Journal of Global Optimization **22:** 191–203, 2002.

Bayesian heuristic approach to global optimization and examples

JONAS MOCKUS
Institute of Mathematics and Informatics, Akademijos 4, Vilnius 2600, Lithuania
(E-mail: jonas@optimum.mii.lt)

Abstract. The traditional numerical analysis considers optimization algorithms which guarantee some accuracy for all functions to be optimized. This includes the exact algorithms. Limiting the maximal error requires a computational effort that in many cases increases exponentially with the size of the problem (Horst and Pardalos, 1995, *Handbook of Global Optimization*, Kluwer). That limits practical applications of the worst case analysis. An alternative is the average case analysis where the average error is made as small as possible (Calvin and Glynn, 1997, *J. Appl. Prob*., 32: 157). The average is taken over a set of functions to be optimized. The average case analysis is called the Bayesian Approach (BA) (Diaconis, 1988, *Statistical Decision Theory and Related Topics*, Springer; Mockus and Mockus, 1987, *Theory of Optimal Decisions*, Nauk, Lithuania). 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*, Kluwer). In this paper a short presentation of the basic ideas of BHA (described in detail in Mockus (1989), *Bayesian Approach to Global Optimization*, Kluwer and Mockus (2000), *A Set of Examples of Global and Discrete Optimization*, Kluwer) is given using the knapsack problem as an example. The application potential is illustrated by the school scheduling example. In addition the new heuristic algorithm for solving a bimatrix game problem is investigated. The results ae applied while solving real life optimization problems and also as examples for distance graduate level studies of the theory of games and markets in the Internet environment.

1. Direct Bayesian approach (DBA)

There are several ways of applying the BA in optimization. The Direct Bayesian Approach (DBA) is defined by fixing a prior distribution P on a set of functions $f(x)$ and by minimizing the Bayesian risk function $R(x)$ (DeGroot, 1970; Mockus, 1989). 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 $f(x)$ might be a deterministic or a stochastic function. This is very important feature of the Bayesian approach that shows the equivalence between uncertain deterministic and the corresponding stochastic functions (Savage, 1954; Lindley, 1965; DeGroot, 1970; Fine, 1983; Zilinskas, 1986). For example, if only the values $z_i = f(x_i)$, $i = 1, \ldots, n$ are known, the level of uncertainty of some deterministic function $f(x)$, $x \neq x_i$ 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 case (Mockus, 1989), assuming that the $(n + 1)$th observation is the last one

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

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 $i = 1, \ldots, n$ and $\epsilon > 0$ is a correction parameter. This parameter is introduced to improve 'one-step-ahead' approximation (1) of the multi-stage decision process. Therefore, it would be reasonable to define ϵ as an decreasing function of iteration number. The convergence depends on ϵ, too (2).

The Wiener process is the simplest stochastic model applying the DBA in the one-dimensional case $m = 1$ (Kushner, 1964; Saltenis, 1971; Torn and Zilinskas, 1989). The Wiener model (see Figure 1) 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.

The Wiener model is extended to multi-dimensional case, too (Mockus, 1989). 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 favorable exceptions are the Markov processes including the Wiener one. 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 $s_n(x)$

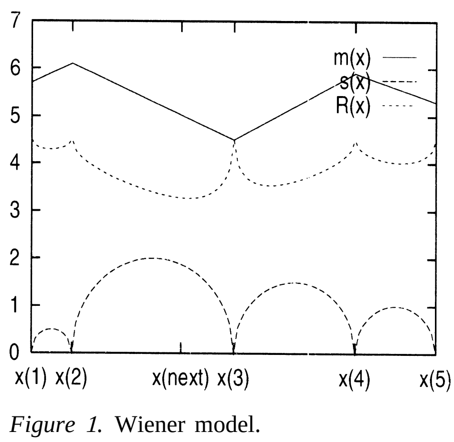

Figure 1. Wiener model.

the following simple expression of $R(x)$ is obtained using the results of Mockus (1989).

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

The aim of DBA (designed mainly for continuous cases) is to provide as small average error as possible. In addition, DBA has some good asymptotic properties, too. It is shown (Mockus, 1989) 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 is average density of x_i, f^* is the optimal value of $f(x)$, f_a is its average value, and ϵ is the correction parameter. That means that DBA provide 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 DBA the good asymptotic behavior should be regarded just as an interesting 'by-product'. The reason is that Bayesian decisions are for the small size samples where asymptotic properties are not noticeable.

