Walras Competition Model as an Example of Global Optimization

Jonas Mockus,
Kaunas Technological University,
Vilnius Technical University,
Institute of Mathematics and Informatics,
Akademijos 4, Vilnius LT-2600, Lithuania.
e-mail: mockus@soften.ktu.lt

June 18, 2008

Abstract

Walras theory is well known and widely used in models of market economy. Various iterative methods are developed to search for the equilibrium conditions.

In this paper a new approach is proposed and implemented where the search for Walras equilibrium is defined as a stochastic global optimization problem. This way random nature of customer arrivals is represented and the convergence to equilibrium is provided if equilibrium exists.

This paper describes a part of a Web-based integrated system for scientific cooperation and distance graduate studies of theories of optimization, games and markets which aim is to provide researchers and graduate students with hands-on experience on effective use of software. The objectives are to provide a tool for scientific collaboration and to stimulate creative abilities of graduate students to work as independent researchers. The web-site $http://soften.ktu.lt/\tilde{}mockus$ includes a family of economic and finnacial models regarding them all as examples of the the general optimization theory.

Key words: Walras Model, Nash Equilibrium, Stochastic Global Optimization, Internet environment.

AMS subject classification: 90C15, 90A20, 90D40, 93E05.

1 Introduction

Future enterprises will be characterized by a focus on total quality, globalization, an object-oriented approach, and a business process-oriented approach. The globalization will lead to the "Virtual Enterprise". The virtual enterprise can obtain a competitive position by defining and re-engineering its business processes. However, such re-engineering requires an enterprise model.

A well known example is the Factory of the Future (FOF) project[Rolstadas, 1995]. It is based on a generalization of the Walras model, and defines a number of design choices and performance indicators.

There are well known and widely used iterative algorithms to define equilibrium in Walras models [Scarf and Hansen, 1973, Herings, 1994] under some conditions. In this paper we consider the search for the Nash equilibrium[Nash, 1950] of the Walras model of the theory of games and markets [Rosenmuller, 1981] under different conditions. The main difference is stochasic demad and stochastic service time. This is important considering "markets" represented by a collection of small number of independent servers.

Each server tries to maximize its profit by setting optimal prices for services and resources. The service rate is the average number of customers that would be served in nonstop operations. Thus, we can consider this rate as the server capacity.

First, an agreement about prices of services and resources - the "Contract Vector" (CV) - is made. Then profits of individual servers are maximized by changing individual prices under the assumption that all their partners respect the prices set by CV. This way one transforms the Contract-Vector into the "Fraud-Vector" (FV)[1].

The optimization problem is to search for such CV that minimizes the deviation of the Fraud-Vector from the Contract-Vector. This makes the fraud less relevant. The fraud is irrelevant and the Nash equilibrium is achieved, if the deviation is zero. Then servers cannot increase their profits by changing service and resource prices defined by the contract.

For simplicity the quality of service is defined as the average time lost by customers while waiting for services[2]. A customer prefers the server with lesser total service cost. The total cost is defined as the service price plus estimated waiting losses (expressed in the same money as the service price). A customer goes away, if the total cost exceeds a certain critical level. A flow of customers is stochastic. Service times are stochastic, too. There is no known analytical solution for this model. The results are obtained by Monte-Carlo simulation.

It is supposed that the server capacity depends on several resources. Each server owns a minimal amount of all resources for its own needs. It is assumed that each server allocates a fixed amount of single resource for sale. Thus, the

[1] Often FV is called as the "Best Response", however we think that the name "Fraud Vector" describes the situation better because that is not necessarily the best response.

[2] Expected waiting time is just one of many parameters defining the quality of services. This parameter is easy to estimate and important in some cases, for example in the network servers and crowded supermarkets.

number of different resources in the "market" is equal to the number of servers. The distribution of the market resources are controlled by their prices[3].

In the model, the servers share each other resources. Therefore, here we are looking for the equilibrium of both the prices for services and the prices for shared resources.

This market model illustrates the possibilities and limitations of the optimization theory and numerical techniques in models of competitive environments. Initially the model was developed as a test function for the Bayesian algorithms of stochastic global optimization [Mockus, 2000]. However, this simple model may help to design more realistic ones. The simple models help to understand processes of competition better. In addition it shows the relation between the optimization and equilibrium.This is important for studies of the Operations Research and the Theory of Games and Markets.

The model is implemented as an Java applet on the web-site: $http://soften.ktu.lt/\tilde{}mockus$ and some mirrors: $http://eta.ktl.mii.lt/\tilde{}mockus$, $http://optimum2.mii.lt/\tilde{}jonas2$, $http://mockus.us/optimum$. That means that anybody can test the presented examples and ssome new ones, too. Thus the scientific cooperation is natural and easy. In this sense the models implemented as Java applets are similar the analytical ones. The difficulties of development are similar, too. The first version was implemented by [Perlibakas, 1999], the Profit Analysis using the Wiener filter was improved by [Sviderskaite, 2003], the updated version was a part of magister thesis by [Skruodys, 2004]. The final version is the result of long process of coding and tesing by [Treigys, 2003, Treigys, 2004].

The software implementation of the Walras model is a specific and very important problem. The main theoretical and practical difficulty is to balance the accuracy and efficiency. Some aspects of this problem are discussed in this paper. The complete description will by published as a separate paper by the software author Povilas Treigys [Treigys, 2004].

1.1 Profit Functions

In the Walras model, the capacity w_i of servers $i = 1, ..., m$ depends on the resource vector $x_i + g_i = (x_{ij} + g_{ij}),\ i, j = 1, ..., m)$ defining the consumption of different resources. Each server i may sale a single resource up to a limit b_i charging p_i for a unit of this resource to partner-servers. Therefore:

$$b_i = \sum_{j=1,...,m} x_{ij},\ i = 1, ..., m \tag{1}$$

The server i controls the vector $x_i = x_{ij},\ j = 1, ..., m$. The component x_{ij} denotes the amount of resource b_j used by the server i. The server i also controls the price y_i for services. and the price p_i that is charged for resource p_i.

Assume that the server capacity w_i is an increasing function of the resources.

$$w_i = \phi_i(x_i + g_i). \tag{2}$$

[3] This model is similar, but not identical, to the traditional Walras model [Walras, 1874].

A simple example of this function

$$w_i = k_i \prod_{j=1}^{m} (1 - \exp(-k_{ij}(x_{ij} + g_{ij}))). \quad (3)$$

The resource component $x_{ij} + g_{ij}$ denotes the total amount of resource j used by server i including both the "market" resource x_{ij} and the "local" resource g_{ij}. The coefficient k_{ij} shows how useful is the resource b_j for the server i. The coefficient k_i defines the capacity limit when $x_{ij} \to \infty$.

The profit of the i-th server:

$$u_i = u_i(x_j, y_j, p_j,\ j = 1, ..., m) = \\ a_i y_i + p_i \sum_{j \neq i} x_{ij} - \sum_{j \neq i} p_j x_{ij}. \quad (4)$$

Here i is the server index. a_i is the rate of customers. y_i is the service price. $x_j = (x_{jk},\ k = 1, ..., m)$ is the resource vector determining the capacity w_j of server j.

The rate of customers a_i of each server i is defined by the total service cost

$$c_i = y_i + \gamma_i, \quad (5)$$

where γ_i is waiting cost[4] at the server i. A customer goes to the server i, if

$$c_i < c_j,\ j = 1, ..., m,\ j \neq i,\ c_i \leq c_0. \quad (6)$$

A customer goes away, if

$$\min_i c_i > c_0, \quad (7)$$

where c_0 is the critical cost. The rate a of incoming consumers flow is fixed:

$$a = \sum_{i=0}^{m} a_i, \quad (8)$$

where a_0 is the rate of lost customers.

From the balance condition follows $x_{ii} = b_i - \sum_{j \neq i} x_{ji}$. Note, the profit u_i of each individual server i depends on the parameters x_j, y_j, p_j of all m servers $j = 1, ..., m$.

Here the first component $a_i y_i$ defines the income of a server i collected from customers of for its services. The second component $p_i \sum_{j \neq i} x_{ij}$ defines the sum payed by other servers using the resource b_i. The third component $\sum_{j \neq i} p_j x_{ij}$ shows the expenses for resources obtained from other servers.

