INFORMATICA, 1997, Vol. 8, No. 4, 495–526

A SET OF EXAMPLES OF GLOBAL AND DISCRETE OPTIMIZATION: APPLICATION OF BAYESIAN HEURISTIC. APPROACH II

Jonas MOCKUS

Institute of Mathematics and Informatics
Akademijos 4, 2600 Vilnius, Lithuania
E-mail: mockus@ktl.mii.lt

Abstract. The following topics are important teaching operation research: games theory, decision theory, utility theory, queuing theory, scheduling theory, discrete optimization.

These topics are illustrated and the connection with global optimization is shown considering the following mathematical models:

- competition model with fixed resource prices, Nash equilibrium,
- competition model with free resource prices, Walras equilibrium,
- inspector's problem, multi-stage game model,
- "Star War" problem, differential game model,
- "Portfolio" problem, resource investment model,
- exchange rate prediction, Auto-Regression-Moving-Average (ARMA) model,
- optimal scheduling, Bayesian heuristic model,
- "Bride's" problem, sequential statistical decisions model.

The first seven models are solved using a set of algorithms of continuous global and stochastic optimization. The global optimization software GM (see [19]) is used. The underlying theory of this software and algorithms of solution are described in [19, 17]. The last model is an example of stochastic dynamic programming.

For better understanding, all the models are formulated in simplest terms as "classroom" examples. However, each of these models can be regarded as simple representations of important families of real-life problems. Therefore the models and solution algorithms may be of interest for application experts, too.

The paper is split into two parts. In the part one [18] the first five models are described. In this part the rest three models and accompanyiing software are considered.

Key words: operations research, Bayesian, heuristic, optimization, global.

1. Exchange rate forecasting, time series model

1.1. Introduction. Modeling economic and financial time series using the autoregressive moving average (ARMA) method was described in [6, 3, 5, 4, 13, 20]. In estimating the parameters of the ARMA models, three approaches have been used: Maximum Likelihood (ML) [22], approximate ML [14, 7, 9, 10], and two-step procedures [8, 11]. In all the cases local optimization techniques were used. In this case, the optimization results depend on the initial values, what implies that one cannot be sure if a global maximum is found.

The global optimization is very difficult in almost all the cases[1]. The reason is a high complexity of multi-modal optimization problems in general. It is well known [12] that optimization of real functions cannot be done in polynomial-time, unless $P = NP$[2]. In practice, this means that we need an algorithm of exponential time to obtain the ε-exact solution. The number of operations in exponential algorithms grows exponentially with the accuracy of solution and dimensions of the optimization problem.

A common approximate approach in estimating the parameters of ARMA models is Least Squares (LS). We minimize the log-sum of square residuals instead of maximizing log-likelihood (see [20, 19]).

Subsection 1.2 deals with the ARMA models and estimation methods. Subsection 1.3.6 compares the average prediction results of ARMA and the Random Walk (RW) models. Subsections 1.5 and 1.4 investigate a bilinear, and an artificial neural networks models correspondingly. Subsection 1.6 regards "multi-day" predictions using semi-Monte Carlo simulation techniques. Subsection 1.7 illustrates the multi-modality.

1.2. Auto-regression moving-average models (ARMA)

1.2.1. Definitions. We define an ARMA model as

$$w_t = \sum_{i=1}^{p} a_i w_{t-i} + \sum_{i=1}^{q} b_i \varepsilon_{t-i} + \varepsilon_t. \qquad (1)$$

We assume that

$$z_{t-i} = 0,\ w_{t-i} = 0,\ \varepsilon_{t-i} = 0,\ \text{ if } t \leqslant i. \qquad (2)$$

1.2.2. Definition of residuals. One of the advantages of residual minimization is that one may see directly how the objective depends on unknown parame-

[1] By 'difficult' we mean the time measure of computational complexity, that is, the minimum length of time would be needed for a standard universal computer to perform a task.

[2] The notation $P = NP$ means the existence a polynomial-time algorithm P for solving NP-complete problems. That is merely a theoretical possibility.

ters. Using equalities (1) we define residuals by recurrent expressions:

$$\begin{aligned}
\varepsilon_1 &= w_1 \\
\varepsilon_2 &= w_2 - a_1 w_1 - b_1 \varepsilon_1 \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\varepsilon_t &= w_t - a_1 w_{t-1} - \ldots - a_p w_{t-p} - b_1 \varepsilon_{t-1} - \ldots - b_q \varepsilon_{t-q}. \qquad (3)
\end{aligned}$$

Next the sum

$$f(x) = \log f_m(x), \qquad f_m(x) = \sum_{t=1}^{T} \varepsilon_t^2 \qquad (4)$$

is minimized.

The logarithm is used to decrease the objective variation by improving the scales.

1.3. Minimization of residuals of ARMA models. We consider an algorithm for optimization of parameters of the ARMA model. This model is simple and may be regarded as a good first approximation. Denote by y_t the value of y at the moment t. Denote by $a = (a_1, \ldots, a_q)$ a vector of Auto-Regression (AR) parameters, and by $b = (b_1, \ldots, b_q)$ a vector of Moving-Average (MA) parameters.

$$y_t - \sum_{i=1}^{p} a_i y_{t-i} = \varepsilon_t - \sum_{j=1}^{q} b_j \varepsilon_{t-j}, \quad t = 1, \ldots, T. \qquad (5)$$

The residual

$$\varepsilon_t = y_t - \sum_{i=1}^{p} a_i y_{t-i} + \sum_{j=1}^{q} b_j \varepsilon_{t-j} \qquad (6)$$

or

$$\varepsilon_t = B_t + \sum_{i=1}^{p} a_i A_t(i). \qquad (7)$$

Here

$$B_t = y_t + \sum_{j=1}^{q} b_j B_{t-j-1} \qquad (8)$$

and

$$A_t(i) = -y_{t-i-1} + \sum_{j=1}^{q} b_j A_{t-j-1}, \qquad (9)$$

where $t - i > 0$ and $t - j > 0$.

1.3.1. Optimization of AR parameters. Denote

$$S(a,b)=\sum_{t=1}^{T}\varepsilon^2, \tag{10}$$

where $a=(a_1,\dots,a_p)$ and $b=(b_1,\dots,b_q)$.

From expressions (10) and (7) the minimum condition is

$$\frac{\partial S(a,b)}{\partial a_j}=2\sum_{t=1}^{T}\varepsilon_t A_t(j)=0,\quad j=1,\dots,p \tag{11}$$

or

$$\sum_{i=1}^{p}A(i,j)a_i=-B(j),\quad j=1,\dots,p, \tag{12}$$

where

$$A(i,j)=\sum_{t=1}^{T}A_t(i)A_t(j) \tag{13}$$

and

$$B(j)=\sum_{t=1}^{T}A_t(j)B_t. \tag{14}$$

The minimum of expression (10) at fixed parameters b is defined by a system of linear equations:

$$a(b)=A^{-1}B. \tag{15}$$

Here matrix $A=(A(i,j),\ i,j=1,\dots p)$ and vector $B=(B(j),\ j=1,\dots p)$, where elements $A(i,j)$ are from (13), components $B(j)$ are from (14), and A^{-1} is an inverse matrix A. This way one define the vector $a(b)=(a_i(b),\ i=1,\dots,p)$ that minimize sum (10) at fixed parameters b.

1.3.2. Optimization of MA parameters The sum of squared residuals (10) is a nonlinear non-convex function of MA parameters b. Thus we have to consider the global optimization algorithms. Denote

$$f(x)=\log S\big(a(x),x\big), \tag{16}$$

where $x=b$ and $S(a,b)$ is from (10) at optimal parameter $a=a(b)$. Denote

$$b^0=x^0=\arg\min_x f(x). \tag{17}$$

1.3.3. Advantages and disadvantages of Squared Residuals Minimization (SRM). An advantage of the SRM approach is that objective (4) is explicitly expressed as a function of variables a, b. It is important in the investigation of multi-modality problems while looking for an efficient method of global optimization. Model (10) is flexible. For example, one may conveniently consider the sum of ARMA model with some other models. In any case, a simple model (10) is a good test problem while developing some global optimization techniques that may be used for likelihood maximization, too.

1.3.4. Predicting "next-day" rate. We minimize the expected "next-day" squared deviation ε_{t+1} using the data available at the moment t

$$y_{t+1}^0 = \arg\min_{y_{t+1}} \mathbf{E}\varepsilon_{t+1}^2. \tag{18}$$

Here

$$\mathbf{E}\varepsilon_{t+1}^2 = \mathbf{E}\left(B_{t+1} + \sum_{i=1}^{p} A_{t+1}(i)a_i(b^0)\right)^2, \tag{19}$$

where the optimal parameter b^0 was obtained using the data available at the day t. Variance (19) is minimal, if

$$y_{t+1}^0 = B_{t+1} + \sum_{i=1}^{p} A_{t+1}(i)a_i(b^0), \tag{20}$$

because the expectation of y_{t+1} is $B_{t+1} + \sum_{i=1}^{p} A_{t+1}(i)a_i(b^0)$ under the assumptions.

