Journal of Global Optimization (2006) 35: 637–651
DOI 10.1007/s10898-005-5369-0

A System for Distance Studies and Applications of Metaheuristics

JONAS MOCKUS
Department of Optimization, Institute of Mathematics and Informatics, Akademijos 4, LT-2600 Vilnius, Lithuania (e-mail: jmockus@gmail.com)

(Received 17 November 2005; accepted 21 November 2005)

Abstract. The efficiency of metaheuristics depends on parameters. Often this relation is defined by statistical simulation and have many local minima. Therefore, methods of stochastic global optimization are needed to optimize the parameters. The traditional numerical analysis considers optimization algorithms that guarantee some accuracy for all functions to be optimized. This includes the exact algorithms. Limiting the maximal error requires a computational effort that often increases exponentially with the size of the problem [Horst and Pardalos (1995), *Handbook of Global Optimization*, Kluwer Academic Publisher, Dordrecht/Boston/London]. That limits practical applications. An alternative is the average analysis where the expected error is made as small as possible [Calvin and Zilinskas (2000), *JOTA Journal of Optimization Theory and Applications*, 106, 297–307]. The average is taken over a set of functions to be optimized. The average analysis is called the Bayesian Approach (BA) [Diaconis (1988), *Statistical Decision Theory and Related Topics*, Springer-Verlag, Berlin, pp. 163–175, Mockus and Mockus (1987), *Theory of Optimal Decision*, Vol. 12, Institute of Mathematics and Cybernetics, Akademia Nauk Lithuanian SSR, Vilnius, Lithuania pp. 57–70]. Application of BA to optimization of heuristics is called the Bayesian Heuristic Approach (BHA) [Mockus (2000), *A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach*, Kluwer Academic Publishers, Dordrecht, ISBN 0-7923-6359-0]. If the global minimum is known then the traditional stopping condition is applied: stop if the distance to the global minimum is within acceptable limits. If the global minimum is not known then the different approach is natural: minimize the average deviation during the fixed time limit because there is no reason to stop before. If the distance from the global minimum is not known the efficiency of method is tested by comparing with average results of some other method. “Pure” Monte Carlo is a good candidate for such comparison because it converges and does not depend on parameters that can be adjusted to a given problem by using some expert knowledge or additional test runs. In this paper a short presentation of the basic ideas of BHA [described in detail in Mockus (2000, A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach, Kluwer Academic Publishers, Dordrecht, ISBN 0-7923-6359-0) and Mockus (1989, Bayesian Approach to Global Optimization, Kluwer Academic Publishers, Dordrec ht-London-Boston)] is given. The simplest knapsack problem is for initial explanation of BHA. The possibilities of application are illustrated by a school scheduling problem and other examples. Designed for distance graduate studies of the theory of games and markets in the Internet environment. All the algorithms are implemented as platform independent Java applets or servlets therefore readers can easily verify and apply the results for studies and for real life heuristic optimization problems. To address this idea, the paper is arranged in a way convenient for the direct reader participation. Therefore, a part of the paper is written as some “user guide”. The rest is a short description of optimization algorithms and models. All the remaining information is on web-sites, for example *http://pilis.if.ktu.lt/~mockus*.

Key words: Bayesian, Distance Studies, Metaheuristics, Optimization

1. Introduction: How to Write Papers About Heuristics?

The answer for mathematical papers is simple:
– theorem – proof – example.

In papers about heuristics the algorithms and models can be regarded as "theorems". The comparison with other heuristics for some test examples can be regarded as an efficiency proof.

This is the weak part of reports about heuristics. The reason is that the results of heuristics depends on the proper choice of parameters. That is difficult because the parameters depends on both the model to be optimized and on the time available for optimization. If the optimization time is long then the convergence properties of heuristics can be significant. If the time is short then the average deviation is important. Thus heuristics work better in the hands of authors as usual. Therefore reported tables and graphs depend on both the quality of heuristics and expert knowledge of authors defining the parameters. There exist some theoretical recommendations concerning these parameters. These recommendations depend on some other parameters that are often unknown. For example, in the Simulated Annealing the initial "temperature" depends on the difference between the maximal and minimal values.

The graphs that show the relations of results on parameteras are useful for those who regard similar problems. Others may obtain different results.

These are reasons why in this paper a different approach is regarded. Traditional tables and graphs are replaced by a short description of algorithms and models with references to web-sites where complete descriptions are presented. The web-sites contain a set of Java applets and servlets thus all the comparisons can be made by the readers. To start Java applets the web browser must have some Java plug-in. For Java servlets any browser works.

For advanced users a Global Minimization Java framework (GMJ) is developed. Applying GMJ readers can add their own models and methods and to test them directly.

The main scientific tool is the Bayesian Approach (BA). The BA is based on some statistical model of the function to be minimized and can be used in stochastic global optimization. That is needed for optimization of heuristic parameters. The application of BA for this task we call the Bayesian Heuristic Approach (BHA). Users can run other algorithms by the GMJ, too.

Many examples are simplified economic and social models transformed into the optimization problems. The models are not difficult for understanding. The computing time does not exceed reasonable limits as usual.

The full set of examples and the complete theoretical explanation is on the six mirror web-sites. This increases the reliability of Internet communications.

No "perfect" examples in these web-sites. All examples has some advantages and some disadvantages. Improvement of "non-perfect" models is useful for students and interesting for colleagues.

