Experimental Investigation of Distance Graduate Studies of the Open Source Environment by Models of Optimal Sequentional Decisions and the Bayesian Approach

Jonas Mockus

[1] Institute of Mathematics and Informatics,
Akademijos 4, LT-2600 Vilnius, Lithuania,
[2] Kaunas Technological University,
K.Donelaiio g. 73, LT-44029 Kaunas, Lithuania,
jmockus@gmail.com

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

The general consideration of this problem is difficult. Therefore we start by relevant case studies. In this paper we consider models of optimal sequential decisions with multiple objective functions as an example. The aim is to show that models can be implemented and updated by graduate students themselfs. That reflects the usual procedures of the open source development. This way students not only learn the underlinig model but obtains the experience in the development of open source software.

In this case the step-by-step improvement of the model and software is at least as important as the final result that is never achieved in open source environment as usual. A short presentation of the basic ideas is in [1]. Note that doing this we accumulate some experience in the completely new field of education when all the information can be easily obtained by internet. The users are doing just the creative part by filtering and transforming the information to meet their own objectives, to build their own models. The natural way is computer experimentaton.

To make the task as easy as possible all the algorithms considered in this paper are implemented as platform independent Java applets or servlets therefore readers can easily verify and apply the results for studies and for real life optimization models.

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*.

Sequential Decisions, Recurrent Equations, Bayesian Approach, Distance Studies

1 Introduction

In this paper traditional tables and graphs are replaced by short description of algorithms and models with references to web-sites where on-line models 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

The main scientific tool is the theory of optimal sequential statistical decisions using the Bayesian approach [2, 3]. The 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 the on-line examples and the complete theoretical explanation is on the six mirror web-sites. This increases the reliability of Internet communications.

There are no "perfect" examples in these web-sites. All examples has some advantages and some disadvantages. Improvement of "non-perfect" models is interesting both for students and for colleagues. The main objective of this paper is to establish scientific collaboration in the Internet environment with distant colleagues and students.

The paper can be regarded as an extended introduction. The complete set of on-line models 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 two well-known optimization topics - Sequential Decisions [2] and Bayesian Approach [3] are regarded using different optimization models as examples. We regard that as a reasonable first step because both these topics are related to most of real life decisions. We often have to make decisions with incomplete information. The future is uncertain, too, as usual.

2 Sequential Statistical Decisions

Most of the decisions in a personal life and in organizations are made sequentially. That means that, at any given moment, one either makes the final decision, or postpones it hoping for better times. Statistical part of sequential decisions represents uncertainties of future events and a limited reliability of our observations.

The Bride problem is a good academic example of sequential statistical decisions [4, 2]. The dynamic programming is a conventional technique to optimize sequential decisions [5]. Applying the dynamic programming to specific problems, one develops specific algorithms, as usual. The algorithms, similar to these of bride's decision making, can be applied choosing the optimal time to buy durable goods, such as cars, houses, computers, to start new business, e.t.c..

Typical industrial applications of sequential decisions are the optimal stopping rules. For example, one can consider stopping of some nuclear plant as a marriage. One can evaluate stopping cost directly. Risks involved in running the plant, when one observes some deviations from the normal routine, can be considered as risks of waiting for the next proposal. That means that one hopes for something better while risking a bad outcome.

The important personal example is buying a PC. The feature of this example is rapid growth of average goodness of the product. We can buy much better machine after a year or two for the same money. Hovewer we may lost a lot of time working with an obsolete equipment.

We start by considering the simplest "single bride" example (no divorce). The underlying reason is the simplicity of calculations. We may solve this problem by textbook recurrent equations [5]. That is a good introduction. The "no-change" condition is not the realistic one therefore all the remaining examples regard the more realistic "multiple-bride" problems. Here changes are permitted for some price.

We often need many parameters to evaluate the real objective. That means a vector optimization problem. The important task is corect separation of the objective and the subjective factors. Both are important for decision maming but in the different ways. In good decision making systems users can include all the available objective information and may easily represent the subjective opinion. A simple way is by fixing some "weights" defining the subjective importance of various objective factors. The objective is a multi-modal, in general. Hovewer we will consider mainly uni-modal problems for simplicity.

First some examples of The "Demo" systems are regarded. Here the goodness of future grooms are defined by random numbers generator. The demo exapmles are extended by adding the "Expert" versions where the actual parameters of the object can be entered by users. The systems response is not "mandatory". A user may accept or reject the systems "advice" depending on his/her subjective judgement.

We finish the study by some generalization of the brides problem. The example is the RAM problem. Here the zero-one case (yes or no) is generalized to multi-valued problem; how many RAM units to order. We start by simple case when both the retail and the wholesale prices are known in advance. Then we consider more realistic case when only statistical distribution of future wholesale prices is known. The optimal retail prices are obtained by market elasticity. The market elasticity shows how the demand depends on rte retail price.

In the last example we consider the Bayesian optimization of parameters of some Logistic Model representing the real case in the static mode since the dynamic solution is dificult and not easily understandable.

Note that each semester most of the examples are tested and updated. Observed bugs are eliminated, accuracy and user interface are improved. The illustration is the interval of numerical integration. When expected values change we need to change this interval, too. Now that is implemented just in " the well tested example of optimal sequential statistical decisions using dynamic programming" (see example "Optimal Marriage Time Problem" , part 'Discrete Optimizatiom', web-site). In other examples the interval is fixed. This limits the 'trend' - the rate of change of expected values. In the 'Buy-a-PC' example the trend is the most important factor. Scalarization of the vector objective function is not always correct and convenient. Some important parameters are 'hidden' so recompilation is needed to change them. That shows how different studends understand the vector optimization problem. There are comments about the advantages and disadvantages of examples. Testing and updating the existing examples is important part of graduate studies. The result is general improvement of the web-sites. That reflects the ideas of the open source development.

3 Bayesian Approach (BA)

The Bayesian Approach (BA) is defined by fixing a prior distribution P on a set of functions $f(x)$ and by minimizing the Bayesian risk function $R(x)$ [3, 6]. 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, 3, 8, 9, 10]. That is important feature of the Bayesian approach in this setup. For example, if several values of some deterministic function $z_i = f(x_i),\ i = 1, ..., n$ 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. The aim of BA is to provide as small average error as possible.

4 Average Utility

The Bride problem is to maximize the average utility of marriage by the optimal choice of a groom. Denote the actual goodness of a groom i by ω_i. Denote by s_i the bride's impression about the groom i. $p(\omega_i)$ is a prior probability density of goodness ω_i. $p_s(s_i|\omega_i)$ is a probability density of impression s_i. Assume that goodness of different grooms are independent and identically distributed. This means that a prior probability density of goodness is

$$\begin{aligned} p(\omega_i, \omega_j) &= p(\omega_i)p(\omega_j), \\ p(\omega_i) &= p(\omega_j) = p(\omega). \end{aligned} \tag{1}$$

Suppose the probability density of an impression s_i, given the goodness ω_i, is

$$\begin{aligned} p_s(s_i, s_j|\omega) &= p_s(s_i|\omega)p_s(s_j|\omega), \\ p_s(s_i|\omega) &= p_s(s_j|\omega) = p(s|\omega). \end{aligned} \tag{2}$$

Assume the Gaussian prior probability density of goodness

$$p(\omega) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-1/2(\frac{\omega-\alpha_0}{\sigma_0})^2}. \tag{3}$$

Here σ_0 is a prior standard deviation and α_0 is a prior mean of goodness. For example, $\alpha_0 > 0$ shows an optimistic bride. $\alpha_0 < 0$ shows that a bride is pessimistic. Suppose, that a prior probability density of bride impressions is

$$p(s|\omega) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1/2(\frac{s-\omega}{\sigma})^2}. \tag{4}$$

Here σ^2 is a variance of impressions around the true goodness ω.

We tacitly assumed that both the groom goodness and the bride impression are random variables depending on many independent factors. This explains the Gaussian distributions (3) and (4). One defines a posterior probability density of goodness ω, given the impression s, by the Bayesian formula [11]

$$p(\omega|s) = \frac{p(s|\omega)p(\omega)}{p_s(s)}. \tag{5}$$

Here

$$p_s(s) = \int_{-\infty}^{\infty} p(s|\omega)p(\omega)d\omega. \tag{6}$$

5 Single-Marriage Case

Denote by d_i the bride's decision about the groom i

$$d_i = \begin{cases} 1, & \text{if bride marry the groom } i, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Suppose that

$$\sum_{i=1}^{m} d_i = 1. \tag{8}$$

The last condition means that brides marry, and marry only once. Formally condition (8) defines the set of feasible decisions when the n-th groom proposes

$$D_{N-n} = \begin{cases} 0 \; and \; 1, & \text{if } g_{N-n} = 0 \\ 0, & \text{if } g_{N-n} = 1, \end{cases} \tag{9}$$

Here g_{N-n} is the marriage index and N is a number of the last proposal.

$$g_{N-n} = 1 - \sum_{i=1}^{N-n-1} d_i. \tag{10}$$

The marriage index is zero, if the bride is marred. This prevents repeated marriages.

5.1 Bellman's Equations

The expected utility function is $u(s)$. Here s is the impression made by the successful groom[3].

