INFORMATICA, 2006, Vol. 17, No. 2, 259–278

Investigation of Examples of E-Education Environment for Scientific Collaboration and Distance Graduate Studies, Part 1

Jonas MOCKUS

Institute of Mathematics and Informatics
Akademijos 4, 08663 Vilnius, Lithuania
e-mail: jmockus@gmail.com

Received: January 2005

Abstract. The objective is to investigate two emerging information technologies in graduate studies and scientific cooperation. Internet is the first technology. The open source is the second. They help each other in many ways. We investigate the joint influence of both.

Results of complexity theory show the limitations of exact analysis. That explains popularity of heuristic algorithms. It is well known that efficiency of heuristics depends on the parameters. Thus we need some automatic procedures for tuning the heuristics. That helps comparing results of different heuristics. This enhance their efficiency, too.

An initial presentation of the basic ideas is in (Mockus, 2000). Preliminary results of distance graduate studies are in (Mockus, 2006a). Examples of optimization of sequential statistical decisions are in (Mockus, 2006b).

In this paper the theory and applications of Bayesian Heuristic Approach are discussed. In the next paper examples of Bayesian Approach to automated tuning of heuristics will be regarded. The examples of traditional methods of optimization including applications of linear and dynamic programming will be investigated in the last paper. These papers represents three parts of the same work. However each part can be read independently.

All the algorithms are implemented as platform independent Java applets or servlets. Readers can easily verify and apply the results for studies and for real life optimization models.

The information is on the main web-site `http://pilis.if.ktu.lt/∼jmockus` and four mirrors.

Key words: Bayesian approach, global optimization, tuning heuristics, scientific collaboration, distance studies, e-education.

1. Introduction

The rapid development of the internet power accumulating knowledge and providing communication services challenge the traditional way of studies. The traditional way based on textbooks and formal lectures is natural for theoretical studies of mathematics, physics etc. That is not true for new technologies of informatics and computer sciences. Obviously a different approach is needed. The problem is to define the right approach.

The objective is to investigate potential possibilities of two emerging information technologies on graduate studies and scientific cooperation. Internet is the first technology. The open source is the second. They help each other in many ways. We investigate the joint influence of both.

The general properties of the Internet are the speed and the convenience of sharing scientific information. For scientific collaboration the possibility to run software developed by colleagues directly by Internet is essential. We can test directly other results by running their software with our data. Therefore algorithms, software and results published in the scientific papers can be investigate independently. This possibility is not widely used yet. But the potential appears great. In this sense Java applets are similar to proofs of theorems in mathematics since we test theorems by reading the proofs. The snapshots of graphical user interfaces are useful for testing the results, too. The snapshots help to do calculations exactly as authors intended.

In a limited way the experimental approach can be applied to asymptotic results, too. We can test actual convergence process by corresponding computer simulation.

That can be done by both the commercial and the open software. However the efficiency is not the same. It is well known that the commercial software works best in the proprietary environment, as usuall. That is natural.

Java is a platform independent language running software on remote computers. The others are perl, python, php. However Java is more efficient for scientific calculations.

In the Internet environment computer simulation is the main tool of experimental research. The well known results of algorithm complexity show the limitations of exact analysis. That explains popularity of heuristic algorithms.

Investigating heuristic algorithms subjective factors are important. It is well known that efficiency of heuristics depends on some parameters. Researchers often work hard to define the best parameters for the proposed heuristic method before submitting a paper. Othervise the paper may be rejected. Published results reflects not just the quality of proposed heuristic method but authors experience, too. Thus we need some automatic procedures for tuning the parameters of heuristics. That helps comparison of different heuristics. This enhance their efficiency, too.

A problem is to define the reasons explaining the fast open source development without any apparent financial rewards. A contributing element is the natural need of self expression. Another element is a desire to show others what we can do. The first element is useful for self esteem. The second helps building a positive professional image.

This was noticed by commercial software companies, too. Some well known software companies support the development of open source software anticipating a profit from services. Some users may run the free software completely by themselves. Others may prefer to pay for services and help.

Results are suprising. The rate of some major open source shares are growing faster comparing with the commercial ones. An IBM expert explained this as a general trend to adapt the software to the specific needs instead of using ’out-of-shelf’ products. Then profits of software companies are generated mainly by service sectors.

Another explanation is security. The open source software helps to locate and destroy malicious software directly without asking for any outside help. Desirable degree of safety is defined flexibly just by limiting rights of different users including ourselves.

Rates of shares of commercial Microsoft and open-source Red-Hat are shown in the Fig. 1. Shares of Microsoft and Google are compared in Fig. 2. Note that the rate scale is logarithmic.

Development and applications of the open source software is new and dynamic field. Important changes often happen in days and weeks. Thus some new non-traditional approaches of education should be investigate to meet the needs of open software adequately.

An initial presentation of the basic ideas is in (Mockus, 2000). Preliminary results of distance graduate studies are in (Mockus, 2006a). Examples of optimization of sequential statistical decisions are in (Mockus, 2006b).

It is difficult to regard the problem in general. Therefore we investigate a set of relevant examples. To make this a part of more general E-education environment one needs a theoretical background and some basic software tools first. All the examples should be united by some general concept.

We regard various examples as optimization models. That is the general concept. First we investigate heuristic algorithms that reflects real life conditions. Comparing various heuristics and improving the efficiency we need specific optimization methods. A convenient theoretical concept is the Bayesian approach. We apply this approach for automatic tuning of heuristic parameters and for search of optimal mixtures of heuristics.

Doing that in the Internet environment a specific software framework is developed using Java. Students and researchers can conveniently include their own examples as separate tasks of this unified framework. Representing the optimization results specific graphical analysis objects are added to some general display tools. New methods developed by users can be easily included in addition to the Bayesian ones.

In this paper the theory and applications of Bayesian Heuristic Approach are discussed. In the next paper examples of Bayesian Approach to automated tuning of heuristic parameters will be regarded. The examples of traditional methods of optimization including applications of linear and dynamic programming will be investigated in the last paper. These papers represents three parts of the same work. However each part can be read independently.

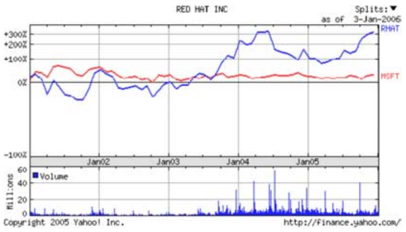


Fig. 1. Comparing RedHat/Microsoft rates.