The main objective of this paper is to establish scientific collaboration in the Internet environment with distant colleagues and students.

The paper represents the extended introduction to the main results. The complete description is available on the web-sites. Important part of this introduction is the presentation of the main ideas and goals for developing those web-sites.

In this paper parameters of two well-known metheuristics – Simulated Annealing (SA) and Genetic Algorithm (GA) – are optimized using the knapsack problem as an example. The results are compared with the default values of SA and GA. The investigation of these and other metaheuristics using different test functions is important future task.

1.1. BAYESIAN APPROACH (BA)

The BA is defined by fixing a prior distribution P on a set of functions $f(x)$ and by minimizing the Bayesian risk function $R(x)$ [6, 7]. The risk function describes the average deviation from the global minimum. The distribution P is regarded as a stochastic model of $f(x)$, $x \in R^m$, where $f(x)$ can be a deterministic or a stochastic function. This is possible because using BA uncertain deterministic functions can be regarded as some stochastic functions [7–11]. That is important feature of the BA in this setup. For example, if several values of some deterministic function $z_i = f(x_i)$, $i = 1, \dots, n$ are known then the level of uncertainty can be represented as the conditional standard deviation $s_n(x)$ of the corresponding stochastic function $f(x) = f(x, \omega)$, where ω is a stochastic variable.

In the Gaussian distribution [6], assuming that the $(n+1)$th observation is the last

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

Here $c_n = \min_i z_i - \epsilon$, $z = f(x)$, $m_n(x)$ is the conditional expectation at the point x given the values of z_i, x_i $i = 1, \dots, n$ and $\epsilon > 0$ is a correction parameter. This parameter improves "a-step-ahead" approximation (1) of the multi-stage decision process. Therefore ϵ is a decreasing function of iteration number. The convergence depends on ϵ, too (2).

The Wiener process is the simplest stochastic model in the uni-dimensional case $m=1$ [12–14].

The Wiener model implies that almost all the sample functions $f(x)$ are continuous, 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, and that $f(x)$ is Gaussian $(0,\sigma x)$ at any fixed $x>0$. Note that the Wiener process originally provided a mathematical model of a particle in Brownian motion. Figure 1 shows the general shape of the functions $m_n(x)$, $s_n(x)$, and $R(x)$.

The Wiener model can be extended to multi-dimensional problems, too [6]. However, simple approximate stochastic models are preferable if $m>1$. The simple models are designed by replacing the traditional Kolmogorov consistency conditions. These conditions require the inversion of matrices of nth order for computing the conditional expectation $m_n(x)$ and variance $s_n(x)^2$. The Markov processes (including the Wiener process) are favorable exceptions. Extending the Wiener process to $m>1$ the Markovian property disappears.

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 simple expression of $R(x)$ is obtained using the results of [6].

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

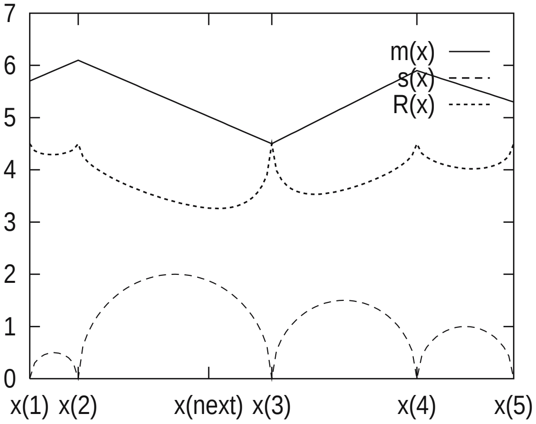


Figure 1. Wiener model.

The aim of BA (designed mainly for continuous cases) is to provide as small average error as possible. In addition, BA has some good asymptotic properties, too. It is shown [6] that

$$d^*/d_a = \left(\frac{f_a - f^* + \epsilon}{\epsilon}\right)^{1/2}, \quad n \to \infty, \tag{2}$$

where d^* is density of x_i around the global optimum, d_a average density of x_i, f^* the optimal value of $f(x)$, f_a the average value, and ϵ is the correction parameter. Thus BA provides convergence to the global minimum for any continuous $f(x)$ and greater density of observations x_i around the global optimum if n is large and ϵ is small.

Note that the correction parameter ϵ has a similar influence as the temperature in simulated annealing. However, that is a superficial similarity since, using BA the good asymptotic behavior should be regarded just as a "by-product". The reason is that Bayesian decisions are designed for the small size samples where asymptotic properties are not noticeable as usual.

The optimal point x_{n+1} for the next iteration of BA is the solution of some auxiliary optimization problem minimizing the average deviation $R(x)$ from the global optimum (see Figure 1). That makes the BA useful mainly for the computationally expensive functions of a few ($m < 20$) continuous variables.

If the number of variables is large and the objective function is not expensive the BHA is preferable. That is true in many discrete optimization problems. These problems are solved using heuristics based on an expert opinion as usual.

1.2. BAYESIAN HEURISTIC APPROACH (BHA)

Using the BHA we fix a prior distribution P on a set of auxiliary functions $f(x) = f_K(x)$. Auxiliary functions defining the best values obtained using K times some heuristic $h(x)$.

The aim of the heuristic $h(x)$ is to optimize an original function $v(y)$ of variables $y \in R^n$ [15]. As usual the components of y are discrete variables. The heuristic $h(x)$ represents an expert opinion about the decision priorities. The heuristics or their "mixture" depend on some continuous parameters $x \in R^m$, where $m < n$ as usual.

