Duel Problem

J. Mockus

September 26, 2009

1 Introduction

In all three competition models, the Nash, the Walras, and the Inspector, the steady-state conditions were considered, for simplicity. In real life competition processes, dynamics are essential. Dynamical competition models may be represented as differential games. They are difficult for numerical and visual analysis, as usual [2, 3, 4, 5, 6, 13, 10, 11, 12]. Therefore, one tries to make the dynamic competition models as simple as possible.

Consider, as a simple illustration, two objects in space trying to destroy each other. Assume, as a first approximation, that the control parameters are: initial points z_0, w_0, "climbing" rates a, b and firing times t_1, t_2. These parameters are set before the start for both objects. Denote the control parameters by vectors $x = (x_i,\ i = 1, 2, 3)$ and $y = (y_i,\ i = 1, 2, 3)$, correspondingly. Denote by (a_i^x, a_i^y) the lower bounds of these parameters. Denote by (b_i^x, b_i^y) the upper bounds. In the illustrative example $x_1 = z_0, x_2 = a, x_3 = t_1$, and $y_1 = w_0, y_2 = b, y_3 = t_2$, where $-1 \leq z_0 \leq 1, -1 \leq w_0 \leq 1, -1 \leq a \leq 1, -1 \leq b \leq 1$. Movements of the objects are described in two-dimensional "hight-time" space by the first-order differential equations

$$\begin{aligned} dz(t)/dt &= az(t), & (1)\\ dw(\tau)/d\tau &= bw(\tau),\ \tau = 2 - t. & (2) \end{aligned}$$

The solution of these equations defines trajectories of the objects

$$\begin{aligned} z(t) &= z_0 e^{at}, & (3)\\ w(\tau) &= w_0 e^{bt}. & (4) \end{aligned}$$

As usual, simple models of moving objects use homogenuos second-order differential equations, at least

$$\begin{aligned} d^2z(t)/dt^2 + gdz(t)/dt &= az(t), & (5)\\ d^2w(t)/dt^2 + hdw(\tau)/d\tau &= bw(\tau),\ \tau = 2 - t. & (6) \end{aligned}$$

The solution of these equations defines more realistic trajectories of the objects because this includes acceleration, too. These trajectories depend on the roots r_1, r_2 of the charasteristic equation

$$r^2 + gr - a = o. \quad (7)$$

Here

$$r_1 = -g/2 + \sqrt{(g/2)^2 + a},\ r_2 = -g/2 - \sqrt{(g/2)^2 + a}. \quad (8)$$

The trajectory of the first object

$$z(t) = \begin{cases} C_1 e^{r_1 t} + C_2 e^{r_2 t}, & \text{if } r_1 \neq r_2\\ (C_1 + C_2 t)e^{-(g/2)t}, & \text{if } r_1 = r_2\\ e^{-(g/2)t}(C_1 cos\beta t + sin\beta t), & \text{if } r_1 = -g/2 + \beta i, r_2 = -g/2 - \beta i. \end{cases} \quad (9)$$

Here $\beta = \sqrt{-a-(g/2)^2}$.

The constants C_1, C_2 are defined by initial or boundary conditions. Suppose that the initial higth $z(0) = z_0$ are fixed by operator of the first flying object. Then

$$z_0 = \begin{cases} C_1 e^{r_1 t} + C_2 e^{r_2 t} = C_1 + C_2, & \text{if } r_1 \neq r_2 \\ (C_1 + C_2 t) e^{-(g/2)t} = C_1, & \text{if } r_1 = r_2 \\ e^{-(g/2)t}(C_1 cos\beta t + sin\beta t) = C_1, & \text{if } r_1 = -(g/2) + \beta i, r_2 = -(g/2) - \beta i \end{cases} \quad (10)$$