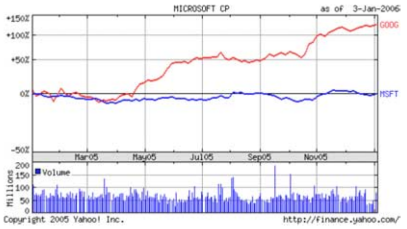


Fig. 2. Comparing Microsoft/Google rates.

All three papers show how optimization models can be implemented and updated by graduate students themselves. That reflects the usual procedures of the open source development. This way students not just learn the underlining model but obtain the experience in the development of open source software. The step-by-step improvement of the model and software is at least as important as the final result.

This way we accumulate some experience in the completely new field of education when all the information can be easily obtained by internet. The internet users are filtering and transforming the information to meet their own objectives, to build their own models. Here creative approach is needed. No well defined patterns and no well tested models yet. The natural way of research is by computer experimentation. This approach is convenient for scientific collaboration, too. We meet similar patterns in collective development of scientific projects. Here researchers need fast ways to test and to apply results obtained by the colleagues.

The development of open source software is regarded as "quite revolution". That contradicts many traditional notions in business administration and economics. Excellent results appear without any noticeable economic incentives. Sometimes open source software exceeds the commercial products. This trend is reflected by dynamics of share rates. Two examples are shown in Figs. 1 and 2.

In the papers algorithms and models are described with references to web-sites of on-line models. Examples of economic, social, and engineering models are regarded as optimization problems. Simplified versions of these models are presented for better understanding. The computing time is reasonable, as usual.

No "perfect" examples in these web-sites. All examples has some advantages and some disadvantages. Improvement of "non-perfect" models is useful both for students and for colleagues. The main objective of this paper is to help establish scientific collaboration in the Internet environment with distant colleagues and students by creating an environment of E-education and scientific collaboration in the fields related to optimization.

Bayesian Approach (BA)

Tuning of heuristics is regarded as an optimization problem. We are looking for such parameters $x \in R^m$ of heuristics $h = h(x)$ that provide best results. Denote the original function to be optimized as $\phi(\omega)$, $\omega \in R^{m_o}$. We know just that this function belongs to some family Φ of functions. Thus we cannot define the optimization quality by just a single sample $\phi(\omega) \in \Phi$. All the family Φ have to be regarded. Assume that search time is limited and defined as the number n of iterations.

Denote by (ϕ^n, ω^n) the results obtained applying n times a heuristic h to a function $\phi(\omega) \in \Phi$. Here $\phi^n = \phi(\omega^n)$ and $\omega^n = \omega^n(h)$, where $\omega^n(h)$ is the final decision of heuristics h after n iterations. Formally heuristic h after n iteration transforms the original function $\phi(\omega) \in \Phi$ into another function $f(x) \in F$. Here $f(x) = \phi(\omega^n(h(x)))$ belongs to a family F of functions ϕ transformed by heuristics h. Transformed function

$f(x)$ shows how the value of the original function $\phi(\omega)$ obtained applying n times the heuristic h depends on the heuristic parameters x.

Denote results of nth iteration as (y^n, x^n). Here $y^n = f(x^n)$ and $x^n = (x_1^n, \dots, x_m^n)$. Using the same heuristics parameter x we obtain different results $f(x)$ depending on which sample function ϕ optimize. Thus here is optimization under uncertainty. In the optimization under uncertainty two different approaches are widely used: the Min-Max and the Bayesian.

The Min-Max approach minimizes the maximal deviation from the solution. Here the worst case is regarded. The Bayesian approach searches for minimal expected deviation. Thus the average case is investigated. If the family F is a small set of well defined functions $f(x)$ then the Min-Max approach is efficient. Otherwise we regard the average case because the worst case is too bad.

Applying the Bayesian approach we fix a prior distribution P on a set F of functions $f(x)$. We need that defining average results. A prior distribution is transformed into a posterior distribution P^n on a subset F^n of the family of F. We can do that by conditional probabilities.

$$P^n(y, x) = P\big(f(x) < y \mid f(x^i) < y^i,\ i = 1, \dots, n\big). \tag{1}$$

Here $P(f(x) < y | f(x^i) < y^i,\ i = 1, \dots, n)$ shows how the probability P of event $f(x) < y$ depends on calculation results $y^1, \dots, y^n$.

Now we can define a risk function $R_0(x)$. The risk function $R_0(x)$ shows how the expected deviation $r(x)$ from the solution depends on parameters x. Here $r(x) = f(x) - f(x^*)$, where x^* is the optimal and x is a current values of heuristic parameters. Owing to linearity of deviation $r(x)$ the risk function $R_0(x)$ can be expressed as the difference $R_0(x) = \mathbf{E}f(x) - \mathbf{E}f(x^*)$. The second component is a constant associated with the set F and a prior distribution P. We see that $\mathbf{E}f(x^*)$ does not depend on the decisions x therefore we disregard this component. Then we can define the risk function as the expected value of $f(x)$

$$R(x) = \mathbf{E}f(x) = \int_F f(x)\, \mathrm{d}P(f(x)). \tag{2}$$

The Bayesian Approach (BA) is defined by fixing a prior distribution P on a set of functions $f(x)$ and by minimizing the risk function $R(x)$ (DeGroot, 1970; Mockus, 1989; Diaconis, 1988; Berger, 1985; Kadane and Wasilkowski, 1985). The risk function shows how the average deviation depends on parameters x. The distribution P is regarded as a stochastic model of $f(x)$, $x \in R^m$, where $f(x)$ may be a deterministic or a stochastic function. This is possible because using Bayesian approach uncertain deterministic functions can be regarded as some stochastic functions (Lindley, 1971; DeGroot, 1970; Savage, 1954; Fine, 1973; Zilinskas, 1986). That is essential feature of the Bayesian approach in this setup. For example, if several values of some deterministic function $y_i = f(x_i)$, $i = 1, \dots, n$ are known then the level of uncertainty can be represented as the conditional standard deviation $s_n(x)$ of the corresponding stochastic function $f(x) = f(x, \nu)$

where ν is a stochastic variable. The aim of BA is to provide as small average deviation as possible.