The aim of BA is to search for such parameters x that minimize the expected deviation from the global minimum of the auxiliary function $f(x)$.

In BHA, the expert knowledge is included by defining the heuristics. Here the Bayesian decision theory is to optimize parameters of the heuristics by BA.

The popular heuristics are randomization procedures depending on some empirically defined parameters. For example, the parameter x is the initial temperature, if the SA is applied. If the mixture of m different algorithms is used then the heuristic parameters $x=(x_1, \ldots x_m)$ are probabilities of different randomization algorithms.

In these problems, the BA is a convenient tool for optimization of the continuous parameters of various heuristic techniques. That is called the BHA [15].

1.3. IMPROVING EXPERT HEURISTICS

The main objective of BHA is to improve any given heuristic or the "mixture" of different heuristics by defining the best parameters. The examples indicate that heuristic decision rules mixed and adapted by BHA often outperform the best individual heuristics. In addition, BHA provides almost sure convergence. However, the final results of BHA depend on the quality of the specific heuristics including the expert knowledge. Therefore, the BHA should be regarded as a tool for enhancing the heuristics but not for replacing them.

Many well known optimization algorithms, such as GAs [16], GRASP [17], and Tabu Search [18], may be regarded as metaheuristics that can be improved using BHA.

The GAs [16] is an important "source" of useful stochastic search heuristics. It is well known that the results of the GAs depend on the mutation and cross-over parameters. The BHA can be used to optimize those parameters.

In the GRASP system [17], the heuristic is repeated many times. During each iteration a greedy randomized solution is constructed and the neighborhood around that solution is searched for a local optimum. The "greedy" component constructs a solution by adding single elements until the final solution is constructed. A possible application of the BHA in GRASP is in optimizing a random selection of a candidate to be in the solution because different random selection rules may be used and their best parameters should be defined. The BHA can be useful as a local component, too, by randomizing the local decisions and optimizing the corresponding parameters.

In Tabu search the issues of identifying best combinations of short and long-term memory and best balances of intensification and diversification strategies can be obtained using BHA.

Hence, the BHA can be applied to improve various stochastic or heuristic algorithms of discrete optimization. The proved convergence of a discrete search method is an asset. If not, then the convergence conditions are provided by tuning the BHA parameters [15].

A large collection of examples for comparison of different methods of global and discrete optimization are in [19].

Another source of heuristics are exact algorithms developed by mathematical means when applied outside the original scope. For example, the Gupta heuristic is the well-known exact algorithm for the two-machine flow-shop problem. For general flow-shop problem this algorithm is just a good heuristic [15].

2. Optimization of Mixed Heuristics

2.1. KNAPSACK EXAMPLE

The example of knapsack problem illustrates the application of BHA in discrete optimization. Given a set of objects $j=1,\dots,n$ with values c_j and weights g_j, find the most valuable collection of limited weight

$$\max_y v(y), \qquad v(y)=\sum_{j=1}^{n} c_j y_j, \qquad \sum_{j=1}^{n} g_j y_j \leqslant g.$$

Here the objective function $v(y)$ depends on n Boolean variables $y=(y_1,\dots,y_n)$, where $y_j=1$ if object j is in the collection, and $y_j=0$, otherwise.

2.1.1. *Greedy Heuristics*

Greedy heuristics build a system from scratch. The well-known greedy heuristic $h_j=c_j/g_j$ is the specific value of object j. The greedy heuristic algorithm: "take the greatest feasible h_j", is fast but may get stuck in some non-optimal decision.

We move out of such non-optimal decisions by taking decision j with probability $r_j=\rho_x(h_j)$, where $\rho_x(h_j)$ is an increasing function of h_j and $x=(x_1,\dots,x_m)$ is a parameter vector. Here the BA is to optimize the parameters x by minimizing the best result $f_K(x)$ obtained applying K times the randomized heuristic algorithm $\rho_x(h_j)$. That is the expensive operation of BHA. Therefore, the parallel computations of $f_K(x)$ are used, if possible. That reduces the computing time in proportion to the number of parallel processors.

Optimization of x adapts the heuristic algorithm $\rho_x(h_j)$ to a given problem. Let us illustrate the parameterization of $\rho_x(h_j)$ by three randomization functions: $r_i^l=h_i^l/\sum_j h_j^l,\ \ l=0,1,\infty$. Here the upper index $l=0$ denotes the Monte Carlo component (randomization by the uniform distribution). The index $l=1$ defines the linear component of randomization. The index ∞ denotes the pure heuristics with no randomization:// $r_i^{\infty}=1$

if $h_i = \max_j h_j$, and $r_i^\infty = 0$, otherwise. Here parameters $x = (x_0, x_1, x_\infty)$ define the probabilities of using randomizations $l = 0, 1, \infty$. The optimal x may be applied solving different but related problems, too [15]. That is important in the "on-line" optimization and illustrates some "learning" of Bayesian methods.

Figure 2 shows the output window of "mixture" optimization using three heuristics.