1.3.5. "Continuous" model. If we wish to keep the "continuity" of sample functions as time unit tends to zero, we have to consider a special case when $a_1 = 1$. In such a case, one ought to change expressions (5)–(20), correspondingly. The popular Random Walk (RW) model:

$$y_{t+1} = y_t + \varepsilon_t, \tag{21}$$

may be regarded as a special case of the "continuous" model when $q = 0$.

1.3.6. Evaluation of ARMA prediction errors. We compare the "next-day" ARMA prediction results and a popular Random Walk (RW) model, where the conditional expectation of $y_{t+1} = y_t$. Table 1 shows the difference between the mean square deviations of ARMA and RW models using DM/\$ and \$/£ exchange rates and AT&T and Intel Co. stocks closing rates for the period of $T = 460$ days. Let us denote by $T_0 = T/4 = 115$ the number of days, used

Table 1. The average prediction results of ARMA and RW models

Data	ARMA	RW	ARMA-RW
$/£	2.299928e-02	2.262379e-02	3.754923e-04
DM/$	3.912910e-02	3.968943e-02	-5.603285e-04
AT&T	1.677580e+02	1.644688e+02	3.289204e+00
Intel Co	4.063805e+02	4.133900e+02	-7.009471e+00

for the "initial" parameter estimation employing global optimization methods. This number seems sufficient for the initial estimation. The "initial" estimates are updated later on by local methods. ARMA and RW denote the mean square deviations of ARMA and RW "next-day" predictions. The difference is denoted as ARMA − RW.

Table 1 shows the average over 345 "next-day" predictions. The table demonstrates that the ARMA model predicts the DM/$ exchange rate and the Intel Co. closing rate better than RW. For the $/£ exchange rate and the AT&T closing rate the opposite is true. The difference is slight but not so insignificant since the average of 345 "next-day" predictions is shown.

There are formal significance tests to answer to this type of questions. However, the results depend on the estimate distribution which is not well-defined in multi-modal cases. The reason is the discontinuity of multi-modal estimates since even slight data changes my cause jumps of estimates.

The results of traditional significance tests depend on the observation numbers, too. For instance, it is shown (see [16]) that any positive difference will become "significant", if the observation number is sufficiently large. Therefore, let us merely define the average prediction errors (see Table 1) and declare that the number of observations is 345.

Table 2 shows optimal parameters $a(b) = (a_0(b), \dots, a_9(b))$ and $b = (b_0, b_1)$ of the ARMA model used for the first 10 of 345 "next-day" predictions of the DM/$ exchange rate. The values of the objective function $f(x)$, $x = (b_0, b_1)$ are denoted by v.

1.4. Artificial Neural Networks Models (ANN). If we are interested mainly in the non-linearities, then we may apply many other non-linear models, including the ones that are non-linear regarding the data, too. In this book we will discuss two of them. In this subsection the ANN model will be considered. In the next section, we shall introduce a bilinear term into the ARMA model.

Table 2. The optimal parameters a and b of ARMA model for 10 predictions

```
a= 1.6261e+00 -1.5934e+00 9.650e-01 0.000e+00 -1.999e+00
   0.0000e+00 -1.9986e+00 0.0000e+00 0.0000e+00 0.0000e+00
     b= 6.2827e-01 -9.7024e-01 v= -4.9132e+00
a= 1.6126e+00 -1.5691e+00 9.5426e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 6.0840e-05 6.2966e-05 -7.8000e-03
     b= 6.1477e-01 -9.5945e-01 v= -4.9053e+00
a= 1.6085e+00 -1.5632e+00 9.5250e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 7.6840e-05 7.1677e-05 -4.0000e-03
     b= 6.1063e-01 -9.5766e-01 v= -4.9041e+00
a= 1.6197e+00 -1.5866e+00 9.6464e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 4.8488e-04 4.2047e-04 -2.0200e-02
     b= 6.2186e-01 -9.6989e-01 v= -4.8595e+00
a= 1.6284e+00 -1.6028e+00 9.7224e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 5.9513e-04 4.9755e-04 -1.0500e-02
     b= 6.3051e-01 -9.7745e-01 v= -4.8502e+00
a= 1.6322e+00 -1.6079e+00 9.7354e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 6.5913e-04 5.5094e-04 -8.0000e-03
     b= 6.3427e-01 -9.7880e-01 v= -4.8437e+00
a= 1.6388e+00 -1.6254e+00 9.8438e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 8.2813e-04 7.1553e-04 1.3000e-02
     b= 6.4082e-01 -9.8975e-01 v= -4.8243e+00
a= 1.6348e+00 -1.6177e+00 9.8062e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 8.5517e-04 7.4322e-04 -5.2000e-03
     b= 6.3693e-01 -9.8600e-01 v= -4.8212e+00
a= 1.6351e+00 -1.6236e+00 9.8619e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 1.0597e-03 8.8185e-04 -1.4300e-02
     b= 6.3722e-01 -9.9145e-01 v= -4.8049e+00
a= 1.6382e+00 -1.6311e+00 9.9063e-01 0.0000e+00 -1.9991e+00
   0.0000e+00 -1.9986e+00 1.1319e-03 9.0880e-04 -8.5000e-03
     b= 6.4026e-01 -9.9593e-01 v= -4.8020e+00
```

We apply ANN by involving the non-linear activation function ϕ into the standard Auto-Regression (AR(l)) model

$$w_t = \phi\left(\sum_{i=1}^{l} a_i w_{t-i}\right) + \varepsilon_t. \tag{22}$$

The idea lurking behind ANN-AR(l) model is that the activation function ϕ roughly represents the activation of a real neuron. We minimize the sum

$$f_m(x) = \sum_{t=1}^{T} \varepsilon_t^2, \tag{23}$$

where the objective $f_m(x)$ depends on l unknown parameters represented as a l-dimensional vector $x = (x_k, k = 1, \ldots, l) = (a_i, i = 1, \ldots, l)$.

From expression (22) it is clear that the residuals ε_t are nonlinear functions of parameters a_t if the activation function ϕ is nonlinear[3]. This means that the minimum conditions

$$\frac{\partial f_m(x)}{\partial a_i} = 0, \quad i = 1, \ldots, p \tag{24}$$

is a system of nonlinear equations that may have a multiple solution.

An interesting activation function is derrived using the Gaussian distribution function

$$\phi(w_t(l)) = \frac{\beta}{\sqrt{2\phi}\sigma} \int_{-\infty}^{w_t(l)} e^{-\frac{w-\mu}{\sigma}} dw. \tag{25}$$

Here $w_t(l) = \sum_{i=1}^{l} a_i w_{t-i}$ and c is a scale parameter.

Obviously, using ANN one meets the multi-modality problem as it was in a case of ARMA model (see non-linear equations (10)). The multi-modality problems of ANN models are discussed in [19].

1.5. Bilinear models. It is well known that for the adequate description of some phenomena additional non-linear terms of the time series could be of use. A simple example is to add a bilinear term (see [21, 15]). Here are bilinear time series extending the ARMA model:

$$w_t = \sum_{i=1}^{p} a_i w_{t-i} + \sum_{i=1}^{q} b_i \varepsilon_{t-i} + \sum_{i=1}^{s} \sum_{j=1}^{r} c_{ij} z_{t-i} \varepsilon_{t-j} + \varepsilon_t. \tag{26}$$

For an illustrative example of the bilinear time series see [19].

1.6. Examples of semi-Monte Carlo simulation A simple way of visual "validation" of a model is by Monte-Carlo simulation. The objective of simulation is to compare the real data with the simulated results of the statistical model.

[3] Assuming the linear activation function ϕ the ANN model is reduced to the standard Auto-Regression model.

An obvious way to do that is by using some type of "Semi-Monte Carlo" simulation technique defined in short as "Simulated Forecasting" (SF).

Using SF we fix the estimated values of unknown parameters μ_0, μ, σ common for both the model 1 (see Fig. 1) and the model 2 (see Fig. 2). We also fix the "individual" parameters a, b, d for each of two ARFIMA models[4].

The residuals ε_t (see expression (3)) are determined up to the simulation starting moment $t(s)$ using the observed data. The rest of residuals $\varepsilon_t,\ t \geqslant t(s)$ are generated by a Gaussian distribution with zero mean and variance σ^2. We repeat the simulation 10 times for each ARMA model separately. The results are presented in Fig. 1 and 2.