The difficulties depends on the heuristics. In general a heuristic is some decision rule defining how the next point of observation depends on observed values. Observation means calculation of $\phi(\omega)$ at fixed ω. Simplest are passive heuristics when all the calculation points $(\omega_1, \dots, \omega_n)$ are fixed (Traub *et al.*, 1988; Traub and Wozniakowski, 1992; Packel and Wozniakowski, 1987; Sukharev, 1971). Well known examples are 'grids', both deterministic and Monte Carlo. In this paper we investigate sequential heuristics where next observation depends on observed results (Wald, 1947; Wald, 1950; Bellman, 1957). Regarding these heuristics we meet difficulties.

Solutions of recurrent equations of sequential statistical analysis are defined just in some special problems (Wald, 1947; Shiryayev, 1978). No general solution is known.

We by-pass this difficulty by reducing multi-stage decision function to simple 'two-step' rule. Here defining coordinates of the next observation we assume that next observation will be the last one. And so on until the end.

In the Gaussian model, assuming (Mockus, 1989) that the $(n+1)$th observation is the last one

$$R(x) = \frac{1}{\sqrt{2}\pi s_n(x)} \int_{-\infty}^{+\infty} \min(c_n, z) \mathrm{e}^{-\frac{1}{2}(\frac{y - m_n(x)}{s_n(x)})^2} \, \mathrm{d}z. \tag{3}$$

Here $c_n = \min_i y_i - \varepsilon$, $y_i = f(x_i)$. $m_n(x)$ is a conditional expectation with respect to observed values y_i, $i = 1, \dots, n$. $s_n^2(x)$ is a conditional variance, and $\varepsilon > 0$ is a correction parameter. The minimum of the risk function $R(x)$ is obtained (Mockus, 1989) at the point

$$x_{n+1} = \arg\max_x \frac{s_n(x)}{m_n(x) - c_n}. \tag{4}$$

The Wiener process (Kushner, 1964a; Kushner, 1964b; Saltenis, 1971; Torn and Zilinskas, 1989) is a convenient stochastic model in the one-dimensional case $m = 1$.

Fig. 3 shows the conditional expectation, the conditional standard, and the risk function with respect to available evaluations.

The Wiener model implies continuity of almost all sample functions $f(x)$. The model assumes that increments $f(x_4) - f(x_3)$ and $f(x_2) - f(x_1)$, $x_1 < x_2 < x_3 < x_4$, are stochastically independent. Here $f(x)$ is Gaussian $(0, \sigma * 2(x)$, where $\sigma * 2(x)$ is variance, at any fixed $x > 0$.

Note, that the Wiener process originally provided a mathematical model of a particle in the Brownian motion.

Calculations of conditional probabilities (1) are difficult, in general. The problem is keeping consistency conditions. In a special one-dimensional case this problem is solved by Wiener model (Zilinskas, 1978b; Zilinskas, 1978a; Torn and Zilinskas, 1989; Calvin and Zilinskas, 2000; Kushner, 1964b; Saltenis, 1971; Mockus and Zilinskas, 1972). The Wiener model assumes that function $f(x)$ to be optimized is a sample of some Wiener process. Here we exploit the Markovian property of the Wiener process providing consistency of the stochastic model by relatively simple calculations.

The Wiener model can be extended to many dimensions, too (Mockus, 1989). However, the Markovian property disappears. Therefore approximate model is designed by replacing the traditional Kolmogorov consistency conditions. These conditions require the inversion of matrices of nth order for computing conditional expectations $m_n(x)$ and conditional variances $s_n^2(x)$.

Replacing the regular consistency conditions by:

– continuity of the risk function $R(x)$,

– convergence of x_n to the global minimum,

– simplicity of expressions of $m_n(x)$ and $s_n(x)$,

the following simplified expression of $R(x)$ is obtained using the results of (Mockus, 1989)

$$R(x) = \min_{1\leqslant i\leqslant n} z_i - \min_{1\leqslant i\leqslant n} \frac{\|x - x_i\|^2}{z_i - c_n}. \tag{5}$$

Simplified Bayesian algorithm minimizes this risk function

$$x(n+1) = \arg\min_x R(x). \tag{6}$$

That means we replace a single stochastic function by a sequence of stochastic functions. This is a natural because the “two-step” decision procedure means a similar approximation. Thus the terms “conditional” expectation and “conditional” variance do not corespond to their standard meaning. Fig. 4 shows that these parameters differ considerably from their standard counterparts in Fig. 3. However the risk functions are similar in the both models. The points of the next observations are close. That is what matters.

The Bayesian statistical methods are designed to optimize decisions when not many observations are available. The good asymptotic behavior is not the aim. Nevertheless there are some asymptotic results, too. Expressions (Mockus, 1989; Mockus *et al.*, 1997) show how the simplified Bayesian algorithm (6) converges to the global minimum of a

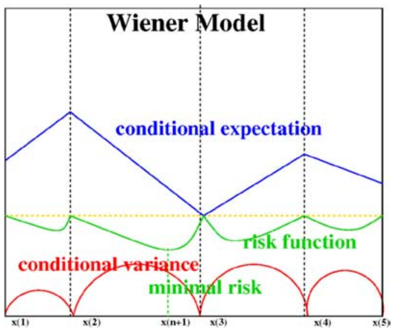


Fig. 3. Wiener model.

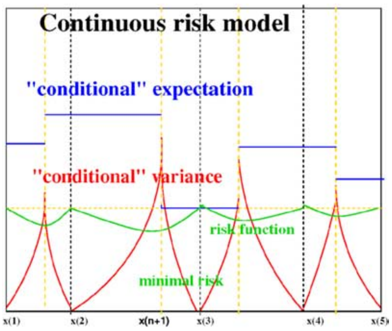


Fig. 4. Continuous risk model.

continuous function:

$$d^*/d_a = \Big(\frac{f_a - f^* + \varepsilon}{\varepsilon}\Big)^{1/2}, \quad n \to \infty. \tag{7}$$

Here d^* is a density of points x_i around the global optimum. d_a is an average density of x_i in the feasible area. f_a is an average value of $f(x)$ in this area. f^* is an average value of $f(x)$ around the global minimum. ε is the correction parameter in expression (3). That means that the Bayesian algorithm converges to the global minimum for any continuous $f(x)$ and provides greater density of observations x_i around the global optimum.

The correction parameter ε has a similar influence as the temperature in simulated annealing. However, that is a superficial similarity. The good asymptotic behavior is just some "by-product" of Bayesian approach. The reason is that Bayesian decisions are designed for small size samples where asymptotic properties are not noticeable.

Minimization of $R(x)$ is a complicated auxiliary optimization problem. That means that Bayesian methods are efficient just for complicated functions of a few ($m < 20$) continuous variables.