[3] By the "successful groom" we mean the groom that a bride marries.

$$u(s) = \int_{-\infty}^{\infty} \omega p(\omega|s) d\omega. \tag{11}$$

Denote by $u_N(s)$ the expected utility function, if the impression of the last groom is s

$$u_N(s) = \int_{-\infty}^{\infty} \omega p(\omega|s) d\omega. \tag{12}$$

Comparing (11) and (12) one observes that

$$u(s) = u_N(s). \tag{13}$$

This follows from independence assumptions (1) and (2). Denote by u_{N-1} the expected utility, if the impression of the $(N-1)$ th groom is s and a bride is making the optimal decision $d = d_{N-1}(s) \in D_{N-1}$

$$u_{N-1}(s) = \max_d (du(s) + (1-d)u_N), \tag{14}$$

$$d_{N-1}(s) = arg \max_d (du(s) + (1-d)u_N). \tag{15}$$

Here

$$u_N = \int_{-\infty}^{\infty} u_N(s) p_s(s) ds. \tag{16}$$

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

$$u_{N-n}(s) = \max_d (du(s) + (1-d)u_{N-n+1}), \tag{17}$$

$$d_{N-n}(s) = arg \max_d (du(s) + (1-d)u_{N-n+1}). \tag{18}$$

Here

$$u_{N-n+1} = \int_{-\infty}^{\infty} u_{N-n+1}(s) p_s(s) ds. \tag{19}$$

Note, that the utility $u(s)$ of accepting a proposal in expressions (17) and (18) is a function only of impression s. It does not depend on the proposal number $N-n$. That follows from independence assumptions (1) and (2). Solving these recurrent equations we define the sequence of optimal decision functions $d_{N-n}(s) \in D_{N-n}$ and the expected utilities $u_{N-n}(s)$. This should be done for all possible impressions $s \in (-\infty, \infty)$ and for all numbers $n = 1, ..., N-1$. We cannot do that in continuous case. Therefore, we use a discrete approximation

5.2 Discrete Approximation

From expressions (11) and (12), replacing the integrals by sums we obtain that

$$u_N(s) = u(s) = 2M/K \sum_{k=1}^{K} \omega_k p(\omega_k|s). \tag{20}$$

From expression (16)

$$u_N = 2M/K \sum_{k=1}^{K} u_N(s_k) p_s(s_k). \tag{21}$$

From expression (19)

$$u_{N-n+1} = 2M/K \sum_{k=1}^{K} u_{N-n+1}(s_k) p_s(s_k). \tag{22}$$

Here $\omega_k \in [-M + \alpha_0, M + \alpha_0],\ \omega_1 = -M + \alpha_0,\ \omega_K = M + \alpha_0$ and $s_k \in [-M + \alpha_0, M],\ s_1 = -M + \alpha_0,\ s_K = M + \alpha_0$ where α_0 is a prior mean of goodness. Note that α_0 often is a function of time $\alpha_0 = \alpha_0(t)$ That is a discrete approximation of the recurrent equations. All possible impressions $s_k \in [-M + \alpha_0, M + \alpha_0]$ and all numbers $n = 1, ..., N-1$ are considered. We set the number of iterations K by the accuracy needed.

The results are sequences of optimal decision functions $d_{N-n}(s_k)$ and the expected utilities $u_{N-n}(s_k)$. These sequences are stored in a set of arrays which define how the optimal decisions d and the expected utilities u depend on the possible impressions $s_k,\ k = 1, ..., K$. Doing this, one avoids the repeated calculations. Large arrays is a disadvantage of this procedure.

5.3 Including the Waiting Cost

The waiting losses are important in many real-life sequential decision problems. Denote by c the loss of waiting for the next groom. Including this parameter into Bellman equations one obtains

$$u_{N-1}(s) = \max_d (du_N(s) + (1-d)(u_N - c)), \tag{23}$$

$$d_{N-1}(s) = arg \max_d (du_N(s) + (1-d)(u_N - c)). \tag{24}$$

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

$$u_{N-n}(s) = \max_d (du_N(s) + (1-d)(u_{N-n+1} - c)), \tag{25}$$

$$d_{N-n}(s) = arg \max_d (du_N(s) + (1-d)(u_{N-n+1} - c)). \tag{26}$$

The other expressions remains the same.

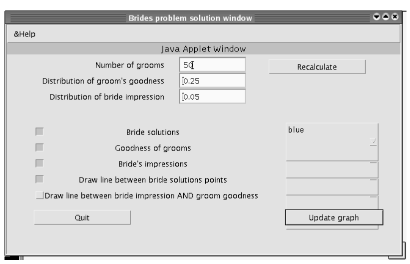
5.4 Non-Linear Case

Expression (12) was defined assuming the linear bride's utility function. It was supposed that bride's utility is equal to the goodness of groom $u(\omega) = \omega$. In the real life, utility functions are nonlinear [1] and non-convex, as usual. Then expression (12) is replaced by this integral

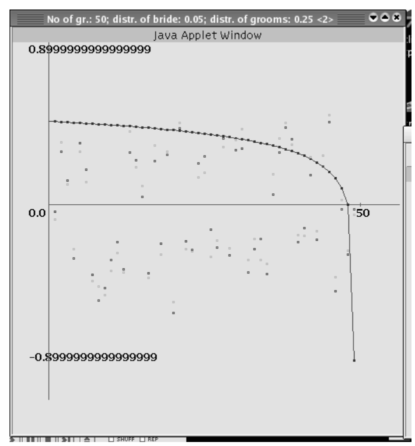
$$u_N(s) = \int_{-\infty}^{\infty} u(\omega) p(\omega|s) d\omega. \tag{27}$$

5.5 Software of "Single-Marriage"

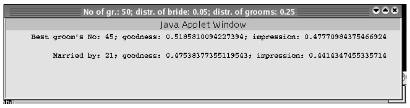
Considering software we refere to parts of web-sites with corresponding applets or servlets. Applets implementing examples of this paper are in "Discrete Optimization". The "Single-Marriage" example is in
"Optimal Marriage Time Problem":
examples of optimal sequential statistical decisions using dynamic programming
Bride-1, Java1
Figure Mini. 1., shows the input window of the "Sigle-Marriage" example, the simulated numeric results are in Mini. 2. and the optimal decision function defining how the critical "goodness" of groom depends on time is in Mini. 3. The time is defined as a groom number. That means "uniform arrival" of grooms.



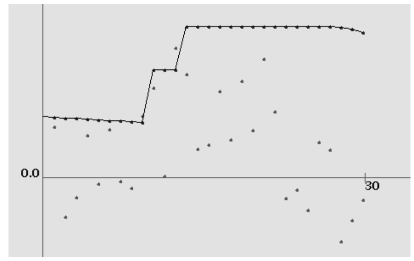
Mini. 1. Input window of "Single-Marriage"'



Mini. 2. Decision function of "Single-Marriage".



Mini. 3. Numeric results of "Single-Marriage".



Mini. 4. Decision function of "Multi-Marriage"

The figure Mini. 4. shows the Multi-Marriage example for comparison.

Condition (8) implies that brides marry and marry only once. No divorce is permitted. That is not realistic assumption made just to simplify the Bellman equations.

6 "Multi-Marriage"

There are several sofware examples of the "Multi-Marriage" problem. The most accurate is
"the well tested example of optimal sequential statistical decisions using dynamic programming"
Bride-Multi, Java2
The screenshots of this example are in figures 1,2

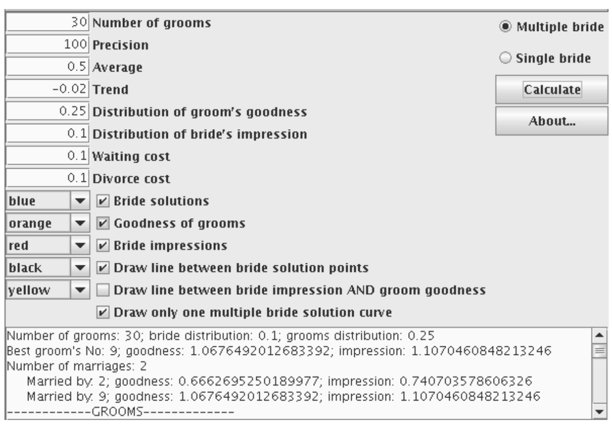


Fig. 1. The output window of the "Grooms" problem.

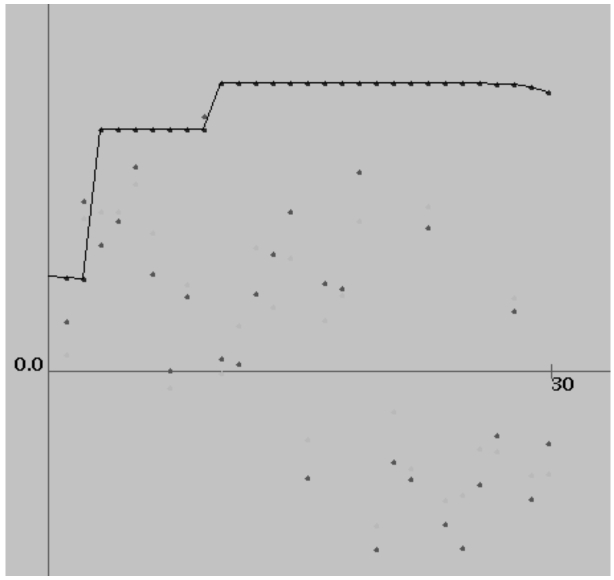


Fig. 2. The decision function of "Grooms" problem.