Considering the speed $dz(t)/dt$ one can write

$$dz(t)/dt = C_1 r_1 e^{r_1 t} + C_2 r_2 e^{r_2 t}, if\ r_1 \neq r_2 \quad (11)$$

$$dz(t)/dt = -(g/2)(C_1 + C_2 t) e^{-(g/2)t} + C_2 e^{-(g/2)t}, if\ r_1 = r_2 \quad (12)$$

(13)

$$dz(t)/dt = -(g/2) e^{-g/2t}(C_1 cos\beta t + C_2 sin\beta t) + e^{-g/2t}(-\beta C_1 sin\beta t + \beta C_2 cos\beta t) \\ if r_1 = -(g/2) + \beta i, r_2 = -(g/2) - \beta i \quad (14)$$

From here assuming that initial speed $dz(0)/dt = v_0$

$$v_0 = \begin{cases} C_1 r_1 + C_2 r_2, & \text{if } r_1 \neq r_2 \\ -(g/2) C_1 + C_2, & \text{if } r_1 = r_2 \\ -(g/2) C_1 + \beta C_2, & \text{if } r_1 = \alpha + \beta i, r_2 = \alpha - \beta i \end{cases} \quad (15)$$

This way we obtained three pairs of linear equations corresponding to three different pairs of roots $r_1, 2$ defining two unknown constants C_1, C_2.
The trajectory of the second object $w(t)$ is defined in the same way.
To prevent the trajectories going "underground" during the time T we define the constants by the boundary conditions $z(0) = z_0 > 0$ and $z(T) = 0$. Then

$$z(0) = C_1 + C_2 = z_0 \\ Z(T) = C_1 e^{r_1 T} + C_2 e^{r_2 T} = 0. \quad (16)$$

From (16)

$$C_2 = z_0 \frac{e^{r_1 T}}{e^{r_1 T} - e^{r_2 T}}, \\ C_1 = z_0 - C_2. \quad (17)$$

Defining the parameters a, g we keep the condition $a > -(g/2)^2$. Note that here the speed dz/dt and acceleration d^2z/dt^2 depends on time t and are defined as functions of parameters a and g,

First, we shall apply first-order linear differential equations (1) and (2) as a first approximation to explain and test algorithms of optimization and modeling. Denote by $d(t)$ a distance between objects at the moment t

$$d(t) = ||(w(t), \tau) - (z(t), t)||. \tag{18}$$

Denote by $p(t)$ the hitting probability

$$p(t) = 1 - (d(t)/D)^{\alpha}. \tag{19}$$

Here $D \geq \max_t d(t)$ and $\alpha > 0$. The expected utility function of the first object

$$U(x,y) = \begin{cases} u_1 p(x_3) - u_2(1-p(x_3))p(y_3), & \text{if } x_3 < y_3, \\ -u_2 p(y_3) + u_1(1-p(y_3))p(x_3), & \text{if } x_3 > y_3, \\ u_1 p(x_3) - u_2 p(y_3), & \text{if } x_3 = y_3, \end{cases} \tag{20}$$

The expected utility function of the second object

$$V(x,y) = \begin{cases} v_1 p(y_3) - v_2(1-p(y_3))p(x_3), & \text{if } x_3 > y_3, \\ -v_2 p(x_3) + v_1(1-p(x_3))p(y_3), & \text{if } x_3 < y_3, \\ v_1 p(y_3) - v_2 p(x_3), & \text{if } x_3 = y_3. \end{cases} \tag{21}$$

Expressions (20) and (21) of expected utilities follows from expression (19) of hitting probabilities. It is assumed that utilities to hit the hostile object are plus u_1 and v_1. Utilities to be hit are minus u_2 and v_2. Here u_1, u_2 denote utilities of the first object and v_1, v_2 define utilities of the second one.