Choosing the optimal point x_{n+1} for the next iteration by DBA one solves a complicated auxiliary optimization problem minimizing the expected deviation $R(x)$ from the global optimum (see Figure 1). That makes the DBA 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 Bayesian Heuristic Approach (BHA) is preferable. That is the case in many discrete optimization problems. As usual these problems are solved using heuristics based on an expert opinion.

2. Bayesian Heuristic Approach (BHA)

The Bayesian Heuristic Approach (BHA) means fixing a prior distribution P on a set of auxiliary functions $f_K(x)$ 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 $y \in R^n$ (Mockus et al., 1997). As usual the components of y are discrete variables. The heuristic $h(x)$ defines an expert opinion about the decision priorities. It is assumed that the heuristics or their 'mixture' depend on some continuous parameters $x \in R^m$, where $m < n$.

In BHA the expert knowledge is involved by defining the heuristics. Here the Bayesian decision theory is exploited just to optimize parameters of the heuristics by DBA.

Heuristics often involve randomization procedures depending on some empirically defined parameters. The examples of such parameters are the initial temperature, if the simulated annealing is applied, or the probabilities of different randomization algorithms, if their mixture is used. In these problems, the DBA is a convenient tool for optimization of the continuous parameters of various heuristic techniques. That is called the Bayesian Heuristic Approach (BHA) (Mockus et al., 1997).

3. Knapsack example

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

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

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

3.1. 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 algorithms: 'take the greatest feasible h_j', is very fast but it may get stuck in some non-optimal decision.

One removes the heuristic algorithm 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 functions of h_j and $x = (x_1, \ldots, x_m)$ is a parameter vector. The DBA is used 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 most expensive operation of BHA. Therefore, the parallel computations of $f_K(x)$ are used, if possible. That reduces the computing time in proportion to a 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 $l = 0, 1, \infty$ correspondingly. The optimal x may be applied solving different but related problems, too (Mockus et al., 1997). That is important in the 'on-line' optimization.

3.2. PERMUTATION HEURISTICS

Permutation heuristics improve some initial expert decision by making permutations.

Here the expert knowledge is involved in the initial decision. Applying BHA, 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 solution y^i and the original one y^0.

The well known simulated annealing 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'.

4. School scheduling

Another example is scheduling of traditional schools where curriculum is defined by school authorities with no choice for students. Here a good initial schedules are available, as usual, thus the permutation heuristics is the natural search algorithm.

The sequence of teaching subjects, regarded as tools, can be changed. Different classes are considered as different tasks. The classrooms, including the computer and physics rooms and studies, are the limited resources. The objective is the number of 'empty' hours, so called 'teacher windows' when a teacher waits for the next scheduled lecture. One searches for such schedules that reduce the sum of teacher windows.

There are five constraints: no 'student windows', a teacher can deliver only one lesson at a time, a student can attend only one lesson ar a time, no 'double' lectures (the sequence of two lectures of the same subject),[1] not more than 7 h per day. The optimization starts from the existing school schedule.

The search for better schedules follows a general pattern of permutation algorithms related to the Bayesian Heuristic Approach (Mockus et al., 1997): select a teacher i with probability x and perform a feasible permutation by exchanging the window class[2] with the convenient one. The GMJ systems described in (Mockus, 2000) are used for optimization of the selection probability x.

The algorithm is very simple and natural. It mimics, in a sense, the usual ways to improve existing schedules. However, the algorithm does not satisfy the convergence conditions (Mockus et al., 1997). Therefore, a penalty function approach is used for profile school scheduling where students of eleventh and twelfth grades are choosing just 16 from 64 available subjects. A completed school scheduling models are in web-sites. Users data files can be uploaded by browser.

5. Improving expert heuristics

The main objective of BHA is improving any given heuristic by defining the best parameters and/or the best 'mixtures' of different heuristics. The heuristic decision

[1] There are some exceptions.
[2] A window class is the class in the teacher window.

rules mixed and adapted by BHA often outperform (in terms of speed) even the best individual heuristics as judged by the considered examples. 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. That means, the BHA should be regarded as a tool for enhancing the heuristics but not for replacing them.

Many well known optimization algorithms, such as Genetic Algorithms (Goldberg, 1989), GRASP (Mavridou et al., 1998), and Tabu Search (Glover, 1994), may be regarded as generalized heuristics that can be improved using BHA.