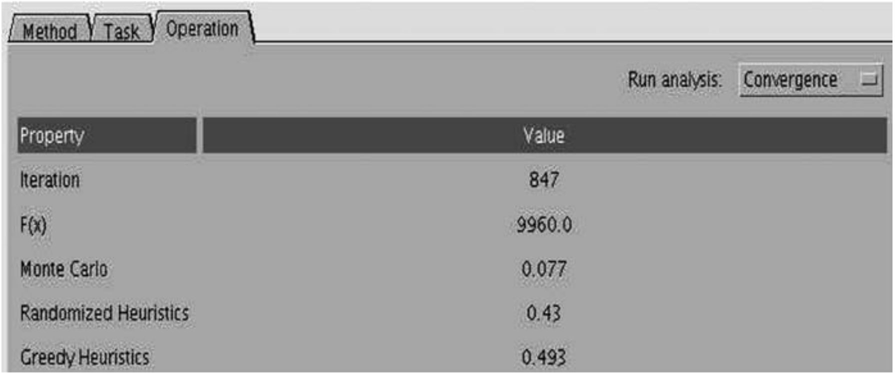
2.1.2. *Permutation Heuristics*

Improving some initial expert decision is the objective of permutations heuristics. Here the expert knowledge is involved in the initial decision. Applying BHA for global minimization different permutations of some feasible solution y^0 are tested. Heuristics are defined as the difference $h_i = v(y^i) - v(y^0)$ between the permuted y^i and the original y^0 solutions.

The simplest version of SA algorithm illustrates the parameterization of $\rho_x(h_j)$ depending on a single parameter x. The probability of accepting worse solutions is $e^{-h_i/x}$, where x is the "annealing temperature".

2.2. OPTIMIZATION OF SIMULATED ANNEALING (SA)

A good example is optimization of profiled school schedule model. The vector objective is evaluated by linear scalarization. The relative importance of various factors are defined by the school authorities. The initial schedule is provided by the school in the "csv" format, because the data are prepared by "Excel" as usual. Note that i/o by the standard "txt" format works well for all tested environments including various versions of Linux and Windows. The i/o by "csv" works for some environments.



Run analysis: Convergence

Property	Value
Iteration	847
F(x)	9960.0
Monte Carlo	0.077
Randomized Heuristics	0.43
Greedy Heuristics	0.493

Figure 2. Optimal mixture.

2.2.1. *Optimization by Permutations*

The initial schedule is improved by permutations. Permutations are done by trying to close the teacher "windows" (empty times between lectures). The best obtained schedule is recorded after each iteration. Changes to worse schedules are made with some probabilities. That is for improving convergence conditions.

Three different heuristics are regarded. Two heuristics with fixed parameters and the third heuristic with parameters obtained by global stochastic optimization.

Using the first heuristic we keep the current schedule until the better schedule is found. The parameter x_1 is the probability to pass by the next teacher. This way we can reach just a sort of local minimum.

A natural first step to provide convergence to the global minimum is the SA: we move from the current schedule i to the permuted schedule $i+1$ with probability

$$r_{i+1} = \begin{cases} e^{\frac{-h_{i+1}}{x_2/\ln(1+N)}}, & \text{if } h_{i+1} > 0, \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

Here N is the iteration number and x_2 is the "initial temperature." The logarithmic "cooling schedule" $\ln(1+N)$ follows from convergence conditions [20].

The difference from the traditional SA is that here we want to improve the average results for some fixed number of iterations $N=K$. Thus the cooling rate should be regarded, too. A way to do it is by introducing the cooling rate parameter x_3.

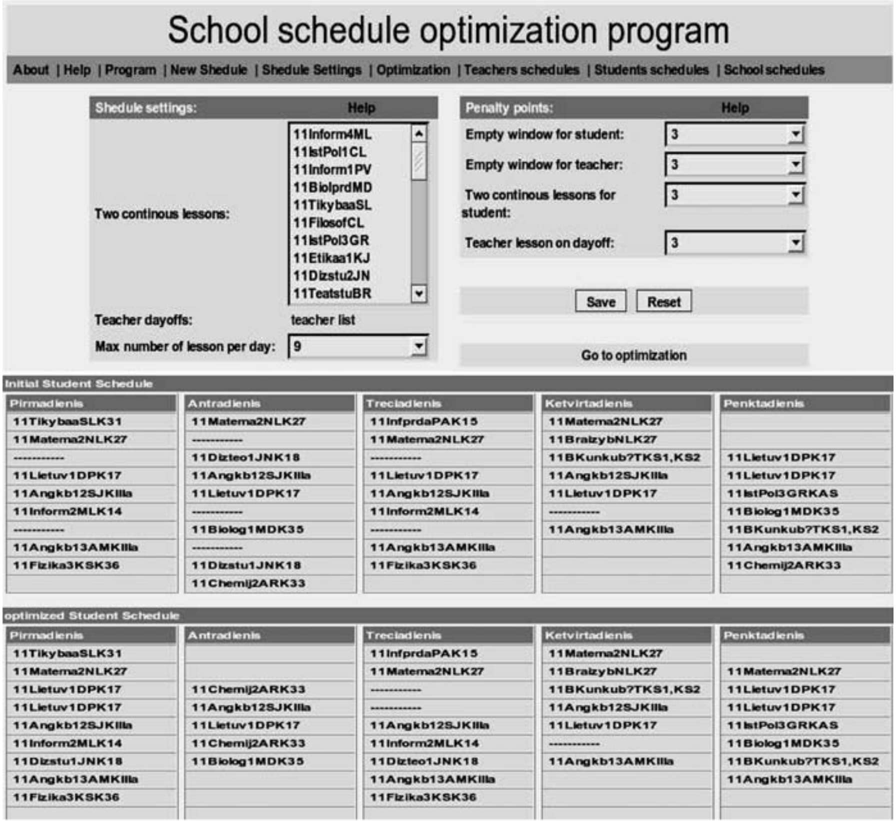
This transforms expression (3)

$$r_{i+1} = \begin{cases} e^{\frac{-h_{i+1}}{x_2/\ln(1+x_3 N)}}, & \text{if } h_{i+1} > 0, \\ 1, & \text{otherwise,} \end{cases} \tag{4}$$

where $x_2 \geqslant 0$ defines an "initial temperature" of SA and $x_3 \geqslant 0$ describes the "cooling rate".