2 Convex Version

Expected utilities (20) and (21) are not convex functions of variables $p(x_3)$ and $p(y_3)$. The convex version may be obtained assuming that the object destroys the enemy, if it is not hit itself. Suppose that $u_1 = u_2 = v_1 = v_2 = 1$. Then it follows from expressions (20) and (21) that the expected utility function of the first object

$$U(x,y) = \begin{cases} p(x_3) - (1-p(x_3)), & \text{if } x_3 < y_3, \\ -p(y_3) + (1-p(y_3)), & \text{if } x_3 > y_3, \\ p(x_3) - p(y_3), & \text{if } x_3 = y_3, \end{cases} \tag{22}$$

The expected utility function of the second object

$$V(x,y) = \begin{cases} p(y_3) - (1-p(y_3)), & \text{if } x_3 > y_3, \\ -p(x_3) + (1-p(x_3)), & \text{if } x_3 < y_3, \\ p(y_3) - p(x_3), & \text{if } x_3 = y_3. \end{cases} \tag{23}$$

Expected utilities (20) and (21) are convex functions of probabilities $p(x_3)$ and $p(y_3)$ at the equilibrium point

$$\begin{aligned} p(x_3) &= 0.5, \\ p(y_3) &= 0.5. \end{aligned} \tag{24}$$

The convexity of expected utilities (20) and (21) as functions of probabilities $p(x_3)$ and $p(y_3)$ does not provide their convexity as functions of firing times x_3 and y_3 and other parameters. The convexity is a part of sufficient equilibrium conditions. Therefore, the equilibrium may not exist.

Expected utilities reach the maximal values at the ends of intervals of parameters z_0, a, w_0, b, as usual. For the first object there are four pure strategies corresponding to various arrangements of these ends
$(z_0 = z_{min}, a = a_{min}), (z_0 = z_{min}, a = a_{max}), (z_0 = z_{max}, a = a_{min}),$
$(z_0 = z_{max}, a = a_{max})$.
The pure strategies for the second object:
$(w_0 = w_{min}, b = b_{min}), (w_0 = w_{min}, b = b_{max}), (w_0 = w_{max}, b = b_{min}),$
$(w_0 = w_{max}, b = b_{max})$.
We may search for the pure equilibrium of bimatrix games (22)(23) corresponding to each fixed pair $(x_3.y_3)$ of firing times using simple discrete algorithm. However, the probability to obtain the pure equilibrium is high only for large bimatrix games [1]. Therefore, we consider mixed strategies, first.

3 Mixed Strategies

If expected utilities reach the maximal values at the ends of intervals of parameters z_0, a, w_0, b then mixed strategies could be helpful. The mixed strategies mean that each object chooses at random a combination of lower and upper bounds. Define those bounds as zero-ones for simplicity. There are four events: $(z_0 = 0, a = 0), (z_0 = 0, a = 1), (z_0 = 1, a = 0), (z_0 = 1, a = 1)$. Probabilities of these events are defined by first four components
$x(1), ..., x(4),\ x(i) \geq 0,\ \sum_{i=1}^{4} x(i) = 1$ of a vector
$x = (x(i),\ i = 1, ..., 5)$[1].
The fifth component $x(5) = t_1$ defines the firing time. A vector
$y = (y(j),\ j = 1, ..., 5)$ has a similar meaning for the second object.

Applying mixed strategies $x = (x_i,\ i = 1, ..., 5)$ and
$y = (y_j,\ j = 1, ..., 5)$, the utility functions of the first and the second objects are

$$U_a(x, y) = \sum_{r \in R} \sum_{s \in S} x(r) u(r, s) y(s), \tag{25}$$

$$V_a(x, y) = \sum_{r \in R} \sum_{s \in S} x(r) u(r, s) y(s). \tag{26}$$

Here $R = \{1, 2, 3, 4\},\ S = \{1, 2, 3, 4\}$, where $1, 2, 3, 4$ denotes the four events $(z_0 = 0, a = 0), (z_0 = 0, a = 1), (z_0 = 1, a = 0), (z_0 = 1, a = 1)$. The matrix

$$u(r, s) = u_{t_1, t_2}(r, s) \tag{27}$$

[1]The variables x, y obtain different meanings, if one uses mixed strategies. That reduces the number of symbols. To avoid confusion, the utilities $U_a(x, y), V_a(x, y)$ are supplied with the subscript a. If the subscript a is absent then x, y defines the parameters z_0, w_0, a, b, t_1, t_2. Otherwise, x, y denotes mixed strategies.