In two-server cases

$$u_1 = u_1(x_{12}, y_1, p_1, x_{21}, y_2, p_2) = \\ a_1 y_1 + p_1 x_{21} - p_2 x_{12}. \quad (9)$$

[4]It is. assumed for simplicity that the waiting cost is equal to an average waiting time.

and

$$u_2 = u_2(x_{21}, y_2, p_2, x_{12}, y_1, p_1) = \\ a_2 y_2 + p_2 x_{12} - p_1 x_{21}. \quad (10)$$

Here x_{11} and x_{22} are not included explicitly because they are defined by the balance conditions

$$x_{11} = b_1 - x_{21},\ x_{22} = b_2 - x_{12}. \quad (11)$$

By fixing the lower and upper limits
$a_{p_i}, a_{x_{ij}}, a_{y_i}, b_{p_i}, b_{x_{ij}}, b_{y_i}\ i, j = 1, ...m$ we obtain the inequalities

$$a_{p_i} \leq p_i \leq b_{p_i},\ a_{x_{ij}} \leq x_{ij} \leq b_{x_{ij}}, \\ a_{y_i} \leq y_i \leq b_{y_i},\ i, j = 1, ..., m \quad (12)$$

1.2 Walras Model

Let us consider m servers providing the same service

$$u_i = u_i(x_1, y_1, ..., x_m, y_m) = a_i y_i - x_i,\ i = 1, ..., m, \quad (13)$$

where u_i is the profit, y_i is the service price, a_i is the rate of customers, x_i is the running cost, and i is the server index. Assume that a server capacity w_i is an increasing function of the running cost x_i:

$$w_i = \phi_i(x_i). \quad (14)$$

Therefore we shall describe the Walras model in the terms similar to those of the Nash model (see expression (13)). A simple example of this function

$$w_i = k_i(1 - \exp(-k_{i0} x_i)). \quad (15)$$

Here k_i defines the maximal server rate and k_{i0} shows the efficiency of resource x_i. Then the total service cost

$$c_i = y_i + \gamma_i, \quad (16)$$

where γ_i is the average waiting cost. To simplify the expressions assume that the waiting cost is equal to an average waiting time. Suppose that arriving customers estimate the average waiting time as the relation of the number of waiting customers n_i to the server capacity w_i

$$\gamma_i = n_i / w_i \quad (17)$$

A customer goes to the server i, if

$$c_i \leq c_j,\ j = 1, ..., m,\ j \neq i,\ c_i \leq c_0. \quad (18)$$

A customer goes away, if

$$\min_i c_i > c_0, \tag{19}$$

where c_0 is the critical cost. The rate a of incoming consumers is fixed

$$a = \sum_{i=0}^{m} a_i, \tag{20}$$

where a_0 is the rate of lost customers. Conditions (18) and (19) separate the flow of incoming customers into $m+1$ flows. This makes the problem very difficult for analytical solution. The separated flow is not simple one, even in the case when the incoming flow is Poisson [Gnedenko and Kovalenko, 1987]. Thus, we need the Monte Carlo simulation, to define average rates of customers $a_i,\ i = 0, 1, ..., m$, by conditions (18) (19), and average profits $u_i,\ i = 1, ..., m$ by expression (13).

1.3 Nash Equilibrium

First we fix some initial values of CV[5] $z^0 = (x_i^0, y_i^0,\ i = 1, ..., m)$. Then the values of the corresponding FV $z^1 = (x_i^1, y_i^1,\ i = 1, ..., m)$, are obtained maximizing profits of each server i separately. The maximization is performed under the assumption that other partners $j \neq i$ honor the contract $(x_j^0, y_j^0,\ j = 1, ..., m,\ j \neq i)$

$$(x_i^1, y_i^1) = arg \max_{x_i, y_i} u_i(x_i, y_i, x_j^0, y_j^0,\ j = 1, ..., m,\ j \neq i),\ i = 1, ..., m. \tag{21}$$

Formally, condition (21) transforms the vector
$z^n = (x_i^n, y_i^n,\ i = 1, ..., m) \in B \subset R^{2m},\ n = 0, 1, 2, ...$ into the vector z^{n+1}. To make expressions shorter denote this transformation by T

$$z^{n+1} = T(z^n),\ n = 0, 1, 2, ... \tag{22}$$

The equilibrium is at the fixed point z^n, where

$$z^n = T(z^n). \tag{23}$$

The fixed point z^n exists, if both the feasible set B and all the profit functions are convex [Michael, 1976]. Traditional way to reach the equilibrium is by iterations (22). That is possible if the transformation T is contracting [Neuman and Morgenstern, 1953]. Othervise minimization of the deviation from equilibrium is needed. This is not a simple task considering stochastic arrivals and service times. The problem is very difficult if the deviation is not only stochastic but multimodal, too[6].

[5]In this paper the a sequence of initial CV is defined by optimization methods saerching for equilibrium.

[6]The deviation is a sum of non-convex functions. It is well known that in this case the sum is not unimodal if the feasible set is sufficiently large.

1.4 Search for Equilibrium

The equilibrium is achieved, if the minimum

$$\min_{z \in B} \parallel z - T(z) \parallel . \tag{24}$$

is zero. If the minimum (24) is positive then the equilibrium does not exist. That is a theoretical conclusion. In numerical calculations involving statistical modeling, some deviations are inevitable. Therefore, we assume that the equilibrium exists, if the minimum is not greater then modeling errors.

One can minimize deviation (24) by the usual stochastic approximation techniques [Ermoljev and Wets, 1988], if the deviation (24) is an unimodal function of z. If not, then the techniques of global stochastic optimization [Mockus, 1989] should be used. The global stochastic optimization may outperform the local one in the unimodal case, too. That happens, if the noise level is great because the Bayesian global stochastic optimization methods are less sensitive to large noise levels.

The norm $\parallel z - T(z) \parallel$ is not convenient for numerical optimization. Square of norm is better in this respect

$$\min_{z \in B} \parallel z - T(z) \parallel^2 . \tag{25}$$

The same result one obtains by the following condition.

$$\min_{z \in B} \sum_i (u_i(T(z)) - u_i(z)). \tag{26}$$

Here the difference $u_i(T(z)) - u_i(z)$ shows the profit obtained by i-th server by breaking the contract z. We call the sum (26) as a fraud profit. Minimization of the fraud profit seems natural in economical terms and is convenient for computations. In these terms equilibrium means such a contract where fraud is not profitable.

1.5 Existence of Nash Equilibrium

The first existence theorem is due to Nash [Nash, 1951] and dates back to 1951. Many generalizations appeared since then. Finding less and less restrictive sufficient conditions have been an active field of research [Forgo et al., 1999]. The proofs of these conditions are based on the various fixed point theorems [Brouwer, 1912, Kakutani, 1941, Browder, 1968]. Considering the examples of this book, we prefer simple existence conditions to the general ones. Testing the existence conditions, we express them in terms of the profit functions u instead of the operators T.

For example, the stable equilibrium exists, if the profit $u(z)$ is strictly convex function of all the components of its parameters $z \subset Z$, and Z is a convex set [Michael, 1976]. In such cases, small changes of z components will not change the maximum points considerably.

The situation would be different in the non-strictly-convex cases. Here even very small change of some parameters z, may change the maximum point z^* of $u(z)$ considerably. For example, in linear cases this point may jump from minimal to maximal limits. In multi-modal cases the point z^* can jump from one local minimum to another one. These sharp changes violate the continuity of the transformation T. The continuity of T is needed in the Brouwer's fixed point theorem [Brouwer, 1912]. In other theorems, such as Kakutani's [Kakutani, 1941] or Browder's [Browder, 1968], the fixed-point conditions are less restrictive. However, testing these conditions is not a trivial task.

2 Search for Nash Equilibrium

First fix a contract-vector $(x_i^0, y_i^0, p_i^0,\ i = 1, ..., m)$. Then the fraud-vector $(x_i^1, y_i^1, p_i^1,\ i = 1, ..., m)$ is obtained by maximizing the profits of each server i assuming that all other partners $j \neq i$ will respect the contract $(x_i^0, y_i^0, p_i^0,\ i = 1, ..., m)$

$$(x_i^1, y_i^1, p_i^1,) = \\ arg \max_{x_i, y_i, p_i} u_i(x_i, y_i, p_i,\ x_j^0, y_j^0, p_j^0,\ j = 1, ..., m,\ j \neq i). \tag{27}$$

Here the profit function $u_i(x_i, y_i, p_i,\ x_j^0, y_j^0, p_j^0,\ j = 1, ..., m,\ j \neq i)$ is defined by expression (4). There are rectangular constraints (13). A server i optimizes only components $x_{ij},\ i \neq j$ because x_{ii} is defined by the balance condition $x_{ii} = b_i - \sum_{j \neq i} x_{ji}$.

In the two-server case

$$(x_{12}^1, y_1^1, p_1^1,) = \\ arg \max_{x_{12}, y_1, p_1} u_1(x_{12}, y_1, p_1, x_{21}^0, y_2^0, p_2^0), \tag{28}$$

and

$$(x_{21}^1, y_2^1, p_2^1,) = \\ arg \max_{x_{21}, y_2, p_2} u_2(x_{21}, y_2, p_2, x_{12}^0, y_1^0, p_1^0). \tag{29}$$

Condition (27) transforms vectors $z^n,\ n = 0, 1, 2, ...$ into vectors z^{n+1}, where $z^n = (x^n, y^n, p^n)$, $x^n = (x_1^n, ..., x_m^n)$, $y^n = (y_1^n, ..., y_m^n)$, and $p^n = (p_1^n, ..., p_m^n)$. Denote this transformation by T

$$z^{n+1} = T(z^n),\ n = 0, 1, 2, ... \tag{30}$$

Here the vector $z = (x_i, y_i, p_i, v_i\ i = 1, ..., m) \in B \subset R^{m^2+2m}$. We reach the equilibrium[7] at the fixed point z^n, where

$$z^n = T(z^n). \tag{31}$$

[7] If the equilibrium exists.