6.1 "Buy-a-PC" example

A good illustration of Bellman equations describing the "Multi-marriage" problem.

Define PC parameters by a vector $g = (g_0, g_1, g_2, g_3)$. Here g_0 is the market price of PC in \$, g_1 is the speed of CPU in MHz, g_2 is the volume of RAM in MB, and g_3 is the volume of HD in GB.
Express a subjective utility of PC by the weighted sum

$$\omega = a_1 g_1 + a_2 g_2 + a_3 g_3. \tag{28}$$

Here a_i is user evaluations of utility of quality parameter g_i expressed in \$ per unit. The users evaluation ω differs from the market price g_0, as usuall.
The PC utility defined as a sum of utilities of different components is just first approximation. Therefore, we need to evaluate different PC configurations separately.

The problem is how to define user evaluation based on the utility theory.
Step 1: define a set of events E as a set of PC described by different feasible vectors (g_1, g_2, g_3, g_4)
Step 2: define a sequence of events $E_i, i = 1, ..., I$ ordered by the condition $E_{i-1} \leq E_i \leq E_{i+1}$. Condition $E_i \leq E_{i+1}$ means that we prefer PC E_{i+1} to E_i.
Step 3: set the normalized utility functions $u_0(E_1) = 0$ and $u_0(E_I) = 1$
Step 4: define the remaining normalized utility functions $u_0(E_i) = p_i$ where p_i is the 'hesitation' probability determined by the lottery:

$$E_i \sim \{p_i E_I + (1 - p_i) E_1\}. \tag{29}$$

Define by h rhe highest price a user is ready to pay for the best PC. Then the general utility $u(E_i)$ of the PC of lesser configuration E_i is

$$u(E_i) = h * u_0(E_i) - g_{0i} \tag{30}$$

Or, in case of linear approximation:

$$u(E_i) = \omega_i - g_{0i} \tag{31}$$

where ω_i is goodness of PC E_i calculated by (28) and g_{0i} is the market price.

Considering real-lfe examples we need two GUI (Graphic User Interface) versions. We call them "Demo" and "Expert". In the Demo version 3 the "state of nature" (for example MB, GB, and GHz) is defined by some random number generator. In the Expert versions 4, 6 theese values are entered by users, represents the real date and reflects subjective opinions. The final decision is defined by users and can be different from the computer suggestions.

6 Assume that a prior probability density of PC utilities is Gaussian

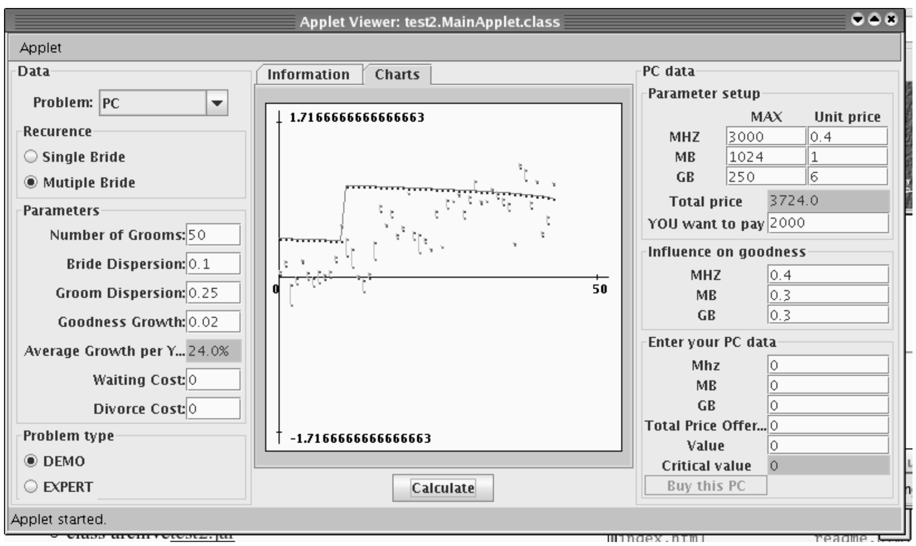


Fig. 3. The window of the demo "Buy-a-PC".

$$p_t(\omega) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-1/2(\frac{\omega - \alpha_t}{\sigma_0})^2}. \tag{32}$$

Here σ_0 is a prior variance and α_t is a prior mean of PC utilities. Suppose that $\sigma_0 = constant$ and that $\alpha_t = \alpha_0 + \alpha_1 t$. This means that expected PC utility is increasing linearly. The expected diversity remains the same.

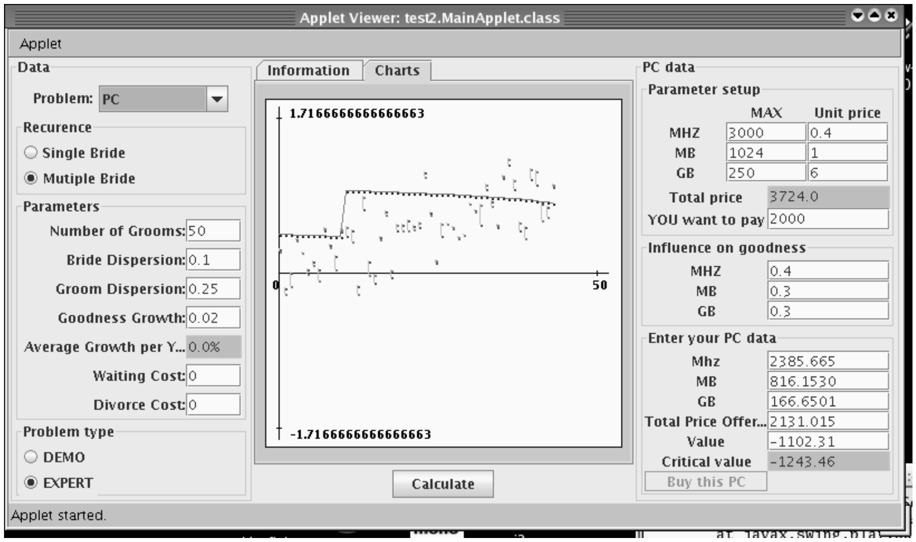


Fig. 4. The input window of the expert "Buy-a-PC".

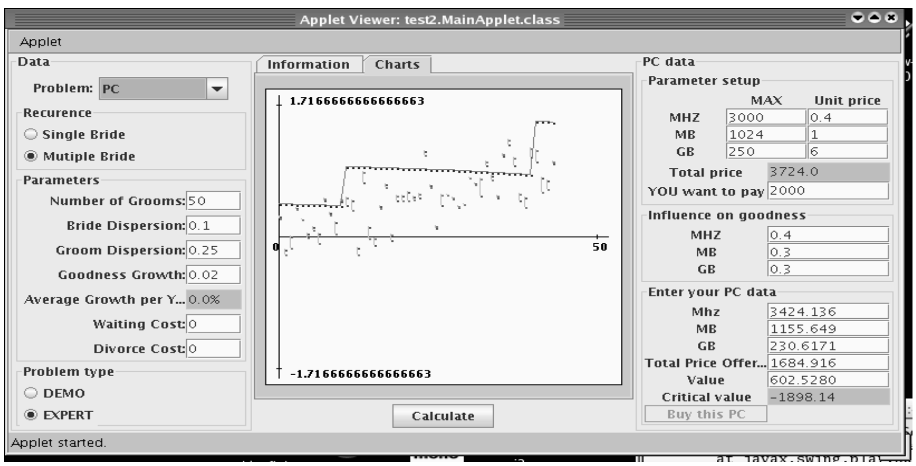


Fig. 5. The output window of the expert rejecting "Buy-a-PC" recomendations.

Suppose that the price g_0 of PC depends linearly on the weighted sum

$$g_0 = b_1 g_1 + b_2 g_2 + b_3 g_3. \tag{33}$$

Here parameters b_i are defined in \$ per unit and reflect the market prices of CPU, RAM, and HD. Assume that one observes the utility ω exactly. This means that the impression $s = \omega$ and that impression errors $\sigma = 0$ in expression (2). That simplifies the problem.