is the winning of the first object, if the pure strategy denoted by the quadruple (t_1, t_2, r, s) is used. The matrix

$$v(r,s) = v_{t_1,t_2}(r,s) \tag{28}$$

is the winning of the second object, if the pure strategy (t_1, t_2, r, s) is used. Expanding expressions (25)

$$\begin{aligned} U_a(x,y) = {} & x(1)u(1,1)y(1) + x(2)u(2,1)y(1) + \\ & x(3)u(3,1)y(1) + x(4)u(4,1)y(1) + \\ & x(1)u(1,2)y(2) + x(2)u(2,2)y(2) + \\ & x(3)u(3,2)y(2) + x(4)u(4,2)y(2) + \\ & x(1)u(1,3)y(3) + x(2)u(2,3)y(3) + \\ & x(3)u(3,3)y(3) + x(4)u(4,3)y(3) + \\ & x(1)u(1,4)y(4) + x(2)u(2,4)y(4) + \\ & x(3)u(3,4)y(4) + x(4)u(4,4)y(4). \end{aligned} \tag{29}$$

Here winnings $u(r,s)$ are defined by expression (27). Expression (26) illustrates how the function $U_a(x,y)$ is defined. The function $V_a(x,y)$ is defined similarly.

One defines bimatrix games (25) and (26) for each pair of firing times $(t_1,\ t_2)$. Therefore, for fixed pair $(t_1,\ t_2)$, the same methods of search for the equilibrium points may be used as in the inspector model (**??**). Here winnings $u(r,s)$ and $v(r,s)$ depend on firing times t_1, t_2. This is the important difference from the inspector game. One updates bimatrix games while changing firing times.

Therefore, there are only two parameters t_1 and t_2 to be found by the fraud minimization (see section 4.1). One can do that by applying methods, such as the *Bayes*, to search for the pair (t_1, t_2) that minimizes the fraud.

4 Search for Equilibrium

4.1 Bimatrix Game Algorithm

Consider the bimatrix game (25) and (26) defined by fixing a pair $(t_1,\ t_2)$ of firing times. We search for the equilibrium of this game. Here the matrix is small thus the proportion of pure strategy equilibrium points will not be considerable [1]. Therefore we start by looking for such a mixed strategy that makes all the partner strategies equally profitable for him. That makes the partner fraud irrelevant. At fixed pair $(t_1,\ t_2)$ a mixed strategy $x(r)$ of the first object is "fraud irrelevant" if

$$\begin{aligned} \sum_{r\in R} x(r)v(r,s) = V,\ s \in S, \\ \sum_{r\in R} x(r) = 1,\ x(r) \geq 0. \end{aligned} \tag{30}$$

There are four linear inequalities, five linear equations and five variables: $V,\ x(r),\ r \in R$, because the set R has four elements.

A mixed strategy $y(s)$ of the second object is fraud irrelevant if

$$\sum_{s\ inS} u(r,s)y(s) = U,\ r \in R, \tag{31}$$

$$\sum_{s\in S} y(s) = 1.y(s) \geq 0.$$

These expressions include inequalities what is usual in the linear programming (LP). Applying LP, the variables U and V should be expesed as differences of two non-negative variables $U = u_1 - u_2$ and $V = v_1 - v_2$. Then from the expressions (30), (31) one obtains the LP problem:

$$\max_{x,y,u,v} (u_1 - u_2 + v_1 - v_2) \tag{32}$$

$$\sum_{j=1}^{m} u(i,j)y_j = u_1 - u_2,\ i = 1,...,m, \tag{33}$$

$$\sum_{i=1}^{m} v(i,j)x_i = v_1 - v_2,\ j = 1,...,m, \tag{34}$$

$$\sum_{j=1}^{m} y_j = 1, \tag{35}$$

$$\sum_{i=1}^{m} x_i = 1 \tag{36}$$

$$x_i \geq 0,\ y_j \geq 0,\ i,j = 1,...,m, u_1 \geq 0,\ u_2 \geq 0, v_1 \geq 0, v_2 \geq 0.. \tag{37}$$

Here $x = (x_1,...,x_m),\ y = (y_1,...,y_m),\ u = (u_1,u_2),\ v = (v_1,v_2)$.