The third heuristic optimizes all three parameters $x=(x_1, x_2, x_3)$ for a fixed optimization time defined by the number of "internal"[1] iterations $N=K$. Figure 3 shows the output window of the scheduler with school preferences, initial, and optimal schedules for some student. The specific feature is the separation of objective factors and subjective evaluation of their relative importance. The objective factors include "Max number of lessons per day", "Teacher dayoffs", "Empty window for student", "Empty

[1] By "internal" we call the number of iterations done at fixed parameters x, by "external" we call the number of iterations used in optimization of x.


School schedule optimization program
About | Help | Program | New Shedule | Shedule Settings | Optimization | Teachers schedules | Students schedules | School schedules
Help
Help
Save
Reset
Go to optimization

Shedule settings:

Two continous lessons: 11Inform4ML, 11IstPol1CL, 11Inform1PV, 11BiolprdMD, 11TikybaaSL, 11FilosofCL, 11IstPol3GR, 11Etikaa1KJ, 11Dizstu2JN, 11TeatstuBR

Teacher dayoffs: teacher list

Max number of lesson per day: 9

Penalty points:

Empty window for student: 3

Empty window for teacher: 3

Two continous lessons for student: 3

Teacher lesson on dayoff: 3

Initial Student Schedule

Pirmadienis	Antradienis	Treciadienis	Ketvirtadienis	Penktadienis
11TikybaaSLK31	11Matema2NLK27	11InfprdaPAK15	11Matema2NLK27	
11Matema2NLK27	------------	11Matema2NLK27	11BraizybNLK27	
------------	11Dizteo1JNK18	------------	11BKunkub?TKS1,KS2	11Lietuv1DPK17
11Lietuv1DPK17	11Angkb12SJKIIla	11Lietuv1DPK17	11Angkb12SJKIIla	11Lietuv1DPK17
11Angkb12SJKIIla	11Lietuv1DPK17	11Angkb12SJKIIla	11Lietuv1DPK17	11IstPol3GRKAS
11Inform2MLK14	------------	11Inform2MLK14	------------	11Biolog1MDK35
------------	11Biolog1MDK35	------------	11Angkb13AMKIIla	11BKunkub?TKS1,KS2
11Angkb13AMKIIla	------------	11Angkb13AMKIIla		11Angkb13AMKIIla
11Fizika3KSK36	11Dizstu1JNK18	11Fizika3KSK36		11Chemij2ARK33
	11Chemij2ARK33			

optimized Student Schedule

Pirmadienis	Antradienis	Treciadienis	Ketvirtadienis	Penktadienis
11TikybaaSLK31		11InfprdaPAK15	11Matema2NLK27	
11Matema2NLK27		11Matema2NLK27	11BraizybNLK27	11Matema2NLK27
11Lietuv1DPK17	11Chemij2ARK33	------------	11BKunkub?TKS1,KS2	11Lietuv1DPK17
11Lietuv1DPK17	11Angkb12SJKIIla	------------	11Angkb12SJKIIla	11Lietuv1DPK17
11Angkb12SJKIIla	11Lietuv1DPK17	11Angkb12SJKIIla	11Lietuv1DPK17	11IstPol3GRKAS
11Inform2MLK14	11Chemij2ARK33	11Inform2MLK14	------------	11Biolog1MDK35
11Dizstu1JNK18	11Biolog1MDK35	11Dizteo1JNK18	11Angkb13AMKIIla	11BKunkub?TKS1,KS2
11Angkb13AMKIIla		11Angkb13AMKIIla		11Angkb13AMKIIla
11Fizika3KSK36		11Fizika3KSK36		

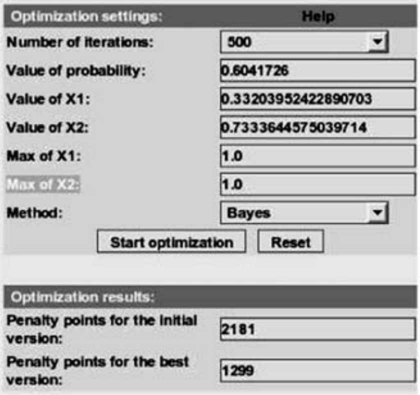
Figure 3. School preferences, initial, and optimal student schedules.

window for teacher", "Two continuous lectures". The subjective importance is defined by "Penalty points". Optimization starts by "Go to optimization" button. The comparing the initial and optimized schedules of some (randomly chosen) student we can evaluate the results. In general the efficiency of the scheduler is high. The apparent reason is the complexity of manual scheduling.

Computer experiments of school scheduling show that the results do not improve regarding simplest "close-window" algorithm if we use SA with default parameters $x_0 = x_1 = x_2 = 0.5$. However we obtain significant improvement by the same SA when we apply parameters optimized by the Bayesian algorithm.

