class of models used in this paper, namely Mixture Transition Distribution models. Then, we describe the different algorithms which are used for their optimization.

2.1 Mixture Transition Distribution models

Consider a situation in which each point of a data set can have been generated by one out of k different distributions, the distribution effectively used for each data point being unknown. We have then a mixture model whose basic equation is

$$f(x) = \sum_{g=1}^{k} \lambda_g \, f_g(x)$$

where $f(x)$ is the marginal density of the data set x, $f_g(x)$ is the density of the g-th possible distribution, and λ_g is the weight associated with the g-th distribution. In practice, the weights are unknown.

Mixture models can be used in many situations. In this paper, we deal with time series and we use a mixture model called Mixture Transition Distribution (MTD) model. Let $\{X_t, t \in \mathbb{N}^*\}$ be a sequence of random variables taking values in $\mathbb{R}$. Let X_{t-a}^{t-1} denote the past observations between t-a and t-1. Assume that the whole observed time series was generated by k different submodels (or *components*). The MTD model is then written

$$F(x_t|x_1^{t-1}) = \sum_{g=1}^{k} \lambda_g \, G_g(x_t|x_{t-r_g}^{t-1})$$

where $F(x_t|x_1^{t-1})$ is the cumulative distribution function of x_t given the past, $G_g(x_t|x_{t-r_g}^{t-1})$ is the cumulative distribution function of x_t given a part of the past (from x_{t-r_g} to x_{t-1}, $r_g \geq 1$), and λ_g is the weight of the g-th component, with $\sum_{g=1}^{k} \lambda_g = 1$, $\lambda_g > 0$, $\forall g$. Different specifications can be used for the components, but here we consider only Gaussian distributions and we write

$$G_g(x_t|x_{t-r_g}^{t-1}) = \Phi\left(\frac{x_t - \mu_{g,t}}{\sigma_g}\right)$$

In order to take into account the dependence relation between successive observations, the expectation of each Gaussian component is written as a function of the past. The expectation of the g-th component at time t is specified by the following autoregressive model:

$$\mu_{g,t} = \varphi_{g,0} + \sum_{i=1}^{p_g} \varphi_{g,i} \, x_{t-i} \, , \quad p_g \geq 1$$

The standard deviation of each Gaussian component can also be written as a function of the past of the process (see Berchtold (2002)), but we do not use this possibility here.

MTD models were defined first for the modeling of high-order Markov chains (Raftery, 1985). Then, they were generalized to continuous data (Le, Martin & Raftery, 1996, Wong & Li, 2000, Berchtold, 2002). See also Berchtold & Raftery (1999) for a survey of this methodology. The choice of the MTD model was guided by two main considerations: the developing of mixture models in non-traditional areas, and the wish to provide results completing the existing ones. Even if we use here a particular mixture model, our results can be directly generalized to any other mixture model because the optimization algorithms are the same. Moreover, our results are coherent with the results given in Biernacki, Celeux & Govaert (2001).

2.2 Optimisation algorithms

The EM algorithm is an iterative method using alternatively two steps to maximise the log-likelihood of a model: During the Expectation (E) step, the parameters of the model are supposed known and the probability that each data point came from each of the k possible components is computed. During the Maximisation (M) step, the parameters are reestimated by maximisation of the log-likelihood. The log-likelihood is increasing at each iteration until stationarity, what insure the convergence to a critical point of the solution space. The number of iterations needed to reach convergence can be infinite. See e.g. McLachlan & Krishnan (1996) for a complete treatment of the EM algorithm.

The critical point toward which the EM algorithm converges depends on the starting values. This fact can lead to very difficult problems. Figure 1 presents a simplified situation in which a model depends on only one parameter, λ. On this figure, the log-likelihood has three maxima. We call *attraction basin* of a maximum the set of starting values of the parameter λ leading to this particular maximum when using the EM algorithm. On this example, we see that each of the three maxima has a quite large attraction basin. So, when starting the EM algorithm with a random value of λ, it is possible to reach any of the three local solutions with a non-null probability.

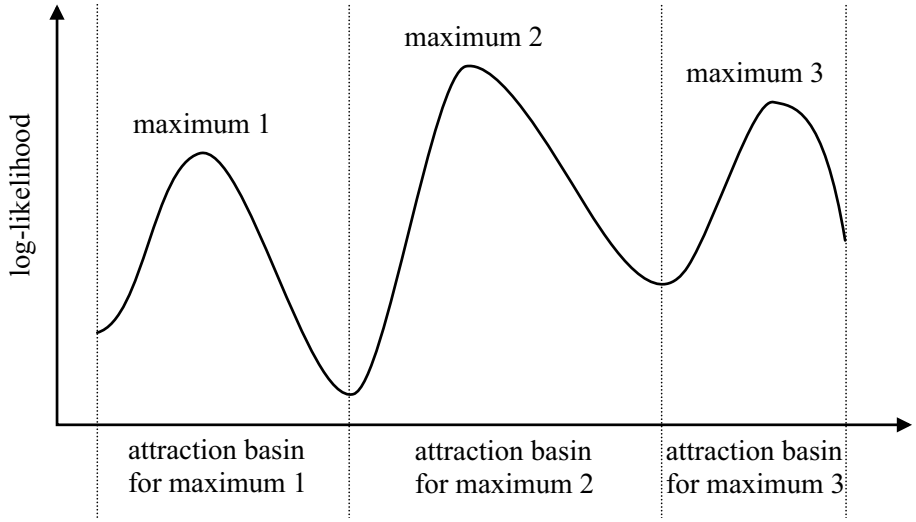

Figure 1. This figure shows the log-likelihood of a model depending on only one parameter, λ. We see the three maxima of the solution space along with their respective attraction basins.

Figures 2 and 3 show two other examples. On Figure 2, the number of maxima is high and the probability to reach one of them when starting the EM algorithm with a random value of λ is small, especially for the global maximum. Finally, Figure 3 presents a situation with only two maxima, but since the local one has a very large attraction basin, it is difficult to reach the global maximum when starting the EM algorithm with a random value of the parameter λ.

These examples show that finding the global maximum is directly related to the size of its attraction basin. The larger its attraction basin, the higher the probability to reach it. So, when the solution space is such that the global maximum does have a small attraction basin, an algorithm such as EM can perform poorly because it is too starting values dependent. Other optimization strategies are then required.

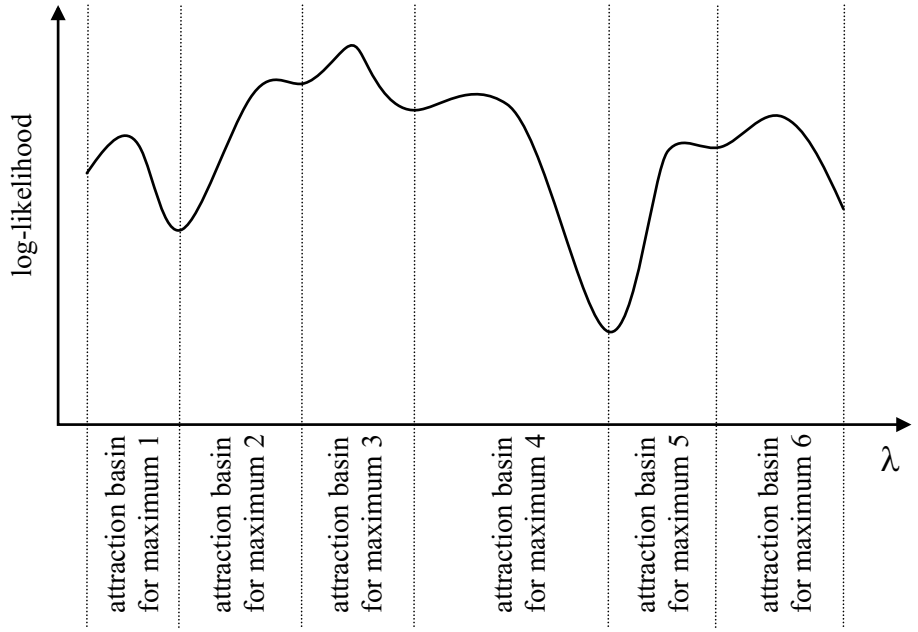

Figure 2. This figure shows the log-likelihood of a model depending on only one parameter, λ. We see the six maxima of the solution space along with their respective attraction basins.