We may obtain the Nash equilibrium directly by simple iterations (30), if the transformation T is contracting [Neuman and Morgenstern, 1953]. There are more sophisticated and efficient iterative procedures [Herings, 1994].

If the equilibrium exists but the transformation T is not contracting then one minimizes the square deviation

$$\min_{z \in B} \parallel z - T(z) \parallel^2 . \tag{32}$$

The equilibrium is achieved, if the minimum (32) is zero.

The alternative way to achieve equilibrium is by minimizing the fraud profit

$$\min_{z \in B} \sum_i (u_i(T(z)) - u_i(z)). \tag{33}$$

Here the difference $u_i(T(z)) - u_i(z)$ shows the profit obtained by i-th server by breaking the contract z. The advantage of (33) is clear economic meaning.

The constraints (13) limits the Walras model. Therefore, setting these constraints one should resolve the following contradiction. Wider limits means more computing for optimization but less restriction for the Walras model, and vice versa. A way to get around this contradiction is to start with narrow bounds (13). One widens them if some of these bounds obviously restrict the profit maximum.

If the equilibrium test (32) fails additional testing of the sufficient existence conditions is needed. It is well known, that the equilibrium exists, if the profit u is strictly convex function of parameters (y, x, p) [Michael, 1976]. Therefore, the "non-strict-convexity" of this function could be a reason of the failure to obtain the equilibrium.

Expressions (4) and (10) show that profits u_i are linear functions of resource prices p_i. Linear functions are not strictly convex. For example, if this function is nearly constant then a small change of some parameters may switch the optimal resource price p_i from zero to upper limit or vice versa.

Thus the Nash equilibrium may not exist in this case. Besides this model is not practical because the obligation to buy a fixed amount of resources regardless of the price in obviously not the realistic one.

To avoid this difficulty two versions of the "Look-Ahead" (LA) equilibrium models are considered in the following sections. In the LA models the control parameters p, y, x are divided into two groups: contract and free. For example in the model (27)- (33) all the parameters were included into the contract. The parameters that are not included into the contract are called as free. The natural example of the free parameter is resource consumptions x.

The important problem of models with free parameters is than one must anticipate the free parameters of all competitors.

We consider estimations of free parameters based on two different assumptions: Nash (NLA) and Greedy (GLA). In the Nash (NLA) case we assume that all the servers select the free parameters at given contract parameters by conditions Nash equilibrium. In the Greedy (GLA) case servers select such free parameters that maximize their profits at fixed contract parameters. The NLA models may provide genuine Nash equilibrium if servers estimate competitors profit functions well enough. However GLA models seems to be more realistic.

Look Ahead LA versions differs by the number of parameters to be set free. Denote by LA(p,y) the case when only the vector of resource demand x is free. Denote by LA(p) the case when both vectors x and y are free.

3 "Nash-Look-Ahead" (NLA) Equilibrium

Definition of NLA equilibrium is similar to that of Nash equilibrium. The difference is that competitors responses. are anticipated while searching for the equilibrium prices. We consider two versions of NLA.

In the first version, NLA(p) for short, both the resource consumption x_j and the service charge y_j are predicted for all servers at given resource price p. It is assumed that competing servers $j \neq i$ define parameters (x_j, y_j) by the usual Nash equilibrium conditions such as (24) at fixed resource prices $p = (p_i,\ i = 1, ..., m)$ [8]. For example, they increase consumption if the prices fall.

In the second version that is denoted as NLA(p,y), only the resource j consumptions x_{ij} are predicted for all servers i. Here servers i define resource j demand x_{ji} by the usual Nash equilibrium conditions at fixed both the service charges $p = (p_j,\ j = 1, ..., m)$ and service prices $y = (y_j,\ i = 1, ..., m)$. For example, they not only increase resource consumption x_j if the prices p_j fall but adapt the service charges y_j to changed resource prices as well.

Using both versions NLA while determining resource prices p one transforms linear profit functions of p_j[9] into nonlinear ones. This way one may satisfy the necessary equilibrium conditions. That is true in both the cases: NLA(p) and NLA(p,y). It seems that NLA(py) better describes the actual economical behavior of participants and demands less calculations.

3.1 First Version: NLA(p)

3.1.1 Search for NLA(p) Equilibrium

Defining Nash Profit Using NLA(p) the Nash profit is defined as the profit function $U(p)$ satisfying Nash conditions at fixed resource prices p. Then the profit of the i-th server:

$$U_i(p) = u_i(x_j^* y_j^*, p_j,\ j = 1, ..., m) = \\ a_i y_i^* + p_i \sum_{j \neq i} x_{ij}^* - \sum_{j \neq i} p_j x_{ij}^*. \tag{34}$$

Here i is the server index. a_i is the rate of customers. $y_i^* = y_i(p)$ is the equilibrium price of services. $x_j^* = (x_{jk}^*,\ k = 1, ..., m)$, $x_{jk}^* = x_{jk}(p)$ is the

[8]That explains the acronym NLA(p).

[9]As defined by expressions (4) and (10).

equilibrium resource vector that defines the capacity w_j of server j. All depend on resource prices p.

From the balance condition follows that $x^*_{ii} = b_i - \sum_{j \neq i} x^*_{ji}$. The vector (x^*, y^*) is defined by Nash conditions (33) for fixed resource prices p by this condition

$$\min_{z \in B} \sum_i (u_i(T(z)) - u_i(z)), \tag{35}$$

where $z = (x, y)$. Denote Nash equilibrium at fixed p as $z^* = z(p)$, then

$$x^* = x(p), y^* = y(p). \tag{36}$$

In two-server cases

$$U_1(p_1, p_2) = u_1(x^*_{12}, y^*_1, p_1, x^*_{21}, y^*_2, p_2) = \\ a_1 y^*_1 + p_1 x^*_{21} - p_2 x^*_{12}. \tag{37}$$

and

$$U_2(p_1, p_2) = u_2(x^*_{21}, y^*_2, p_2, x^*_{12}, y^*_1, p_1) = \\ a_2 y^*_2 + p_2 x^*_{12} - p_1 x^*_{21}. \tag{38}$$

Here x^*_{11} and x^*_{22} are not included explicitly because they are defined by the balance conditions

$$x^*_{11} = b_1 - x + 21^*, \ x^*_{22} = b_2 - x^*_{12}. \tag{39}$$

One solves (35) many times for each fixed resource price p before the equilibrium value $p = p^*$ is reached.

Defining NLA(p) Profit First a contract-vector $(p^0_i, \ i = 1, ..., m)$ is fixed. Then the fraud-vector $(p^1_i, \ i = 1, ..., m)$ is obtained by maximizing the profits of each server i assuming that other partners $j \neq i$ keep the contract resource prices.

$$p^1_i = arg \ \max_{p_i} U_i(p_i, \ p^0_j, \ j = 1, ..., m, \ j \neq i), \ i = 1, ..., m. \tag{40}$$

Here $p^0_j, \ j \neq i$ are contract resource prices of competing servers $j \neq i$.

In the two-server case

$$p^1_1 = arg \max_{p_1} U_1(p_1, p^0_2), \tag{41}$$

and

$$p^1_2 = arg \max_{p_2} U_2(p_2, p^0_1). \tag{42}$$

Condition (40) transforms vectors $p^n,\ n = 0, 1, 2, ...$ into vectors p^{n+1}, where $p^n = (p_i^n,\ i = 1, ..., m)$, Denote this transformation by T

$$p^{n+1} = T(p^n),\ n = 0, 1, 2, ... \tag{43}$$

We reach the equilibrium[10] at the fixed point p^n, where

$$p^n = T(p^n). \tag{44}$$

If the equilibrium exists but the transformation T is not contracting then we minimize the square deviation

$$\min_{p \in B} \parallel p - T(p) \parallel^2 . \tag{45}$$

The equilibrium is achieved, if the minimum (45) is zero.

The alternative way to achieve equilibrium is by minimizing the fraud profit

$$\min_{p \in B} \sum_i (U_i(T(p)) - U_i(p)). \tag{46}$$

Here the difference $U_i(T(p)) - u_i(p)$ shows the profit obtained by i-th server by deviating from the NLA equilibrium p^*.
The final equilibrium values of (x, y) are defined substituting p^* into expression (36)

3.2 Second Version: NLA(p,y)

3.2.1 Search for NLA(p,y) Equilibrium

Defining Nash Profit Using NLA(p,y) the profit of the i-th server:

$$U_i(p, y) = u_i(x_j^* y_j, p_j,\ j = 1, ..., m) = \\ a_i y_i^* + p_i \sum_{j \neq i} x_{ij}^* - \sum_{j \neq i} p_j x_{ij}^*. \tag{47}$$

Here i is the server index. a_i is the rate of customers. $x_j^* = (x_{jk}^*,\ k = 1, ..., m)$, $x_{jk}^* = x_{jk}(p, y)$ is the equilibrium resource vector that defines the capacity w_j of server j. All depend on prices p and y. From the balance condition follows $x_{ii}^* = b_i - \sum_{j \neq i} x_{ji}^*$.

The vector x^* is defined at fixed prices (p, y) by this condition

$$\min_{x \in B} \sum_i (u_i(T(x)) - u_i(x)). \tag{48}$$

Denote Nash equilibrium at fixed price vector (p, y) as

$$x^* = x(p, y). \tag{49}$$

[10]If the equilibrium exists.