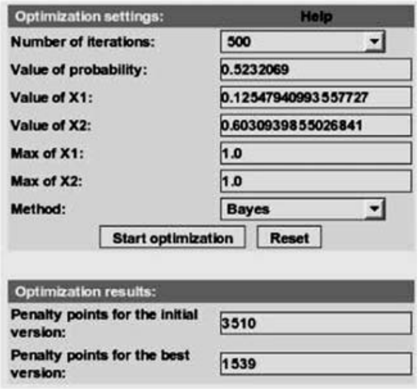
The Table I shows the results of 20 runs by 500 iterations each. The minimal, the maximal, and the expected values are given.

These results are illustrated by three pairs of figures (denoted "Mini"). The first figure of each pair corresponds to the worst results, the right figure shows the best results of 20 runs. The first pair Mini. 1 and Mini. 2 is for the BHA when all three parameters are optimized by the Bayesian algorithm. The second pair Mini. 3 and Mini. 4 is for SA with default



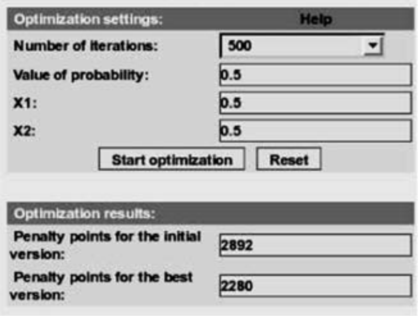
Number of iterations: 500
Value of probability: 0.6041726
Value of X1: 0.33203952422890703
Value of X2: 0.7333644575039714
Max of X1: 1.0
Max of X2: 1.0
Method: Bayes
Penalty points for the initial version: 2181
Penalty points for the best version: 1299

Mini. 1. Worst BHA results



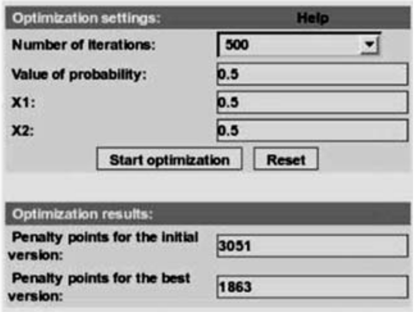
Number of iterations: 500
Value of probability: 0.5232069
Value of X1: 0.12547940993557727
Value of X2: 0.6030939855026841
Max of X1: 1.0
Max of X2: 1.0
Method: Bayes
Penalty points for the initial version: 3510
Penalty points for the best version: 1539

Mini. 2. Best BHA results



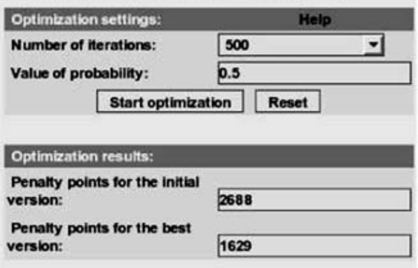
Number of iterations: 500
Value of probability: 0.5
X1: 0.5
X2: 0.5
Penalty points for the initial version: 2892
Penalty points for the best version: 2280

Mini. 3. Worst SA results



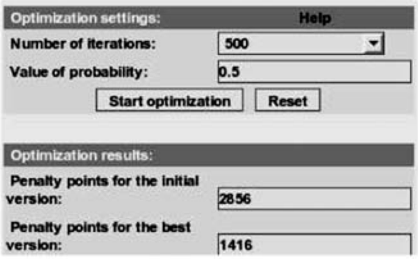
Number of iterations: 500
Value of probability: 0.5
X1: 0.5
X2: 0.5
Penalty points for the initial version: 3051
Penalty points for the best version: 1863

Mini. 4. Best SA results



Number of iterations: 500
Value of probability: 0.5
Penalty points for the initial version: 2688
Penalty points for the best version: 1629

Mini. 5. Worst 'Close-Window' results



Number of iterations: 500
Value of probability: 0.5
Penalty points for the initial version: 2856
Penalty points for the best version: 1416

Mini. 6. Best 'Close-Window' results

Table I. Comparing three algorithms

Iterations $IT = 500$, Runs $R = 20$			
What algorithm	Worst results	Average results	Best results
Bayesian heuristics BHA	882	1269	1961
Simulated annealing SA	612	940	1188
“Close-Window”	675	1027	1440

parameters. The third pair Mini. 5 and Mini. 6 is for simplest “close-window” algorithm.

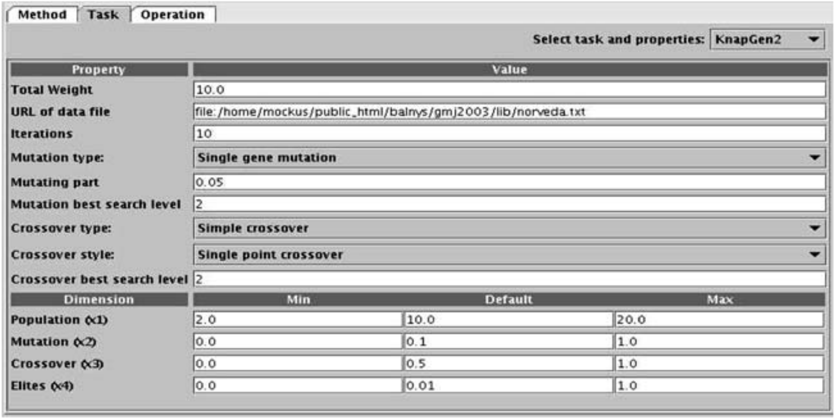
2.3. OPTIMIZATION OF GENETIC ALGORITHMS (GA)