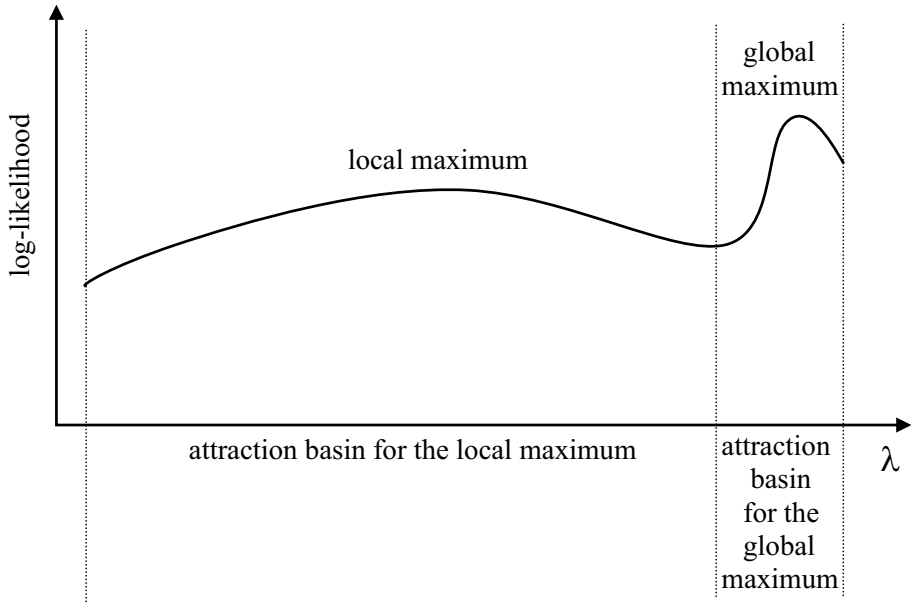

Figure 3. This figure shows the log-likelihood of a model depending on only one parameter, λ. We see the two maxima of the solution space along with their respective attraction basins. Note that the attraction basin of the local maximum is much larger than the one of the global maximum.

Several algorithms derived from the basic EM have been proposed. The Classification EM (CEM) algorithm (Celeux & Govaert, 1992) is used to assign quickly each data point to one of the k

components of the model. After the E step, each data point is attributed to the component having the greatest probability to have generated this observation. This algorithm is known to converge in a finite number of iterations and it creates a partition of the observations in k groups. On the other hand, the model corresponding to this partition is not optimal and its log-likelihood is not a maximum of the solution space.

The Stochastic EM (SEM) algorithm (Celeux & Diebolt, 1985) is another variant of the basic EM algorithm. After the E step, each observation is assigned randomly to one of the k components on the basis of the probability distribution computed for each observation during the E step. This method generates a Markov chain of successive solutions which is expected to be around a maximum of the solution space.

A Genetic Algorithm (GA) obeys to a completely different principle. The process is initialized by the creation of a population of several possible solutions whose parameters are generally chosen at random. A measure of fitness is computed for each member of this population. In our case, such a measure can be based on the log-likelihood, the members of the population being closer to the global maximum having a greater log-likelihood than the others. Then, a new population is generated from the previous one using genetic operators, that is selection of some members of the previous population (the ones with the greatest fitness values having a better probability to be selected), crossover of the parameters of two members of the population in order to obtain two children having both inherited a part of the parameters of both parents, and random mutation of some parameter values. The fitness value of each member of the new population is then computed and the process iterates. The probability for a member of the population to reach the global maximum of the solution space generally increases after each iteration. On the other hand, the speed of convergence is very slow, and a quasi-infinite number of iterations can be required. See e.g. Holland (1975) and Vose (1999) for more details and results on genetic algorithms. Compared to the EM algorithm, the use of a genetic algorithm suppress the attraction basin problem: even if no starting point is in the attraction basin of the global maximum, this particular extremum can still be reached in a few iterations.

2.3 Relative cost of the different algorithms

The main difficulty in defining strategies combining different optimization methods is to compare the relative computing cost of each algorithm. The idea is to allocate to each strategy a comparable number of iterations. We take as a reference a population of size one, and 100 iterations of an EM algorithm. As the population size increases, the number of EM iterations applied on each member of the population decreases accordingly. For instance, in a population of size 10, each member has only 10 EM available iterations, and so on.

The use of algorithms derived from EM does not imply a major change. Since the additional computations required by both CEM and SEM are very small in regard of the basic EM algorithm, we consider that one CEM or one SEM iteration is equal to one EM iteration. For instance, it is equivalent to perform 100 EM iterations, or 50 CEM iterations followed by 50 EM iterations.

The case of genetic algorithm is completely different. First of all, this method requires the use of a large population. Here, we do not consider populations of size smaller than 10. Basically, on a same population size, a GA iteration is less time consuming than an EM iteration does, because the required computations are lighter. However, it is difficult to compare exactly the two methods for at least three reasons:

1. On the contrary of EM, the use of a GA requires several parameters to be chosen (crossover rate, mutation rate, number of digits used to code each parameter into binary form, and so on). Even if standard values exist, these parameters stay problem-dependent and each choice can influence the speed of the algorithm (see e.g. Davis et al. (1991)).

2. When the population size increases, the total number of available iterations being the same, the GA becomes more time-consuming. For instance, it takes more time to perform 5 GA iterations on a population of size 20, than 10 GA iterations on a population of size 10.

3. As the number of data increases, the GA becomes proportionnally quicker than the EM does. When multipliying by two the number of data will approximately multiply by the same factor the computation time required by EM, the increase in time will be a lot smaller for GA.

Considering all of these factors together, it appears that it is not possible to insure a perfect comparison between EM and GA. So, in this paper, we took a conservative approach: First of all, we used only standard settings for the genetic algorithm, that is a crossing rate of 50%, a mutation rate of 1%, elitism (the best member of a population becomes automatically part of the next population), and a recoding of each real parameter into a binary string of 20 bits. Then, we determined from short samples of 50 data that performing 10 EM iterations on each member of a population of size 10 was as time-consuming as performing 30 GA iterations on this same population. This is the basis for our comparisons. Of course, as the number of data increases, this comparison basis can become unfair for GA, but considering that the genetic algorithm is here a new method compared to EM, this choice suppress the risk of a bias in favor of the genetic algorithm. Moreover, the analysis of our simulations show that allocating more iterations to GA would not have dramatically modified our findings.

3 Numerical simulations, part I

3.1 21 strategies

Table 1 presents 21 different strategies to be compared through numerical simulations. Strategies 1 to 4 are EM-only strategies. A population of size 1 to 50 is randomly generated, then each member of this population is improved through a fixed number of EM iterations. The member of the population reaching the largest log-likelihood is retained as the solution. On the other hand, strategies 5 to 9 are GA-only strategies. A population of size 10 to 300 is randomly generated, then new populations are created using the genetic operators.

Strategies 10 to 21 are two-phases strategies. During the first phase, a CEM, SEM, or GA algorithm is used to explore the solution space in order to determine an optimal set of starting values. During the second phase, an EM algorithm is applicated on each of these starting points in order to reach the global maximum of the log-likelihood. Strategies 10 and 11 are using a CEM algorithm to find starting values. Note that since the CEM algorithm is known to converge in a finite number of iterations, we retained the following approach: each time the CEM converged, it was started again with random values, and the best result achieved after the available number of iterations was used to start the EM part of the strategy. Obviously, during the initialization phase, strategies 10 and 11 are in fact using a population of size greater than the one given in Table 1.

Table 1. Description of the 21 optimization strategies. The symbol x can take any positive integer value. It acts as a multiplicator on the number of available iterations. For instance, when x=1, the number of iteration available for strategy 1 is 100. When x=2, there are 200 available iterations, and so on.

Number	Population size	Strategy used on each member of the population
1	1	x100 EM
2	10	x10 EM
3	20	x5 EM
4	50	x2 EM
5	10	x30 GA
6	20	x15 GA
7	50	x6 GA
8	100	x3 GA
9	300	x1 GA
10	1	x50 CEM - x50 EM
11	10	x5 CEM - x5 EM
12	1	x50 SEMmax - x50 EM
13	10	x5 SEMmax - x5 EM
14	1	x50 SEMmoy - x50 EM
15	10	x5 SEMmoy - x5 EM
16	10	x15 GA - x5 EM
17	50	x3 GA - x1 EM
18	10 - 1	x15 GA - x50 EM (on the best result obtained after GA)
19	50 - 1	x3 GA - x50 EM (on the best result obtained after GA)
20	10	x5 [3 GA - 1 EM]
21	50	x1 [3 GA - 1 EM]

Strategies 12 and 13 are using a SEMmax algorithm. During the initialization phase, SEM iterations are performed, then the largest log-likelihood reached during these iterations is elected and the corresponding parameters are used to start the EM part of the strategy. Strategies 14 and 15 are using a SEMmoy algorithm. During the initialization phase, a certain number of SEM iterations are performed, then an average of the parameters obtained after each iteration is computed and this mean value of the parameters is used to initialize the EM part of the strategy. In order to improve the method, we took the average on the second half only of the SEM iterations.

Strategies 16 to 21 combine a genetic and an EM algorithms in different ways. Strategies 16 and 17 are using first a GA to find starting values. Then, each member of the population is improved

through a few iterations of the EM algorithm. Strategies 18 and 19 are similar, but only the best member of the final population obtained after the GA part is improved using EM. Obviously, the number of EM iterations attributed to this best member of the population is larger. Finally, strategies 20 and 21 are using iteratively the GA and EM algorithms.