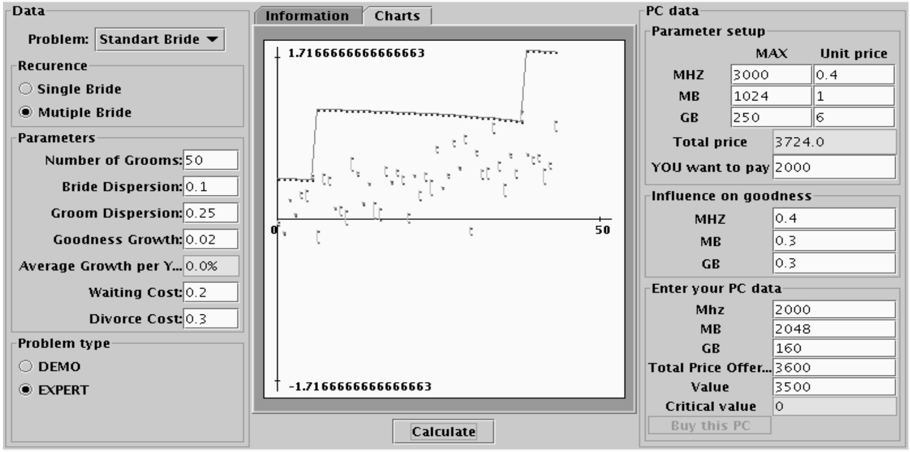
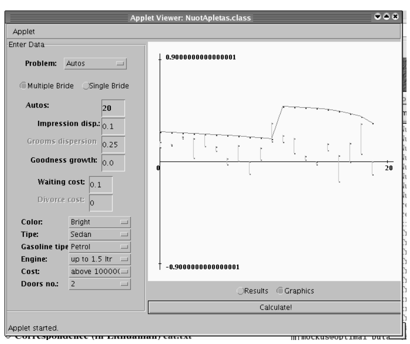


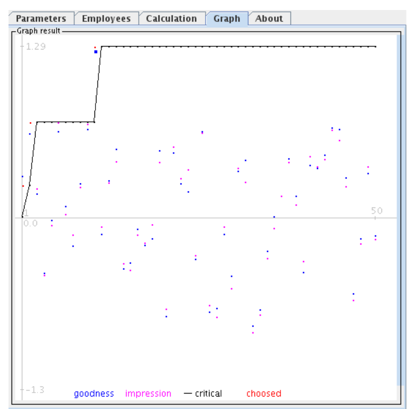
Fig. 6. The decision function.

6.2 "Buy-a-Car"

The "Buy-a-Car" example Mini. 5. and Mini. 6. applies modified "Multi-Marriage" software . There are three options:
- autos "Buy-a-Car".
- workers "Employ-a-person",
- grooms "Marry-a-Groom".
The implementation of Bellman equations is just an initial approximation that should be improved.



Mini. 5. The decision function of "Buy-a-Car".



Mini. 6. The decision function of "Employ-a-Person".

6.3 "Employ-a-Person"

Another "Multi-Marriage" example is "Employ-a-Person" problem. Here the corrected version of Bellman equations is used and both the demo and expert vesions are implemented.. Figure 6.2 shows the decision function. Figure 7 illustrates the input of demo version

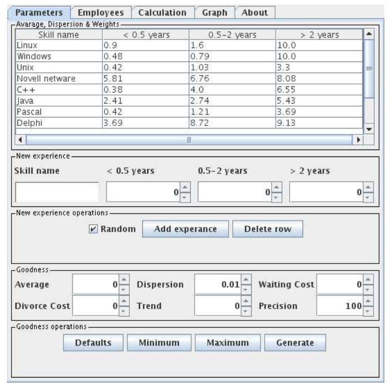
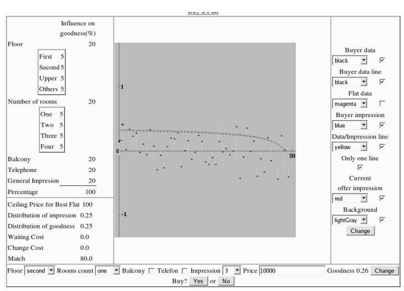


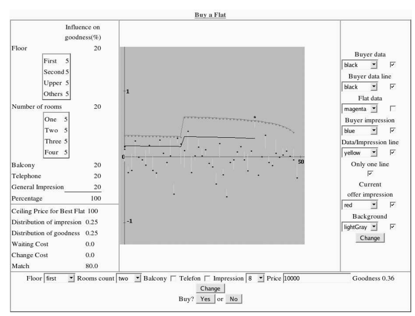
Fig. 7. Initial data of "Employ-a-Person"

6.4 "Buy-a-Flat"

A "Buy-a-Flat" example implements "Multi-Marriage" model as an Java servlet . In the servlet mode all the calculations are performed at the server side. The users provide data and regard results Mini. 7. (Demo mode) and Mini. 8. (Expert mode). The user interface performs the scalarization of the vector objective in a "friendly" way.



Mini. 7. The initial window of "Buy-a-Flat".



Mini. 8. The next window of "Buy-a-Flat".

6.5 Bellman's Equations

The expected utility function is $u(\omega, q)$. Here ω is the utility of a new PC. q is the utility of the old PC, to be replaced by the new one. Consider a "horizon" of N years. During a year one can change PC only once, if one wishes.

Denote by $u_N(\omega, q)$ the maximal expected utility in the year N

$$u_N(\omega, q) = \max_d (d\omega + (1-d)(q_N - c_N)). \tag{34}$$

There are two possible decisions $d = \{0, 1\}$. The decision $d = 1$ means to buy a new PC. The utility of the new PC is ω. The utility of the old one is q.

The utility of the decision $d = 0$, to keep the old PC in the last year N, is $q_N + c_N$. Here $c_N = \tau_N - g_0(N)$ is the penalty of refusing to buy a new PC. This includes the waiting losses τ_N minus the price $g_0(N)$ of new PC in the year N defined. It is assumed that we abandon the old PC as soon as we obtain the new one. Therefore, one "wins" the price g_0 of the new PC by using the old PC. The optimal decision in the last year N

$$d_N^* = \begin{cases} 1, & \text{if } \omega_N > q_N - c_N, \\ 0, & \text{if } \omega_N \le q_N - c_N. \end{cases} \tag{35}$$

Denote by $u_{N-1}(\omega, q)$ the maximal expected utility in the year $N-1$.

$$u_{N-1}(\omega, q) = \max_d (d\omega + (1-d)(u_N(q_N) - c_{N-1})). \tag{36}$$

Here $u_N(q)$ is the maximal expected utility in the year N, if the utility of the old PC is q.

$$u_N(q) = \int_{-\infty}^{\infty} u_N(\omega, q) p_N(\omega) d\omega. \tag{37}$$

$p_N(\omega)$ is a prior probability density of ω defined by expression (32) at a time $t = N$.

$$q_N = \begin{cases} \omega_{N-1}, & \text{if } q_N < q_{N-1}^*, \\ q_{N-1}, & \text{if } q_N \ge q_{N-1}^*. \end{cases} \tag{38}$$

Here q_{N-1}^* is obtained from this equation

$$\omega_{N-1} = u_N(q_{N-1}^*) - c_{N-1} \tag{39}$$

The optimal decision in the year $N-1$

$$d_{N-1}^* = \begin{cases} 1, & \text{if } \omega_{N-1} > u_N(q_N) - c_{N-1}, \\ 0, & \text{if } \omega_{N-1} \le u_N(q_N) - c_{N-1}. \end{cases} \tag{40}$$

Denote by $u_{N-i}(\omega, q)$ the maximal expected utility in the year $N-i, \ i = 1, ..., N-1$. Then

$$u_{N-i}(\omega, q) = \max_d (d\omega + (1-d)(u_{N-i+1}(q_{N-i+1}) - c_{N-i})). \tag{41}$$

Here $u_{N-i+1}(q)$ is the maximal expected utility in the year $N-i+1$, if the utility of the old PC is q.

$$u_{N-i+1}(q) = \int_{-\infty}^{\infty} u_{N-i+1}(\omega, q) p_{N-i+1}(\omega) d\omega. \tag{42}$$

$p_{N-i+1}(\omega)$ is a prior probability density of ω defined by expression (32) at a time $t = N - i + 1$.

$$q_{N-i+1} = \begin{cases} \omega_{N-i}, & \text{if } q_{N-i+1} < q^*_{N-i}, \\ q_{N-i}, & \text{if } q_{N-i+1} \geq q^*_{N-i}. \end{cases} \tag{43}$$

q^*_{N-i} is obtained from the equation

$$\omega_{N-i} = u_{N-i+1}(q^*_{N-i}) - c_{N-i} \tag{44}$$

The optimal decision in the year $N - i$

$$d^*_{N-i} = \begin{cases} 1, & \text{if } \omega_{N-i} > u_{N-i+1}(q_N - i + 1) - c_{N-i}, \\ 0, & \text{if } \omega_{N-i} \leq u_{N-i+1}(q_{N-i+1}) - c_{N-i}. \end{cases} \tag{45}$$

Here the decision functions d^*_{N-i} depend on two variables ω, q defining the corresponding utilities of the new and the old computer. That is difficult for both the calculations and grafical representations.

One applies the discrete approximation such as in Section 5.2. The optimal decision functions depend on two variables ω and q. Therefore, one needs K times more calculations to obtain the same accuracy as in the single marriage case (see Section 5.2).

To solve real life problems the "directly unobservable" factors should be included into the "Buy-a-PC" model. For example , one can estimate the expected life-span $T(s)$ of PC, given the impression s, by the Bayesian formula

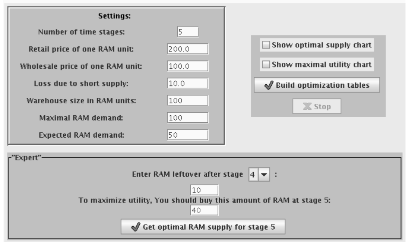
$$T = \int_{-\infty}^{\infty} \tau p(\tau|s) d\tau, \tag{46}$$

$$p(\tau|s) = \frac{p(s|\tau)p(\tau)}{p_s(s)}. \tag{47}$$

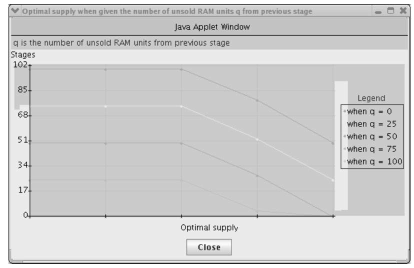
Here τ is the life-span of PC and s is the user's impression about its reliability. The impression s depends on the warranty, the reputation of the manufacturer and on some subjective factors, too.