Software Framework for Global Optimization (GMJ)

In optimization problems theory and software are interconnected. The final results depend on the mathematical theory of optimization and the software implementation. Thus we have to regard them both.

Representing a set of examples as a part of E-education environment we need a basic software tools. The examples should be united by some common framework. We call that Global Minimizer by Java (GMJ). Java is platform independent and almost as efficient as "C++" for difficult optimization problems. We implement the Bayesian Heuristic Approach as a convenient theoretical concept. We apply this approach for automatic tuning of heuristic parameters and for search of optimal mixtures of heuristics.

That is just an initial part of the GMJ. Important is to make GMJ open for development by users. Users contribute their own optimization methods in addition to the Bayesian ones. User optimization models are included as GMJ tasks. The results of optimization are represented by GMJ analysis objects. A minimal set of methods, tasks, and analysis objects is implemented by default. The rest depends on users.

Sample of GMJ

Here is the default applet tag with additional contributed task of cut optimization:

Applet Tag

```
<html><head>
<meta http-equiv="Content-Type" content="text/html;
```

```
charset=ISO-8859-13">
<body bgcolor="#d0d0d0"><h1>GMJ</h1><p>

<applet
code="lt.ktu.gmj.ui.GMJext.class" codebase="lib"
align="bottom" high="800" width="800" height="450"
archive=
"gmj2.jar,url.jar,Sinc.jar,gmj.cut.jar">
<param name="TASKS"
value=
"lt.ktu.gmj.tasks.Sinc|lt.ktu.gmj.tasks.Cut">
<param name="METHODS"
value=
"lt.ktu.gmj.methods.Bayes|lt.ktu.gmj.methods.Mig1">
<param name="ANALYSIS"
value=
"lt.ktu.gmj.analysis.cutAnalyser|lt.ktu.gmj.analysis.Convergence">
 <p>You need Java compatible browser to see this applet </p>
</applet>

</p></body></html>
```

All the default classes are in archives ’gmj2.jar’ and ’url.jar’. Contributed task is in ’cut.jar’. The parameter “TASKS” defines a default sinusoidal task ’Sinc’ and the contributed task ’Cut’.

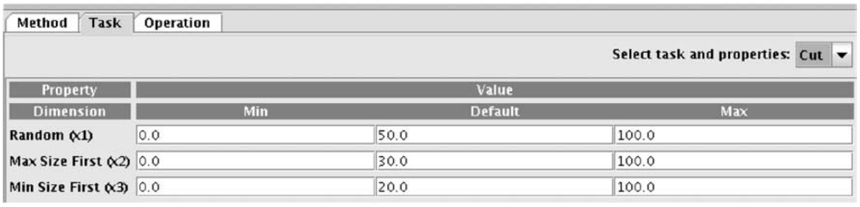
The parameter “METHODS” defines two default methods: Bayesian ’Bayes’ and Monte Carlo ’Mig1’

The parameter “Analysis” defines the contributed object ’CutAnalysis’ and the default object ’Convergence’ .

Examples of Methods

Bayesian Multi-Dimensional Search, “Bayes”

The “Bayes” is a multi-dimensional search procedure by the continuous risk model (6). Fig. 4 shows the shape of the risk and other functions of this model. The aim is to minimize the average deviation from the global minimum at fixed number of iterations. That involve both the active observations (calculations of objective function at fixed variables)



Select task and properties: Cut

Property	Value		
Dimension	Min	Default	Max
Random (x1)	0.0	50.0	100.0
Max Size First (x2)	0.0	30.0	100.0
Min Size First (x3)	0.0	20.0	100.0

Fig. 5. Cut optimization, input.

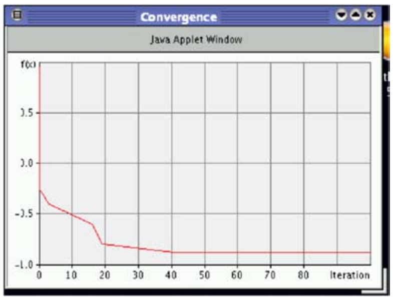
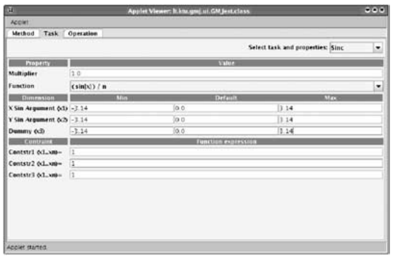


Fig. 6. Cut optimization, convergence.

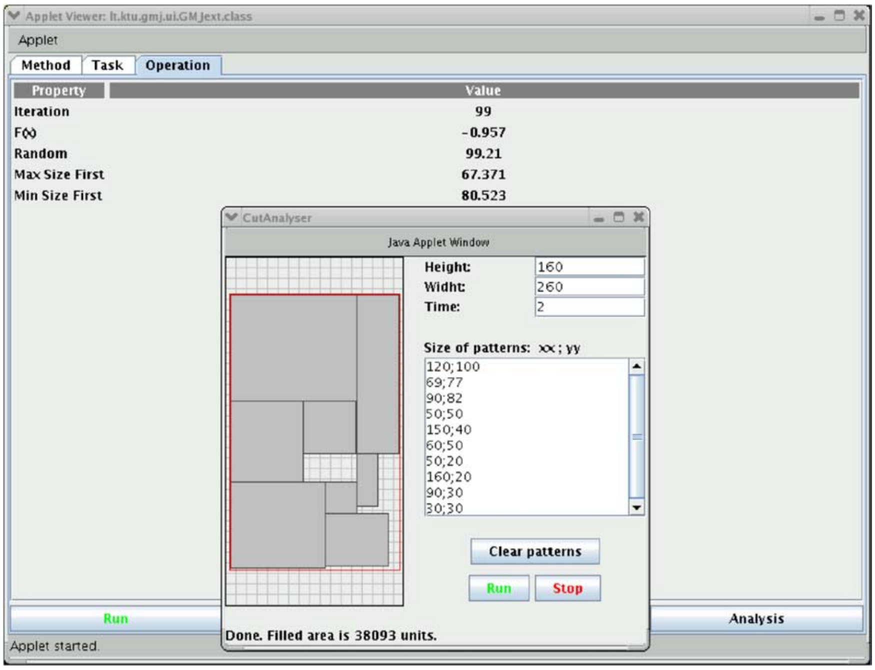


Select task and properties: Sinc

Property	Value		
Multiplier	1.0		
Function	(sin(x)) / n		
Dimension	Min	Default	Max
X Sin Argument (x1)	-3.14	0.0	3.14
Y Sin Argument (x2)	-3.14	0.0	3.14
Dummy (x3)	-3.14	0.0	3.14
Constraint	Function expression		
Contstr1 (x1..xn)=	1		
Contstr2 (x1..xn)=	1		
Contstr3 (x1..xn)=	1		

Applet started.