Note that, when approaches similar to strategies 1 to 4 and 10 to 15 are presented in Biernacki, Celeux & Govaert (2001), we do not know of any other study using the other strategies in the context of mixture models. The three following subsections present detailed results for two numerical simulations. Note that more simulations were performed, but we provide here only the two most representative of them. Nevertheless, the analysis of section 3.4 is consistent with all of our results.

3.2 Simulation 1

The first simulation uses the data appearing on Figure 4. This is a simulated time series of 105 data points. We computed the following MTD model:

$$F(x_t|x_1^{t-1}) = \lambda\,\Phi\left(\frac{x_t - \varphi_{1,0} - \varphi_{1,1}\,x_{t-1} - \varphi_{1,2}\,x_{t-2}}{\sigma_1}\right) + (1-\lambda)\,\Phi\left(\frac{x_t - \varphi_{2,0} - \varphi_{2,1}\,x_{t-1} - \varphi_{2,2}\,x_{t-2}}{\sigma_2}\right)$$

There are 9 independent parameters and we put the following constraints: $\lambda \in [0.01, 0.99]$, $\varphi \in [-10, 10]$, $\sigma^2 \in [0.5, 50]$. The model was computed on the last 100 data points, conditionally on the beginning of the series. Main results are provided in Table 2 and on Figures 5 and 6. Discussion is postponed until section 3.4.

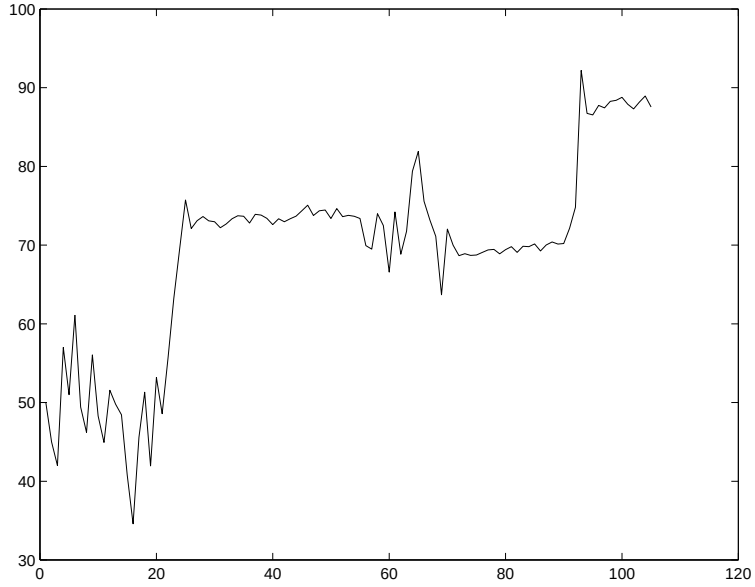

Figure 4 Series of 105 simulated data.

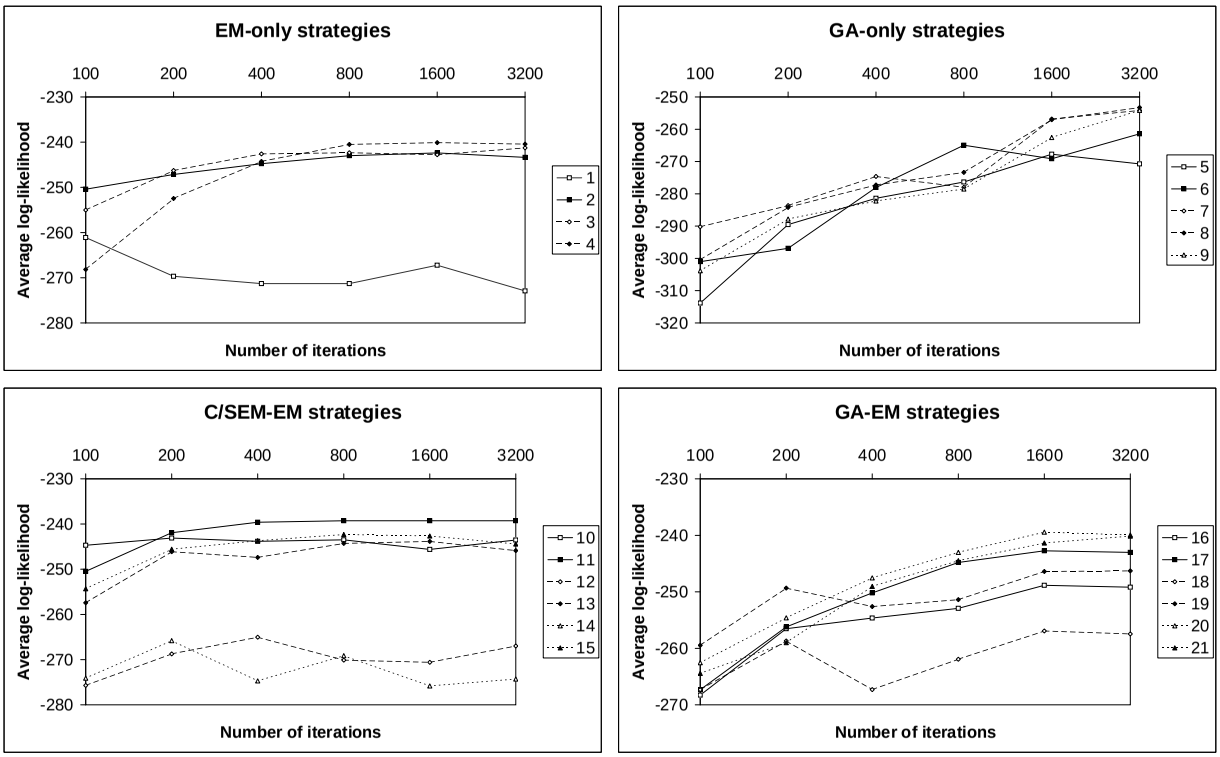

Figure 5 Complete results for simulation 1. Each strategy type appears on a separate graph.

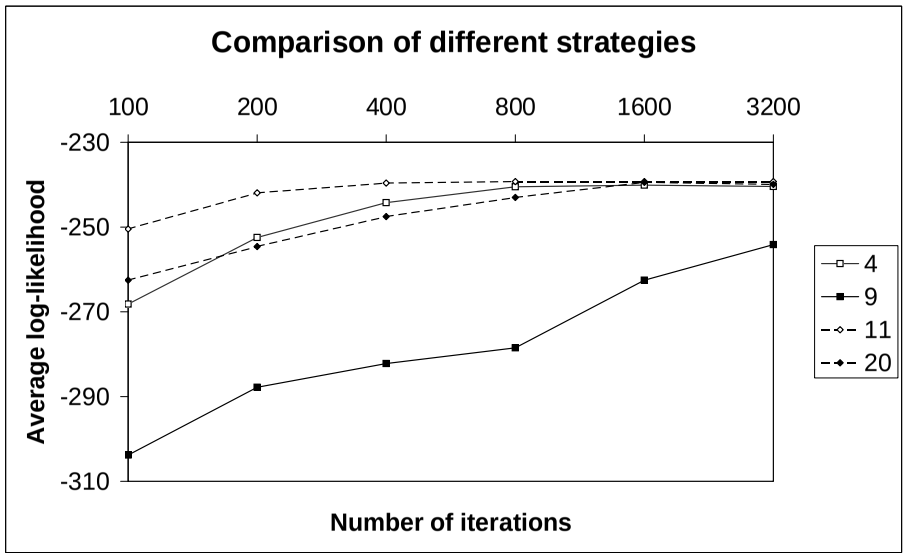

Figure 6. Summary of the results for simulation 1. For each of the four type of strategies, only the strategy obtaining the best average result with 3200 available iterations is presented.

Table 2. Results for simulation 1. For each of the 21 strategies and each number of available equivalent-EM iterations, we give the average of the best observed log-likelihoods obtained from 25 replications, and the corresponding standard error (into brackets). The best result obtained for each number of available iterations and each simulation type is printed in bold.