The lines denoted as "real" show the real data (the rial/$ exchange rate transformed by the variable-structure model (see [19]). The "mean" lines show the average results of SF prediction of $y_t + \mu_0$. The "min" and "max" lines denote the lower and the upper values of simulation. Therefore, these lines are referred to as "SF-confidence intervals", meaning that if the model is true, one may expect those "intervals" to cover the real data with some "SF-confidence level" $\alpha(SF)$. It is very difficult to define $\alpha(SF)$ exactly. Assuming that "interval deviations" may be regarded as independent and uniformly distributed random variables, we obtain $\alpha(SF) = 1 - k$. Here k is the number of Monte-Carlo repetitions. In our case, $k = 10$, thus $\alpha(SF) = 0.9$

Unfortunately, this assumption "over-simplifies" the statistical model. Therefore, one may regard "SF-confidence levels" $\alpha(SF)$ merely as a Monte-Carlo approximation.

1.7. Examples of squared residuals minimization

1.7.1. Multi-modality examples. We consider $/£ and DM/$ daily exchange rates and AT&T and Intel Corporation stocks closing rates as examples (see Fig. 3, 4, 5, 6). Estimating unknown ARMA parameters we minimize a log-sum of squared residuals defined by expression (4).

The Fig. 7 and 8 show how log-sum (4) depends on the parameters b_0 and b_1, considering the $/£ exchange rate.

The Fig. 9 shows the relation on b_1 considering the DM/$ exchange rate.

The Fig. 10 shows the relation on b_0 regarding the AT&T stocks closing rate.

The Fig. 11 and 12 show the relation on b_0 and b_1 obtained from the Intel Corporation closing rate.

Parameters a are estimated by expression (12).

[4] The ARFIMA model is an extension of ARMA modelincluding an additional parameter d, see [19].

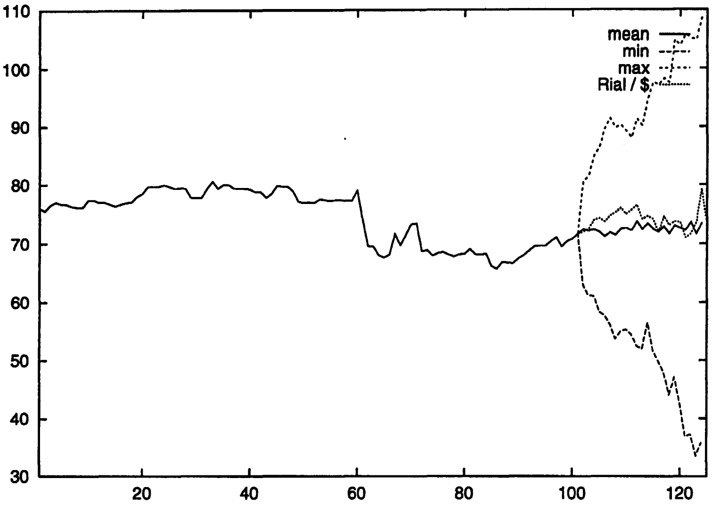

Fig. 1. Results of semi-Monte Carlo Simulation of the transformed rial/$ monthly exchange rate (1965–1976), the model 1.

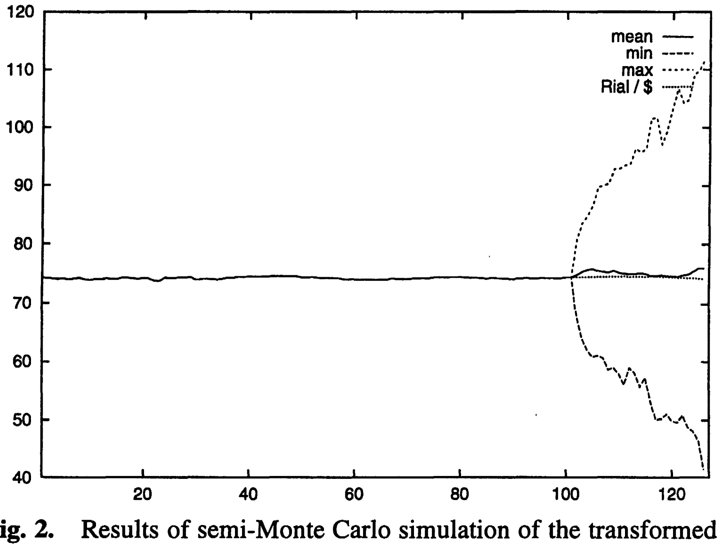

Fig. 2. Results of semi-Monte Carlo simulation of the transformed rial/$ monthly exchange rate (1977–1988), the model 2.

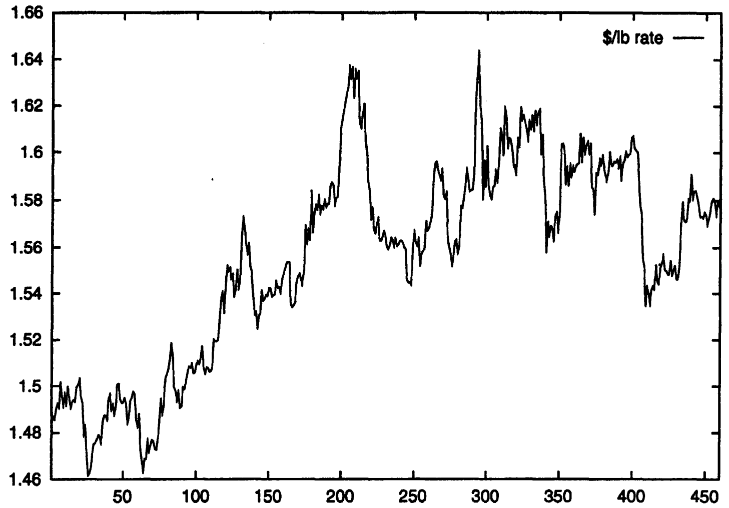

Fig. 3. $/£daily exchange rate (starting from September 13, 1993).

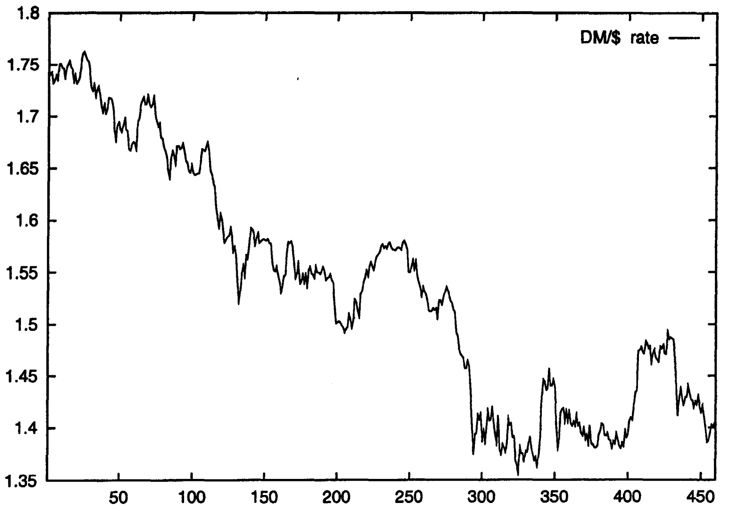

Fig. 4. DM/$ daily exchange rate (starting from September 13, 1993).

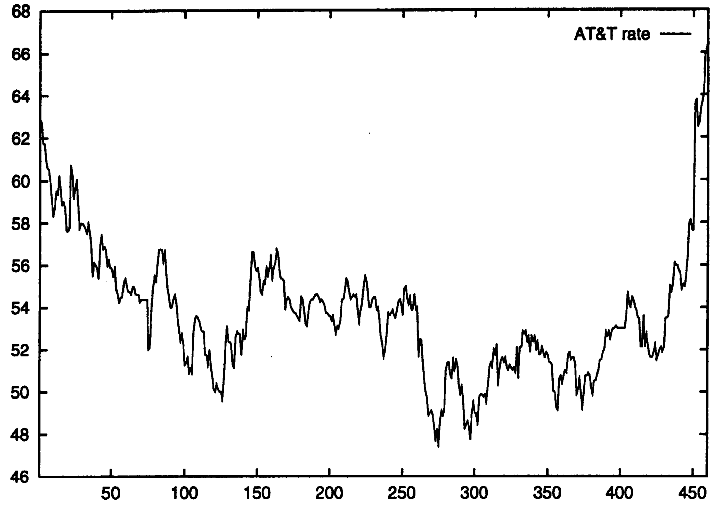

Fig. 5. AT&T stocks closing rate (starting from August 30, 1993).