Fig. 7. Default task, “Sinc”.



Property	Value
Iteration	99
F(x)	-0.957
Random	99.21
Max Size First	67.371
Min Size First	80.523

Height: 160
Widht: 260
Time: 2

Size of patterns: xx ; yy
120;100
69;77
90;82
50;50
150;40
60;50
50;20
160;20
90;30
30;30

Done. Filled area is 38093 units.

Applet started.

Fig. 8. Cut optimization, sample.

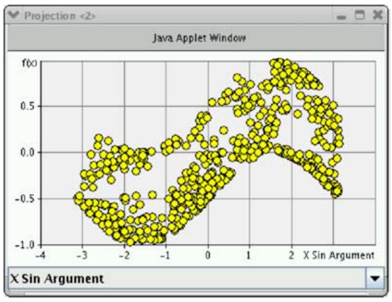


Fig. 9. Clustering method “Globt”, x-projection.

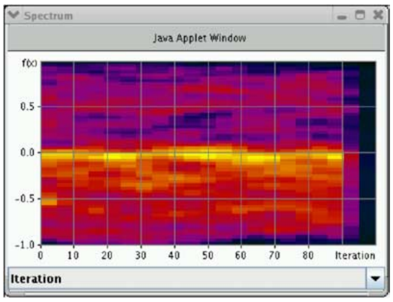


Fig. 10. Clustering method “Globt”, spectrum.

depending on available results and a few passive uniformly distributed initial observation. The initial observations are for estimation of search parameters.

Coordinate Search by Wiener Model, "Exkor"

The "Exkor" performs a multi-dimensional coordinate search. The search starts from some initial point. The optimization is done by changing single variables separately using the Wiener model (Zilinskas, 1978b; Mockus and Zilinskas, 1972). This is the first loop:

$$\begin{aligned} x_1^{n+1} &= \arg\min_{x_1} f(x_1, x_2^n, \dots, x_m^n), \\ x_2^{n+1} &= \arg\min_{x_2} f(x_1^{n+1}, x_2, x_3^n, \dots, x_m^n), \\ &\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_m^{n+1} &= \arg\min_{x_m} f(x_1^{n+1}, x_2^{n+1}, \dots, x_{m-1}^{n+1}, \dots, x_m). \end{aligned} \tag{8}$$

If $x^{n+1} = x^n$ exkor stops, otherwise next loop is made:

$$\begin{aligned} x_1^{n+2} &= \arg\min_{x_1} f(x_1, x_2^{n+1}, \dots, x_m^{n+1}), \\ x_2^{n+2} &= \arg\min_{x_2} f(x_1^{n+2}, x_2, x_3^{n+1}, \dots, x_m^{n+1}), \\ &\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_m^{n+2} &= \arg\min_{x_m} f(x_1^{n+2}, x_2^{n+2}, \dots, x_{m-1}^{n+2}, \dots, x_m). \end{aligned} \tag{9}$$

Here $\arg\min_{x_i}$ denotes the value of the variable x_i obtained using the Wiener model of optimization.

The method converges to some point of attraction depending on the initial point. Some *ε-accuracy* can be provided by optimizing a monotone transformation $\nu(f(x))$ such that

$$\nu(f(x)) = \sum_{i=1}^{m} \nu_i(x_i) + \varepsilon \nu_0(x). \tag{10}$$

Here $\varepsilon > 0$, the component $\nu_0(x)$ depends on all variables, and components $\nu_i(x_i)$ depend just on a single variable x_i. For example, if $\nu(f(x)) = f(x)$ then from (10)

$$f(x)) = \sum_{i=1}^{m} f_i(x_i) + \varepsilon f_0(x). \tag{11}$$

If $\nu(f(x) = \log f(x)$ then

$$f(x) = \varepsilon f_0(x) \prod_{i=1}^{m} f_i(x_i). \tag{12}$$

The specific feature of multi-dimensional coordinate search are good projections. The nth projection shows how values of objective function depend on the nth variable x_n,

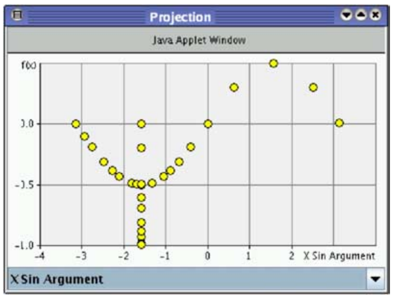


Fig. 11. Method of coordinate optimization “Exkor”, x-projection.

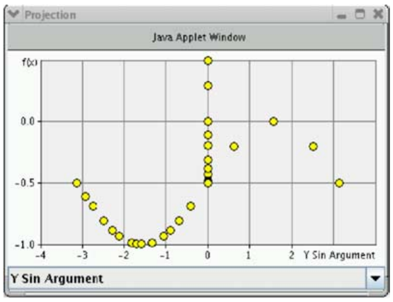


Fig. 12. Methot “Exkor”, y-projection.

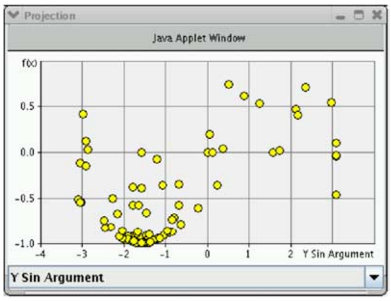


Fig. 13. Bayesian method “Bayes”, x-projection.

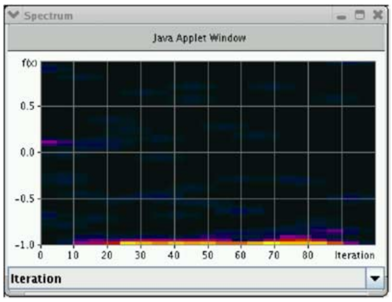


Fig. 14. Method “Bayes”, spectrum.

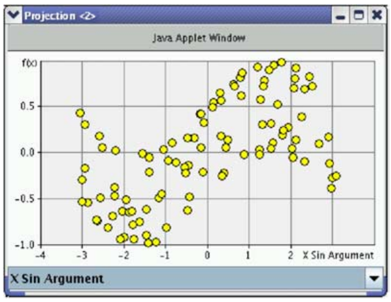


Fig. 15. Monte Carlo method “Mig1”, x-projection.