Genetic Algorithms (Goldberg, 1989) is an important 'source' of interesting and useful stochastic search heuristics. It is well known that the results of the genetic algorithms depend on the mutation and cross-over parameters. The Bayesian Heuristic Approach could be used in the optimizing those parameters.

In the GRASP system (Mavridou et al., 1998) 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, one element at a time until a 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 could be used and their best parameters should be defined. BHA might 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 may be obtained using BHA.

Hence, the Bayesian Heuristics Approach may be considered when applying almost any stochastic or heuristic algorithm of discrete optimization. The proven convergence of a discrete search method (Andradottir, 1996) is an asset. Otherwise, the convergence conditions are provided tuning the BHA (Mockus et al., 1997). A large collection of examples for comparison of different methods of global and discrete optimization are in (Floudas et al., 1999).

An important source of heuristics are approximate algorithms developed by mathematical means. For example, the Gupta heuristic was the best one while applying BHA to the flow-shop problem (Mockus et al., 1997). To illustrate the point, a new heuristic algorithm for solving a bimatrix game called the 'Inspector Problem' will be considered.

6. Inspector problem

The free market competition defining equilibrium prices and qualities (Nash, 1950; Mockus, 2000) represents only a part of economical and social activities. Another part is the inspection that one needs to provide tax collection, health and environment protection, etc. We consider a simple model that illustrates the competitions between inspector and violator.

6.1. BIMATRIX GAME

Denote by $x = (x_1, \ldots, x_m)$, $x_i \geqslant 0$, $\Sigma_i\, x_i = 1$ the inspection vector and by $y = (y_1, \ldots, y_m)$, $y_j \geqslant 0$, $\Sigma_j\, y_j = 1$ the violation vector. Here x_i denotes the inspection probability of the area i. y_j means the violation probability in the area j. Denote by $u(i, j)$ the inspection utility function when the object i is inspected and the object j is violated. Denote by $v(i, j)$ the violation utility function when the object i is inspected and the object j is violated. Functions $U(x, y)$ and $V(x, y)$ denote expected values of the inspection and violation utility functions using inspection and violation vectors x, y. These vectors define probabilities of inspection and violation. For example,

$$u(i, j) = \begin{cases} p_i g_i q_i\,, & \text{if } i = j \\ 0\,, & \text{otherwise}\,, \end{cases} \tag{3}$$

and

$$v(i, j) = \begin{cases} -q_j p_i g_j + (1 - p_i) q_j g_j\,, & \text{if } i = j \\ q_j g_j\,, & \text{otherwise}\,. \end{cases} \tag{4}$$

Here p_i is the probability of detecting the violation, if it happens in the area i. q_i is the probability of completing the violation,[3] if violation occurs in the area i. g_i is the utility of the completed violation in the area i.

Expression (3) means that if the violation is completed and detected[4] then the inspector premium is equal to the utility of violation.[5] Expression (4) shows that if the violation is completed and detected, the violator utility is negative. The utility is positive, if it is completed and not detected.

The average utility functions at fixed inspection and violation vectors x and y

$$U(x, y) = \sum_{i,j} x_i u(i, j) y_i\,, \tag{5}$$

and

$$V(x, y) = \sum_{i,j} x_i v(i, j) y_i\,, \tag{6}$$

Here $U(x, y) \neq V(x, y)$. Therefore, the inspection model is a bimatrix game (Forgo et al., 1999).

[3] For example, killing prey.
[4] Prey is killed and a poacher is caught.
[5] The price of the killed prey.

7. Direct Search Algorithm (DSA)

A simple way to obtain equilibrium in pure strategies is the Direct Search Algorithm (DSA). The set IJ of equilibrium points in pure strategies is the intersection of two sets I and J defined by the following conditions

$$IJ = I \cap J\,, \tag{7}$$

$$I = \cup_j I(j)\,, \qquad J = \cup_i J(i) \tag{8}$$

$$I(j), \subseteq \arg\max_i u(i,j) \tag{9}$$

$$J(i), \subseteq \arg\max_j v(i,j)\,. \tag{10}$$

Here $I(j)$ is a set of maximal elements at each column of the matrix $u(i,j)$. $J(i)$ is a set of maximal elements at each row of the matrix $v(i,j)$. If equilibrium exists and the intersection IJ is empty, one needs mixed strategies to obtain the equilibrium (Owen, 1968).