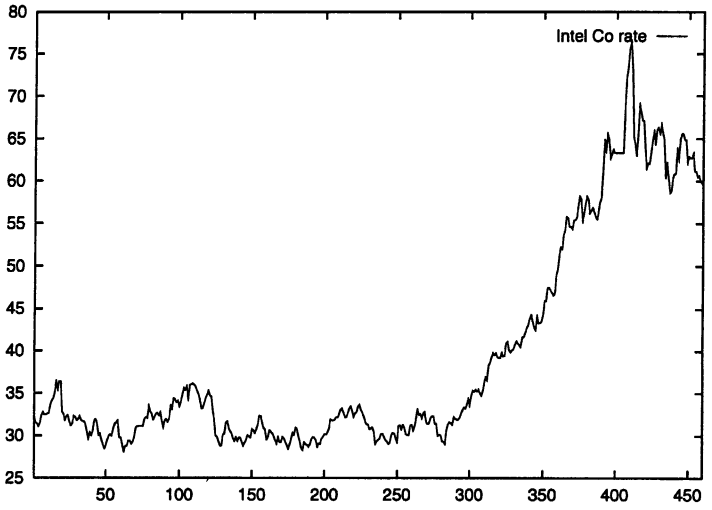

Fig. 6. Intel Co. stocks closing rate (starting from August 30, 1993).

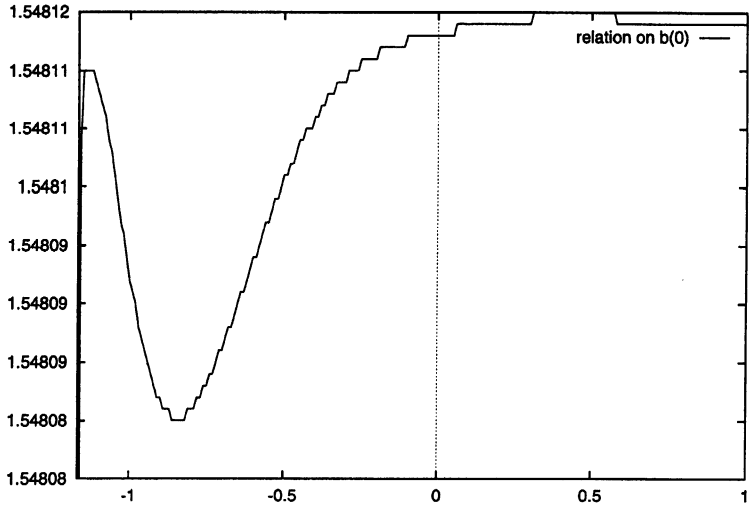

Fig. 7. Log-sum (4) as a function of the parameter $b_0 \in [-1.167, 1]$ regarding the $/£ exchange rate.

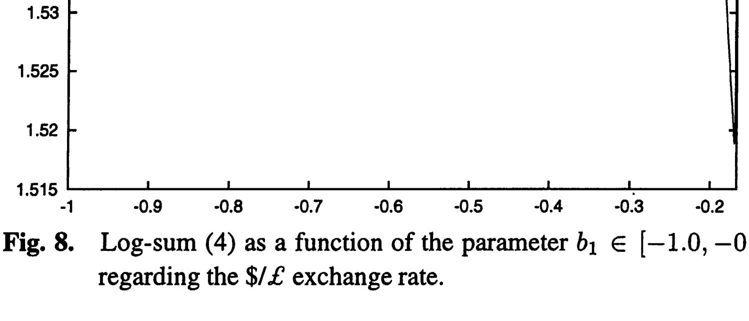

Fig. 8. Log-sum (4) as a function of the parameter $b_1 \in [-1.0, -0.167]$ regarding the $/£ exchange rate.

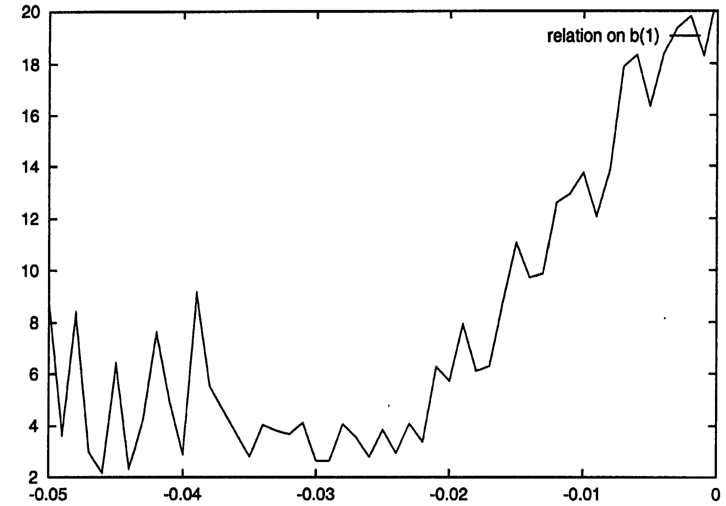

Fig. 9. Log-sum (4) as a function of the parameter $b_1 \in [-0.05, 0.]$ regarding the DM/$ exchange rate.

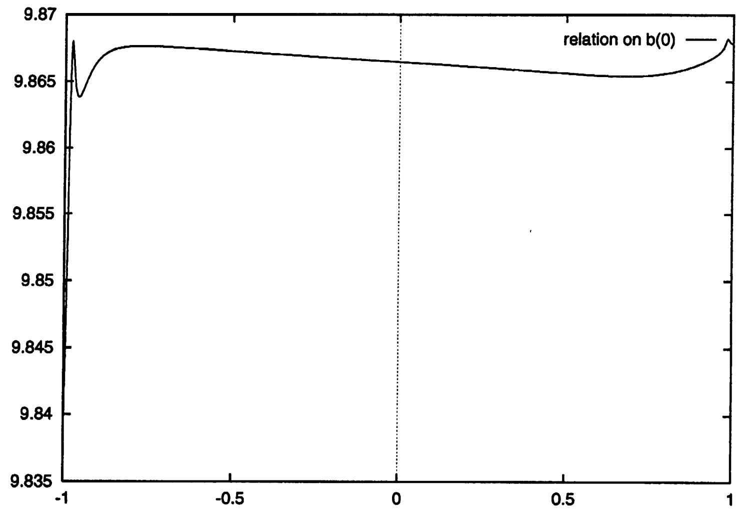

Fig. 10. Log-sum (4) as a function of the parameter $b_0 \in [-1., 1.]$ regarding the AT&T stocks closing rate.

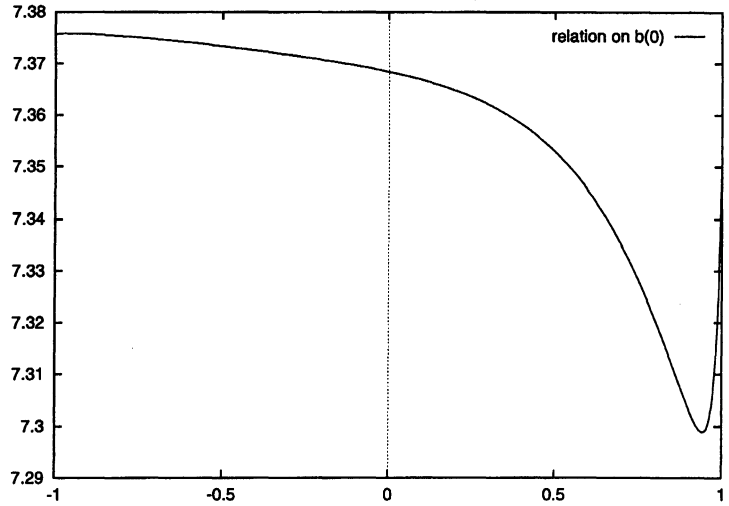

Fig. 11. Log-sum (4) as a function of the parameter $b_0 \in [-1., 1.]$ regarding the Intel Co stocks closing rate.

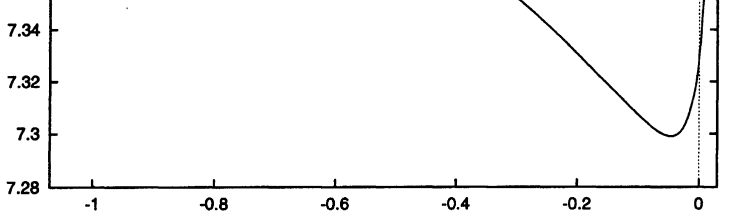

Fig. 12. Log-sum (4) as a function of the parameter $b_1 \in [-1.07, 0.03]$ regarding the Intel Co stocks closing rate.