Two versions of genetic algorithm are illustrated. The parameters of the first version are set to default values. The parameters of the second version are optimized by the Bayesian algorithm. The iteration number $iIT\ n$ both cases is 200. The knapsack problem is used as a test-function. The figure Mini. 7 i shows the default values of the first version. The Mini. 8 the initial window of the second version. The Figure 4 shows the initial window of the knapsack problem adapted for the GA. The Figure 5 illustrates the output of the first version. The Figure 6 shows the output of the second version. The Table II shows the worst, the average, and the best results of both GA versions after 20 runs. The number of iterations is 200.

2.4. DISTANCE STUDIES

For the graduate level distance and class-room studies of the theory of games and markets in the Internet environment a set of examples of global and discrete optimization was implemented using platform independent Java applets and servlets. Here is the list of web-sites:

http://pilis.if.ktlu.lt/˜jmockus;
http://optimum2.mii.lt/˜jonas2;
http://eta.ktl.mii.lt/˜mockus;
http://proin.ktu.lt/˜mockus;
http://mockus.us/optimum.

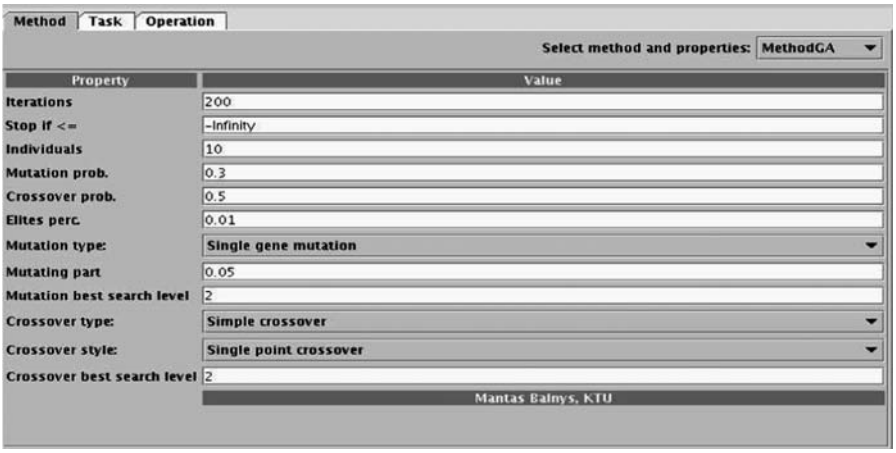


Select task and properties: KnapGen2

Property	Value
Total Weight	10.0
URL of data file	file: /home/mockus/public_html/balnys/gmj2003/lib/norveda.txt
Iterations	10
Mutation type:	Single gene mutation
Mutating part	0.05
Mutation best search level	2
Crossover type:	Simple crossover
Crossover style:	Single point crossover
Crossover best search level	2

Dimension	Min	Default	Max
Population (x1)	2.0	10.0	20.0
Mutation (x2)	0.0	0.1	1.0
Crossover (x3)	0.0	0.5	1.0
Elites (x4)	0.0	0.01	1.0

Figure 4. Initial window of the knapsack-GA problem.

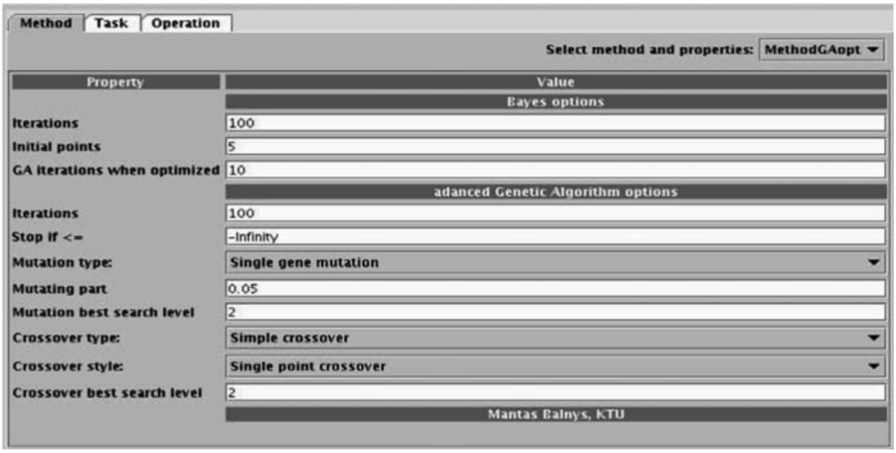


Select method and properties: MethodGA

Property	Value
Iterations	200
Stop if <=	-Infinity
Individuals	10
Mutation prob.	0.3
Crossover prob.	0.5
Elites perc.	0.01
Mutation type:	Single gene mutation
Mutating part	0.05
Mutation best search level	2
Crossover type:	Simple crossover
Crossover style:	Single point crossover
Crossover best search level	2

Mantas Balnys, KTU

Figure 5. Default values of the first GA version.