7 Optimal Ordering of Supplies

The important application a of sequential statistical decisions is optimal ordering of supplies. Consider, for example, same computer shop that orders d units of RAM (Random Access Memory) at fixed times $t = 1, ..., N$ dividing the time into N stages of unit length each.



Mini. 9. The recommended decision of ”Order-Ram”.



Mini. 10. A set of decision functions of ”Order-Ram”.

The recommended decision is in Mini. 9. A set of decision functions are in Mini. 10.

In this example the wholesale price and the demand of RAM units is fixed. Usually we have just some statistical data about the distribution of wholesale prices. The demand is defined by market elasticity. The elasticity shows how the demand depends on the retail prices. The updated ”genOrder-RAM” model regards both: the uncertainty of wholesale prices and the elasticity of market. The results are in Fig. 8, Fig. 9, and Fig. 10. Figures Fig. 11, Fig. 12 show how the utility function and the optimal supply depends on the wholesale price.

Fig. 8. Initial data of ”genOrder-RAM”

7.1 Uncertain Demand

One optimizes the supply d when the demand ω is unknown. Denote by ω a number of RAM units demanded during one stage. Denote the retail price of one RAM unit by c_r and the wholesale price by $c_w \leq c_r$. The utility function-the profit-at the $(N - i)$-th stage

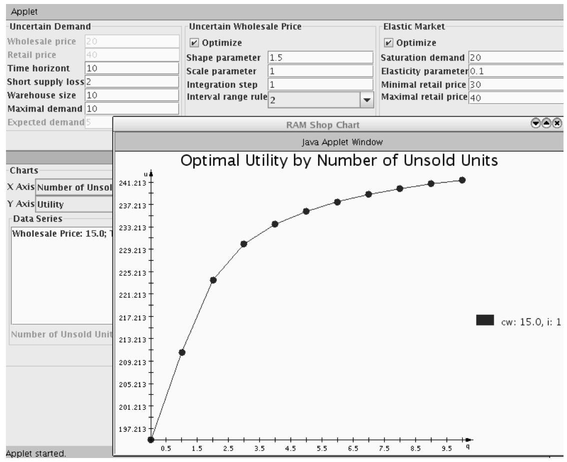


Fig. 9. Utility-to-unsold units, "genOrder-RAM".

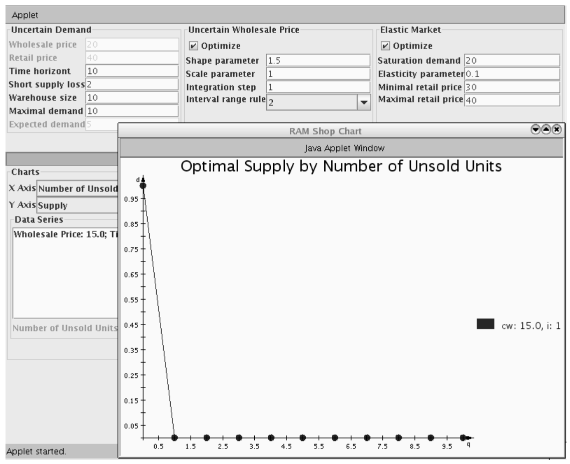


Fig. 10. Supply-to-unsold units, "genOrder-RAM".

$$v_{N-i}(\omega, d, q) = \begin{cases} c_r\omega - c_w d, & \text{if } b \geq \omega, \\ c_r\omega - c_w d + L(b-\omega), & \text{if } b < \omega. \end{cases} \tag{48}$$

Here $c = c_r - c_w$ and $b = d + q$, q is the number of unsold units.
L is the loss due to short supply including fines for breaking supply contracts and/or damages to reputation as an reliable retailer.
All these parameters correspond to the stage $N-i,\ i = 0, 1, ..., N-1$, meaning that
$\omega = \omega_{N-i},\ d = d_{N-i}, q = q_{N-i}$
where ω_{N-i} is the demand at the stage $N - i$,
$d = d_{N-i} \in D$ is the supply belonging to a feasible set $D = D_{N-i}$ at this

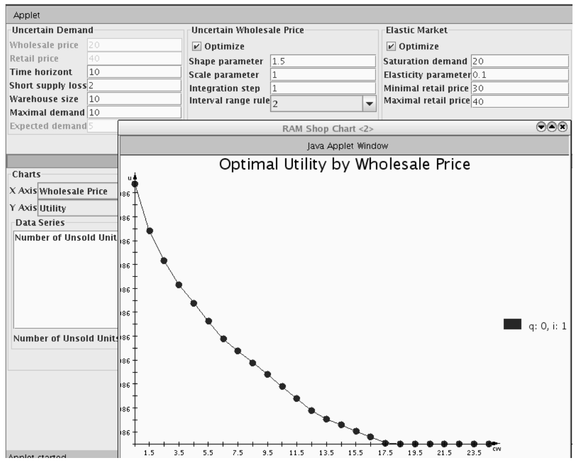


Fig. 11. Utility-to-wholesale price, "genOrder-RAM"

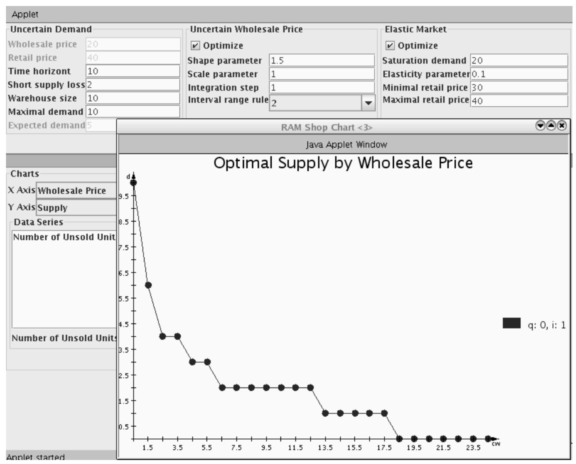


Fig. 12. Supply-to-wholesale price, "genOrder-RAM".

stage, and
$q = q_{N-i-1} = d_{N-i-1} + q_{N-i-2} - \omega_{N-i-1}$
is the number of unsold RAM units, left from the previous stage $N - i - 1$. Here and later we omit the lower indices to make expressions shorter.

Denote by $p_{N-i}(\omega)$ the probability of demand $\omega_{N-i} = \omega$ at the stage $N - i$.
Usually the demand $\omega = 0, 1, 2, ...$ during the time τ is described by the Poisson distribution [13]

$$p_{N-i}(\omega) = e^{-\lambda\tau} \frac{(\lambda\tau)^{\omega}}{\omega!}. \tag{49}$$

Considering stages of unit time $\tau = 1$ and

$$p_{N-i}(\omega) = e^{-\lambda} \frac{\lambda^{\omega}}{\omega!}, \tag{50}$$

where $\lambda = \lambda_{N-i}$ is the expected demand at the stage $N-i$. Then the expected utility at the stage $N-i,\ i = 0, ..., N-1$

$$u_{N-i}(d,q) = \sum_{\omega} v_{N-i}(\omega, d, q)\ p_{N-i}(\omega) \tag{51}$$

The maximal expected utility at the last stage N

$$u_N(q) = \max_{d \in D_N} \sum_{\omega} v_N(\omega, d, q)\ p_N(\omega) \tag{52}$$

where $q = q_{N-1}$ is a number of unsold RAM units available at the beginning of the N-th stage[4],
$d = d_N$ is a number of RAM units ordered at the beginning of the N-th stage, $\omega = \omega_N$ is the demand at the stage N, D_N is a set of feasible supplies at the stage N.

Denote by $d_N(q)$ the optimal decision function maximizing the expected utility (52) at any fixed q.

Denote by $u_{N-1}(q)$ the maximal expected utility at the $(N-1)$-th stage

$$u_{N-1}(q) = \max_{d \in D_{N-1}} (u_{N-1}(d,q) + \sum_{\omega}\ u_N(q)\ p_N(\omega)). \tag{53}$$

Note, that the meaning of parameter q depends on the index of the function u, for example, $q = q_{N-1}$, if the index is N, and $q = q_{N-2}$, if this index is $N-1$, etc.

The maximum of (53) is obtained by the optimal decision function $d_{N-1}(q)$ where $q = q_{N-2}$.

Extending (53) to $i = 0, ..., N-1$ one obtains

$$u_{N-i}(q) = \max_{d \in D_{N-i}} (u_{N-i}(d,q) + \sum_{\omega}\ u_N(q)\ p_{N-i+1}(\omega)) \tag{54}$$

Solving recurrent equations (52)-(54) one defines the sequence of optimal decision functions
$d_{N-i}(q),\ i = 0, ..., N-1$
and the optimal expected utility functions $u_{N-i}(q)$.
This is done for all possible remaining RAM units q and for all stages $i = 1, ..., N-1$.

In the software terms that means calculating recurrently the sequence $i = 0, 1, ..., N-1$ of arrays that shows how the optimal supply $d = d_{N-i}(q)$

[4] The meaning of the argument q of the function $u_N(q)$ is defined by the index N, for example, $q = q_{N-1}$, if this index is N, etc.

and the maximal profit $u = u_{N-i}(q)$ depend on the unsold RAM units q, left from the previous stages. This way one reduces computing time considerably, because array access operations need much less time as compared to multi-dimensional optimization.