Similar linear fraud irrelevant equations were considered in the inspection problem. Here the fraud irrelevant equations (33) and (34) depend on two unknown parameters t_1 and t_2. Therefore, linear problems (33) and (34) are just a part of general non-linear problem. We use longer notations, such as $U = U(t_1,t_2)$ and $V = V(t_1,t_2)$, to show that U,V depend on t_1,t_2.

If there is no solution of fraud irrelevant equations, we search for the pure equilibrium strategies using condition (**??**). The Strategy Elimination Algorithm (SE) is a good alternative to obtain equilibrium. A general but difficult way to find the equilibrium is to solve the bilinear problem (**??**) defining the necessary and sufficient equilibrium conditions.

Obtaining equilibrium in mixed strategies $x(r), y(s),\ r \in R,\ s \in S$ or in pure strategies r, s, one reduces the six-dimensional optimization problem to the two-dimensional one.

$$\min_{z\in B} \parallel z - T_t(z) \parallel^2 . \tag{38}$$

Here $z = (t_1^0, t_2^0)$, where (t_1^0, t_2^0) are the "contract" firing times. The operator T_t is defined by the fraud condition

$$t_1^1 = arg \max_{t_1} U(t_1, t_2^0),$$
$$t_2^1 = arg \max_{t_2} V(t_1^0, t_2). \tag{39}$$

Here $U(t_1, t_2^0)$ is the expected winning of the first object. It is the solution of the corresponding bimatrix game at fixed firing times t_1 and t_2^0. $V(t_1^0, t_2)$ is the expected winning of the second object defined as the solution of the bimatrix game at given firing times t_1^0 and t_2. For example, winnings $U(t_1, t_2^0)$ and $V(t_1^0, t_2)$ can be defined by LP problems (33)(34), or by the Direct Search Algorithm (DSA) or by Strategy Elimination Algorithm (SE) (see Section **??**).

The equilibrium is achieved, if the minimum (38) is zero. The minimization of (38) is the task for global optimization algorithms, including the Bayesian ones.

4.2 Matrix Game Algorithm

The bimatrix game algorithm can be applied to both the symmetric and the non-symmetric cases. In the symmetric case the utilities satisfy the zero-sum game condition $U = -V$. Here one obtains the equilibrium by solving the following linear programming problems

$$\max_x \ U \tag{40}$$
$$\sum_{i=1}^{m} x_i u_{ij} \geq U, \ j = 1, ..., m,$$
$$\sum_{i=1}^{m} x_i = 1, \ x_i \geq 0$$
$$U = u_1 - u_2, \ u_1 \geq 0, \ u_2 \geq 0,$$

and

$$\min_y \ V, \tag{41}$$
$$-\sum_{j=1}^{m} y_j u_{ij} \leq V, \ i = 1, ..., m,$$
$$\sum_{i=1}^{m} y_i = 1, \ y_i \geq 0$$
$$V = v_1 - v_2, \ v_1 \geq 0, \ v_2 \geq 0.$$

Here mini-max condition [9]

$$\min_y \ V = \max_x \ U \tag{42}$$

holds for all firing times t_1, t_2. Therefore, a search for the equilibrium is reduced to the condition

$$\min_{t_1,t_2}(U^2 + V^2). \tag{43}$$

The zero-sum equilibrium is reached when the hitting probabilities are equal to 0.5 for both players and their winnings are zero (see the one-dimensional example in the next section).