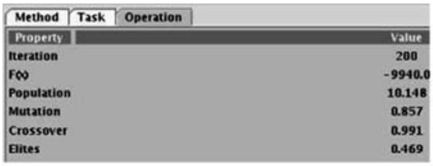


Select method and properties: MethodGAopt

Property	Value
	Bayes options
Iterations	100
Initial points	5
GA iterations when optimized	10
	adanced Genetic Algorithm options
Iterations	100
Stop if <=	-Infinity
Mutation type:	Single gene mutation
Mutating part	0.05
Mutation best search level	2
Crossover type:	Simple crossover
Crossover style:	Single point crossover
Crossover best search level	2

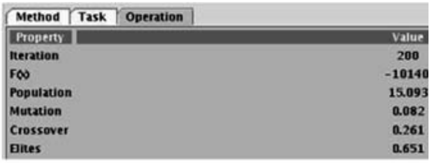
Mantas Balnys, KTU

Figure 6. Default values of the second GA version.



Property	Value
Iteration	200
F(x)	-9940.0
Population	10.148
Mutation	0.857
Crossover	0.991
Elites	0.469

Mini. 7. Output of the first GA version.



Property	Value
Iteration	200
F(x)	-10140
Population	15.093
Mutation	0.082
Crossover	0.261
Elites	0.651

Mini. 8. Output of the second GA version.

The theoretical background and the complete description of the software is in the file “stud2.pdf”. Examples are described in two separate sections: “Global Optimization” for continuous variables, and “Discrete Optimization” for discrete optimization plus applications of linear and dynamic programming.

Table II. Results of default and optimized GA algorithms.

Iterations $IT = 200$, Runs $R = 20$			
What algorithm	Worst results	Average results	Best results
Default genetic algorithm	9460	9805	9940
Optimized genetic algorithm	9940	1046	10140

All the results for international users are in English. Specific examples designed for Lithuanian universities are in Lithuanian.

3. Conclusions

1. The fast growing power of internet opens new possibilities for distant scientific collaboration and graduate studies. Therefore, some new non-traditional ways for presentation of scientific results should be defined.
2. The paper is a try to start a specific style designed for encouragement of new approaches to presentation of scientific results.
3. The objective of the paper is to start the scientific collaboration with colleagues sharing similar ideas.
4. The optimization of well-known SA and GA heuristics for solving the knapsack and the school scheduling problems shows significant improvement.
5. The results of optimization of other well-known heuristics for solving different discrete problems are investigated and will be published in the near future.

References

1. Horst, R. and Pardalos, P. (1995), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht/Boston/London.
2. Calvin, J. and Zilinskas, A. (2000), One-dimensional p-algorithm with convergence rate o (n 3c) for smooth functions, *JOTA Journal of Optimization Theory and Applications* 106, 297–307.
3. Diaconis, P. (1988), Bayesian numerical analysis, *Statistical Decision Theory and Related Topics*, Springer-Verlag, Berlin, pp. 163–175.
4. Mockus, J. and Mockus, L. (1987), Bayesian approach to global optimization and applications, *Theory of Optimal Decision*, vol. 12, Institute of Mathematics and Cybernetics, Akademia Nauk Lithuanian SSR, Vilnius, Lithuania, pp. 57–70 (in Russian).
5. Mockus, J. (2000), *A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach*, Kluwer Academic Publishers, Dordrecht, ISBN 0-7923-6359-0.
6. Mockus, J. (1989), *Bayesian Approach to Global Optimization*, Kluwer Academic Publishers, Dordrecht-London-Boston.
7. DeGroot, M. (1970), *Optimal Statistical Decisions*, McGraw-Hill, New York.

8. Lindley, D. (1971), *Bayesian Statistics: A Review*, SIAM, Philadelphia.
9. Savage, L. (1954), *Foundations of Statistics*, Wiley, New York.
10. Fine, T.L. (1973), *Theories of Probability*, Academic Press, New York.
11. Zilinskas, A. (1986), *Global Optimization: Axiomatic of Statistical Models, Algorithms and Their Applications*, Mokslas, Vilnius, Lithuania (in Russian).
12. Kushner, H. (1964), A new method of locating the maximum point of an arbitrary multi-peak curve in the presence of noise, *Journal of Basic Engineering* 86, 97–100.
13. Saltenis, V. (1971), On a method of multi-extremal optimization. *Automatics and Computers* (Avtomatika i Vychislitelnayya Tekchnika) 33–38 (in Russian).
14. Torn, A. and Zilinskas, A. (1989), *Global optimization*, Springer-Verlag, Berlin.
15. Mockus, J., Eddy, W., Mockus, A., Mockus, L. and Reklaitis, G. (1997). *Bayesian Heuristic Approach to Discrete and Global Optimization*, Kluwer Academic Publishers, ISBN 0-7923-4327-1, Dordrecht-London-Boston.
16. Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
17. Mavridou, T., Pardalos, P., Pitsoulis, L. and Resende, M. (1998), A GRASP for the biquadratic assignment problem, *European Journal of Operations Research* 105, 613–621.
18. Glover, F. Tabu search: improved solution alternatives, *Mathematical Programming. State of the Art 1994*, University of Michigan, MI, 64–92.
19. Floudas, C., Pardalos, P., Adjiman, C., Esposito, W., Z.H. Gu, u., Harding, S., Klepeis, J., Meyer, C. and Schweiger, C. (1999), *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, Dordrecht-Boston-London.
20. Cohn, H. and Fielding, M. (1999), Simulated annealing: searching for an optimal temperature schedule, *SIAM Journal Optimization* 9, 779–802.