7.2 Uncertain Wholesale Price

Denote by $p^c_{N-i}(c_w) = P\{c_w(N-i) = c_w\}$ a probability that the wholesale price $c_w(N-i)$ at the stage $N-i$ is c_w. Denote by $C_w(N-i)$ the expected wholesale price c_w at the stage $N-i$. Assuming the Gaussian distribution of the wholesale price logarithm, one obtains the lognormal density function [13]

$$p^c_{N-i}(c_w) = \frac{1}{\alpha c_w \sqrt{2\pi}} e^{-\frac{(ln(c_w)-\gamma)^2}{2\alpha^2}} \tag{55}$$

where c_w means $c_w(N-i)$, $\alpha > 0$ is the shape parameter, and γ is the scale parameter. The expected value $C_w(N-i)$ and the standard $S_w(N-i)$ of the lognormal distribution

$$C_w(N-i) = e^{\gamma+\alpha^2/2} \tag{56}$$

$$S_w(N-i) = e^{\alpha^2} - 1 \tag{57}$$

The lognormal density is unimodal with maximum at $c_w = exp(\gamma - \alpha^2)$.

In the case of random wholesale prices $c_w = c_w(N-i)$, the profit at the $(N-i)$-th stage

$$v_{N-i}(\omega, d, q) = \begin{cases} c_r\omega - c_w d, & \text{if } b \geq \omega, \\ c_r\omega - c_w d + L(b-\omega), & \text{if } b < \omega. \end{cases} \tag{58}$$

Here $c = c_r - c_w(N-i)$, $q = q_w(N-i-1)$ is the number of unsold units. Then the expected utility at the stage $N-i, \ i = 0, ..., N-1$

$$u_{N-i}(d, q) = \sum_\omega \int_0^\infty v_{N-i}(\omega, d, q) \ p^c_{N-i}(c_w) dc_w \ p_{N-i}(\omega) \tag{59}$$

The maximal expected utility at the last stage N

$$u_N(q) = \max_{d \in D_N} \sum_\omega \int_0^\infty v_N(\omega, d, q) \ p^c_N(c_w) dc_w \ p_N(\omega) \tag{60}$$

where $q = q_{N-1}$ is a number of unsold RAM units left at the beginning of N-th stage,
$c_w = c_w(N)$ is the wholesale price at the stage N.

Denote by $d_N(q)$ the optimal decision function maximizing the expected utility (60) at any fixed q.

Denote by $u_{N-1}(q)$ the maximal expected utility at the $(N-1)$-th stage

$$u_{N-1}(q) = \max_{d \in D_{N-1}} (u_{N-1}(d,q) +$$
$$\sum_{\omega} \int_0^{\infty} u_N(q)\ p_N^c(c_w) dc_w\ p_N(\omega)). \tag{61}$$

That is achieved by the optimal decision function $d_{N-1}(q)$
where $q = q_{N-2}$.

Extending to $i = 0, ..., N-1$ one obtains

$$u_{N-i}(q) = \max_{d \in D_{N-i}} (u_{N-i}(d,q) +$$
$$\sum_{\omega} \int_0^{\infty} u_N(q)\ p_{N-i+1}^c(c_w) dc_w\ p_{N-i+1}(\omega)) \tag{62}$$

Solving recurrent equations (60)-(62) one defines the sequence of optimal decision functions
$d_{N-i}(q),\ i = 0, ..., N-1$
and the optimal expected utilities $u_{N-i}(q)$.
This is done for all possible numbers of RAM units q available at the beginning of stages $i = 1, ..., N-1$.

7.3 Market Elasticity

We call by Market Elasticity the relation $\Omega = \Omega(c_r)$ of expected demand Ω to the retail price $c_r \in C$, where C is a set of feasible retail prices. A simple approximation is the exponential function

$$\Omega(c_r) = be^{-\beta c_r} \tag{63}$$

where $b = b_{N-i}$ is the saturation demand, $\beta = \beta_{N-i}$ is the elasticity parameter, $c_r = c_r(N-i) \in C_{N-i}$ is the retail price, and C_{N-i} is a set of feasible retail prices, all at the stage $N-i$.

In this case the profit function at the $(N-i)$-th stage

$$v_{N-i}(\omega, d, q, c_r) = \begin{cases} c_r\omega - c_w d, & \text{if } b \geq \omega, \\ c_r\omega - c_w d + L(b-\omega), & \text{if } b < \omega. \end{cases} \tag{64}$$

Here c denotes the difference $c = c_r(N-i) - c_w(N-i)$ of retail and wholesale prices at the stage $N-i$.
The expected utility at the stage $N-i,\ i = 0, ..., N-1$

$$u_{N-i}(d,q,c_r) = \sum_{\omega} \int_0^{\infty} v_{N-i}(\omega, d, q)\ p_{N-i}^c(c_w) dc_w\ p_{N-i}(\omega) \tag{65}$$

where $c_r = c_r(N-i)$.
The maximal expected utility at the last stage N

$$u_N(q) = \max_{d \in D_N, c_r \in C_N} \sum_{\omega} \int_0^{\infty} v_N(\omega, d, q, c_r)\ p_N^c(c_w) dc_w\ p_N(\omega) \quad (66)$$

Denote by $d_N(q) = (d_N(q), c_N(q))$, where q denotes q_{N-1}, the two-dimensional optimal decision function maximizing the expected utility (66) at any fixed q.

Denote by $u_{N-1}(q)$ the maximal expected utility at the $(N-1)$-th stage

$$u_{N-1}(q) = \max_{d \in D_{N-1}, c_r \in C_{N-1}} (u_{N-1}(d, q, c_r) + \int_0^{\infty} \sum_{\omega} u_N(q)\ p_N^c(c_w) dc_w\ p_N(\omega)). \quad (67)$$

That is achieved by the two-dimensional optimal decision function $d_{N-1}(q) = (d_{N-1}(q), c_{N-1}(q))$
where q means q_{N-2}.

Extending (67) to $i = 0, ..., N-1$ one obtains

$$u_{N-i}(q) = \max_{d \in D_{N-i}, c_r \in C_{N-i}} (u_{N-i}(d, c_r, q) + \sum_{\omega} \int_0^{\infty} u_N(q)\ p_{N-i+1}^c(c_w) dc_w\ p_{N-i+1}(\omega)) \quad (68)$$

Solving recurrent equations (66)-(68) one defines the sequence of decision functions
$d_{N-i}(q) = (d_{N-i}(q), c_{N-i}(q))\ i = 0, ..., N-1$
and the optimal expected profits $u_{N-i}(q)$.
This is done for all possible numbers q of RAM units q available at the beginning of stages $i = 1, ..., N-1$.

In the case of market elasticity, to represent maximal profits $u_{N-i}(q)$ and two-dimensional optimal decisions $d_{N-i}(q) = (d_{N-i}(q), c_{N-i}(q))\ i = 0, ..., N-1$ as functions of remaining RAM units q one needs to calculate a sequence of N arrays defining the sequence of two-dimensional decision functions. This is difficult, therefore, one starts from the simplest case of uncertain demand (see section 7.1), as usual.

Note that in the actual calculations of expressions (60)-(68) the integrals are replaced by sums and the densities by probabilities using expressions similar to those in the section 5.2.

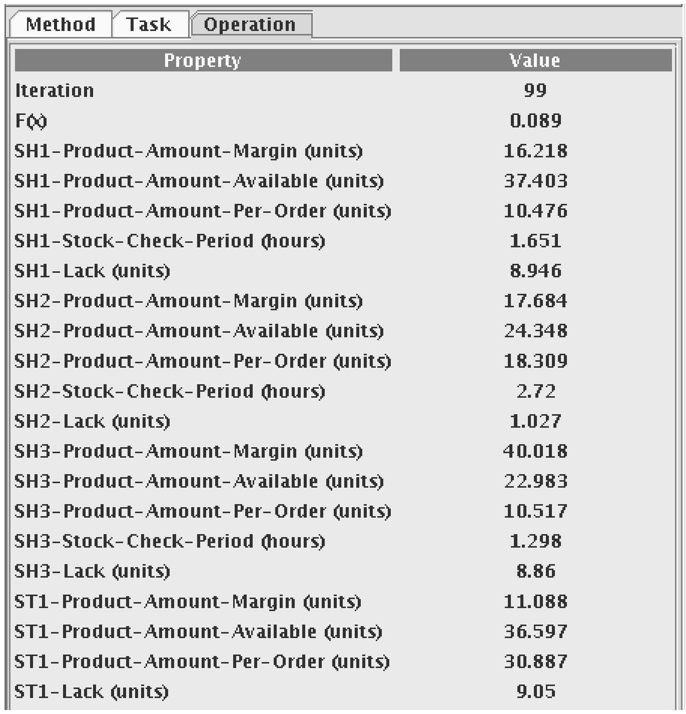
In the demonstration version the "horizon" N means the estimated end of business. In the expert version one uses the "moving horizon" $N_o < N$, repeating the optimization after each stage. This way one simplifies calculations and updates data.

The recurrent equations (52)-(54) include Bride problems as special cases with only two feasible decisions $D = \{0, 1\}$ and different utility functions (48).