In two-server cases

$$U_1(p_1, y_1, p_2, y_2) = u_1(x^*_{12}, y_1, p_1, x^*_{21}, y_2, p_2) = \\ a_1 y_1 + p_1 x^*_{21} - p_2 x^*_{12}. \quad (50)$$

and

$$U_2(p_1, y_1, p_2, y_2) = u_2(x^*_{21}, y_2, p_2, x^*_{12}, y_1, p_1) = \\ a_2 y_2 + p_2 x^*_{12} - p_1 x^*_{21}. \quad (51)$$

Here x^*_{11} and x^*_{22} are not included explicitly because they are defined by the balance conditions

$$x^*_{11} = b_1 - x + 21^*,\ x^*_{22} = b_2 - x^*_{12}. \quad (52)$$

One solves (35) many times for each fixed price vector (p, y) before the equilibrium values $(p = p^*, y^*)$ are reached.

Defining NLA(p,y) Profit First a contract-vector $(p^0_i, y^0_i\ i = 1, ..., m)$ is fixed. Then the fraud-vector $(p^1_i, y^1_i\ i = 1, ..., m)$ is obtained by maximizing the profits of each server i assuming that other partners $j \neq i$ keep the contract resource prices.

$$(p^1_i, y^1_i) = arg \max_{p_i, y_i} U_i(p_i, y_i\ p^0_j, y^0_j\ j = 1, ..., m,\ j \neq i),\ i = 1, ..., m. \quad (53)$$

Here $p^0_j, y^0_j\ j \neq i$ are contact prices of competing servers $j \neq i$.

Condition (40) transforms vectors $p^n, y^n\ n = 0, 1, 2, ...$ into vectors p^{n+1}, y^{n+1}, where $p^n = (p^n_i,\ i = 1, ..., m)$, and $y^n = (y^n_i,\ i = 1, ...m)$. Denote this transformation by T

$$z^{n+1} = T(z^n),\ n = 0, 1, 2, ... \quad (54)$$

where $z = (p, y)$. We reach the equilibrium[11] at the fixed point p^n, where

$$z^n = T(z^n). \quad (55)$$

If the equilibrium exists but the transformation T is not contracting then we minimize the square deviation

$$\min_{z \in B} \parallel z - T(z) \parallel^2 . \quad (56)$$

The equilibrium is achieved, if the minimum (56) is zero.

The alternative way to achieve equilibrium is by minimizing the fraud profit

$$\min_{z \in B} \sum_i (U_i(T(z)) - U_i(z)). \quad (57)$$

Here the difference $U_i(T(z)) - u_i(z)$ shows the profit obtained by i-th server by deviating from the NLA equilibrium z^*.
The final equilibrium values of x^*) are defined substituting (p^*, y^*) into expression (49)

[11] If the equilibrium exists.

3.3 Description of different parts NLA(p,y) Algorithm

3.3.1 Reserve Resources

Consider two-server case for simplicity. The capacity function (3) including reserves x^r_{ij}:

$$w_i = k_i \prod_{j=1}^{m} (1 - \exp(-k_{ij}(x_{ij}) + x^r_{ij})),\ i = 1.2 \tag{58}$$

where $\sum_i x^r_{ij} < b_j,\ i, j = 1, 2$

3.3.2 Equilibrium Resources

Assume, that at fixed prices $p_i,\ y_i, i = 1, 2$ servers obtain additional resources x_{ij} followng the Nash equilibrium conditions using this algorithm.

1. a pair $x^0_{12},\ x^0_{21}$ is fixed
2. individual fraud profits u^1 and v^1 are calculated

$$u^1 = \max_{x_{12}} u(x_{12}, x^0_{21}) \tag{59}$$

$$v^1 = \max_{x_{21}} u(x^0_{12}, x_{21}) \tag{60}$$

3. general fraud profit is defined

$$f^1 = u^1 - u^0 + v^1 - v^0 \tag{61}$$

4. the procedure is repeated for K^2 pairs x_{12}, x_{21} defining a sequence of general fraud profits $f^k,\ k = 1, ..., K^2$ where K depends on fixed calculation error.
5. the pair $x_{1,2}, x_{21}$ with minimal fraud profit f^k is considered as equilibrium

3.3.3 Equilibrium Prices

1. a quadruple of prices $p^0_1, p^0_2, y^0_1, y^0_2$ is fixed
2. individual fraud profits U^1, V^1 and correspnding fraud prices $(p^1_1, y^1_1, p^1_2, y^1_2)$ are calculated

$$U^1 = \max_{p_1, y_1} U(p_1, y_1, p^0_2, y^0_2) \tag{62}$$

$$(p^1_1, y^1_1) = arg \max_{p_1, y_1} U(p_1, y_1, p^0_2, y^0_2), \tag{63}$$

and

$$V^1 = \max_{p_2, y_2} U_2(p_2, y_2 p^0_2, y^0_2) \tag{64}$$

$$(p^1_2, y^1_2) = arg \max_{p_2, y_2} U(p_2, y_2 p^0_2, y^0_2). \tag{65}$$

3. general fraud profit is defined

$$F^1 = U^1 - U^0 + V^1 - V^0 \tag{66}$$

4. the procedure is repeated for a number N pairs x_{12}, x_{21} defining a sequence of general fraud profits $f^k,\ k = 1, ..., N$ where N is the number of iterations of optimiaztion method/footnoteThe optimization of prices p, y is performed by means of GMJ set of methods to make optimization as efficient as possible..

5. the quadruple $p_1^*, p_2^*, y_1^*, y_2^*$ with minimal fraud profit f^* is considered as equilibrium

3.4 Profit Analysis

The sufficient condition of equilibrium is convexity of profit functions. That is tested this way:

1. fix the equilibrium prices $p_1^*, p_2^*, y_1^*, y_2^*$

2. change the first parameter p_1 step-by-step while keeping other three constant, draw two graphs $U(p_1, p_2^*, y_1^*, y_2^*)$ and $V(p_1, p_2^*, y_1^*, y_2^*$, write values to tables

3. do the same for all the parameters p_1, p_2, y_1, y_2

3.5 Modelling Custoers

Intervals between customer arrivals are random and are defined by exponential distribution. The algorithm is described in section 5

Initial testing the software could be more convenient by using regular arrivals of customers by equal intervals.

4 "Greedy-Look-Ahead" (GLA) Equilibrium

Definition of GLA equilibrium is similar to that of NLA because in both the cases competitors responses. are anticipated while searching for the equilibrium prices. The difference is that the 'greedy' servers select the free parameters not by equilibrium conditions but by maximal profit. We consider only one version GLA(py) where the free parameter is a vector of resource consumption x. The resource vector x is predicted for all servers assuming that each server defines resource demand by maximizing profit at fixed both the resource and service price vectors p and y .

Using GLA(py) one transforms linear profit functions of p_j[12] into nonlinear ones. This way one may satisfy the necessary equilibrium conditions if the profit

[12] As defined by expressions (4) and (10).

functions are convex. That is true in both the cases: NLA and GLA. In this sense both versions are equivalent. Thus one may select a version that better describes the actual economical behavior of participants.

Both the NLA and the GLA approaches is based on the important tacit assumption that servers know profit functions of their competitors. This is not true as usual. However, that is a price one pays for making profit functions strictly convex. This is needed to satisfy necessary equilibrium conditions. The price is not so great when servers know at least some approximation of competitor profit functions and behavior

4.0.1 Search for GLA(p,y) Equilibrium

Defining Nash Profit Using GLA(p,y) the profit of the i-th server:

$$U_i(p,y) = u_i(x_j^* y_j, p_j,\ j=1,...,m) = \\ a_i y_i^* + p_i \sum_{j \neq i} x_{ij}^* - \sum_{j \neq i} p_j x_{ij}^*. \quad (67)$$

Here i is the server index. a_i is the rate of customers. $x_j^* = (x_{jk}^*,\ k = 1,...,m),\ x_{jk}^* = x_{jk}(p,y)$ is the greedy resource vector that defines the capacity w_j of server j and is obtained by maximizing the profit function U_i at given contract prices p and y. From the balance condition follows $x_{ii}^* = b_i - \sum_{j \neq i} x_{ji}^*$. At fixed prices (p,y) the vector x^* is defined by these conditions

$$\max_{x_{ij} \in B} U_i(p,y),\ i=1,...,m,\ i \neq j. \quad (68)$$

Denote the Greedy resource consumption vector at fixed price vectors (p,y) as

$$x^* = x(p,y). \quad (69)$$

In two-server cases

$$U_1(p_1,y_1,p_2,y_2) = u_1(x_{12}^*, y_1, p_1, x_{21}^*, y_2, p_2) = \\ a_1 y_1 + p_1 x_{21}^* - p_2 x_{12}^*. \quad (70)$$

and

$$U_2(p_1,y_1,p_2,y_2) = u_2(x_{21}^*, y_2, p_2, x_{12}^*, y_1, p_1) = \\ a_2 y_2 + p_2 x_{12}^* - p_1 x_{21}^*. \quad (71)$$

Here x_{11}^* and x_{22}^* are not included explicitly because they are defined by the balance conditions

$$x_{11}^* = b_1 - x + 21^*,\ x_{22}^* = b_2 - x_{12}^*. \quad (72)$$

One solves (35) many times for each fixed price vector (p,y) before the equilibrium values $(p = p^*, y^*)$ are reached.