Table 3. The initial points obtained in optimizing ARFIMA models by global algorithms

Data	b_0	b_1	$\min \log f(x)$
$/£	-1.195	-0.169	1.51675
DM/$	-1.019	0.0120	1.60065
Rial/$ (1)	0.15	0.45	0.9395
Rial/$ (2)	0.51	-0.45	0.9350
AT&T	-1.017	0.0118	9.83208
Intel Co	0.9975	0.0055	7.35681

The figures indicate the multi-modality of log-sum (4) as a function of parameters b_0, b_1 in all the three cases[5].

1.7.2. "Initial" points. The "initial" points b_0^0, b_1^0 (see Table 3) for drawing all these relations were defined using a sequence of two global methods referred to as BAYES1 and EXKOR (see [17]). BAYES1 denotes a search in accordance with a multi-dimensional Bayesian model [19]. The best result of BAYES1 is a starting point for an one-dimensional coordinate search EXKOR using an one-dimensional Wiener model, see [19]. In both the cases 1000 iterations are used.

The number of auto-regression (AR) parameters is $p = 10$, the number of moving-average (MA) parameters is $q = 2$. The reason is that increasing these numbers we did not succeed to improve the objective function (4) significantly.

Table 3 shows the estimated parameters for $/£, and DM/$ daily exchange rates, and for closing rates of AT&T and Intel Co stocks. Table 3 also shows these parameters for rial/$ monthly exchange rates for two periods: period (1) is before the Iranian revolution, and period (2) is after it.

1.8. Discussion of results. Let us to compare this result with that of traditional approaches to testing for long memory processes (see, for example, [3, 5, 4]). The traditional methods are based on the assumption of continuity: small changes in data do not cause jumps in parameter estimates. This assumption is valid in a linear regression. However, in a non-linear regression, as is the case here, the multi-modality cannot be ignored because a sum of non-convex functions (4) could be multi-modal.

[5] That means a multi-modality of sum (4), too.

In multi-modal cases, a small change in data may result in a jump of the optimal parameter value. Investigating various figures we find that slight data changes cause jumps in parameters b_0 and b_1.

The present research clearly shows that the traditional uni-modality assumption used in the linear regression models, is not sustainable for the data sets used in this study and which show evidence of non-linearity. Ignoring the multi-modality of (4) one cannot obtain reliable estimates of parameters b.

The objective of this work is merely to show a multi-modality of the problem. Therefore to save the computing time the global optimization was carried out approximately using not many of iterations. The results of global optimization were used as a starting point for local optimization. Thus we guarantee the final results at least as good as that of local optimization.

The high-accuracy global optimization is very expensive. As usual, the computing time is an exponential function of accuracy in the global optimization. Therefore, what happens after the high-accuracy global optimization of the objective function, is not yet clear. However, it seems clear that the investigation of multi-modality should be the first step in estimating parameters of non-linear regression models, including the ARFIMA ones[6]. Balancing computing expenses and accuracy of estimation is the important problem of future investigation in the fields of exchange rate prediction and global optimization.

1.9. Structural stabilization

1.9.1. Stabilization of structures of time series. The objective of the traditional time series models considered in the previous sections was to minimize the deviation from the available data. One may call these as the best fit models. The models that fit best to the past data will predict the future data well if no changes happen in the system. Otherwise, the best fit to the past data can be irrelevant or even harmful. Thus a model of stable structure is needed which is not sensitive to the system changes and thus may predict the uncertain future better eliminating the nuisence parts from the structure of the model. We shall consider the structural stabilization of the time series models R that minimizes the prediction errors in the changing environment. The available data $W = (w_t,\ t = 1,\dots,T)$ is divided into two parts $W_0 = (w_t,\ t = 1,\dots,T_0)$ and $W_1 = (w_t\ t = T_0+1,\dots,T)$. The goodness of fit is described by continuous variables C, for example, in the ARMA model (see expression (1)) $C = (a_i,\ i = 1,\dots,p,\ b_j,\ j = 1,\dots,q)$. The model structure is determined by the Boolean variables S meaning that a structural variable is equal to unit if a corresponding component of time series model is included and is equal to zero, otherwise,

[6] Meaning that the sum (4) of the ARMA model is a non-linear function of the parameters b.

for example, in the ARMA model $S = (s_i^a,\ i = 1,\dots,p,\ s_j^b,\ j = 1,\dots,q)$, $s_i^a = 1$, if the parameter a_i is included in the ARMA model and $s_i^a = 0$ if not[7]. Denote by $R_t(S, C, W)$ the predicted value of a model R with fixed parameters S, C using the data $(w_1, \dots, w_{t-1}) \subset W$. The difference between the prediction and the actual data w_t is denoted by $\varepsilon_t(S, C, W) = w_t - R_t(S, C, W)$. Denote by $C_0(S)$ the fitting parameters which minimize the sum of squared deviations $\Delta_0(S)$ using the first data set W_0 at fixed structure parameters S.

$$C_0(S) = \arg\min_C \sum_{t=1}^{T_0} \varepsilon_t^2(S, C, W), \tag{27}$$

$$\Delta_0(S) = \min_C \sum_{t=1}^{T_0} \varepsilon_t^2(S, C, W). \tag{28}$$

We stabilize the structure S by minimizing the sum of squared deviations $\Delta_0(S)$ using the different data set W_1

$$S_1 = \arg\min_S \sum_{t=T_0+1}^{T} \varepsilon_t^2(S, C_0(S), W) \tag{29}$$

$$\Delta_1(S_1) = \min_S \sum_{t=T_0+1}^{T} \varepsilon_t^2(S, C_0(S), W). \tag{30}$$

This way a trade-off is reached between the fitting and the structural parameters. The fitting parameters $C_0(S)$ provide the best fit to the first data set W_0 of some fixed structure S. One stabilizes the structure S by minimizing the prediction errors for the second set of data W_1 while using the fitting parameters $C_0(S)$ obtained from the first set of data W_0. This way the stabilized structure $S = S_1$ of R is obtained eliminating the unstable[8] parameters and parts of the time series.

The usefulness of the stabilization follows from the observation that any optimal estimate of time series parameters using the data W uniting two difrerent parts W_0 and W_1 is optimal for both parts only if all the parameters remain the same. Othervise one may obtain a better estimate eliminating the changing parameters from the model. For example in the case of changing parameters of the ARMA model the best prediction may be obtained by elimination of all the parameters except $a_1 = 1$ (see the first and the third row in Table 1).

[7] Note that describing a real data one needs diverse structures including a number of different models, not just one specific model as in this illustrative example.

[8] The parameters are regarded as unstable if they are different for different data sets.

The stable structure models will be illustrated using the time series considered in this paper. The general idea of structural stability may be directly applied to other time series models, too.

Let us consider a very simple illustration assuming that $p = 2$, $q = 0$, $w_{t-1} = 1.$, $w_{t-2} = 1.$, $a_1^0 = a_1^1 = 1$, $a_2^0 = 1.5$, $a_2^1 = 0.5$. In this case the right prediction is $w_t = 1 + 0.5 = 1.5$. The prediction using just the first data set W_0 is $w_t = 1 + 1.5 = 2.5$ and the prediction omitting the first term $a_2 = 0$ is $w_t = 1$. One may see that in this special case the first term a_1 is not stable in the sense that its elimination improves the prediction (the prediction $w_t = 1$ is closer to the "right" prediction $w_t = 1.5$ as compared with the prediction $w_t = 2.5$ obtained using the first data set).

In the next subsection the stable structure models will be illustrated using most of the time series considered in this book. The general idea of structural stability may be directly applied to other time series models, too.