8. Necessary and sufficient conditions

For a pair (x^0, y^0) to be an equilibrium point of the bimatrix game (A, B), it is necessary and sufficient (Forgo et al., 1999) that there exist real numbers α^0, β^0 such that $(x^0, y^0, \alpha^0, \beta^0)$ satisfies the system

$$\begin{aligned}
&xAx - \alpha = 0\,,\\
&xBy - \beta = 0\,,\\
&Ay - \alpha I \leqslant 0\,,\\
&xB - \beta I \leqslant 0\,,\\
&x \geqslant 0\,, \qquad y \geqslant 0\,, \qquad Ix = 1\,, \qquad Iy = 1\,.
\end{aligned} \tag{11}$$

Here I denotes the unit vector. The condition (11) represents the bilinear problem because it involves products of different variables. The solution of bilinear problems is difficult. Therefore, we consider some approximate algorithms to search for mixed strategies of the equilibrium.

9. Irrelevant Fraud algorithm (IF)

The idea is to define such mixed strategies y^0, x^0 that makes the partner fraud irrelevant.[6] The following expressions define these strategies

[6] The fraud is irrelevant, if average winnings of the partner does not depend on his pure strategy $i = 1, \ldots, m$, and vice versa.

$$\sum_{j=1}^{m} u(i,t)y_j^0 = U\ , \quad i = 1, \ldots, m\ , \tag{12}$$

$$\sum_{i=1}^{m} v(i,t)x_i^0 = V\ , \quad j = 1, \ldots, m\ , \tag{13}$$

$$\sum_{j=1}^{m} y_j^0 = 1\ , \qquad \sum_{i=1}^{m} x_0^i = 1\ , \tag{14}$$

$$x_i^0 \geqslant 0\ , \qquad y_j^0 \geqslant 0\ , \quad i, j = 1, \ldots, m\ . \tag{15}$$

Expressions (12), (13), (15) include inequalities. Therefore, the linear programming (LP) is an appropriate method of solution. From (12), (13), (15) one obtains this LP problem

$$\max_{x,y,u.v} (u_1 - u_2 + v_1 - v_2)\ , \tag{16}$$

$$\sum_{j=1}^{m} u(i,j)y_j = u_1 - u_2\ , \quad i = 1, \ldots, m\ , \tag{17}$$

$$\sum_{i=1}^{m} v(i,j)x_i = v_1 - v_2\ , \quad j = 1, \ldots, m\ , \tag{18}$$

$$\sum_{j=1}^{m} y_j = 1\ , \qquad \sum_{i=1}^{m} x_i = 1\ , \tag{19}$$

$$x_i \geqslant 0, \quad y_j \geqslant 0\ , \quad i, j = 1, \ldots, m\ , \quad u_1 \geqslant 0\ , \quad u_2 \geqslant 0, \quad v_1 \geqslant 0\ , \quad v_2 \geqslant 0\ . \tag{20}$$

Here $x = (x_1, \ldots, x_m)$, $y = (y_1, \ldots, y_m)$, $u = (u_1, u_2)$, $v = (v_1, v_2)$, $U = u_1 - u_2$ and $V = v_1 - v_2$. The solution of linear programming problem (13), (12) not always exists. Then one searches for the equilibrium using the Direct Search algorithm (7). That complements the fraud irrelevant algorithm (19), (20) and defines the equilibrium in pure strategies, if it exists. An alternative is the strategy elimination algorithm.

10. Quadratic Strategy Elimination Algorithm (QSE)

The strategy elimination algorithm includes both Irrelevant Fraud and Direct Search algorithms. In addition it eliminates irrelevant raws and columns:

1. obtain a solution (x^0, y^0) of linear equations (12), (13), (14), including inequalities (15), if a solution of the system is found, go to step 7;
2. search for equilibrium in pure strategies by the Direct Search algorithm (7) if a solution is found, go to step 7;

3. obtain a solution (x^0, y^0) of linear equations (12), (13) and (14), ignoring inequalities (15);
4. reduce the system of linear equations by eliminating columns $j = k$ and raws $i = k$ if at least one variable is not positive k: $y_k \leqslant 0$ or $x_k \leqslant 0$, set to zero the variables $x_k = y_k = 0$;
5. if a solution of the reduced system[7] is found, go to step 7;
6. if the reduced system is not empty, go to step 3;
7. stop, and record the components $x_i > 0$, $y_j > 0$ as equilibrium strategies.