8 "Logistic Model"

In the Logistic Model the heuristic decisions are regarded. The search for the optimal parameters of these heuristics are performed using the Bayesian Heuristic Approach [1]. The results are illustrated by a set of figures. Fig. 13 shows the input data. Fig. 14 shows the results of "Logistic Model" parameters optimization. Fig. 15 shows Step 1. Initialize "Logistic Model". Fig. 16 shows Step 2. Modify Data of "Logistic Model". Fig. 8 shows Step 3. Simulation of "Logistic Model". Fig. 8 shows Step 4. Analyzis of "Logistic Model". Fig. 9 shows the contents of "Help" file .

Fig. 13. Input window, "Logistic Model".



Property	Value
Iteration	99
F(x)	0.089
SH1-Product-Amount-Margin (units)	16.218
SH1-Product-Amount-Available (units)	37.403
SH1-Product-Amount-Per-Order (units)	10.476
SH1-Stock-Check-Period (hours)	1.651
SH1-Lack (units)	8.946
SH2-Product-Amount-Margin (units)	17.684
SH2-Product-Amount-Available (units)	24.348
SH2-Product-Amount-Per-Order (units)	18.309
SH2-Stock-Check-Period (hours)	2.72
SH2-Lack (units)	1.027
SH3-Product-Amount-Margin (units)	40.018
SH3-Product-Amount-Available (units)	22.983
SH3-Product-Amount-Per-Order (units)	10.517
SH3-Stock-Check-Period (hours)	1.298
SH3-Lack (units)	8.86
ST1-Product-Amount-Margin (units)	11.088
ST1-Product-Amount-Available (units)	36.597
ST1-Product-Amount-Per-Order (units)	30.887
ST1-Lack (units)	9.05

Fig. 14. Output window, "Logistic Model".

9 "Buy-a-Flat" by *C#*

All the examples are by Java. The reason is that Java software can run by different OS both in the form of applets (using users computer) and servlets (using server resources). The nearest alternative is *C#* (C-sharp). The C-sharp is a propriatory Microsoft software and runs by Windows. C-sharp can run by Unix, too, using "Mono" system. Hovewer there are no "applets" in C-sharp. Fig. 8 shows the "Buy-a-Flat" example implemented by *C#* and "Mono".

Distance Studies

For the graduate level distance and class-room studies of the theory of optimal statistical decisions in the Internet environment a set of examples of corresponding optimization models was implemented by 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

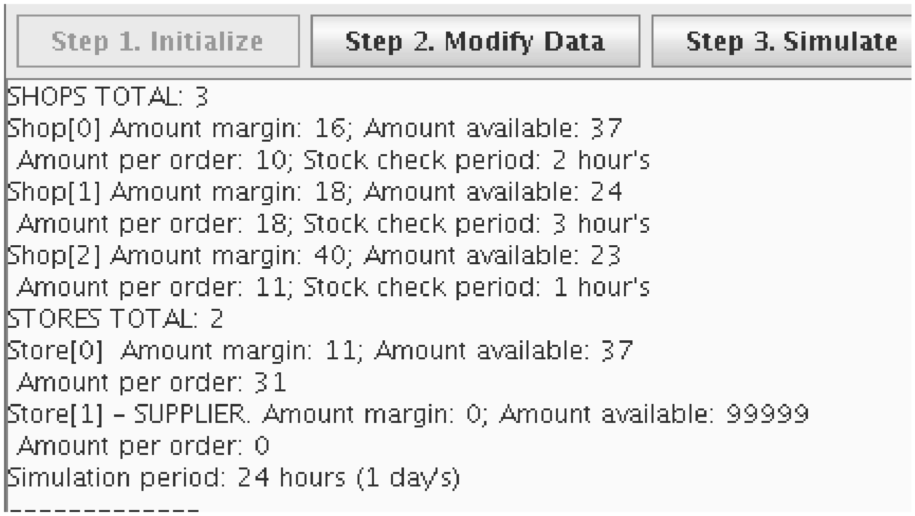


Fig. 15. Step 1. Initialize, "Logistic Model".

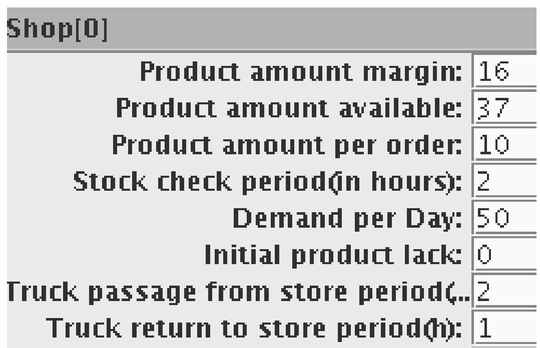


Fig. 16. Step 2. Modify, "Logistic Model".

http://proin.ktu.lt/~mockus
http://mockus.us/optimum
The theoretical background and the complete description of the software is in the file 'stud2.pdf'. Examples are described in the section: "Discrete Optimization" for discrete optimization and applications of linear and dynamic programming.

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

```
SHOPS TOTAL: 3
Shop[0] Amount margin: 16; Amount available: 37
 Amount per order: 10; Stock check period: 2 hour's
Shop[1] Amount margin: 18; Amount available: 24
 Amount per order: 18; Stock check period: 3 hour's
Shop[2] Amount margin: 40; Amount available: 23
 Amount per order: 11; Stock check period: 1 hour's
STORES TOTAL: 2
Store[0]  Amount margin: 11; Amount available: 37
 Amount per order: 31
Store[1] - SUPPLIER. Amount margin: 0; Amount available: 99999
 Amount per order: 0
Simulation period: 24 hours (1 day's)
-------------
TICK[1] - STARTED.
E_SHOPCHECKSTOCK: Shop     [2]      Value=0
    - Shop[2]: Ordering products amount[11] in store 0
    - Shop[2]: Total products ordered:[11]
    - Store[0]: Recieved an order from shop 2
    - Store[0]: Fulfiling order. Will be acomplished after 1 hour's
E_STORECHECKSTOCK: Store     [0]      Value=0
TICK[1] - COMPLETE.
----- RESULTS-----
Shop[0] - Amnt avail.: 33; Demand[4]; Lack: 0; Products ordered: 0
Shop[1] - Amnt avail.: 19; Demand[5]; Lack: 0; Products ordered: 0
Shop[2] - Amnt avail.: 19; Demand[4]; Lack: 0; Products ordered: 11
--------------------
Store[0] - Amnt avail.: 26; Truck Count[10]; Lack: 0; Products ordered: 0
Store[1] - Amnt avail.: 99999; Truck Count[10]; Lack: 0; Products ordered: 0
--------------------
TICK[2] - STARTED.
E_ORDERFULFILCOMLETE: Store     [0]      Value=2
    - Store[0]: Order Accomplished. Truck ready to carry products to shop ID[2]
E_SHOPCHECKSTOCK: Shop     [0]      Value=0
TICK[2] - COMPLETE.
----- RESULTS-----
Shop[0] - Amnt avail.: 29; Demand[4]; Lack: 0; Products ordered: 0
Shop[1] - Amnt avail.: 14; Demand[5]; Lack: 0; Products ordered: 0
Shop[2] - Amnt avail.: 17; Demand[2]; Lack: 0; Products ordered: 11
--------------------
Store[0] - Amnt avail.: 26; Truck Count[9]; Lack: 0; Products ordered: 0
Store[1] - Amnt avail.: 99999; Truck Count[10]; Lack: 0; Products ordered: 0
```

Fig. 17. Step 3. Simulate, "Logistic Model".

10 Conclusions

1. 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.
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 examples of decision making models illustrate of two important f well-known theretical topics: the Sequential Statistical Decisions and the Bayesian Approach.
5. The results of optimization show the possibilities of some nontraditional ways of graduate studies in both the distance and the classroom cases.

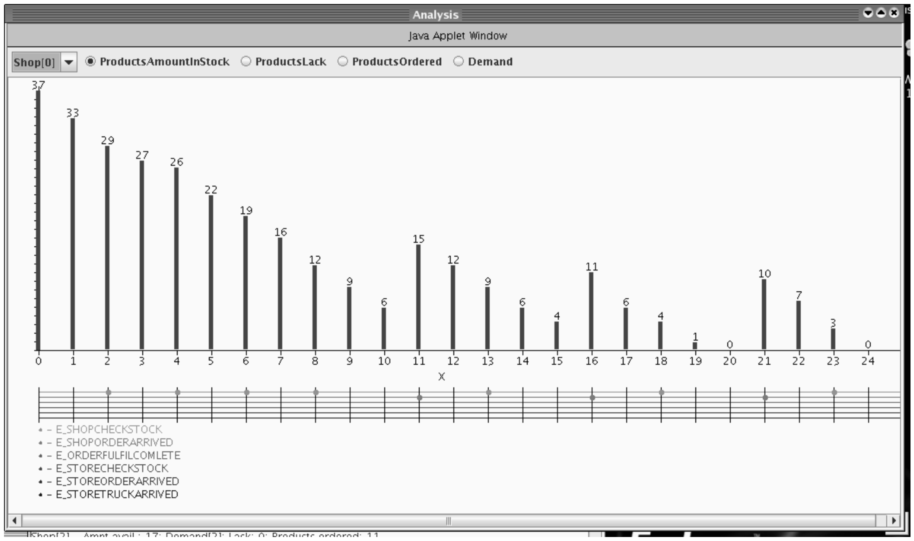


Fig. 18. Step 4. Analyze, "Logistic Model".