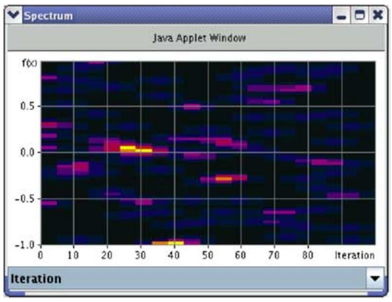


Fig. 16. Method “Mig1”, spectrum.

$n = 1, \ldots, m$. The first projection obtained using Exkor for minimization of sinusoidal function is in Fig. 11. Fig. 12 shows the second projection.

That helps to obtain some initial information about behavior of the objective function. Starting the optimization no such information is available, as usual.

Using other methods we change several variables at once. The visual results are blurred figures. The first projection of the Bayesian search is in Fig. 13. Fig. 15 shows the first projection of the Monte Carlo search.

Monte Carlo, Uniform Search, "Mig1"

Often the calculation of deviation from the exact global minimum is as difficult as the original global optimization problem. We can define some ε*-bounds* by using additional properties of the objective function, such as Lipshitz constant.

Otherwise a different standard should be defined. The uniform Monte Carlo search can be used:

$$x^{n+1} = \arg \min_{1 \leqslant i \leqslant n} f(x^i). \tag{13}$$

Here components x_j^i are generated randomly by uniform distribution on the feasible intervals $a_j \leqslant x_j^i \leqslant b_j,\ j = 1, \ldots, m,\ i = 1, \ldots, n$

Monte Carlo is apparently the simplest method providing convergence with probability one for any continuous objective function. Thus we can test efficiency of global optimization methods by comparing them with Monte Carlo search (13).

The analysis tool "Convergence" defines how the best values of objective functions depend on iteration number. Fig. 6 illustrates convergence of the Bayesian method optimizing heuristics in the cut problem. A method of search can be regarded efficient only if it converges faster than Monte Carlo.

Clustering Search, "Globt"

We use the Torn algorithm of the clustering optimization (Torn and Zilinskas, 1989). The clustering optimization is a version of the general multi-start algorithm. We do local searches from several starting points and accept the best result as a solution. Starting points are generated by a Monte Carlo procedure.

The specific feature of the clustering algorithm is the procedure of eliminating local searches. In short the first stage of clustering algorithm can be described this way:

- generate k starting points $x^i,\ i = 1, \ldots, k$ randomly by uniform distribution on the feasible interval $a_j \leqslant x_j^i \leqslant b_j,\ j = 1, \ldots, m, i = 1, \ldots, k$;
- perform k local searches starting from the points $x^i,\ i = 1, \ldots, k$, denote the results of local searches as $x^i(1)$;
- perform the clustering of the points $x^i(1)$ by grouping the 'neighbors' into clusters;
- eliminate the 'bad' points $x^i(1)$ by some heuristic procedure and denote the remaining points as $x^i(2),\ i = 1, \ldots, k_2 < k$;
- go to second stage.

In the second stage of clustering we make local search from 'good' points $x^i(2),\ i = 1, \ldots, k_2$ Clustering stops at some stage K when $k_K = 1$ and the final local search is made from the single staring point $x^i(K)$.

No convergence is provided. However the clustering algorithm is efficient if there are not many local minima and the region of attraction of global minimum is large.

Examples of Tasks

Sinusoid, "Sinc"

The sinusoid is a well-known multi-modal function and is implemented as a default task "Sinc":
$(sin(x)/n$ where n is a number of variables $x = (x_1, \dots, x_n)$. By default the uni-modal version is defined by fixing bounds $[-3.14, 0]$. In this paper we use a 'two-modal' version by extending the default bounds to $[-3.14, +3.14]$. Fig. 7 shows the input window of "Sinc".

Cut Optimization, "Cut"

This is a simple example of typical contributed task using the Bayesian Approach for tuning heuristics. The two-dimensional "Guillotine" cut is regarded. The 'mixture' of three 'greedy' heuristics is optimized. The term 'Greedy' describes heuristics that build system from scratch by adding elements one-by-one. The 'mixture' means that different heuristics are selected randomly at each iteration by some fixed probabilities.

The first heuristic is denoted as "Random". Here we select cut elements with equal probabilities. The second heuristic is 'Max Size First'. The third heuristic is 'Min Size First'. The names speaks for themselves. Fig. 5 shows the input window. Fig. 6 shows convergence. We search for the best mixture of these tree heuristics at fixed optimization time. The time parameter is important because the optimal mixture depends on the number of iterations. Fig. 8 shows the results. The optimal mixture is described by the numbers of iterations. The mixture probabilities are estimated as corresponding fractions.

Examples of Analysis Tools

Convergence

The 'Convergence' window shows how the best value of the objective depends on the number of iterations, For most methods that is a decreasing step-function. Steps happen at 'improving' iterations. In global optimization the convergence line often is nearly logarithmic, see for example, Fig. 6. That illustrates the high complexity of global optimization, in general.

Projection

The 'Projection' is apparently the simplest two-dimensional visualization of multi-dimensional objective functions. Using the Projection we may obtain some useful information about the objective if the variables are changed separately. The method 'Exkor' does just that.

Other optimization methods change many variables at each iteration. Here the 'Projection' shows how the density of observations depends on variables x. In efficient methods the density is higher around the global minimum.

Comparing "Projection" of different methods we see higher densities around global minima in both Bayesian methods "Exkor" and "Bayes". Method "Exkor" is in Fig. 11 and Fig. 12. Method 'Bayes' is in Fig. 13. In the Monte Carlo method "Mig1" the density of observations is nearly uniform, Fig. 15. In the clustering method "Globt" the observation density is slightly higher around the global minimum, Fig. 9. Therefore the simple one-dimensional projection is a second tool for visual analysis of optimization results in addition to the "Convergence".

The third tool is "Spectrum". That is the last default tool. Other analysis tools are for specific problems.

Spectrum

The 'Spectrum' shows how the 'density' of objective function values depends on the number of iterations. The density is represented by color. Bright yellow color means a high density. Dark color- low density. In the Monte Carlo method "Mig1" the density is close to uniform for all iterations as it should be, Fig. 16.

In the clustering "Globt" method the density is highest around the zero values, Fig. 10.