Strategy	Number of available equivalent-EM iterations					
number	100	200	400	800	1600	3200
1	-261.11	-269.68	-271.29	-271.30	-267.22	-272.92
	(17.24)	(17.00)	(15.86)	(16.65)	(18.50)	(15.74)
2	**-250.40**	-247.15	-244.76	-243.00	-242.37	-243.36
	(7.72)	(7.89)	(7.83)	(3.88)	(4.65)	(4.28)
3	-255.03	**-246.27**	**-242.60**	-242.34	-242.78	-241.26
	(10.42)	(4.63)	(4.09)	(4.12)	(2.53)	(2.07)
4	-268.17	-252.47	-244.25	**-240.50**	**-240.11**	**-240.44**
	(7.30)	(10.36)	(4.93)	(2.01)	(1.77)	(1.93)
5	-313.83	-289.52	-281.40	-276.34	-267.74	-270.73
	(24.97)	(21.43)	(18.10)	(17.74)	(16.10)	(18.69)
6	-301.02	-296.89	-278.06	**-264.94**	-269.13	-261.42
	(23.03)	(17.53)	(14.63)	(18.35)	(18.75)	(14.45)
7	**-290.22**	**-283.66**	**-274.63**	-278.00	**-256.83**	-254.17
	(15.61)	(16.19)	(15.29)	(18.02)	(7.86)	(8.24)
8	-300.42	-284.20	-277.24	-273.36	-257.05	**-253.27**
	(19.63)	(8.78)	(11.82)	(7.69)	(9.49)	(10.72)
9	-303.82	-287.81	-282.22	-278.50	-262.55	-254.13
	(18.06)	(8.18)	(6.44)	(6.36)	(5.90)	(6.96)
10	**-244.70**	-243.11	-243.82	-243.50	-245.62	-243.54
	(7.46)	(2.29)	(3.35)	(3.62)	(5.12)	(4.69)
11	-250.48	**-241.92**	**-239.61**	**-239.28**	**-239.28**	**-239.28**
	(6.31)	(2.57)	(1.34)	(0.81)	(0.81)	(0.81)
12	-275.72	-268.74	-265.05	-270.12	-270.60	-266.98
	(12.20)	(16.47)	(20.25)	(16.17)	(17.92)	(18.26)
13	-257.43	-246.11	-247.38	-244.28	-243.85	-245.89
	(11.85)	(8.27)	(9.88)	(4.30)	(3.99)	(7.52)
14	-274.09	-265.80	-274.70	-269.14	-275.83	-274.29
	(16.45)	(18.50)	(15.63)	(16.23)	(13.26)	(14.55)
15	-254.33	-245.57	-243.70	-242.31	-242.64	-244.41
	(10.57)	(6.57)	(4.09)	(3.52)	(3.39)	(4.69)
16	-268.29	-256.53	-254.64	-252.93	-248.84	-249.18
	(12.43)	(16.61)	(17.15)	(16.55)	(11.37)	(11.33)
17	-267.32	-256.22	-250.20	-244.80	-242.73	-243.02
	(9.20)	(7.74)	(9.84)	(8.07)	(4.11)	(3.82)
18	-267.33	-258.69	-267.31	-261.93	-256.93	-257.46
	(19.03)	(17.79)	(17.36)	(16.40)	(15.70)	(16.86)
19	**-259.47**	**-249.34**	-252.61	-251.38	-246.40	-246.29
	(16.33)	(13.34)	(14.28)	(12.69)	(9.71)	(4.91)
20	-262.54	-254.60	**-247.53**	**-243.02**	**-239.44**	**-239.94**
	(14.57)	(10.42)	(11.83)	(5.44)	(1.12)	(1.65)
21	-264.43	-258.99	-249.03	-244.48	-241.35	-240.13
	(9.22)	(11.76)	(8.48)	(6.95)	(4.28)	(2.81)

3.3 Simulation 2

The second simulation uses the data diplayed on Figure 7. This is a time series of length 179 giving the US annual consumer price inflation level from 1821 to 1999. We computed the following model:

$$\begin{aligned} F(x_t|x_1^{t-1}) &= \lambda_1 \, \Phi\left(\frac{x_t - \varphi_{1,0} - \varphi_{1,1}\, x_{t-1} - \varphi_{1,2}\, x_{t-2} - \varphi_{1,3}\, x_{t-3}}{\sigma_1}\right) \\ &\quad + \lambda_2 \, \Phi\left(\frac{x_t - \varphi_{2,1}\, x_{t-1} - (1-\varphi_{2,1})\, x_{t-2}}{\sigma_2}\right) + (1-\lambda_1-\lambda_2)\, \Phi\left(\frac{x_t - x_{t-1}}{\sigma_3}\right) \end{aligned}$$

There are 10 independent parameters and we put the following constraints: $\lambda_1 \in [0.01, 0.98]$, $\lambda_2 \in [0.01, 0.99 - \lambda_1]$, $\varphi \in [-50, 50]$, $\sigma^2 \in [1, 400]$. The model was computed on the last 175 points of the series, conditionally on the beginning of the series. Main results are provided in Table 3 and on Figures 8 and 9. Discussion is postponed until section 3.4.

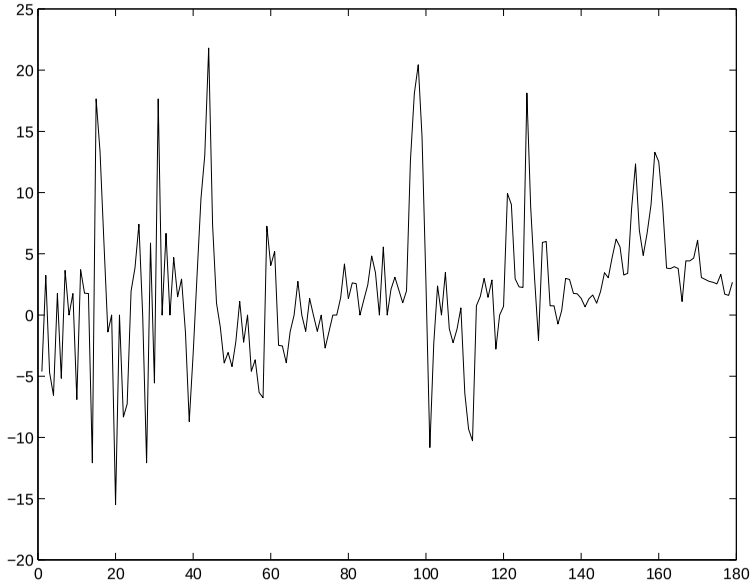

Figure 7 US annual consumer price inflation level from 1821 to 1999.

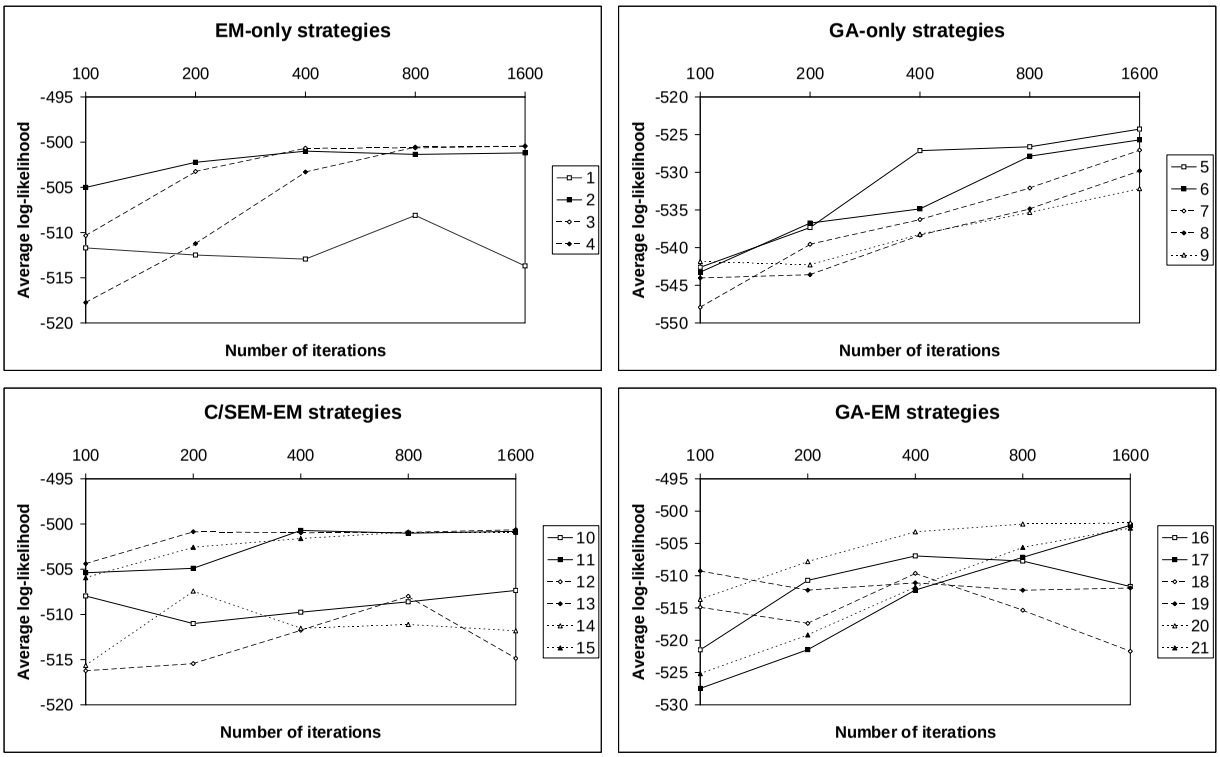

Figure 8 Complete results for simulation 2. Each strategy type appears on a separate graph.