5 One-dimensional Example

Applying expressions (1)-(21) in one-dimensional space the trajectories are

$$z(t) = t, \tag{44}$$
$$w(\tau) = 2 - \tau, \tag{45}$$

and the hitting probability is

$$p(t) = 1 - d(t)/D. \tag{46}$$

Here $D \geq \max_t d(t)$. The control parameters are the firing times $x = t_1$ and $y = t_2$. Assuming that $D = 2$ and applying expression (22), the expected utility function of the first object is

$$U(x,y) = \begin{cases} x - (1-x), & \text{if } x < y, \\ -y + (1-y), & \text{if } x > y, \\ x - y, & \text{if } x = y. \end{cases} \tag{47}$$

The expected utility function of the second object follows from expression (23)

$$V(x,y) = \begin{cases} y - (1-y), & \text{if } x > y, \\ -x + (1-x), & \text{if } x < y, \\ y - x, & \text{if } y = x, \end{cases} . \tag{48}$$

The equilibrium is reached at the firing moment $x = y = 0.5$. The one-dimensional example helps to understand and to test the results of the two-dimensional case.

6 Economic Duel. Nash Model

By the "Economic Duel" we mean a dynamic case of the Nash model. In Chapter **??**, the static case was described.

There are m servers providing the same service

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

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

Suppose that arriving customers estimate the average waiting time as the relation of the number of waiting customers n_i to the server capacity x_i that is assumed to be equal to the running cost

$$\gamma_i = n_i/x_i$$

A customer goes to the server i, if

$$c_i \leq c_j,\ j = 1, ..., m,\ j \neq i,\ c_i \leq c_0.$$

A customer goes away, if

$$\min_i c_i > c_0,$$

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

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

To illustrate general ideas of the dynamic Nash model, we simplify the static one, first.

6.1 Simplified Nash Model. Static Case

We express the customer estimate of the expected waiting time as

$$\gamma_i = a_i/x_i,\ i = 1, 2. \tag{49}$$

Assume the customer balance condition.

$$a_i/x_i + y_i = q. \tag{50}$$

Here q is the customer balance factor. The balance condition follows from (**??**),(**??**), (**??**), if the number of customers is large. The condition is based on the observation that arriving customers prefer servers with lesser sum $y_i + \gamma_i$. From expression (**??**)

$$q \leq c. \tag{51}$$

From (50)

$$a_i = (q - y_i)x_i, \tag{52}$$

and

$$u_i = (q - y_i)x_i y_i - x_i. \tag{53}$$

Consider a monopolistic situation when only the ith server is there. In this case

$$a_i = a,\ if\ \ q \leq c \tag{54}$$

and

$$y_i + a/x_i \leq c. \tag{55}$$

Othervise, customers abandon services $a_i = 0$. The maximal monopolistic profit

$$\max_{x_i, y_i}(ay_i - x_i), \tag{56}$$

$$y_i + a/x_i \leq c. \tag{57}$$

Replacing inequality (57) by equality, we eliminate (57) and maximize the expression

$$max_{y_i}(ay_i - a/(c - y_i)). \tag{58}$$

Now one defines the optimal service price

$$y_i^* = c - 1, \tag{59}$$

the optimal service capacity,

$$x_i^* = a, \tag{60}$$

and the maximal monopolistic profit

$$u_i^* = a(c - 2). \tag{61}$$

6.2 Dynamic Nash Model

In the previous section, considering the static model, it was tacitly assumed that a server can operate steadily while making loss instead of profit. Now we consider a case of insolvency, too. Suppose that a server gets broken, if the accumulated losses exceed some credit threshold. Here the bankrupt server is eliminated and the remaining one assumes a monopolistic position.