Defining GLA(p,y) Profit First a contract-vector $(p_i^0, y_i^0\ \ i = 1, ..., m)$ is fixed. Then the fraud-vector $(p_i^1, y_i^1\ \ i = 1, ..., m)$ is obtained by maximizing the profits of each server i assuming that other partners $j \neq i$ keep the contract resource prices.

$$(p_i^1, y_i^1) = arg \max_{p_i, y_i} U_i(p_i, y_i\ p_j^0, y_j^0\ j = 1, ..., m,\ j \neq i),\ i = 1, ..., m. \tag{73}$$

Here $p_j^0, y_j^0\ j \neq i$ are contact prices of competing servers $j \neq i$.

In the two-server case

$$(p_1^1, y_1^1) = arg \max_{p_1, y_1} U_1(p_1, y_1, p_2^0, y_2^0), \tag{74}$$

and

$$(p_2^1, y_2^1) = arg \max_{p_2, y_2} U_2(p_2, y_2 p_2^0, y_2^0). \tag{75}$$

Condition (40) transforms vectors $p^n, y^n\ n = 0, 1, 2, ...$ into vectors p^{n+1}, y^{n+1}, where $p^n = (p_i^n,\ i = 1, ..., m)$, and $y^n = (y_i^n,\ i = 1, ...m)$. Denote this transformation by T

$$z^{n+1} = T(z^n),\ n = 0, 1, 2, ... \tag{76}$$

where $z = (p, y)$. We reach the equilibrium[13] at the fixed point p^n, where

$$z^n = T(z^n). \tag{77}$$

If the equilibrium exists but the transformation T is not contracting then we minimize the square deviation

$$\min_{z \in B} \parallel z - T(z) \parallel^2 . \tag{78}$$

The equilibrium is achieved, if the minimum (56) is zero.

The alternative way to achieve equilibrium is by minimizing the fraud profit

$$\min_{z \in B} \sum_i (U_i(T(z)) - U_i(z)). \tag{79}$$

Here the difference $U_i(T(z)) - u_i(z)$ shows the profit obtained by i-th server by deviating from the NLA equilibrium z^*.
The final equilibrium values of x^* are defined substituting (p^*, y^*) into expression (49)

[13] If the equilibrium exists.

5 Monte-Carlo Simulation

5.1 Search for Equilibrium

The analytical solution of both the described market models is not practical. Therefore, we briefly consider the basic steps of an algorithm of the statistical simulation using Monte-Carlo techniques. The algorithm implements two basic tasks:

- generates the next event time t,
- updates the state of queuing system defined by the vector of waiting customers $n(t) = (n_i(t),\ i = 1, ..., m)$,
- updates the vector $h(t) = (h_i(t),\ i = 1, ..., m)$ of the service cost including the money charged and the time lost.

There are $2m + 2$ types of event times t:

- the time t when a customer arrives into the system,
- the time t when a customer arrives into the i-th server, $i = 1, ..., m$,
- the time t when a customer departs from the i-th server,
- the time t when a customer abandons the service (departs from the system without being served).

Here $i = 1, ..., m$. The system state is updated at each event time t. Two vectors define the system state:

- a vector $n = n(t)$ with m components $n = (n_1, ..., n_m)$, where $n_i = n_i(t)$ shows the number of customers waiting for the service of the i-th server,
- a vector $h = h(t)$ with m components $h = (h_1, ..., h_m)$, where $h_i(t) = y_i + \gamma_i,\ \gamma_i = n_i(t)/w_i$ shows the total customer expenses, y_i is money charged for the service, and γ_i is the time lost waiting for the service of the i-th server[14].

There are no state changes between events. The basic steps of the Monte Carlo algorithm:

1. fix the zero event time $t = t^0 = 0$ when the first customer arrives,
2. define the zero state vector n^0 by the condition: $n_i^0 = 0,\ i = 1, ..., m$ and the zero state vector h^0 by the condition : $h_i^0 = y_i = 0,\ i = 1, ..., m$ because there are no customers waiting for service yet,

[14]For simplicity, it is assumed that "time-is-money" and that a unit of time cost a unit of money, in the real life cases the corresponding coefficients should be included

3. define the next arrival into the system by the expression

$$\tau_a = -1/a \; ln(1-\eta) \tag{80}$$

where η is a random number uniformly distributed in the interval [0,1],

4. chose the best server i^0 for the first customer by the condition $i^0 = arg \min_{i=0,\dots,m} \; h_i$ where $h_i = y_i$, because $\gamma_i = 0, \; i = 1, ..., m$ since there are no customers waiting yet, $i^0 = 0$ means that the customer abandons the service,

5. define the time of event when the first customer will be served by the server i^0 using the expression

$$\tau_{i^0} = -1/x_{i^0} \; ln(1-\eta), \tag{81}$$

6. define the next event t^1 by comparing the arrival time τ_a and the service time τ_{i^0} :
if $\tau_a < \tau_{i^0}$ then $t^1 = \tau_a$,
if $\tau_a > \tau_{i^0}$ then $t^1 = \tau_{i^0}$,

7. define the system state at the next event t^1:
if $t^1 = \tau_a$ then $n_{i^0} = 1$ and $n_i = 0, \; i = 1, ..., m, \; i \neq i^0$,
consequently $h_{i^0} = y_{i^0} + 1/w_{i^0}$ and $h_i = y_i, \; i = 1, ..., m, \; i \neq i^0$,
if $t^1 = \tau_{i^0}$ then $n_i = 0, \; i = 1, ..., m$, and $h_i = y_i, \; i = 1, ..., m, \; i \neq i^0$.

Definition of later events and system states is longer but the main idea remains the same. For illustration, we update the fourth step for the nth customer:

- chose the best server i^0 for the nth customer by the condition
$i^0 = arg \min_{i=0,\dots,m} \; h_i$,
where $h_i = y_i + \gamma_i$, $\gamma_{i^0} = n_i/\omega_{i^0}$, and n_i is the number of customers waiting for server i.

The algorithm can be directly adapted to the Monte-Carlo simulation of the Nash model with two servers, too. For example, that can be done this way:
- set to unit both the resource charges $p_i = 1, \; i = 1, 2$,
- set to zero resources exchanges $x_{21} = x_{12} = 0$,
- assume that $x_{11} = x_1, \; x_{22} = x_2$, where variables $x_1, \; x_2$ are from expression (13).
If the number of severs $m > 2$ then some modification of the described algorithm is needed.

5.2 Testing Equilibrium Conditions

In the Monte-Carlo simulation, equilibrium tests (32) should be relaxed by accepting some simulation error ϵ:

$$\min_{z \in B} \parallel z - T(z) \parallel^2 \leq \epsilon, \tag{82}$$

The alternative way to achieve equilibrium is by minimizing the fraud profit (33). Then the test is

$$\min_{z \in B} \sum_i (u_i(T(z)) - u_i(z)) \leq \epsilon \tag{83}$$

To test the convexity of profit functions, some smoothing is desirable. The smoothing eliminates the random deviations due to Monte-Carlo simulation. Both the convolution and the Wiener filters are applied for smoothing the profit functions (possibly multi-modal). The convolution filter defines the function at some fixed point as an average of values in the neighborhood of this point. The more sophisticated Wiener filter is implemented, too.

5.3 Wiener Filter

If the objective function $f(x)$ is defined by Monte Carlo simulation, some noise is present. That means that one observes the sum

$$\phi(x) = f(x) + \xi, \tag{84}$$

where ξ is a random number called the noise.

If finding the optimum of a convex function $f(x)$ is the only goal, we can apply some stochastic optimization algorithms [Ermoljev and Wets, 1988]. These algorithms converge to the optimum of $f(x)$ by filtering the noise during the optimization process.

To test properties of $f(x)$, such as convexity, unimodality e.t.c., we need specific smoothing algorithms that eliminate false local optima. In one-dimensional cases, a convenient smoothing function is the conditional expectation of the Wiener process with noise [Kushner, 1964, Senkiene, 1980]. It is assumed that the optimization parameter $x \in [0,1]$, the Wiener parameter is a unit, and the noise ξ is Gaussian with zero mean and variance S_i at the points $x^i \in [0,1],\ i = 1,...,n$. Then the conditional expectation $\mu_k = \mu_n(x^k)$ of the objective function $y_k = f(x^k)$ at some fixed point x^k with respect to the observations results $y_i = \phi(x^i),\ i = 1,...,n$, can be expressed this way

$$\mu_k = \frac{\sum_{i=1}^k b_i y_i + \frac{b_k}{c_k} \sum_{i=k+1}^n c_i y_i}{\sum_{i=1}^k b_i + \frac{b_k}{c_k} \sum_{i=k+1}^n c_i}. \tag{85}$$

Here

$$b_1 = 1,\ B_1 = 1, \tag{86}$$

$$b_2 = \frac{S_1^2 + r_{1,2}}{S_2^2},\ B_2 = B_1 + b_2, \tag{87}$$

....

$$b_k = \frac{S_{k-1}^2 b_{k-1} + r_{k,k-1} B_{k-1}}{S_k^2},\ B_k = B_{k-1} + b_k, \tag{88}$$

$$c_n = 0, \tag{89}$$

$$c_k = \frac{S_{k+1}^2 c_{k+1} + r_{k,k+1} C_{k+1}}{S_k^2}, C_{k+1} = \sum_{i=k+1}^{n} c_i, \tag{90}$$

$$r_{i,j} = |x^i - x^j|. \tag{91}$$

It is convenient to assume that

$$S_i = S,\ i = 1, ..., n, \tag{92}$$

where S can be considered as a smoothing parameter.