Often (QSE) obtains solutions of bilinear problems (11). The algorithm defines both the pure and the mixed strategies. Therefore, QSE may replace both the Direct Search algorithm (7) and the Irrelevant Fraud one (19), (20).

Testing for equilibrium one has to test this QSE solution x^0, y^0 against all possible mixed strategies x and y. This is performed by solving the following auxiliary LP problems

$$\max_x \sum_{i=1,\ j=1}^{m} x_i u(i, j) y_j^0 \tag{21}$$

$$\begin{aligned} &\max_y \sum_{i=1,\ j=1}^{m} x_i^0 v(i, j) y_j \\ &\sum_{j=1}^{m} y_j = 1 , \qquad \sum_{i=1}^{m} x_i = 1 , \quad x_i \geqslant 0 , \quad y_j \geqslant 0 , \quad i, j = 1, \ldots, m . \end{aligned} \tag{22}$$

11. Preventing unauthorized deals

The game will not be played by rules if the players may win more by breaking them. Unauthorized deals (for example, bribes) are usual tools breaking the rules of non-zero-sum games where $v(i, j) \neq -u(i, j)$. An example of such deal is sharing the killed pray between the inspector and violator. Values of the optimal deal $(\bar{u}, \bar{v})$ are defined by the Nash bargaining conditions (Owen, 1968).

The optimal deal $(\bar{u}, v)$ depends on the set of feasible deals D, and on max-min values defining what the players will get if the deal fails

$$\begin{aligned} u^* &= \max_x \min_y U(x, y) \\ v^* &= \max_x \min_y V(x, y) . \end{aligned} \tag{23}$$

Here, u^* is the maximal expected guarantee win of the first player, v^* is that of the second player. In the deal the first player gets $\bar{u}$, the second one obtains $\bar{v}$. Under some natural assumptions (Owen, 1968), the optimal deal $(\bar{u}, \bar{v}) = \phi(D, u^*, v^*)$ is

[7] The reduced system is described by quadratic matrix, this explains the acronym QSE.

uniquely defined by the following conditions. If there is a pair $(u, v) \in D$, such that $u > u^*$, $v > v^*$, then the optimal deal

$$(\bar{u}, \bar{v}) = \arg\max_{u,v} (u - u^*)(v - v^*) , \tag{24}$$

where

$$(u, v) \in D , \quad u \geqslant u^* , \quad v \geqslant v^* \tag{25}$$

If the sum of expected wins of both players is restricted by c then

$$D = \{(u, v) : u + v \leqslant c\} , \tag{26}$$

and the optimal deal

$$(\bar{u}, \bar{v}) = ((c + u^* - v^*)/2, (c + v^* - u^*)/2) \tag{27}$$

An obvious way to prevent unauthorized deals is by introducing penalties (w_1, w_2) that makes the deal unprofitable

$$(w_1, w_2) \geqslant (\bar{u} - u^*, \bar{v} - v^*) \tag{28}$$

From (27)

$$w_i \geqslant c - u^* - v^*, \quad i = 1, 2 . \tag{29}$$

Assuming that the deal is arranged before the game

$$c = \max_i \; g_i q_i \tag{30}$$

Here c is the maximal expected utility to be divided between the bargaining players.

12. Software example

Figure 2 shows the input window. Data is recorded by adding forests and by entering their parameters *P, Q*, *G*. To prevent unauthorized deals one enters ‘Fine’ and ‘Fine’s probability’ defining the average penalties.

13. Future developments

In the inspector problem the QSE algorithm performed well, thus, no additional randomization and parameters optimization was needed. Therefore, an interesting future problem is to apply the QSE algorithm developed for the inspector problem to other bimatrix games. Then the randomization and optimization of its parameters could be helpful.

14. Distance studies

For the graduate level distance and class-room studies of the theory of games and