Denote by $U_i(t)$ a profit accumulated by the ith server at a time t. Assume that

$$U_i(t) = \int_{t_0}^{t} u_i(t)dt, \tag{62}$$

where $u_i(t)$ defines a profit at a moment t

$$u_i(t) = a_i(t)y_i(t) - x_i(t). \tag{63}$$

Here $a_i(t)$ is a rate of customers of the ith server at a moment t. This rate is formally defined as a limit of the fraction

$$a_i(t) = \lim_{\delta \to 0} \frac{A_i(t + \delta) - A_i(t)}{\delta}, \tag{64}$$

where $A_i(t)$ is a total number of customers arrived during an interval $(0,t),\ t \leq T$. The zero denotes the starting time of the system. T is the end of the operation period, the "horizon". In real life $A_i(t)$ is defined at fixed moments, for example, minutes, hours, days e.t.c.. We consider it as a continuous parameter because we consider a continuous model.

Denote the bankruptcy threshold of an ith server by U_i^*. Denote a bankruptcy moment (if it happens) of the ith server by t_i^*. Then

$$a_i(t) \quad = \quad \begin{cases} a, & \text{if } t \geq t_j^*,\ t \leq t_i^*, i,j = 1,2,\ i \neq j, \\ 0, & \text{if } t \leq t_j^*,\ t \geq t_i^*\ \ i,j = 1,2,\ i \neq j. \end{cases} \tag{65}$$

Otherwise, customer rates $a_i(t)$ are defined by the customer balance condition (52)[2] .

From (61) the profit

$$u_i(t) \quad = \quad \begin{cases} u_i^*, & \text{if } t \geq t_j^*,\ t \leq t_i^*, i,j = 1,2,\ i \neq j, \\ 0, & \text{if } t \leq t_j^*,\ t \geq t_i^*\ \ i,j = 1,2,\ i \neq j. \end{cases} \tag{66}$$

Otherwise, the profit $u_i(t)$ is defined by (63).

Assume that both servers are planning their dynamic competition in advance. Each server i defines trajectories of the service price $y_i(t)$ and the service capacity $x_i(t)$. These trajectories show how the service prices and capacities depend on the time $t \in (0,T)$. Objectives are to maximize profits

$$U_i = U_i(T),\ i = 1,2. \tag{67}$$

Here $U_i(T)$ defines the accumulated profit (62) of ith server at the end of period $t = T$.

We illustrate the idea by a simple example. Assume that servers control initial values $y_i(0),\ x_i(0)$ and rates of change $b_{y_i},\ b_{x_i}$ of parameters $y_i(t)\ x_i(t)$. Then the trajectories are defined by differential equations

$$dy_i(t)/dt = b_{y_i} y_i(t), \tag{68}$$

and

$$dx_i(t)/dt = b_{x_i} x_i(t). \tag{69}$$

The corresponding solutions are

$$y_i(t) = y^i(0) \exp(b_{y_i} y_i(t)), \tag{70}$$

and

$$x_i(t) = x^i(0) \exp(b_{x_i} x_i(t)). \tag{71}$$

[2]This means that the customer balance follows the changes of parameters $y_i(t)$ and $x_i(t)$ almost without delay.

6.2.1 Search for Nash Equilibrium

The search is similar to that in the Nash model (see Section **??**). We fix the the initial values, the "Contract-Vector"
$z^0 = (x_i(0)^0, b^0_{x_i}, y_i(0)^0, b^0_{y_i},\ i = 1, ..., m)$. The transformed values, the "Fraud-Vector"
$z^1 = (x_i(0)^1, b^1_{x_i}, y_i(0)^1, b^1_{y_i},\ i = 1, 2)$, is obtained by maximizing the profits of each server i. The maximization is performed assuming that all partners $j \neq i$ honors the contract
$(x_i(0)^0, b^0_{x_j}, y_i(0)^0, b^0_{y_j},\ j = 1, 2\ j \neq i)$

$$(x_i(0)^1, b^1_{x_i}, y_i(0), b^1_{y_i}) =$$
$$arg \max_{x_i(0), b_{x_i}, y_i(0), b_{y_i}} U_i(x_i(0), b_{x_i}, y_i(0), b_{y_i}, x_j(0)^0, b^0_{x_j}, y_i(0)^0, b^0_{y_j},$$
$$j = 1, 2\ j \neq i),\ i = 1, 2. \tag{72}$$