In the "Bayes" method the density is highest close to minimal objective values, Fig. 14.

This way the "Spectrum" provides some additional information about the behavior of different methods.

Distance Studies

On-Line

The video-conferencing is regular: each Friday from 8:00 until 9:30 EET (EEST). Now the language is Lithuanian is because no foreign students are connected. However essential part of the web-site is in English so the English broadcasts are possible, too. The main difficulty for English speaking students apparently are early hours (7:00–8:30 CET) .
The connection is free. Static IP and NetMeeting is used. Advance notification is assumed. Interactive mode is easy using some standard Audio and Video tools. Fig. 17 shows that using just sound and data low speed connection (128 kbps) is enough. Using video we need at least 256 kbps, Fig. 18 and Fig. 19.

The snapshot in Fig. 19 is a slide explaining optimality of the uniform search with some "on-line" graphical explanations made by the "Acrobat pencil" in response to students questions.

Fig. 20 is a snapshot of interactions with a single student at her home PC.

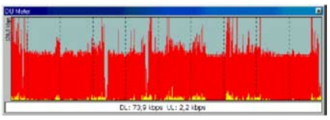


Fig. 17. Audio.

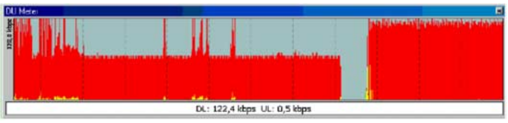
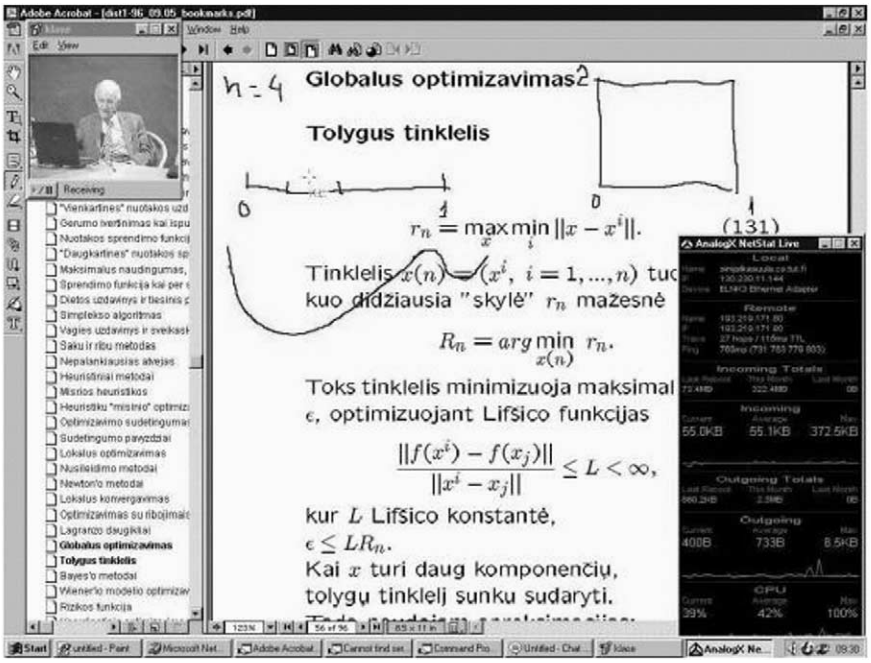


Fig. 18. Video.



"Vienkartines" nuotakos uzd
Gerumo ivertinimas kai ispu
Nuotakos sprendimo funkcij
"Daugkartines" nuotakos sp
Maksimalus naudingumas,
Sprendimo funkcija kai per
Dietos uzdavinys ir tiesinis
Simplekso algoritmas
Vagies uzdavinys ir sveikask
Saku ir ribu metodas
Nepalankiausias atvejas
Heuristiniai metodai
Misrios heuristikos
Heuristiku "misinio" optimiz
Optimizavimo sudetingumas
Sudetingumo pavyzdziai
Lokalus optimizavimas
Nusileidimo metodai
Newton'o metodai
Lokalus konvergavimas
Optimizavimas su ribojimais
Lagranzo daugikliai
Globalus optimizavimas
Tolygus tinklelis
Bayes'o metodai
Wiener'o modelio optimizav
Rizikos funkcija

Globalus optimizavimas2

Tolygus tinklelis

$$r_n = \max_x \min_i ||x - x^i||.$$

Tinklelis $x(n) = (x^i,\ i = 1, ..., n)$ tuc
kuo didžiausia "skylė" r_n mažesnė

$$R_n = arg \min_{x(n)} r_n.$$

Toks tinklelis minimizuoja maksimal
ϵ, optimizuojant Lifšico funkcijas

$$\frac{||f(x^i) - f(x_j)||}{||x^i - x_j||} \le L < \infty,$$

kur L Lifšico konstantė,
$\epsilon \le LR_n$.
Kai x turi daug komponenčių,
tolygų tinklelį sunku sudaryti.

(131)

Fig. 19. Video-conferencing, explaining global optimization.

Off-Line

All the video-conferences are recorded and put on web-site

`http://distance.ktu.lt/vips/join.php?sr=242`

Here is the web-site

`http://pilis.if.ktlu.lt/~jmockus`

and four mirror-sites:

`http://kopustas.elen.ktu.lt/~jmockus`
`http://eta.ktl.mii.lt/~mockus`
`http://proin.ktu.lt/~mockus`
`http://mockus.us/optimum`

The theoretical background and the complete description of the software is in the web-site section "General Description". Software tools are in the section "Software Systems" Examples of continuous global optimization are in "Global Optimization". The section "Discrete Optimization" is for examples of discrete optimization and linear and dynamic

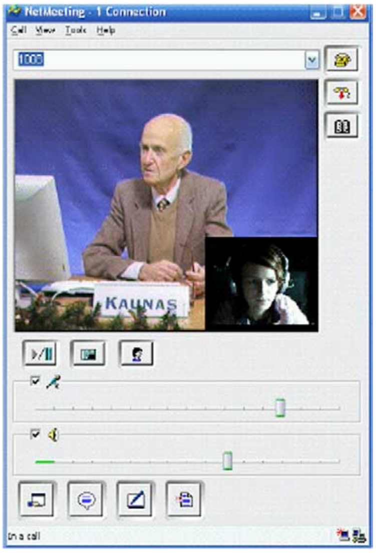


Fig. 20. “Student-at-home”.

programming.