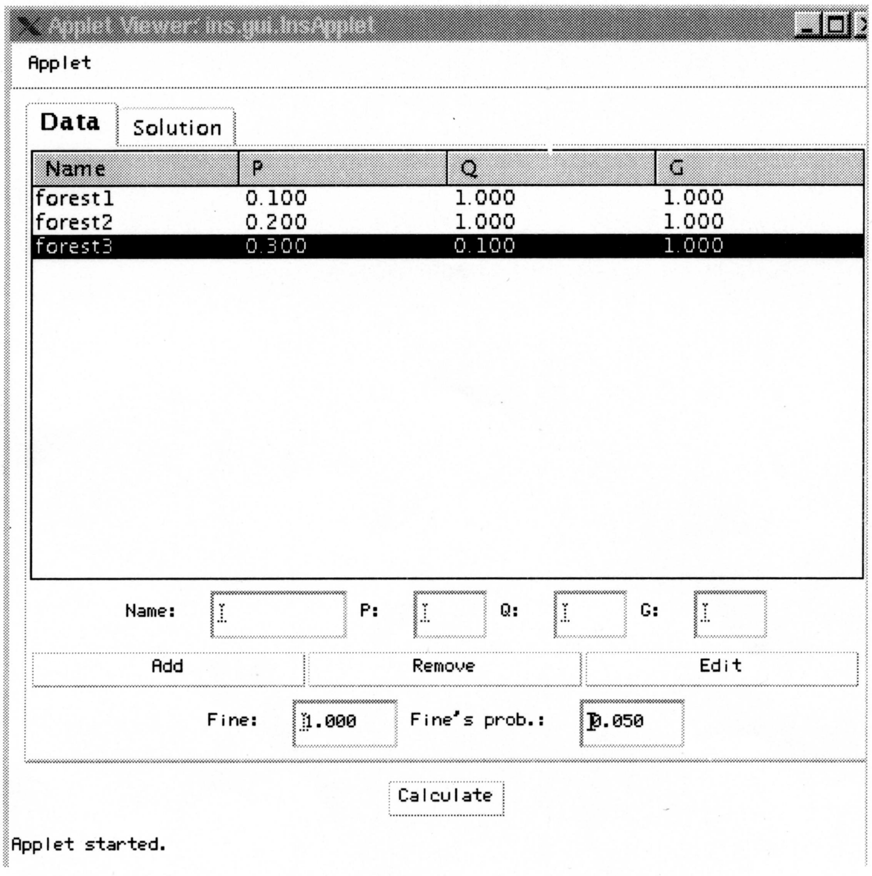

Figure 2. The input window of inspector game.

markets in the Internet environment a set of examples of global and discrete optimization was implemented using Java language. A family of web-sites was developed:

http://optimum2.mii.lt/ ~ jonas2
http://soften.ktu.lt/ ~ mockus
http://mockus.org/optimum

The theoretical background and the complete description of the software is in the file 'stud2.ps'. Examples of discrete optimization, including the knapsack and the school scheduling problems are in the web-side section 'Discrete Optimization and Linear and Dynamic Programming'.

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

References

Andradottir, S. (1996), A Global Search Method for Discrete Stochastic Optimization. *SIAM Journal Optimization*, 6: 513–530.

Calvin, J. and Glynn, P. (1997), Average Case Behavior of Random Search for the Maximum. *J. Apply. Prob.*, 32: 157–163.

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

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

Fine, T. (1983), *Theories of Probability*. Academic Press, New York.

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.

Forgo, F., Szep, J., and Szidarovszky, F. (1999), *Introduction to the Theory of Games*. Kluwer Academic Publishers, Dordrecht.

Glover, F. (1994), Tabu search: improved solution alternatives. In: *Mathematical Programming. State of the Art 1994*, pp. 64–92. University of Michigan.

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

Horst, R. and Pardalo, P. (1995), *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht-Boston-London.

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

Lindley, D. (1965), *Introduction to Probability and Statistics from a Bayesian Viewpoint*. Cambridge University Press, London.

Mavridou, T., Pardalos, P., Pitsoulis, L. and Resende, M. (1998), A GRASP for the biquadratic assignment problem. *European Journal of Operatons Research*, 105: 613–621.

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, Dordrecht.

Mockus, J. and Mockus, L. (1987), Bayesian approach to global optimization and applications. In *Theory of Optimal Decision*, volume 12, pp. 57–70. Institute of Mathematics and Cybernetics, Akademia Nauk Lithuania SSR, Vilnius, Lithuania (in Russian).

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

Nash, J. (1950), Equilibrium points in n-person games. *Proc. Natl. Acad. Sci. USA*, 36: 48–49.

Owen, G. (1968), *Game Theory*. W.B. Saunders, Philadelphia, PA.

Saltenis, V. (1971), On a method of multi-extremal optimization. *Automatics and Computers* (*Automatika i Vychislitelnayya Tekchnika*), (3): 33–38 (in Russian).

Savage, L. (1954), *The Foundation of Statistics*. John Wiley, New York.

Torn, A. and Zilinskas, A. (1989), *Global Optimization*. Springer, Berlin.

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