1.9.2. Example

1.9.2.1. Taylor–Fourier–ARMA–ANN Model. Choosing the stable structure one should consider diverse structures including different models. In this example the deterministic part of the structure includes m terms of the Taylor series and n terms of the Fourier series. The stochastic part involves the traditional ARMA(p, q) model (see expression (1)) and a version of artificial neural network model ANN-AR(r) (see expression(22)). In such a case

$$\begin{aligned} R_t(S, C, W) = {} & \sum_{i=1}^{m} s_i^g g_i\, t^{i-1} + \sum_{i=1}^{n} s_i^{h(0)} h_i(0) \sin(s_i^{h(1)} h_i(1) \\ & + s_i^{h(2)} h_i(2)\, t) + \sum_{i=1}^{p} s_i^a a_i w_{t-i} + \sum_{i=1}^{q} s_i^b b_i \varepsilon_{t-i} \\ & + s^\beta \frac{\beta}{\sqrt{2\phi\sigma}} \int_{-\infty}^{w_t(p(\nu))} e^{-\frac{w-\mu}{\sigma}} dw, \end{aligned} \tag{31}$$

where $w_t(p(\nu)) = \sum_{i=1}^{p(\nu)} s_i^\nu a_i(\nu) w_{t-i}$. Here C is the array of $m + 3n + p + q + p(\nu) + 3$ continuous variables, namely

$$\begin{aligned} C = \big(& g_1, \dots, g_m, h_1(0), h_1(0), \dots, h_n(0), h_1(1), \dots, h_n(1), h_1(2), \dots, \\ & h_n(2), a_1, \dots, a_p, b_1, \dots, b_q, a_1(\nu), \dots, a_{p(\nu)}(\nu), \beta, \mu, \sigma\big), \end{aligned} \tag{32}$$

and S is the array of $m + 3n + p + q + p(\nu) + 3$ Boolean variables

$$S = (s_1^g, \dots, s_m^g, s_1^{h(0)}, \dots, s_n^{h(0)}, s_1^{h(1)}, \dots, s_n^{h(1)}, s_1^{h(2)}, \dots, s_n^{h(2)},$$

$$s_1^a,\dots,s_p^a,s_1^b,\dots,s_q^b,s_1^\nu,\dots,s_{p(\nu)}^\nu,s^\beta,s^\mu,s^\sigma). \tag{33}$$

One optimizes the continuous fitting variables C using the data set W_0 and applying the Direct Bayesian Approach described in[17][9]. Optimization of the Boolean structural variables S is carried out using the different data set W_1 and applying the Bayesian Heuristic Approach and the permutation heuristics described in [19] and in the next section of this paper.

At present there are no theoretical results defining the best structures. Therefore it is reasonable to start from the simplest continuous model where all the Boolean variables are zero except fixed $s_1^g = 1$ considering corresponding permutations later on. In this case the natural heuristic is the difference between the permuted state and the initial one[10].

1.9.2.2. Simple model. The described model has a great number of continuous and Boolean parameters and is difficult for optimization. Therefore we define a simplified version of the model (31) as an illustrative example.

$$\begin{aligned} R_t(S,C,W) &= s_1^g g_1 + s_2^g g_2\, t + s_1^{h(0)} h_1(0)\sin(s_1^{h(1)} h_1(1) \\ &\quad + s_1^{h(2)} h_1(2)\, t) + \sum_{i=1}^{p} s_i^a a_i w_{t-i} + \sum_{i=1}^{q} s_i^b b_i \varepsilon_{t-i}. \end{aligned} \tag{34}$$

Here C is the array of $2+3+p+q$ continuous variables, namely

$$C = (g_1,\dots,g_2,h_1(0),h_1(1),h_1(2),a_1,\dots,a_p,b_1,\dots,b_q,\beta,\mu,\sigma)$$

and S is the array of $M = 2+3+p+q$ Boolean variables

$$S = (s_1^g,\dots,s_2^g,s_1^{h(0)},s_1^{h(1)},s_1^{h(2)},s_1^a,\dots,s_p^a,s_1^b,\dots,s_q^b).$$

1.9.2.3. Simple model including lag. The presence of nearly periodic component is detected in more economical way by introducing the lag variable into the time series model. We illustrate this by the simplified model (34).

$$\begin{aligned} R_t(S,C,W) &= s_1^g g_1 + s_2^g g_2\, t + s_1^{h(0)} h_1(0)\sin\left(s_1^{h(1)} h_1(1) + s_1^{h(2)} h_1(2)\, t\right) \\ &\quad + \sum_{i=1}^{p} s_i^a a_i w_{t-i(k+1)} + \sum_{i=1}^{q} s_i^b b_i \varepsilon_{t-i(k+1)}, \end{aligned} \tag{35}$$

[9] The "linear" variables $a_1,\dots,a_p$ are optimized solving a system of linear equations similar to the system (11) for each fixed values of parameters $g, h, b, a(\nu)$.

[10] With the minus sign, because we minimize.

where the lag parameter $k = 0, \ldots, l-1,\ l \leqslant T_0/p,\ p \geqslant q$. Here C is the array of $2+3+p+q$ continuous variables, namely

$$C = \big(g_1, \ldots, g_2, h_1(0), h_1(1), h_1(2), a_1, \ldots, a_p, b_1, \ldots, b_q, \beta, \mu, \sigma\big)$$

and S is the array of $M = 2+3+p+q$ Boolean variables and one integer variable $k = 0, \ldots, l-1$. One may see that the lag is introduced only into AR and MA parts of the model and $k = 0$ means no lag. Without the lag variable the numbers of p, q should be greater than the period to detect the periodic components of time series. The model is improved by introducing several lag parameters $k = (k_1, \ldots, k_l)$

$$\begin{aligned} R_t(S, C, W) &= s_1^g g_1 + s_2^g g_2\, t + s_1^{h(0)} h_1(0) \sin(s_1^{h(1)} h_1(1) + s_1^{h(2)} h_1(2)\, t) \\ &\quad + \sum_{k_0=1}^{k_l} \sum_{i=1}^{p} s_i^a a_i w_{t-i(k_0+1)} + \sum_{i=1}^{q} s_i^b b_i \epsilon_{t-ik_0}, \end{aligned} \tag{36}$$

where the lag parameters $k_0 = 1, \ldots, k_l,\ k_l \leqslant T_0/p,\ p \geqslant q$. Here C is the array of $2+3+p+q$ continuous variables, namely

$$C = \big(g_1, \ldots, g_2, h_1(0), h_1(1), h_1(2), a_1, \ldots, a_p, b_1, \ldots, b_q, \beta, \mu, \sigma\big)$$

and S is the array of $M = 2+3+p+q$ Boolean variables and k_l integer variables $k_0 = 0, \ldots, k_l - 1$.

1.9.2.4. Algorithm. We consider the simple case of only one lag parameter k and we start from the "trivial" model where all the structural variables are zero except the $s_1^g = 1$. The model is called as trivial because the predicted value is equal to the average. Later on two types of permutations are used. In the "mutation" type the cyclic addition is carried out to all the discrete variables and the "heuristic" $h_i = v_n(i) - v_n(0)$ is defined, where $v_n(i)$ is the objective of ith permutation at nth iteration, and $v_n(0)$ is the objective of the initial state at the nth iteration. The "cross-over" type means dividing at the random point and interchanging two best $0, 1$ sequences including all the structural variables S and representing the integer lag variable in the Boolean form. Thus one defines the "baby" structures in addition to the "mutant" structures defined by cyclic addition of individual variables. We are keeping in memory the best structure obtained as the result of all the previous K permutations. The objective this structure is denoted by $f_K(x)$. Here x represents parameters of some randomized procedures used to chose the initial states for the next iteration. This means that the initial state for the next iteration will not necessarily be the best structure, because by choosing only the best structures one may stuck into some local optimum. The Bayesian Heuristic Approach [19] is used choosing the initial state for each iteration and defining the optimal randomization procedures.

The optimization of continuous variables C at fixed values is carried out using the global methods [17] for all the continuous variables except the MA variables a which are obtained by solving the linear equations at fixed values of remaining variables, like in the equations (11).

2. Optimal scheduling, Bayesian huristic model

2.1. Knapsack problem. A convenient way to explain the BHA is by applying it to a simple NP-complete problem. A good example[11] is a knapsack problem. The knapsack problem is to maximize the total value of a collection of objects when the total weight g of those objects is limited. We denote the value of the object i by c_i and the weight by g_i.

$$\max_y \sum_{i=1}^{n} c_i y_i, \tag{37}$$

$$\sum_{i=1}^{n} g_i y_i \leqslant g, \quad y_i \in \{0,1\}. \tag{38}$$

Here the objective depends on n Boolean variables x_i.

2.1.1. Exact algorithms. The simplest exact algorithm is exhaustive search of all the decisions. The decision m means that object m is selected. The exhaustive search needs $T = C\ 2^n$ time. Here the constant C does not depend on the problem. The search efficiency of exact algorithms is improved by Branch&Bound (B&B) techniques where the time-constant $C_\omega \leqslant C$ depends on the specific problem ω.

2.2. Approximate algorithms. The simplest approximate algorithm is Monte-Carlo where the decision m is taken with probability

$$r(m) = 1/m. \tag{39}$$

This algorithm converges to the exact solution with probability 1 if the number of repetitions $K \to \infty$. However the convergence is very slow.

[11] This example is good just for illustration how BHA works but not for showing the advantages of this approach. The BHA works more efficiently in scheduling problems, for example in flow-sop and batch scheduling problems, see the following sections.

2.2.1. Heuristic algorithms. Define the heuristics $h(m)$ as the specific value of object m

$$h_i(m) = \frac{c_m}{g_m}. \tag{40}$$

This is a well known and widely used "greedy" heuristics. The greedy heuristics prefer the feasible[12] object with highest heuristic $h(m)$. The greedy heuristic algorithm is very fast. However it may stuck in some non-optimal decision.