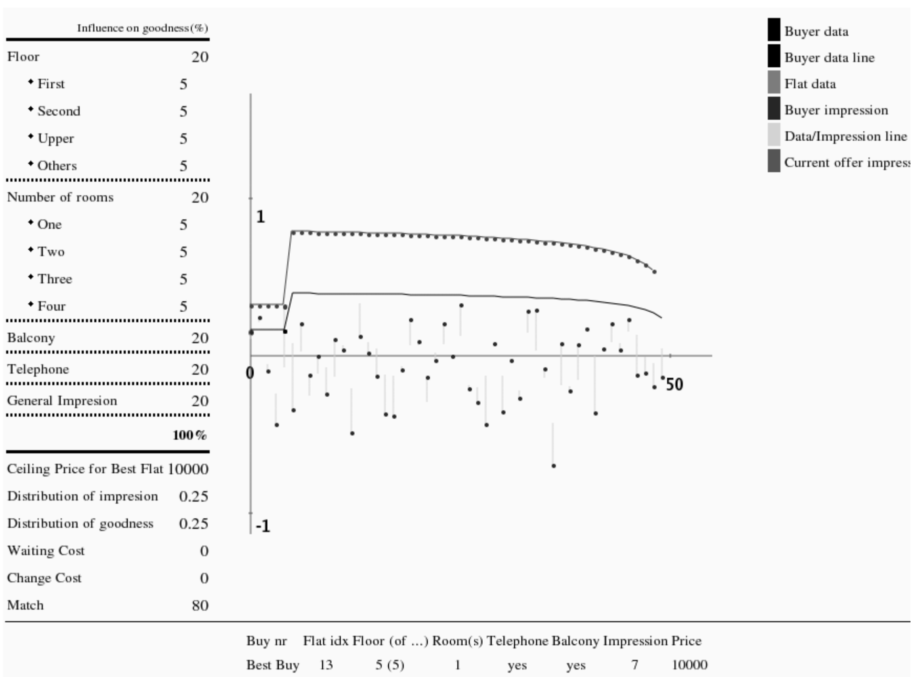
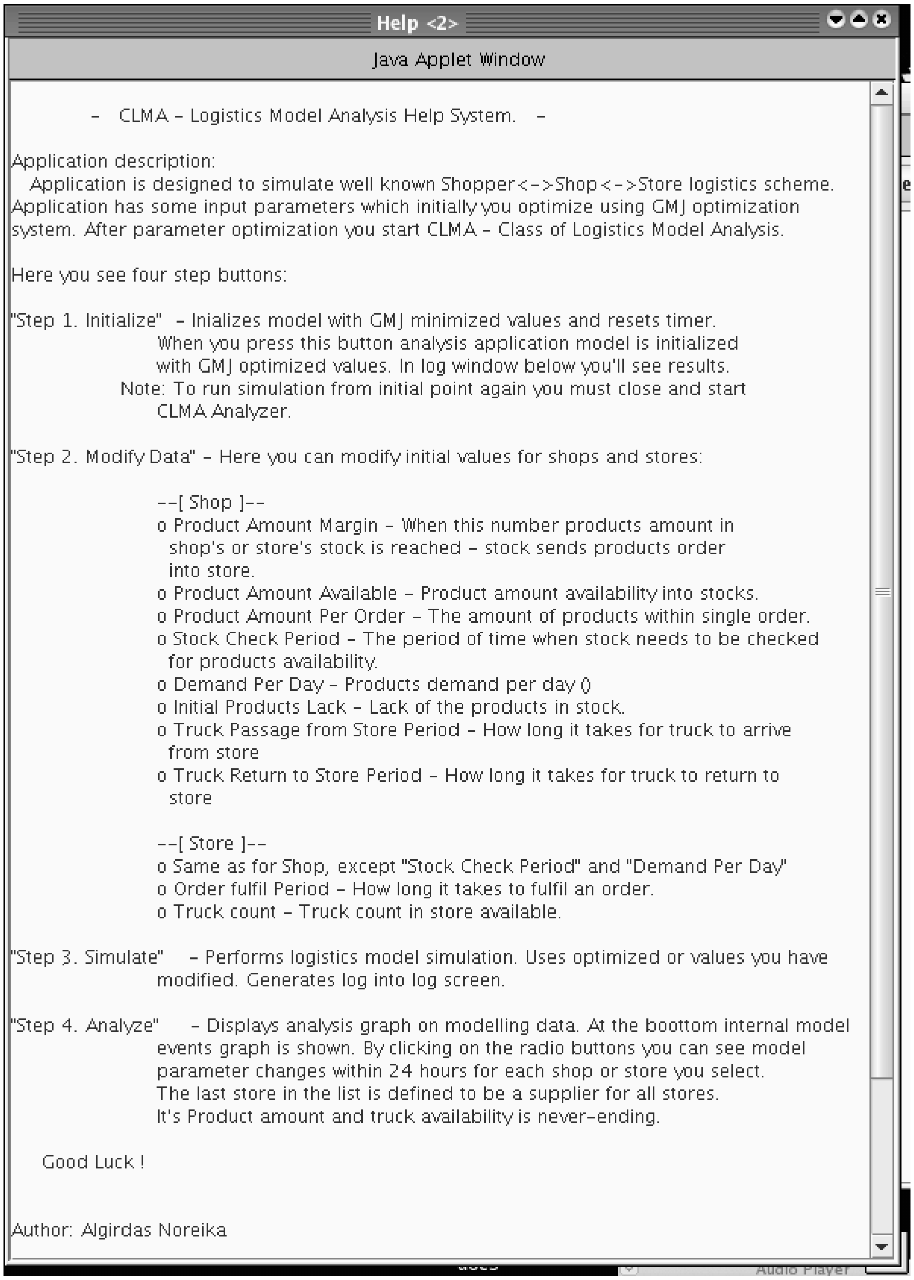


Fig. 19. "Buy-a-Flat" by C-sharp.



Help <2>

Java Applet Window

– CLMA – Logistics Model Analysis Help System. –

Application description:
Application is designed to simulate well known Shopper<->Shop<->Store logistics scheme. Application has some input parameters which initially you optimize using GMJ optimization system. After parameter optimization you start CLMA – Class of Logistics Model Analysis.

Here you see four step buttons:

"Step 1. Initialize" – Inializes model with GMJ minimized values and resets timer.
When you press this button analysis application model is initialized with GMJ optimized values. In log window below you'll see results.
Note: To run simulation from initial point again you must close and start CLMA Analyzer.

"Step 2. Modify Data" – Here you can modify initial values for shops and stores:

--[Shop]--
o Product Amount Margin – When this number products amount in shop's or store's stock is reached – stock sends products order into store.
o Product Amount Available – Product amount availability into stocks.
o Product Amount Per Order – The amount of products within single order.
o Stock Check Period – The period of time when stock needs to be checked for products availability.
o Demand Per Day – Products demand per day ()
o Initial Products Lack – Lack of the products in stock.
o Truck Passage from Store Period – How long it takes for truck to arrive from store
o Truck Return to Store Period – How long it takes for truck to return to store

--[Store]--
o Same as for Shop, except "Stock Check Period" and "Demand Per Day"
o Order fulfil Period – How long it takes to fulfil an order.
o Truck count – Truck count in store available.

"Step 3. Simulate" – Performs logistics model simulation. Uses optimized or values you have modified. Generates log into log screen.

"Step 4. Analyze" – Displays analysis graph on modelling data. At the boottom internal model events graph is shown. By clicking on the radio buttons you can see model parameter changes within 24 hours for each shop or store you select.
The last store in the list is defined to be a supplier for all stores.
It's Product amount and truck availability is never-ending.

Good Luck !

Author: Algirdas Noreika

Fig. 20. Help, "Logistic Model".

References

1. Mockus, J.: A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach. Kluwer Academic Publishers (2000) ISBN 0-7923-6359-0.
2. Wald, A.: Statistical Decision Functions. J. Wiley, New York (1950)
3. DeGroot, M.: Optimal Statistical Decisions. McGraw-Hill, New York (1970)
4. Wald, A.: Sequential Analysis. J. Wiley, New York (1947)
5. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton, New Jersey (1957)
6. Mockus, J.: Bayesian approach to global optimization. Kluwer Academic Publishers, Dordrecht-London-Boston (1989)
7. Lindley, D.: Bayesian Statistics: A Review. SIAM, Philadelphia (1971)
8. Savage, L.: Foundations of Statistics. Wiley, New York (1954)
9. T.L.Fine: Theories of Probability. Academic Press, New York (1973)
10. Zilinskas, A.: Global optimization: axiomatic of statistical models, algorithms and their applications. Mokslas, Vilnius, Lithuania (1986) in Russian.
11. Bayes, T.: An essay towards solving a problem in the doctrine of chances. Phil. Transactions of Royal Society **53** (1783) 370–418
12. Mockus, J.: Bayesian approach to global optimization and application to constrained and multi-objective problems. In: Abstracts, the Fifth International Conference on Stochastic Programming in Ann Arbor, August 13-18, 1989. (1989)
13. Tijms, H.: Stochastic Models. An Algorithmic Approach. Wiley, New York (1994)