If $S = 0$ then no smoothing occurs. The smoothing function (the conditional expectation) is the piece-wise line connecting the observed points (see Figure 1).

If S is large then one obtains a horizontal line corresponding to the average value of observed values y_i. That means a sort of "total smoothing." In modeling the yield of differential amplifiers, the best smoothing was achieved at $S = 10$ [Mockus et al., 1997].

Figure 2 shows how the first server profit u_1 depends on the price p_1 charged for its resources. There are two samples of the same relation. They show the differences between two samples of random arrival times of fifty customers.

The buttons 'smooth' and 'wiener' on the right side are for switching on these filters. The button 'smooth' is for the convolution filter. The button 'wiener' is for the Wiener filter.
The field denoted by 'S' at the top right corner, defines the smoothing parameter S of the Wiener filter (see expression (92).
One can increase the level of smoothing by pressing the 'smooth' or 'wiener' buttons repeatedly.
The 'refresh' button repeats the Monte-Carlo simulation of the same profit function.

Figures 3 show the third sample of the 'refreshed' smoothed graph and the same sample smoothed by the convolution filter.

The underlying profit function is the same in all the Figures, from 2 up to 3. The repeated simulation defines different graphs because of the simulation errors. After smoothing, the level of these errors is lower.

6 Software Example

Java j2sdk implementation [Treigys, 2004] of the Walras model is on the websites.

Figure 4 shows the input page. The method 'Bayes1' is set. The parameters:
the number of initial iterationslt is 5,
the number of iterationsit is 10,
the stocks of server resources b_1 and b_2 is 0.7
the "run-away" threshold $C0$ is 20
the customer rate A is 20,

the number of time units M is 5
the efficiency of all servers z_{ij} is 1.0
the accuracy is 20 %
the lower bounds *min* are 0.0,
the upper limits *max* are 30.0,
t

Figure 5 shows the results of optimization.
In this figure:
iteration denotes the "best' iteration
$F(x)$ means the minimal deviation from the equilibrium point,
y_1, p_1, y_2, p_2 are the optimal values of the parameters.

Figure 6 shows how the profits of the first server depends on the service charge y_1.
The figure 7 shows how the profits of the first server depends on the resource price p_1.
The result is difficult to explain so the additional investigation is needed. That could be made easy by performing profit analysis by separate program at fixed estimates of equilibrium vectors p, y.

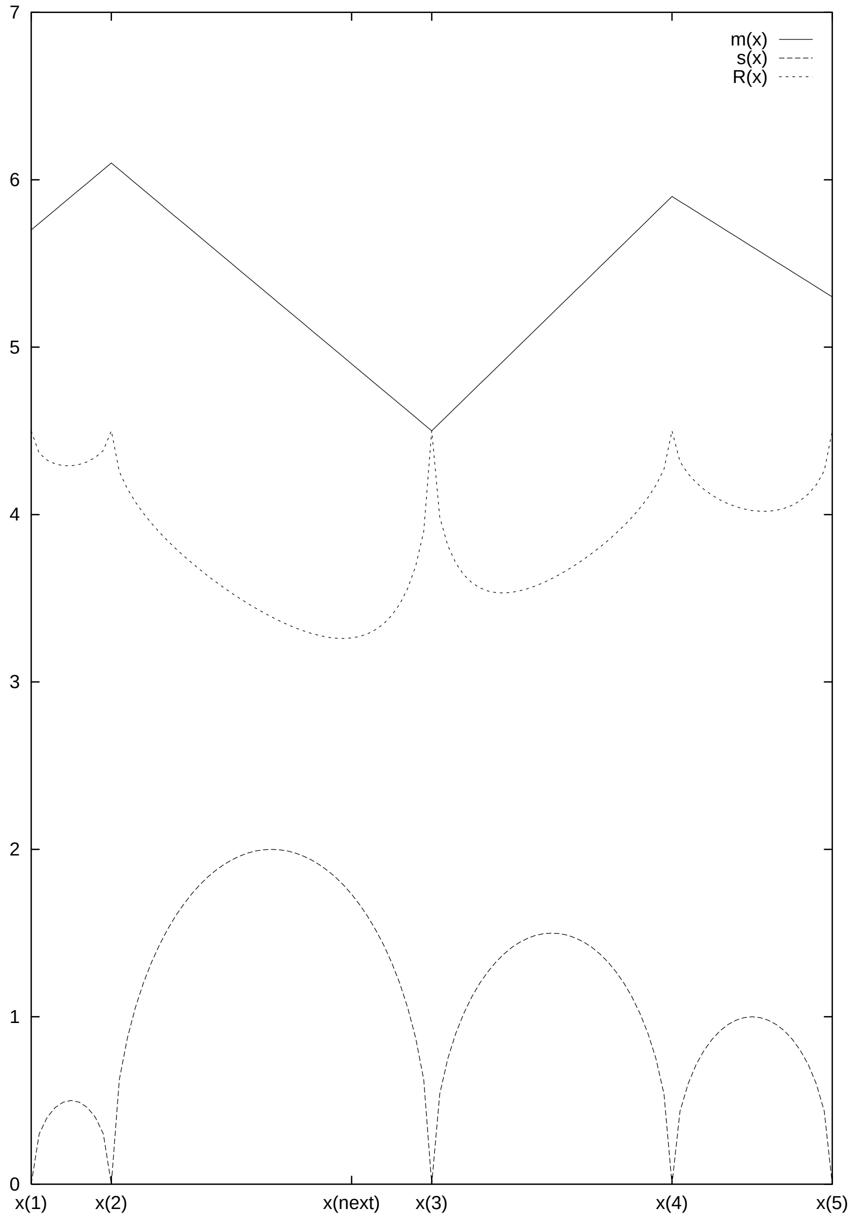


Figure 1: The Wiener Model. The conditional expectation $m(x)$, the conditional standard $s(x)$, and the risk function $R(x)$ regarding fixed values $x(1), y(1), x(2), y(2), ...$