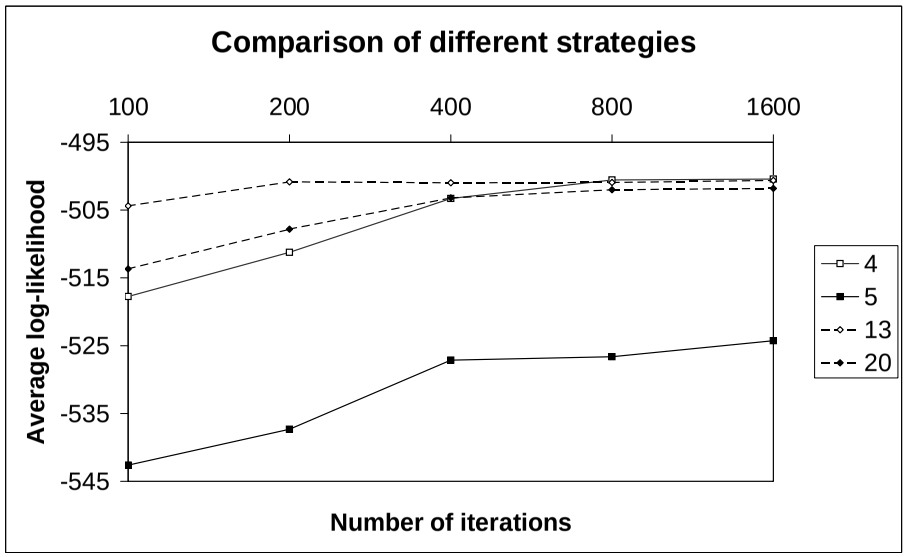

Figure 9. Summary of the results for simulation 2. For each of the four type of strategies, only the strategy obtaining the best result with 1600 available iterations is presented.

Table 3. Results for simulation 2. For each of the 21 strategies and each number of available equivalent-EM iterations, we give the average of the best observed log-likelihoods obtained from 25 replications, and the corresponding standard error (into brackets). The best result obtained for each number of available iterations and each simulation type is printed in bold.

Strategy number	Number of available equivalent-EM iterations				
	100	200	400	800	1600
1	-511.70	-512.49	-512.93	-508.10	-513.69
	(11.36)	(13.72)	(12.78)	(7.62)	(13.26)
2	**-505.01**	**-502.23**	-500.99	-501.35	-501.19
	(3.29)	(2.95)	(1.37)	(2.33)	(1.79)
3	-510.34	-503.25	**-500.67**	-500.63	-500.44
	(2.47)	(1.73)	(0.77)	(0.67)	(0.00)
4	-517.75	-511.23	-503.30	**-500.55**	**-500.43**
	(3.79)	(2.12)	(1.58)	(0.11)	(0.00)
5	-542.61	-537.33	**-527.13**	**-526.62**	**-524.27**
	(9.32)	(5.05)	(7.62)	(8.66)	(8.43)
6	-543.25	**-536.74**	-534.86	-527.86	-525.70
	(9.57)	(6.18)	(4.48)	(7.40)	(6.91)
7	-547.93	-539.56	-536.27	-532.09	-527.06
	(7.45)	(4.67)	(5.84)	(5.14)	(6.23)
8	-544.03	-543.59	-538.34	-534.86	-529.80
	(6.65)	(5.08)	(4.08)	(5.72)	(4.72)
9	**-541.84**	-542.29	-538.21	-535.33	-532.16
	(5.90)	(4.82)	(4.26)	(4.85)	(5.51)
10	-507.96	-511.01	-509.75	-508.60	-507.35
	(7.04)	(10.73)	(5.85)	(5.60)	(4.36)
11	-505.39	-504.90	**-500.71**	-501.02	-500.81
	(2.74)	(3.58)	(1.02)	(1.42)	(0.88)
12	-516.23	-515.44	-511.75	-507.99	-514.86
	(13.83)	(14.78)	(13.84)	(7.48)	(12.05)
13	**-504.39**	**-500.85**	-500.97	-500.92	**-500.63**
	(2.97)	(0.91)	(1.14)	(1.21)	(0.70)
14	-515.64	-507.41	-511.53	-511.11	-511.81
	(13.90)	(7.38)	(9.86)	(10.60)	(13.19)
15	-505.94	-502.57	-501.61	**-500.88**	-500.91
	(3.19)	(2.79)	(2.14)	(1.26)	(1.13)
16	-521.48	-510.73	-506.94	-507.74	-511.67
	(7.05)	(7.76)	(8.15)	(9.38)	(9.25)
17	-527.46	-521.45	-512.22	-507.16	-502.20
	(3.38)	(4.14)	(3.82)	(2.83)	(2.17)
18	-514.89	-517.39	-509.64	-515.35	-521.71
	(11.09)	(10.02)	(8.86)	(11.39)	(10.34)
19	**-509.27**	-512.24	-511.10	-512.26	-511.90
	(6.93)	(7.62)	(9.59)	(8.07)	(9.56)
20	-513.67	**-507.82**	**-503.21**	**-501.99**	**-501.81**
	(5.76)	(6.54)	(5.37)	(4.18)	(2.25)
21	-525.17	-519.17	-511.73	-505.61	-502.63
	(6.06)	(5.23)	(4.25)	(3.90)	(3.08)

3.4 Analysis of the first two simulations

We compare here the 21 strategies on the basis of the results of the first two simulations. We comment first separately the four strategy types, then we compare the best methods and we give general recommendations about the choice of an optimization strategy. It must be noted that, since the simulations were made separately for each considered number of equivalent-EM available iterations, the results for one particular method can decrease when increasing the number of iterations. This is not a mistake, but a consequence of the uncertainty of any numerical method.

Strategies 1 to 4, using a pure EM algorithm, are interesting only when a large number of runs is allowed. The classical one-run EM (strategy 1) performs very poorly. This method should never be used, except maybe when good reasons exist to think that the solution space has only one critical point. Since its convergence speed can be low, the EM algorithm has to be run for a while in order to be powerful. So, when the number of available iterations is small (e.g. 100, 200, ...) only a small population size is appropriate (like 10 for strategy 2). On the other hand, as the number of available iterations increases, it becomes more and more interesting to also increase the population size. When the number of iterations is maximum, we can observe on both examples that the best EM-only strategy is the fourth one, with a population size of 50.

Strategies 5 to 9, using a pure GA algorithm, are clearly not competitive in the context of these simulations (see Figures 6 and 9). As we know (see e.g. Vose (1999)), when the number of runs is infinite and the parameters (population size, mutation rate, and so on) are well-chosen in regard of the given optimization problem, a GA will find almost surely the global optimum. However, this does not correspond to our situation. Here, we use standard values for the parameters instead of problem-related ones, and more important, the number of available iterations is finite. In this context, we cannot expect a GA to be competitive. Moreover, the two simulations behave differently. On the first example, average size populations give the best results (strategies 7 and 8 with respective populations of size 50 and 100). On the second example, the smallest population size (10, strategy 5) is preferred as soon as the number of available equivalent-EM iterations becomes larger than 200. In that regard, we can not give general recommendations about the most appropriate settings for genetic algorithms.

Strategies 10 to 15 are using a two-phases approach with an initialization phase based on CEM or SEM followed by an EM optimization phase. We observe first that the strategy using a population size of 10 is almost always preferred to the corresponding single run strategy, the only exception being strategy 10 which is preferred to strategy 11 in the 100 equivalent-EM iterations of the first simulation. This is consistent with the observation made before concerning the need to run several times an EM method in order to obtain good results. On the first simulation, strategy 11 using the CEM method during the initialization phase is clearly the best. On the second example, each of the three methods is the best at least once. Globally, the CEM method seems more reliable than the SEMmoy and SEMmax methods in the sense than even when it is outperformed by SEM, the CEM method obtains results close to SEM.

Strategies 16 to 21 are using different combinations of genetic and EM algorithms, with different population sizes. Except in the case of very small numbers of equivalent-EM available iterations, strategy 20 using iteratively several runs of GA and several runs of EM on a population of size 10 is the preferred one. On the other hand, strategy 18 obtains very unstable results, the worst being reached on the second simulation with the maximum of available iterations. This points out the need to use a large population during the whole optimization process and not during the initialization

phase only.

Globally, the best method emerging from simulation 1 is a combination of an initialization phase based on CEM with an optimization phase using EM (strategy 11). On the other hand, the second simulation favors strategy 13 combining SEMmax and EM. However, it must be noted that, since the MTD model used in the second simulation seemed quite easy to optimize, several methods provided very similar results. Moreover, as the number of equivalent-EM available iterations increases, strategies 17, 20, and 21, combining GA and EM, become more and more interesting, their results being in some cases very close to the best ones.

The analysis of the standard errors associated with each batch of 25 simulations shows clearly that strategies having the best average behavior have also small standard errors. More generally, the use of larger population sizes combined with a sufficient number of available iterations leads to more precise results.

4 Numerical simulations, part II

4.1 Selection of 7 strategies

On the basis of the results obtained after the first two simulations, we decided to retain only 7 optimization strategies among the 21 previously defined. Table 4 describes these strategies. For convenience purposes, we kept the strategy numbers used previously.

Table 4. Description of the 7 optimization strategies used for the second set of simulations. The symbol x can take any positive integer value. It acts as a multiplicator on the number of available iterations. For instance, when x=1, the number of iteration available for each member of the population in strategy 2 is 10. When x=2, there are 20 available iterations for each population member, and so on.