2.2.2. Randomized heuristic algorithms. A simple way to force the heuristic algorithm out of such non-optimal decisions is by taking decision m with probability

$$r_1(m) = \frac{h(m)}{\sum_i h(i)}. \tag{41}$$

This algorithm is better than greedy one, because it converges with probability one if K is large. It is better than Monte-Carlo, too, since it includes an expert knowledge by making the decision probabilities dependent on heuristics.

An open question is why consider only the linear randomization (41) ignoring non-linear ones. Expressions (42) and (43) define an example of a set of randomization functions

$$r_l(m) = \frac{h^l(m)}{\sum_i h^l(i)}, \quad l = 0, 1, 2, \dots, L, \tag{42}$$

and

$$r_\infty(m) = \begin{cases} 1, & \text{if } h(m) = \max_i h(i), \\ 0, & \text{otherwise,} \end{cases} \tag{43}$$

Here $l = 0$ denotes Monte-Carlo component, $l = 1$ and $l = 2$ defines linear and quadratic components of randomization. The index ∞ denotes the greedy heuristics with no randomization.

One may define the best randomization function empirically by considering each randomization function separately and estimating the quality of a randomization function by the best decision obtained applying this function K times. That is a traditional way.

We may solve the same problem in a more general set-up by considering a "mixture" of different randomization functions. The mixture means using a randomization function r_l with some probability $x(l)$. Denote the probability distribution as $x = (x(l),\ l = 0, 1, \dots, L, \infty)$. Denote by $f_K(x)$ the best decisions

[12] Satisfying inequality (38).

obtained using K times a mixture x. This way we extend the set of possible decisions x from the discrete set $x(l) \in \{0, 1\}$, $l = 1, \dots, L, \infty$ to a continuous one. That is important because the best results we often obtain using a mixture of functions but not the single one.

The difficulty is that $f_K(x)$ is stochastic and, usually, multi-modal function. A natural way to consider such problems is by regarding a functions $f_K(x)$ as a sample of a stochastic function (defined by some a priori distribution) and obtaining the next observation by minimizing the expected deviation from the exact solution. This technique is called Bayesian Heuristic Approach (BHA).

2.3. Flow-shop problem. The flow-shop problem is a simple example of large and important family of scheduling problems. We denote by J and S the set of jobs and machines. Denote by $\tau_{j,s}$ the duration of operation (j, s), where $j \in J$ denotes a job and $s \in S$ denotes a machine.

Suppose that the sequence of machines s is fixed for each job j. One machine can do only one job at a time. Several machines cannot do the same job at the same moment. The decision $d_i(j) \in D_i$ means the start of a job $j \in J_i$ at stage i. We define the set of feasible decisions D_i as the set J_i of jobs available at the stage i conforming to the flow-shop rules.

The objective function is the make-span v. Denote by $T_j(d)$ the time when we complete job j (including the gaps between operations) using the decision sequence d. Then the make-span for d is

$$v(d) = \max_{j \in J} T_j(d). \tag{44}$$

2.4. Algorithm

2.4.1. Permutation schedule. We can see that the number of feasible decisions for the flow-shop can be very large. The number can be reduced by considering only smaller subset of schedules, the so-called permutation schedules.

The permutation schedule is a schedule with the same job order on all machines. Such a schedule can be defined by fixing job indices 1,2,...,n. We assume the first operation to be on the first machine, the second on the second, and so on. The schedule is transformed by a single permutation of job indices. It is generally assumed that permutation schedules approach the optimal decision sufficiently closely and are easier to implement (see [1]).

Denote

$$\tau_j = \sum_{s=1}^{|S|} \tau_{j,s},$$

Table 4. The results of Bayesian methods using Longer-Job heuristics (45)

$R = 100$, $K = 1$, $J = 10$, $S = 10$, and $O = 10$					
Randomization	f_B	d_B	x_0	x_1	x_2
Delta	6.183	0.133	0.283	0.451	0.266
Taylor 3	6.173	0.083	0.304	0.276	0.420
CPLEX	12.234	0.00	—	—	—

where $|S|$ stands for the number of machines. We define τ_j as the length of the job j.

2.4.2. Heuristics. Define the Longer-Job heuristics

$$h_i(j) = \frac{\tau_j - A_i}{A^i - A_i} + a. \tag{45}$$

Here

$$A_i = \min_{j \in J_i} \tau_j, \qquad A^i = \max_{j \in J_i} \tau_j, \ a > 0. \tag{46}$$

The priority rule (45) prefer a longer job. We optimize a stochastic function $f_K(x)$ defining the minimal make-span (see (44)) found as a result of K repetitions.

2.4.3. Results. Table 4 illustrates the results of the Bayesian method in hundreds of time units after 100 iterations using the Longer-Job heuristic (45) and different randomization procedures. Assume that $J = S = O = 10$, where J, S, O are the number of jobs, machines, and operations, respectively. Lengths and sequences of operations are generated as random numbers uniformly distributed from 0 to 99. The expectations and standard deviations are estimated by repeating optimization of a randomly generated problem 40 times. In Table 4 the symbol f_B denotes a mean, and d_B denotes a standard deviation of make-span. The "Delta" denotes randomization including terms $l = 0, 1, 2, \infty$ (see expressions (42) and (43)), the "Taylor 3" denotes randomization (42) where the number of terms is $L = 3$, and "CPLEX" denotes the results of the well known general discrete optimization software after 2000 iterations (one CPLEX iteration is comparable to a Bayesian observation). The bad results of CPLEX show that the standard MILP technique is not efficient in solving this specific problem of discrete optimization. It is not yet clear how much one improve the results using specifically tailored B&B.

3. Brides problem, sequential statistical decisions model

3.1. Introduction. The Brides problem is a good example of sequential statistical decisions, see [23, 24]. The dynamic programming is a conventional technique optimizing the sequential decisions, see [24]. The usual way applying dynamic programming is to develop specific algorithms for a given family of problems.

The Bride's problem is to maximize the average utility of marriage by the optimal choice of groom. Denote the actual goodness of the groom $i = 1, \dots, N$ by ω_i. Denote by s_i the brides impression about the groom i. Denote an a priori probability density of goodness ω_i as $p(\omega_i)$. Denote a probability density of impression s_i as $p_s(s_i|\omega_i)$. Assume that goodness of different grooms are independent and identically distributed as well as corresponding impressions. This means that an a priori goodness is

$$p(\omega_i, \omega_j) = p(\omega_i)p(\omega_j) = p(\omega), \tag{47}$$

and an impression s_i given the goodness ω_i is

$$p_s(s_i, s_j|\omega) = p_s(s_i|\omega)p(s_j|\omega) = p(s|\omega). \tag{48}$$

Assume the Gaussian distributions

$$p(\omega) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-1/2(\frac{\omega-a}{\sigma_0})^2} \tag{49}$$

and

$$p(s|\omega) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1/2(\frac{s-\omega}{\sigma})^2}. \tag{50}$$

Applying the Bayesian formula [2] we define an a posteriori probability density of goodness ω given the impression s

$$p(\omega|s) = \frac{p(s|\omega)p(\omega)}{p_s(s)}. \tag{51}$$

Here

$$p_s(s) = \int_{-\infty}^{\infty} p(s|\omega)p(\omega)d\omega. \tag{52}$$

Denote by d_i the brides decision regarding the groom i

$$d_i = \begin{cases} 1, & \text{if bride marry the groom } i, \\ 0, & \text{otherwise,} \end{cases} \tag{53}$$

Suppose that

$$\sum_{i=1}^{m} d_i = 1. \tag{54}$$

3.2. Bellman's equations. Denote by $u_N(s)$ the expected utility function if the impression of the last groom[13] is s

$$u_N(s) = \int_{-\infty}^{\infty} \omega p(\omega|s) d\omega \tag{55}$$

Denote by u_{N-1} the expected utility if the impression of the $(N-1)$th groom is s and the bride is making the optimal decision $d = d_{N-1}(s) \in D_{N-1}$

$$u_{N-1}(s) = \max_d \big(du_N(s) + (1-d)u_N\big), \tag{56}$$

$$d_{N-1}(s) = \arg\max_d \big(du_N(s) + (1-d)u_N\big). \tag{57}$$

Here

$$u_N = \int_{-\infty}^{\infty} u_N(s) p_s(s) ds, \tag{58}$$

and D_{N-1} is a set feasible decisions

$$D_{N-1} = \begin{cases} 0 \text{ and } 1, & \text{if} \quad g_{N-1} = 0, \\ 0, & \text{if} \quad g_{N-1} = 1. \end{cases} \tag{59}$$

Here g_{N-1} is a marriage index

$$g_{N-1} = 1 - \sum_{i=1}^{N-2} d_i. \tag{60}$$

Following the same pattern we define the expected utility if the impression of the $(N-n)$th groom is s and the bride is making the optimal decision $d = d_{N-n}(s) \in D_{N-n}$

$$u_{N-n}(s) = \max_d \big(du_N(s) + (1-d)u_{N-n+1}\big), \tag{61}$$

$$d_{N-n}(s) = \arg\max_d \big(du_N(s) + (1-d)u_{N-n+1}\big), \tag{62}$$

where

$$u_{N-n+1} = \int_{-\infty}^{\infty} u_{N-n+1}(s) p_s(s) ds. \tag{63}$$

[13] By the "last groom" we mean the groom which the bride marry.