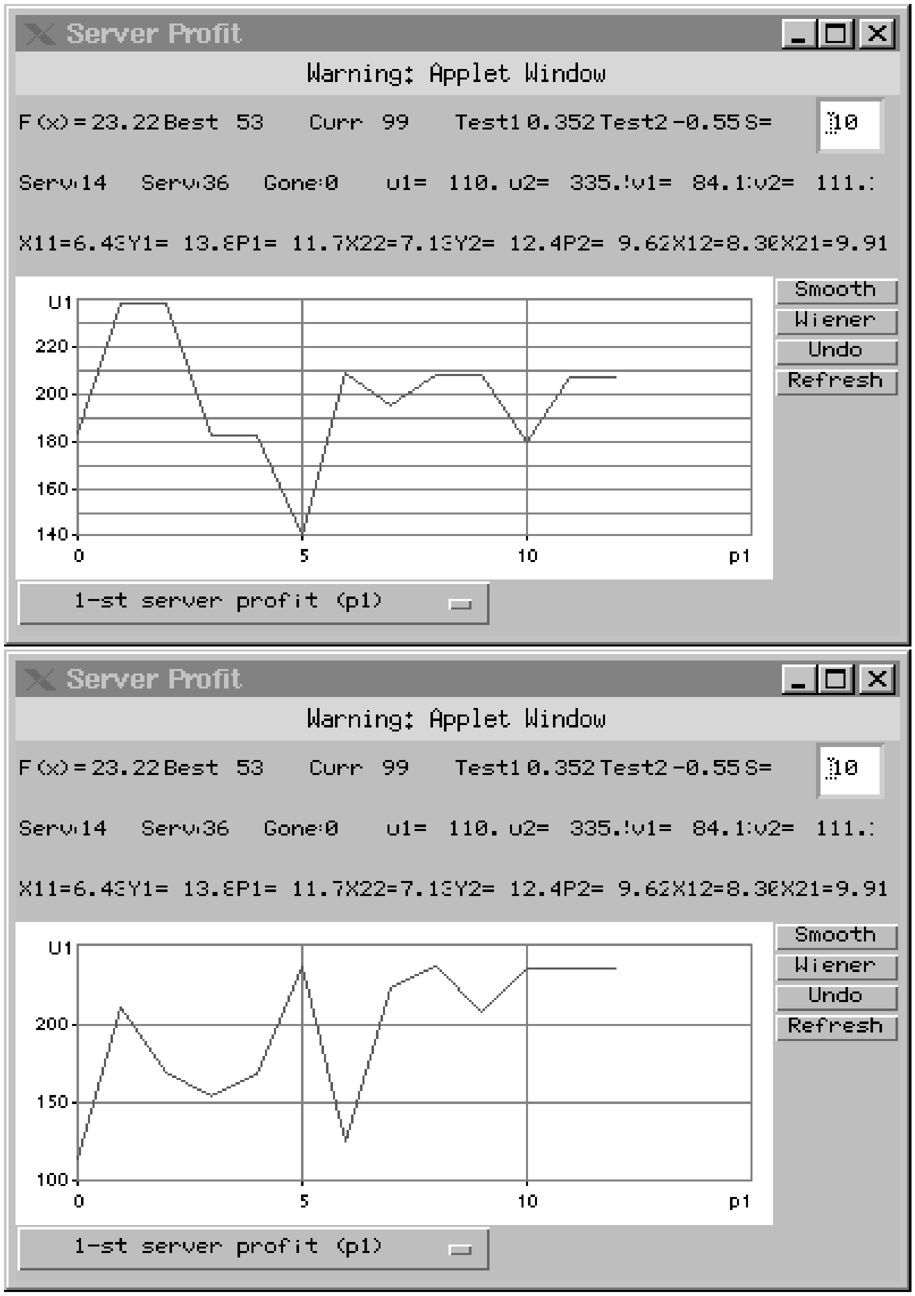


Figure 2: The relation of the profit u_1 on the resource price p_1 (top), the "refreshed" relation of the profit u_1 on the resource price p_1 (bottom).

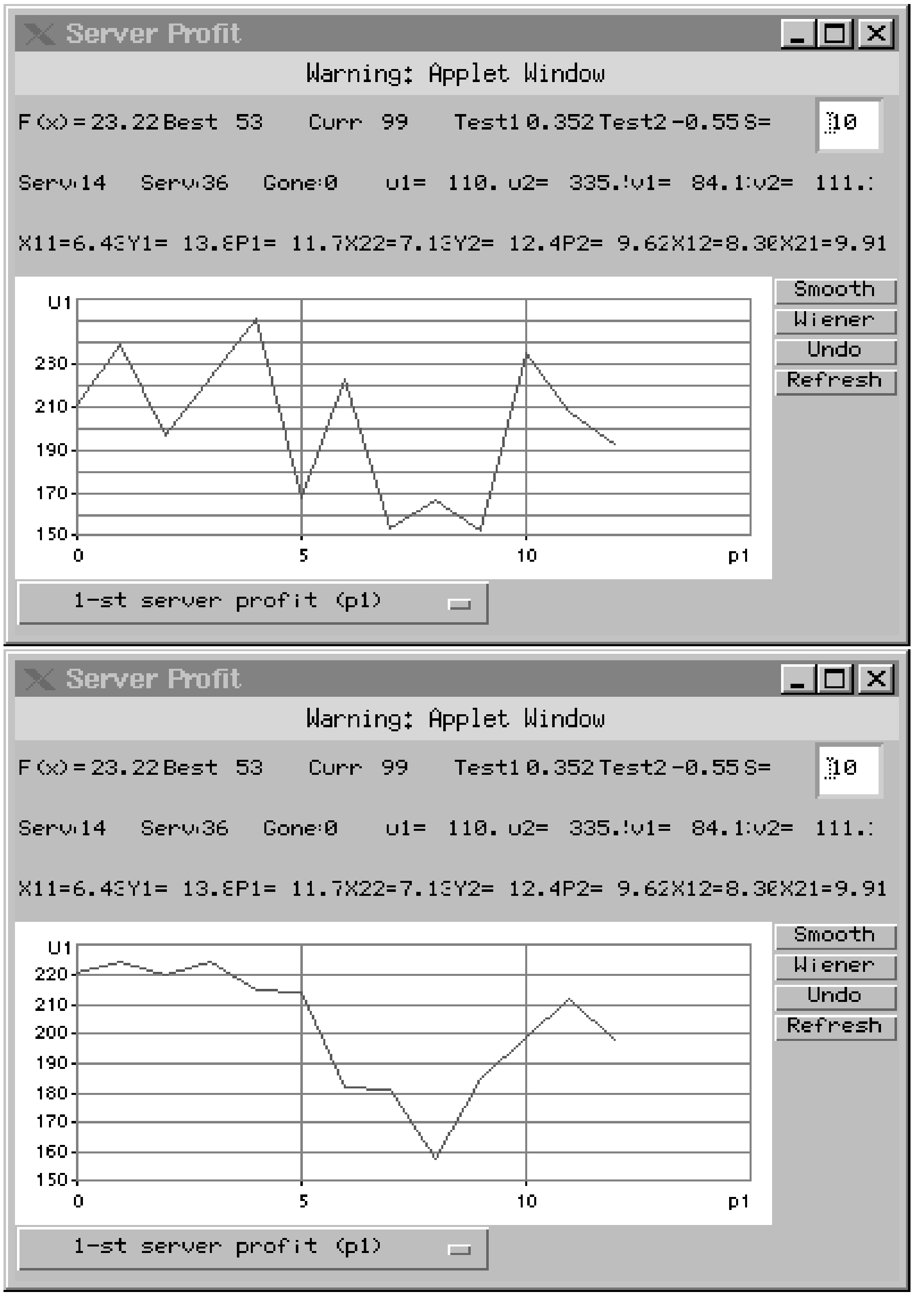


Figure 3: The third sample of the relation of the profit u_1 on the resource price p_1 (top), the relation of the profit u_1 on the price p_1, smoothed by the convolution filter (bottom).

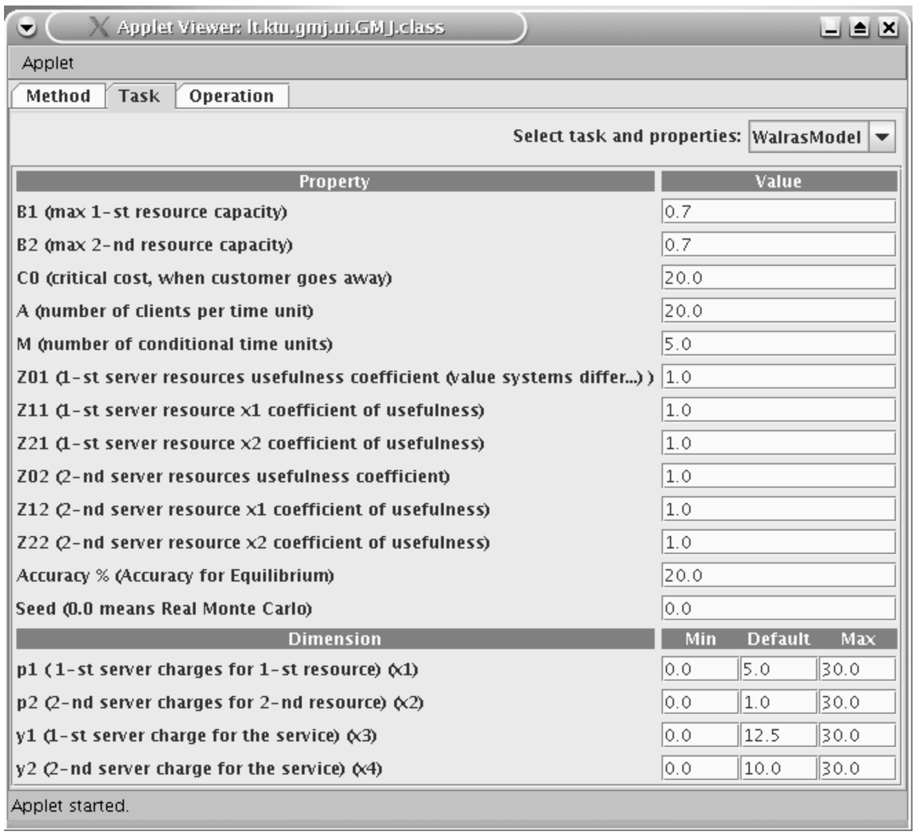


Figure 4: The table of initial parameters.