Number	Population size	Strategie used on each member of the population
2	10	x10 EM
4	50	x2 EM
11	10	x5 CEM - x5 EM
13	10	x5 SEMmax - x5 EM
15	10	x5 SEMmoy - x5 EM
17	50	x3 GA - x1 EM
20	10	x5 [3 GA - 1 EM]

The first two strategies are EM-only strategies with populations of size 10 and 50, respectively. The first of these strategy proved to lead quickly to acceptable results, when the second one is appropriate when a large number of iterations is available. The next three strategies are two-phases strategies using a CEM or a SEM algorithm to initialize a pure EM optimization phase. We retained here only the simulations using a size 10 population, the other simulations having shown poor results. Finally, we kept two strategies combining GA and EM. In spite of average results,

strategy 17 was retained because it uses a two-phases approach which can be easily compared to strategies 11, 13, and 15. Strategy 20 was chosen because it obtained very good results when a large number of iterations was available. In spite of good results, we did not retain strategy 21 because it is too similar to strategy 20. In accordance with our previous results, we did not retain any GA-only strategy.

4.2 Simulations 3 and 4

In order to check whether the complexity of a MTD model can influence the optimization algorithm, we considered again the simulated time series of Figure 4 previously used for the first simulation, and we computed two additional models. The first one is defined as

$$F(x_t|x_1^{t-1}) = \lambda\,\Phi\left(\frac{x_t - \varphi_{1,1}\,x_{t-1}}{\sigma_1}\right) + (1-\lambda)\,\Phi\left(\frac{x_t - \varphi_{2,2}\,x_{t-2}}{\sigma_2}\right)$$

There are 5 independent parameters and we put the following constraints: $\lambda \in [0.01, 0.99]$, $\varphi \in [-10, 10]$, $\sigma^2 \in [0.5, 50]$. The model was computed on the last 100 data points, conditionally on the beginning of the series. This model is simpler than the model used for the first simulation, since the expectation of each Gaussian component depends on only one past observation (x_{t-1} for component 1, and x_{t-2} for component 2). Table 5 and Figure 10 summarize the results.

Table 5. Results for simulation 3. For each of the 7 strategies and each number of available equivalent-EM iterations, we give the average of the best observed log-likelihoods obtained from 25 replications, and the corresponding standard error (into brackets). The best result obtained for each number of available iterations is printed in bold.

Strategy	Number of available equivalent-EM iterations					
number	100	200	400	800	1600	3200
2	-336.05	-335.16	-335.60	-336.05	-336.05	-336.05
	(0)	(4.36)	(2.22)	(0)	(0)	(0)
4	-321.80	-330.61	-334.76	-335.45	-336.05	-334.07
	(15.49)	(12.17)	(4.42)	(2.94)	(0)	(9.70)
11	-325.27	-308.93	-300.37	-289.29	-284.75	-282.25
	(14.58)	(14.73)	(11.25)	(12.76)	(9.31)	(6.34)
13	-329.32	-329.23	-330.87	-329.20	-329.47	-330.45
	(15.07)	(14.06)	(9.31)	(10.58)	(12.21)	(10.83)
15	-332.82	-335.37	-336.05	-335.14	-336.05	-336.05
	(9.18)	(3.34)	(0)	(4.45)	(0)	(0)
17	**-296.94**	**-281.34**	**-274.84**	**-268.19**	-261.38	**-257.59**
	(25.96)	(16.05)	(11.28)	(5.78)	(3.72)	(1.95)
20	-308.68	-293.95	-279.33	-272.59	**-260.90**	-260.38
	(18.78)	(23.11)	(19.16)	(14.02)	(8.92)	(6.73)

This simulation is very interesting since, on the contrary of previous examples, strategies combining GA and EM are clearly preferred to any other method, including the combination CEM-EM.

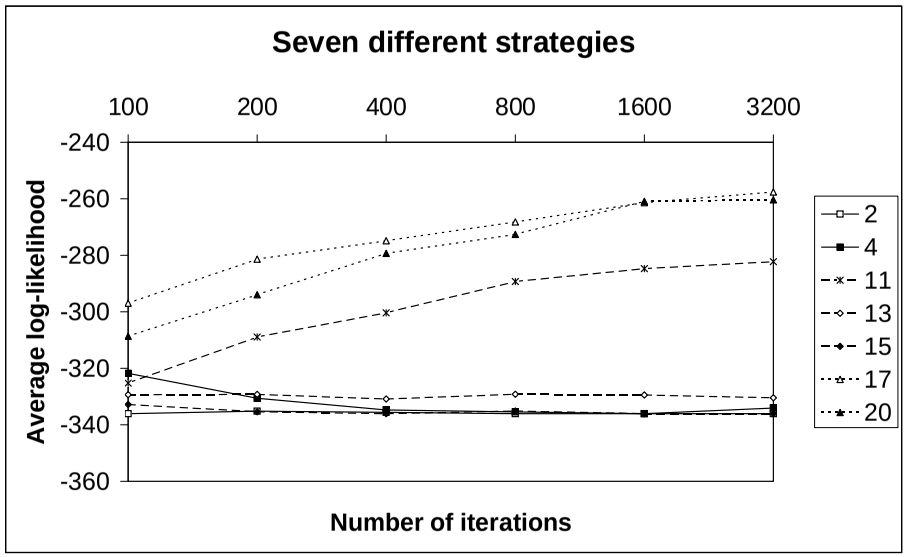

Figure 10 Results for simulation 3.

Moreover, EM-only strategies achieve particularly poor results. To understand this particular behavior, we performed a detailed analysis of the whole solution space. It appeared that one local maximum with a log-likelihood equal to -336.05 presents a very large attraction basin. On the other hand, the attraction basin of the global maximum covers less than one percent of the solution space. In this context, only very specific starting values allow the EM algorithm to reach the best solution. So, a complete exploration of the whole solution space must be performed by the optimization algorithm, what is exactly the principle of genetic algorithms.

Simulation 4 is using a much more complex model defined as

$$\begin{aligned} F(x_t|x_1^{t-1}) &= \lambda_1 \, \Phi\left(\frac{x_t - \varphi_{1,0} - \varphi_{1,1}\, x_{t-1} - \varphi_{1,2}\, x_{t-2} - (1 - \varphi_{1,1} - \varphi_{1,2})\, x_{t-3}}{\sigma_1}\right) \\ &\quad + \lambda_2 \, \Phi\left(\frac{x_t - \varphi_{2,0} - \varphi_{2,1}\, x_{t-1} - \varphi_{2,2}\, x_{t-2}}{\sigma_2}\right) + (1 - \lambda_1 - \lambda_2)\, \Phi\left(\frac{x_t - \varphi_{3,1}\, x_{t-1}}{\sigma_3}\right) \end{aligned}$$

There are 12 independent parameters and we put the following constraints: $\lambda_1 \in [0.01, 0.98]$, $\lambda_2 \in [0.01, 0.99 - \lambda_1]$, $\varphi \in [-10, 10]$, $\sigma^2 \in [0.5, 100]$. The model was computed on the last 100 data points, conditionally on the beginning of the series. Table 6 and Figure 11 summarize the results.

This example shows again that a complete exploration of the solution space is required in order to correctly identify the global optimum of the problem. In this case a combination CEM-EM achieved the best result, the use of GA requiring a larger number of iterations for the same result.

Table 6. Results for simulation 4. For each of the 7 strategies and each number of available equivalent-EM iterations, we give the average of the best observed log-likelihoods obtained from 25 replications, and the corresponding standard error (into brackets). The best result obtained for each number of available iterations is printed in bold.

Strategy	Number of available equivalent-EM iterations					
number	100	200	400	800	1600	3200
2	-263.51	-253.91	-252.39	-252.14	-251.41	-251.14
	(8.47)	(3.86)	(1.01)	(0.71)	(2.11)	(2.43)
4	-266.76	-262.97	-253.03	-251.43	-249.74	-250.20
	(11.90)	(10.53)	(6.18)	(3.53)	(2.59)	(2.80)
11	**-260.64**	**-252.10**	**-247.83**	**-245.27**	**-241.42**	**-241.35**
	(7.70)	(5.55)	(5.34)	(4.71)	(3.81)	(3.59)
13	-264.79	-258.16	-252.67	-251.73	-251.56	-252.27
	(8.65)	(9.10)	(2.79)	(1.44)	(2.53)	(0.80)
15	-263.09	-253.86	-251.45	-252.37	-251.98	-251.85
	(12.64)	(5.54)	(1.92)	(0.73)	(1.12)	(1.74)
17	-276.79	-266.86	-257.93	-251.00	-250.57	-243.19
	(6.74)	(10.82)	(12.51)	(5.29)	(4.17)	(7.10)
20	-280.70	-274.53	-265.31	-254.97	-252.92	-248.30
	(19.34)	(16.06)	(15.81)	(9.96)	(9.57)	(8.29)