Solving these recurrent equations one defines the sequence of optimal decision functions $d_{N-n}(s)$ and the expected utilities $u_{N-n}(s)$ for all possible impressions $s \in (-\infty, \infty)$ for any number $n = 1, \ldots, N-1$. We cannot do that in continuous case. Therefore we use discrete approximation

3.3. Discrete approximation. Replacing the integrals by sums we obtain from expression (56)

$$u_N(s) = 1/K \sum_{k=1}^{K} \omega_k p(\omega_k|s), \tag{64}$$

from expression (58)

$$u_N = 1/K \sum_{k=1}^{K} u_N(s_k) p_s(s_k), \tag{65}$$

and from expression (63)

$$u_{N-n+1} = 1/K \sum_{k=1}^{K} u_{N-n+1}(s_k) p_s(s_k). \tag{66}$$

Here $\omega_k \in [-M, M]$, $\omega_1 = -M$, $\omega_K = M$ and $s_k \in [-M, M]$, $s_1 = -M$, $s_K = M$.

Solving the recurrent equations by discrete approximation one defines the sequences of optimal decision functions $d_{N-n}(s_k)$ and the expected utilities $u_{N-n}(s_k$ for all possible impressions $s_k \in [-M, M)$ for all the numbers $n = 1, \ldots, N-1$. We can do that by defining a set of arrays determining how the optimal decisions d and the corresponding expected utilities u depends on the possible impressions s_k, $k = 1, \ldots, K$. This way we avoid the repeated calculations at the expense of keepinglarge arrays. The number of K is determined by the accuracy needed.

3.4. Including the waiting losses The waiting losses are important in many real-life sequential decision problems. Denote by c the loss of waiting for next groom. Subtracting this parameter from the Bellman equation (56) defining the optimal decisions of $(N-1)$th groom if the impression is s, one obtains

$$u_{N-1}(s) = \max_d \left(d u_N(s) + (1-d)(u_N - c)\right), \tag{67}$$

$$d_{N-1}(s) = \arg\max_d \left(d u_N(s) + (1-d)(u_N - c)\right). \tag{68}$$

In a similar way one defines the expected utility if the impression of the $(N-n)$th groom is s and the bride is making the optimal decision $d_{N-n}(s) \in D_{N-n}$

$$u_{N-n}(s) = \max_d \left(du_N(s) + (1-d)(u_{N-n+1} - nc)\right), \tag{69}$$

$$d_{N-n}(s) = \arg\max_d (du_N(s) + (1-d)(u_{N-n+1} - nc)). \tag{70}$$

The other expressions remains the same. The last expression shows that increasing the waiting losses c we can make the bride's problem less sensitive[14] to a number N which is unknown, as usual.

3.5. Non-linear case. The expression (55) was defined assuming the linear Bride's utility function. It was supposed that the Bride's utility is equal to the goodness of groom $u(\omega) = \omega$. If not then expression (55) should be replaced by the following integral

$$u_N(s) = \int_{-\infty}^{\infty} u(\omega)p(\omega|s)d\omega. \tag{71}$$

REFERENCES

1. Baker, K. R. (1974). *Introduction to Sequencing and Scheduling*. John Wiley & Sons, New York.
2. Bayes, T. (1983). An essay towards solving a problem in the doctrine of chances. *Phil. Transactions of Royal Society*, **53**, 370–418.
3. Cheung, Y.W. (1993). Long memory in foreign exchange rates. *Journal of Business and Economic Statistics*, **1**, 93–101.
4. Cheung, Y.-W., and K. Lai (1993). Fractional co-integration analysis of purchasing power parity. *Journal of Business and Economic Statistics*, **1**, 103–112.
5. Cheung, Y.-W., and K. Lai (1993). Long-run purchasing power parity during the recent float. *Journal of International Economics*, **34**, 181–192.
6. Diebold, F.X., and G. D. Rudebusch (1989). Long memory and persistence in aggregate output. *Journal of Monetary Economics*, **24**, 189–209.
7. Fox, R., and M. Taqqu (1986). Large-sample properties of parameter estimates for strongly dependent stationary gaussian time series. *Annals of Statistics*, **14**, 517–532.
8. Geweke, J., and S. Porter-Hudak (1983). The estimation and application of long memory time series models. *Journal of Time Series Analysis*, **4**, 221–238.
9. Hosking, J.R.M.(1981). Fractional differencing. *Biometrika*, **68**, 165–176.
10. Hosking, J.R.M. (1984). Modeling persistence in hydrological time series using fractional differencing. *Water Resources Research*, **20**, 1898–1908.

[14] Increasing c one increases the second component nc as compared to the first one u_{N-n+1} which depends on N.

11. Janacek, G. (1982). Determining the degree of differencing for time series via the long spectrum. *Journal of Time Series Analysis*, **3** 177–188.
12. Ko K.-I. (1991). *Complexity Theory of Real Functions*. Birkhauser, Boston.
13. Koop, G., E. Ley, J. Osiewalski, M.F.J. Steel. (1994). Bayesian analysis of long memory and persistence using arfima models. *Technical report*, Department of Economics, University of Toronto.
14. Li, W.K., and A. I. McLeod. (1986). Fractional time series modelling. *Biometrika*, **73**, 217–221.
15. Liu, J. (1989) On the existence of general multiple bilinear time series. *Journal of Time Series Analysis*, **10**, 341–355.
16. Mockus, J. (1967). *Multimodal Problems in Engineering Design*. Nauka, Moscow (in Russian).
17. Mockus, J. (1989). *Bayesian Approach to Global Optimization*. Kluwer Academic Publishers, Dordrecht-London-Boston.
18. Mockus, J. (1997). A set of examples of global and discrete optimization: application of bayesian heuristic approach I. *Informatica*, **8**(2), 237–264.
19. Mockus, J., W. Eddy, A. Mockus, L. Mockus, G. Reklaitis (1997). *Bayesian Heuristic Approach to Discrete and Global Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
20. Mockus, J., and A. Soofi (1995). The long-run economic relationships: An optimization approach to fractional integrated and bilinear time series. *Informatica*, **6**, 61–70.
21. Subba Rao, T., and M.M. Gabr (1984). An introduction to bispectral analysis and bilinear time series models. In *Lecture Notes in Statistics*, Vol. 24. Springer-Verlag, Berlin.
22. Sowel, F. (1992) Maximum likelihood estimation of stationary univariate fractionally integrated models. *Journal of Econometrics*, **53**, 165–188.
23. Wald, A. (1947). *Sequencial Analysis*. J. Wiley, New York.
24. Wald, A. (1950). *Statistcal Decision Functions*. J. Wiley, New York.

Received October 1997

J. Mockus graduated Kaunas Technological University, Lithuania, in 1952. He got his Doctor habilitus degree in the Institute of Computers and Automation, Latvia, in 1967. He is a head of Optimal Decision Theory Department, Institute of Mathematics and Informatics, Vilnius, Lithuania, and professor of Kaunas Technological University. Present research interests include theory, applications and software of the Bayesian heuristic approach to global and discrete optimization

GLOBALINIO IR DISKRETINIO OPTIMIZAVIMŲ PAVYZDŽIŲ RINKINYS: BAJESO HEURISTINIŲ METODŲ TAIKYMAS. II

Jonas MOCKUS

Dėstant operacijų tyrimą svarbu susipažinti su lošimų, naudingumo, eilių, tvarkaraščių bei nuoseklių sprendimų teorijomis. Staripsnyje šios teorijos iliustruojamos bei parodomas jų ryšys su globaliniu optimizavimu, nagrinėjant aštuonis pavyzdžius. Visi pavyzdžiai formuluojami lengvai suvokiamais įvairių specialybių studentams terminais, tačiau kiekvienas iš jų atstovauja svarbioms uždavinių šeimoms. Todėl aprašomi modeliai bei jų optimizavimo algoritmai gali būti įdomūs ir patyrusiems atitinkamų sričių ekspertams.