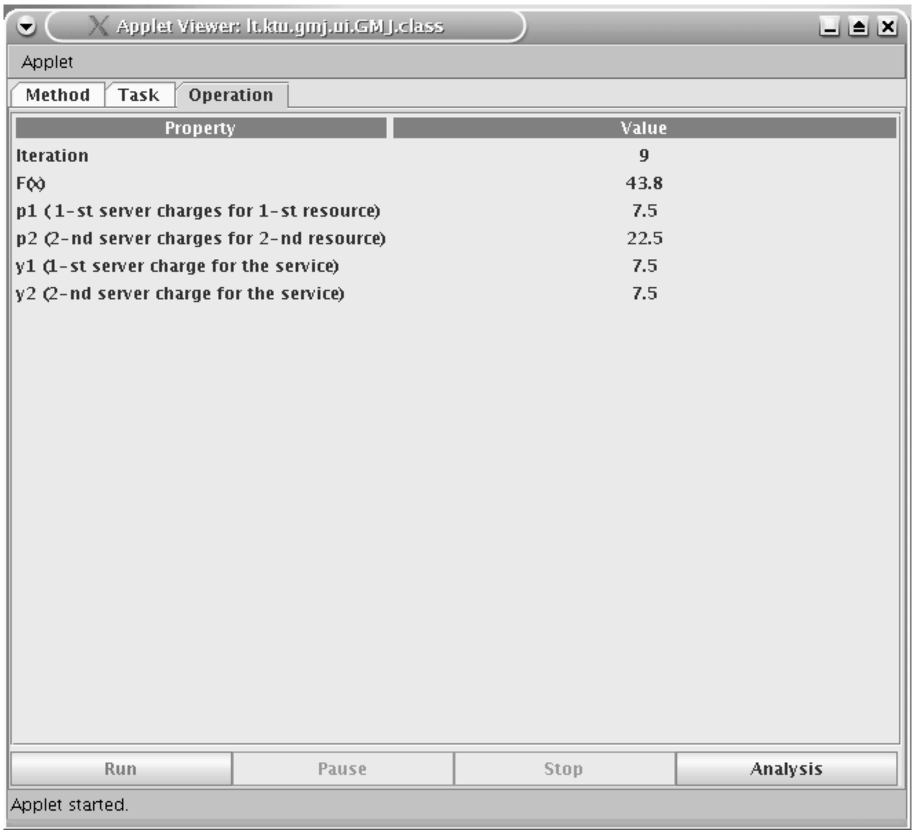


Figure 5: Optimization results using the method Bayes1.

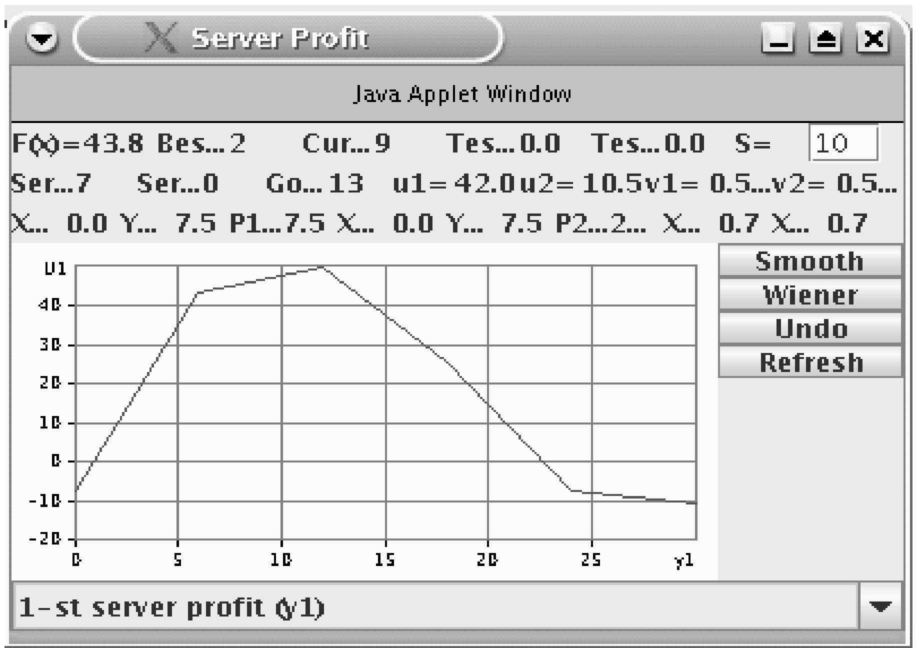
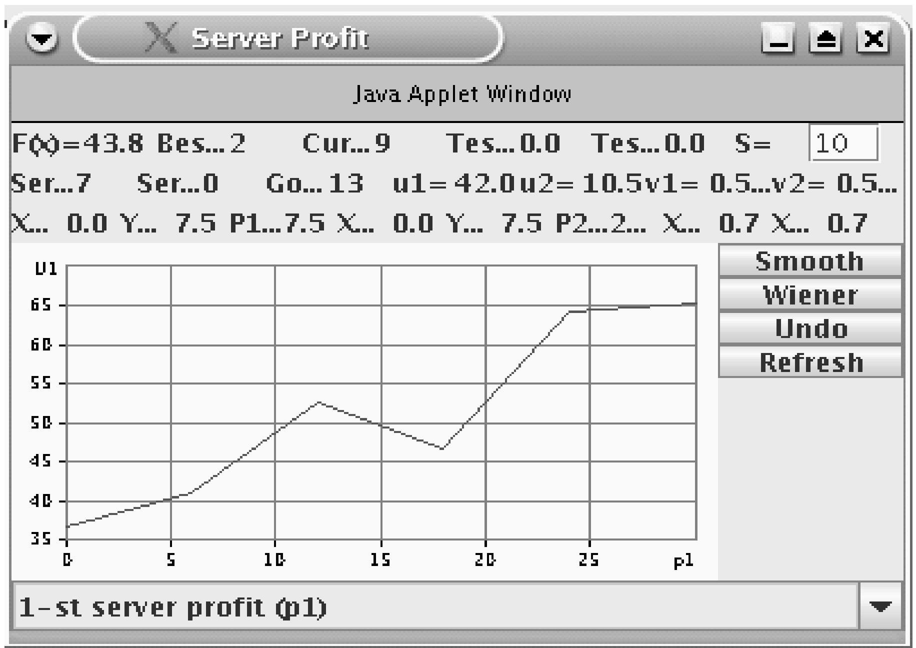


Figure 6: Relation of the profit u_1 on the service charge y_1.



p

Figure 7: Relation of the profit u_1 on the resource price p_1.

References

[Brouwer, 1912] Brouwer, L. (1912). Uber abbildung von mannigfaltigkeiten. *Math. Ann.*, 71:97–115.

[Browder, 1968] Browder, F. (1968). The fixed point theory of multivalued mappings in topological vector spaces. *Math. Ann.*, 177:283–302.

[Ermoljev and Wets, 1988] Ermoljev, Y. and Wets, R.-B. (1988). *Numerical Techniques for Stochastic Optimization*. Springer-Verlag, Berlin- New York-London.

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

[Gnedenko and Kovalenko, 1987] Gnedenko, B. and Kovalenko, I. (1987). *Introduction to the Theory of Queuing*. Nauka, Moscow. (in Russian).

[Herings, 1994] Herings, P. (1994). A gobally and universally stable price ajustment process. Technical Report 9452, Center for Economic Research, Tilburg University, The Netherlands, Tilburg.

[Kakutani, 1941] Kakutani, S. (1941). A generalization of brouwer's fixed point theorem. *Duke Journal of Mathematics*, 8:457–459.

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

[Michael, 1976] Michael, J. (1976). *The Computation of Fixed Points and Applications*. Springer-Verlag, Berlin.

[Mockus, 1989] Mockus, J. (1989). *Bayesian approach to global optimization*. Kluwer Academic Publishers, Dordrecht-London-Boston.

[Mockus, 2000] Mockus, J. (2000). *A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach*. Kluwer Academic Publishers. ISBN 0-7923-6359-0.

[Mockus et al., 1997] 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.

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

[Nash, 1951] Nash, J. (1951). Noncooperative games. *Annals of Mathematics*, 54:286–295.

[Neuman and Morgenstern, 1953] Neuman, J. and Morgenstern, O. (1953). *Theory of Games and Economic Behavior.* Princeton University Press, Princeton.

[Perlibakas, 1999] Perlibakas, V. (1999). Software for the walras problem. Technical report, Kaunas Technological University, Faculty of Informatics, Kaunas, Lithuania. (in Lithuanian).

[Rolstadas, 1995] Rolstadas, A. (1995). Enterprise modeling for competitive manufacturing. *CONTROL ENGINEERING PRACTICE*, 3:43–50.

[Rosenmuller, 1981] Rosenmuller, J. (1981). *The Theory of Games and Markets.* North-Holand, Amsterdam.

[Scarf and Hansen, 1973] Scarf, H. and Hansen, T. (1973). *The Computation of Economic Equilibria.* Yale University Press, New Haven - London.

[Senkiene, 1980] Senkiene, E. (1980). Properties of conditional mean and variance of the wiener process in the presence of noise, and the convergence of bayesian optimization algorithms. In *Theory of Optimal Decision.* Institute of Mathematics and Cybernetics, Akademia Nauk Lithuanian SSR, Vilnius, Lithuania. (in Russian).

[Skruodys, 2004] Skruodys, N. (2004). *Investigation of the Competition Models.* PhD thesis, Kaunas Technological University, Kaunas, Lithuania.

[Sviderskaite, 2003] Sviderskaite, A. (2003). Software for the walras problem, walras profit analysis. Technical report, Kaunas Technological University, Faculty of Informatics, Kaunas, Lithuania. (in Lithuanian).

[Treigys, 2003] Treigys, P. (2003). Software for the walras problem. Technical report, Vilnius Gedimino Technical University, Vilnius, Lithuania. (in Lithuanian).

[Treigys, 2004] Treigys, P. (2004). Updated software for the walras problem. Technical report, Vilnius Gedimino Technical University, Vilnius, Lithuania. (in Lithuanian).

[Walras, 1874] Walras, L. (1874). *Elements d'Economie Politique Pure.* LKorbaz and Company, Lousanne.