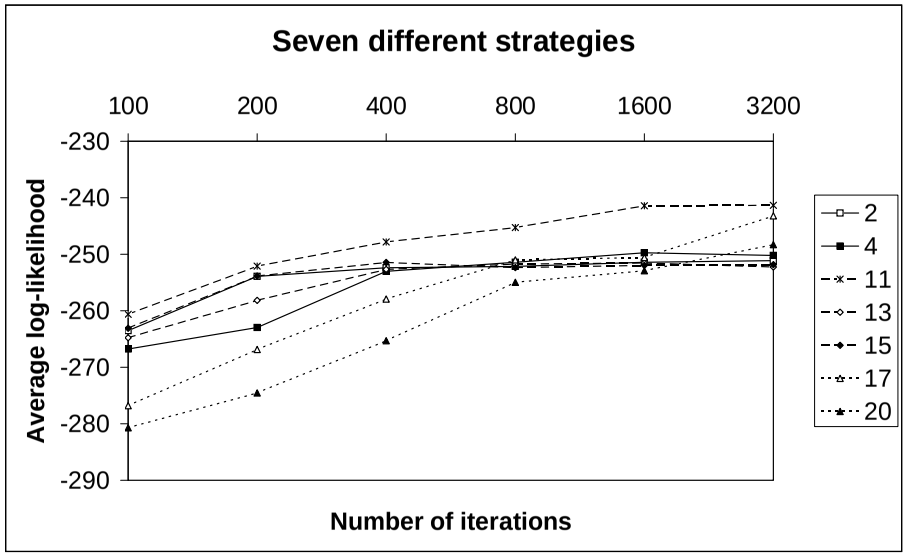

Figure 11 Results for simulation 4.

4.3 Simulation 5

Figure 12 shows the closing price of the Eastman Kodak share from May 12, 1998 to May 2, 2000, for a total of 499 observations. It was found in Berchtold & Raftery (1999) that this time series is well represented by a second-order random walk GMTD, that is a MTD model of the form

$$\begin{aligned} F(x_t|x_1^{t-1}) &= \lambda_1 \Phi\left(\frac{x_t - \varphi_{1,1}\, x_{t-1} - (1-\varphi_{1,1})\, x_{t-2}}{\sigma_1}\right) \\ &\quad + \lambda_2 \Phi\left(\frac{x_t - x_{t-1}}{\sigma_2}\right) + (1-\lambda_1-\lambda_2)\, \Phi\left(\frac{x_t - x_{t-2}}{\sigma_3}\right). \end{aligned}$$

There are 6 independent parameters and we put the following constraints: $\lambda_1 \in [0.01, 0.98]$, $\lambda_2 \in [0.01, 0.99 - \lambda_1]$, $\varphi_{1,1} \in [-20, 20]$, $\sigma^2 \in [0.1, 50]$. The model was computed on the last 495 points of the series, conditionally on the first observations. Table 7 and Figure 13 summarize the results.

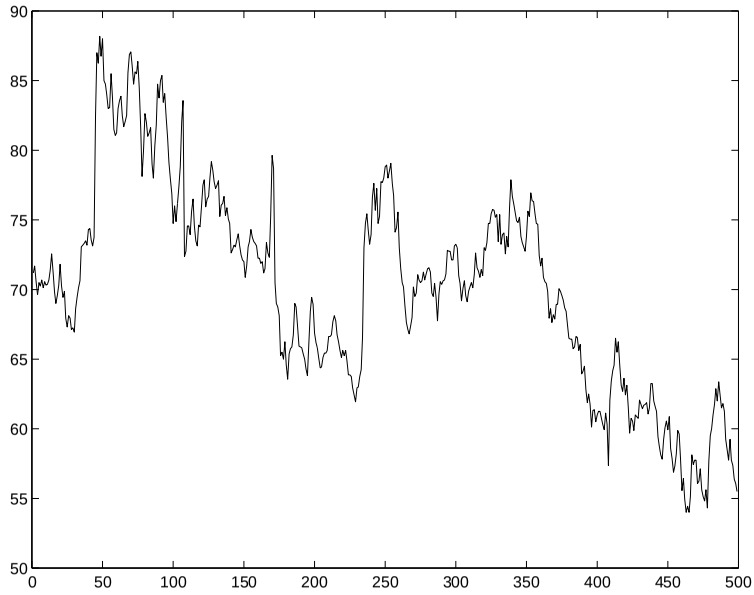

Figure 12. Closing price of the Eastman Kodak share from May 12, 1998 to May 2, 2000 (499 observations).

On the contrary of some of the previous examples, the model computed on the Eastman Kodak data was very easy to optimize. This is established by the fact that pure EM strategies achieved very good results, even with a small number of available iterations. Strategies 4 and 17, using a larger population but with less iterations allocated to each member, needed more time to converge to the global maximum. This fact indicates that the global maximum has a very large attraction basin, so it can be reached from a large number of starting point, the only condition being a sufficient number of available iterations. With 400 or more available iterations, each of the 7 strategies become equivalent. The choice of the optimization strategy is then here less important than in other examples.

Table 7. Results for simulation 5. For each of the 7 strategies and each number of available equivalent-EM iterations, we give the average of the best observed log-likelihoods obtained from 25 replications, and the corresponding standard error (into brackets). The best result obtained for each number of available iterations is printed in bold.

Strategy number	Number of available equivalent-EM iterations 100	200	400	800
2	-809.07	**-806.93**	-806.60	-806.17
	(1.02)	(0.23)	(0.66)	(1.15)
4	-820.35	-811.34	-807.96	-806.85
	(3.56)	(1.31)	(0.66)	(0.12)
11	-808.88	-806.94	-806.71	-805.49
	(1.28)	(0.16)	(0.36)	(1.29)
13	**-808.84**	-806.97	-806.75	-805.98
	(1.54)	(0.21)	(0.23)	(1.20)
15	-809.63	-807.27	-806.71	-805.92
	(1.98)	(1.33)	(0.38)	(1.17)
17	-827.31	-814.80	-808.41	-806.98
	(9.02)	(5.80)	(1.86)	(0.67)
20	-811.32	-807.55	**-806.07**	**-805.35**
	(3.35)	(2.29)	(1.33)	(0.21)

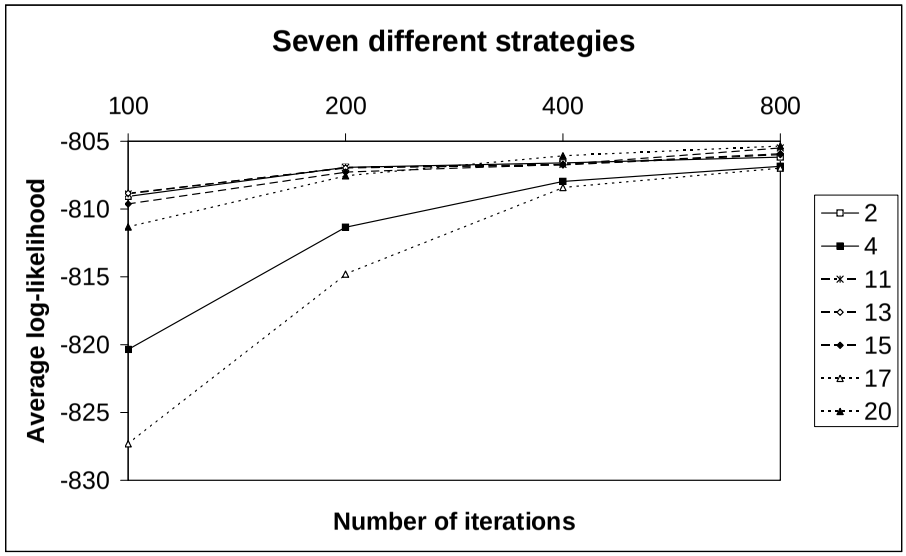

Figure 13 Results for simulation 5.

4.4 Simulation 6

The last simulation was performed on the viscosity series appearing as series D of Box, Jenkins & Reinsel (1994). This is a series of 310 hourly viscosity readings from a chemical process. The data appear on Figure 14.

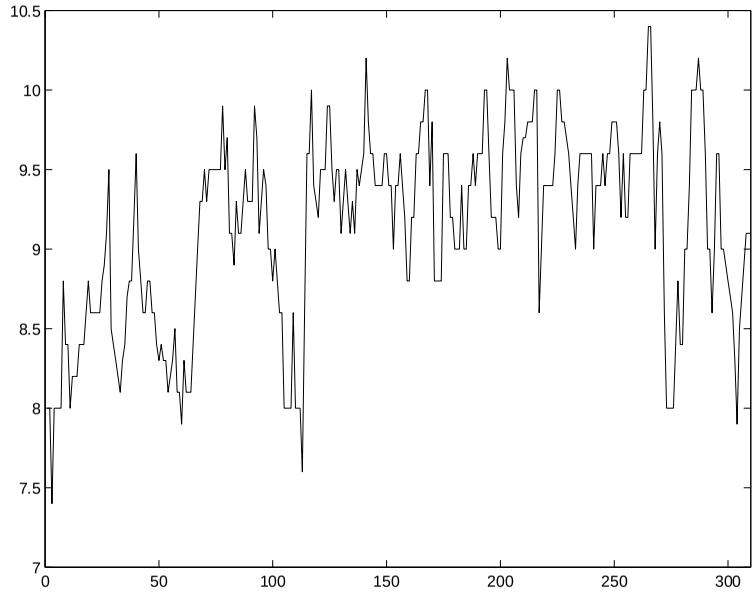

Figure 14 Series of hourly viscosity readings from a chemical process (310 observations).