All the results for international users are in English. Examples intended for Lithuanian universities are described in Lithuanian.

2. Conclusions

1. An objective of the paper is to start the scientific collaboration with colleagues on similar lines.
2. The growing power of internet presents new problems and opens new possibilities for distant scientific collaboration and graduate studies. Therefore some nontraditional ways for presentation of scientific results should be defined.
3. The paper is writen as an example of specific style designed for encouragement of new approaches to presentation and sharing of scientific results.
4. The results of optimization show the possibilities of some nontraditional ways of graduate studies and scientific collaboration by creating and using a specific environment for E-education.

   5. Examples of applications of the Bayesian heuristic approach show the efficiency of automated tuning of heuristics.

References

Berger, J. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, Berlin, Heidelberg.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, New Jersey.

Calvin, J., and A. Zilinskas (2000). One-dimensional p-algorithm with convergence rate ??? for smooth functions. *JOTA Journal of Optimization Theory and Applications*, **106**, 297–307.

DeGroot, M. (1970). *Optimal Statistical Decisions*. McGraw-Hill, New York.

Diaconis, P. (1988). Bayesian numerical analysis. In *Statistical Decision Theory and Related Topics*. Springer Verlag. pp. 163–175.

Fine, T.L. (1973). *Theories of Probability*. Academic Press, New York.

Kadane, J., and G. Wasilkowski (1985). Average case ε-complexity in computer science – a Bayesian view. In *Bayesian Statistics* 2. Elsevier Science Publishers. pp. 361–374.

Kushner, H. (1964a). A new method of locating the maximum point of an arbitrary multi-peak curve in the presence of noise. *J. of Basic Engineering*, **86**, 97–100.

Kushner, H. (1964b). A versatile stochastic model of a function of unknown and varying form. *J. of Mathematical Analysis and Applications*, **5**, 150–167.

Lindley, D. (1971). *Bayesian Statistics*: *A Review*. SIAM, Philadelphia.

Mockus, J. (1989). *Bayesian Approach to Global Optimization*. Kluwer Academic Publishers, Dordrecht–London–Boston.

Mockus, J. (2000). *A Set of Examples of Global and Discrete Optimization*: *Application of Bayesian Heuristic Approach*. Kluwer Academic Publishers.

Mockus, J. (2006a). A system for distance studies and applications of metaheuristics. *Journal of Global Optimization* (in print).

Mockus, J. (2006b). Experimental investigation of distance graduate studies of the open source environment by models of optimal sequentional decisions and the bayesian approach. In J. Zilinskas (Ed.), *Global Optimization*. Springer (in print).

Mockus, J., and A. Zilinskas (1972). *On Bayesian Method for Seeking the Minimum*. Avtomatika i vychislitelnaja technika (in Russian).

Mockus, J., W. Eddy, A. Mockus, L. Mockus and G. Reklaitis (1997). *Bayesian Heuristic Approach to Discrete and Global Optimization*. Kluwer Academic Publishers, Dordrecht–London–Boston.

Packel, E., and H. Wozniakowski (1987). Recent developments in information-based complexity. *Bulletin of the AMS*, **17**, 9–35.

Saltenis, V. (1971). On a method of multi-extremal optimization. *Automatics and Computers* (Avtomatika i Vychislitelnayya Tekchnika), 33–38 (in Russian).

Savage, L. (1954). *Foundations of Statistics*. Wiley, New York.

Shiryayev, A. (1978). *Optimal Stopping Rules*. Springer-Verlag, New York.

Sukharev, A. (1971). On optimal strategies of search of extremum. *Computational Mathematics and Mathematical Physics*, 910–924 (in Russian).

Torn, A., and A. Zilinskas (1989). In *Global optimization*. Springer-Verlag, Berlin.

Traub, J., G. Wasilkowski and H. Wozniakowski (1988). *Information-Based Complexity*. Academic Press, New York.

Traub, J., and H. Wozniakowski (1992). Perspectives on information-based complexity theory. *Bulletin of the AMS*, **26**, 29–52.

Wald, A. (1947). *Sequential Analysis*. J. Wiley, New York.

Wald, A. (1950). *Statistical Decision Functions*. J. Wiley, New York.

Zilinskas, A. (1978a). On statistical models for multi-modal optimization. *Math. Operat. Stat.*, Ser. *Statistics*, **9**, 255–266.

Zilinskas, A. (1978b). Optimization of one-dimensional multi-modal functions, algorithm as-133. *Applied Statistics*, **23**, 367–375.

Zilinskas, A. (1986). *Global Optimization*: *Axiomatic of Statistical Models, Algorithms and their Applications*. Mokslas, Vilnius (in Russian).

J. Mockus graduated Kaunas University of Technology, 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, and professor of Kaunas University of Technology. His research interests include global and discrete optimization.

Eletroninės mokymo aplinkos, skirtos moksliniam bendradarbiavimui ir aukštosioms nuotolinėms studijoms, pavyzdžių tyrimas, 1-ji dalis

Jonas MOCKUS

Tikslas – ištirti naujų informacinių technologijų įtaką aukštosiom studijom ir moksliniam bendradarbiavimui.

Sudėtingumo teorijos rezultatai rodo, kad tiksliųjų metodų galimybės yra ribotos. Tai paaišskina heuristinių metodų paplitimą. Heuristikų efektyvumas priklauso nuo parametrų. Todėl reikalingos automatinės procedūros heuristikų optimizavimui.

Pradinės mintys aprašytos (Mockus, 2000). Preliminarūs nuotolinių studijų rezultatai pateikti (Mockus, 2006a). Nuoseklių statistinių sprendimų pavyzdžiai nagrinėjami (Mockus, 2006b).

Šiame straipsnyje tiriami Bajeso heuristinių metodų taikymai. Kitame straipsnyje bus nagrinėjami pavyzdžiai, skirti šių metodų taikymui optimizuojant heuristikų parametrus. Tradicinių optimizavimo metodų atitinkami pavyzdžiai bus pateikti trečiame straipsnyje. Visi šie straipsniai yra trys vieno bendro darbo dalys. Tačiau juos galima nagrinėti nepriklausomai.

Visi algoritmai realizuoti nepriklausančios nuo platformos Java kalbos appletų bei servletų formoje. Todel skaitytojai gal legvai panaudoti visus rezultatus savųjų optimizacinių modelių studijoms, tyrimui ir tobulinimui. Pilna informacija tinklapyje `http://pilis.if.ktu.lt/∼mockus` bei keturiuose jo atspindžiuose.