We applied the same second-order random walk GMTD model previously used for simulation 5 with the following constraints: $\lambda_1 \in [0.01, 0.98]$, $\lambda_2 \in [0.01, 0.99 - \lambda_1]$, $\varphi_{1,1} \in [-10, 10]$, $\sigma^2 \in [0.05, 10]$. Note that to avoid a possible degeneration of the log-likelihood which could distort our simulations, these constraints are different from the ones used in Le, Martin & Raftery (1996). That explains why we achieved a log-likelihood much lower than theirs. The model was computed on the last 308 data points of the series, conditionally on the first two observations. Table 8 and Figure 15 summarize the results.

This simulation presents another situation, with three strategies (11, 13, 15) obtaining very good results even with only 100 available iterations. On the other hand, pure EM strategies obtain poor results, and combined GA-EM strategies have difficulties to converge. An exhaustive exploration of the solution space revealed the presence of many local maxima. For this very reason, strategy 2 performs very poorly. Nevertheless, the attraction basin of the global optimum is quite large and strategies using CEM or SEM are then appropriate, when strategies 17 and 20, giving less iterations to EM, require too much time to converge.

Table 8. Results for simulation 6. For each of the 7 strategies and each number of available equivalent-EM iterations, we give the average of the best observed log-likelihoods obtained from 25 replications, and the corresponding standard error (into brackets). The best result obtained for each number of available iterations is printed in bold.

Strategy number	Number of available equivalent-EM iterations 100	200	400	800	1600
2	-159.24	-168.34	-148.14	-167.49	-160.11
	(52.36)	(45.23)	(37.92)	(52.58)	(55.19)
4	-105.50	-108.90	-106.28	-114.65	-102.12
	(17.98)	(18.93)	(17.56)	(18.45)	(16.32)
11	-73.13	-72.95	-72.71	**-72.03**	-72.58
	(3.43)	(3.04)	(2.55)	(2.78)	(2.99)
13	-71.84	-72.73	-71.90	-72.91	-73.02
	(2.35)	(3.05)	(3.46)	(3.01)	(3.51)
15	**-70.94**	**-71.70**	**-71.65**	-72.51	**-71.30**
	(3.22)	(3.29)	(2.35)	(3.38)	(2.81)
17	-112.97	-93.65	-90.88	-86.91	-86.66
	(6.93)	(9.32)	(6.68)	(6.50)	(6.15)
20	-120.83	-111.35	-104.36	-102.24	-98.98
	(13.38)	(10.27)	(5.07)	(4.77)	(3.55)

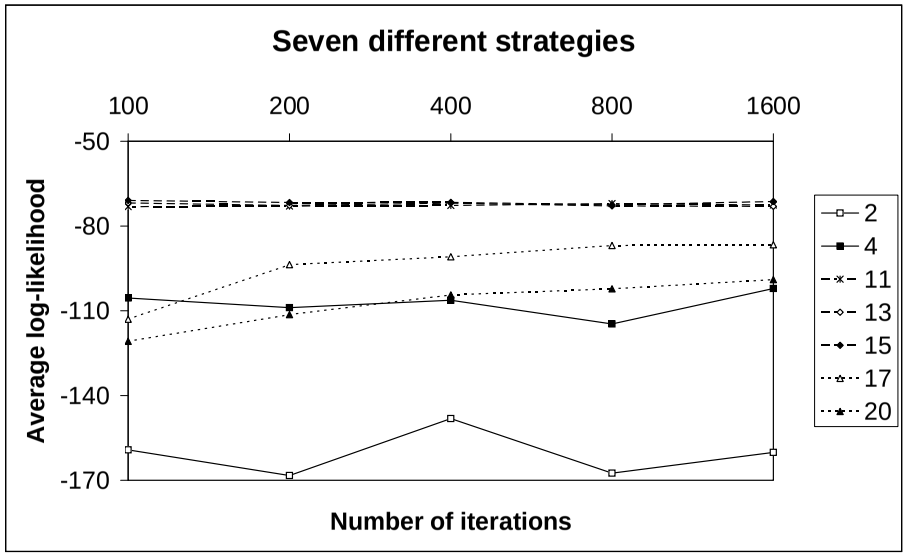

Figure 15 Results for simulation 6.

5 Conclusion

In this paper, we discussed different type of strategies for the optimization of mixture models. Through a set of numerical simulations, we compared the use of two main algorithms: EM and GA. Overall, we found that, in order to obtain reliable results for any possible situation, several rules have to be followed. First of all, it appears that the probability to converge from a randomly chosen starting point to the global maximum of the log-likelihood is often small. So, multiplying the number of trials with different sets of starting values is a good idea. Moreover, in order to explore the entire solution space, a strategy in two phases is appropriate: In a first time, a set of good starting points is determined through either a CEM, SEM or GA. Then, in a second step, each intermediary solution is improved through several iterations of a pure EM algorithm. Using this method, a reliable solution can generally be provided in a reasonable amount of time.

Through our set of numerical simulations, we identified at least three main situations: First, there is only one maximum which can be reached from virtually any possible starting point (see e.g. simulation 5). In this case, a pure EM algorithm is appropriate. A second situation occurs when there are several maxima, the global one having a large attraction basin (see e.g. simulation 4). The use of a combination CEM-EM provides then good results. Finally, it is possible to observe situations where the global maximum has a very small attraction basin (see simulation 3). In this case, it is crucial to insure the exploration of the whole solution space, what can be done very efficiently through the use of a genetic algorithm.

In most cases, we do not know in which of these three main situations we are. So, the question of the choice of an overall optimal optimization strategy stays. If we consider that two-phases strategies are generally much better than pure EM or pure GA methods, the real question is about the choice of the algorithm to be used during the first phase, the EM algorithm being adequate for the second phase. In definitive, we consider that the best solution is to run simulteanously two optimization strategies: a combination CEM-EM, and a combination GA-EM. Following our experience, in each case at least one of these two strategies proved to be appropriate, whatever the model and the data. Their simultaneous use insures then that the global maximum will be reach with a high probability.

Acknowledgements

I would like to thank Gilles Celeux for his very useful comments about the use and the behavior of the EM, CEM, and SEM algorithms.

References

BERCHTOLD A. (2001) Mixture Modeling of Non-Gaussian Probability Distributions. *Proceedings of the 10th International Symposium on Applied Stochastic Models and Data Analysis*, 188-193, Compiègne, France.

BERCHTOLD A. (2002) Mixture Transition Distribution (MTD) Modeling of Heteroscedastic Time Series. Forthcoming in *Computational Statistics & Data Analysis.*

BERCHTOLD, A., RAFTERY, A.E. (1999) The Mixture Transition Distribution (MTD) Model for High-Order Markov Chains and Non-Gaussian Time Series. Forthcoming in *Statistical Science.*

BIERNACKI C., CELEUX G., GOVAERT, G. (2001) Strategies for getting the highest likelihood in mixture models. Rapport de recherche 4255, INRIA, France. Forthcoming in *Computational Statistics & Data Analysis.*

BÖHNING, D. (2001) The Potential of Recent Developments in Nonparametric Mixture Distributions. *Proceedings of the 10th International Symposium on Applied Stochastic Models and Data Analysis*, 6-13, Compiègne, France.

BOX, G.E.P., JENKINS, G.M., REINSEL, G.C. (1994) *Time Series Analysis, Forecasting and Control.* Prentice Hall, London, 3rd edition.

CELEUX, G., DIEBOLT, J. (1985) The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2, 73-82.

CELEUX, G., GOVAERT, G. (1992) A Classification EM Algorithm for Clustering and Two Stochastic Versions. *Computational Statistics & Data Analysis*, 14, 315-332.

DAVIS, L. ET AL. (1991) *Handbook of Genetic Algorithms.* Davis Editor, Van Nostrand Reinhold, New York.

HOLLAND, J.H. (1975) *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor.

LE, N.D., MARTIN, R.D., RAFTERY, A.E. (1996) Modeling Flat Stretches, Bursts, and Outliers in Time Series Using Mixture Transition Distribution Models. *Journal of the American Statistical Association*, 91, 1504-1515.

MCLACHLAN, G.J., KRISHNAN, T. (1996) *The EM Algorithm and Extensions.* John Wiley & Sons, New York.

RAFTERY, A.E. (1985) A model for high-order Markov chains. *Journal of the Royal Statistical Society B*, 47 (3), 528-539.

VOSE, M.D. (1999) *The Simple Genetic Algorithm, Foundations and Theory.* MIT Press, Cambridge.

WONG, C.S., LI, W.K. (2000) On a mixture autoregression model. *Journal of the Royal Statistical Society B*, 62, 95-115.