Here the symbol U_i denotes the accumulated profit (67). Formally, condition (72) transforms the vector
$z^n = ((x_i(0)^n, b^n_{x_i}, y_i(0)^n, b^n_{y_i},\ i = 1, 2) \in B \subset R^8,\ n = 0, 1, 2, ...$ into the vector z^{n+1}. To make expressions shorter denote this transformation by T

$$z^{n+1} = T(z^n),\ n = 0, 1, 2, ... \tag{73}$$

One may obtain the equilibrium at the fixed point z^n, where

$$z^n = T(z^n). \tag{74}$$

The fixed point z^n exists, if the feasible set B is convex and all the profit functions (**??**) are convex [8]. We obtain the equilibrium directly by iterations (73) , if the transformation T is contracting. If not, then we minimize the square deviation

$$\min_{z \in B} \parallel z - T(z) \parallel^2 . \tag{75}$$

The equilibrium is achieved, if the minimum (75) is zero.

One makes the model more flexible by introducing second derivatives of parameters $y_i(t)$ and $x_i(t)$. The second derivatives represent accelerations. Assume that servers control the starting values $y_i(0),\ x_i(0)$, the rates $b_{y_i},\ b_{x_i}$ and accelerations $g_{y_i},\ g_{x_i}$ of parameters $y_i(t)\ x_i(t)$. Then trajectories are defined by differential equations

$$d^2 y_i(t)/dt^2 = g_{y_i} dy_i(t)/dt + b_{y_i} y_i(t), \tag{76}$$

and

$$d^2 x_i(t)/dt^2 = g_{x_i} dx_i(t)/dt + b_{x_i} x_i(t). \tag{77}$$

Here the vector z^n has twelve components.
$z^n = (x_i(0)^n, b^n_{x_i}, g^n_{x_i}, y_i(0)^n, b^n_{y_i}, g^n_{y_i}\ i = 1, 2) \in B \subset R^{12},\ n = 0, 1, 2, ...$ That

means high dimension of the fraud minimization problem (75). In addition, one needs to test the convexity of accumulated profits U_i defined by expressions (67) as functions of parameters
$x_i(0), b_{x_i}, y_i(0), b_{y_i},\ i = 1, 2$. If they are convex then one obtains equilibrium by condition (75). Otherwise, some specific algorithms should be designed. An example is the inspector problem.

7 Software Example

An interactive duel model is implemented using Java JDK1.2 [7]. To run the Java applet go to web-sites (see section **??**), start $GMJ2$ and select the task $DuelStarwars$ (see Figure **??**).
The task window, in Figure 1, shows the domain of firing times $0 \leq t_1 \leq 1,\ 0 \leq t_2 \leq 1$.

One selects the optimization method from the $Methods$ menu (see Figure **??**). The method $Bayes$ is selected (see Figure 2). The total number of iterations is set to forty. The number of initial iterations is set to five.

The optimization results are in Figure 3.

To start the analysis of results one touches the $Analysis$ button (see Figure **??**).
The $Convergence$ mode shows how the deviation from the equilibrium depends on the iteration number (see Figure 4).

7.1 Operating Duel

The duel starts by selecting $DuelStarwars$ mode in the $Analysis$ menu (see Figure **??**).
To run duel press the $Start$ button (see Figure 5).

One selects the operation mode in the operation window (see Figure 6). There are three operation modes:

- Person vs Person,
- Person vs Computer,
- Computer vs Computer.

Figure 6 shows the $Computer\ vs\ Computer$ mode. The
$Current\ situation$ field shows parameters z_0, w_0, a, b generated at random using probabilities defined by equilibrium conditions. The firing times t_1, t_2 are decided by these conditions, directly.

In the $Person\ vs\ Computer$ game the person fix parameters z_0, a, t_1 of the first object. A window to fix them is in the center of the top Figure 7.

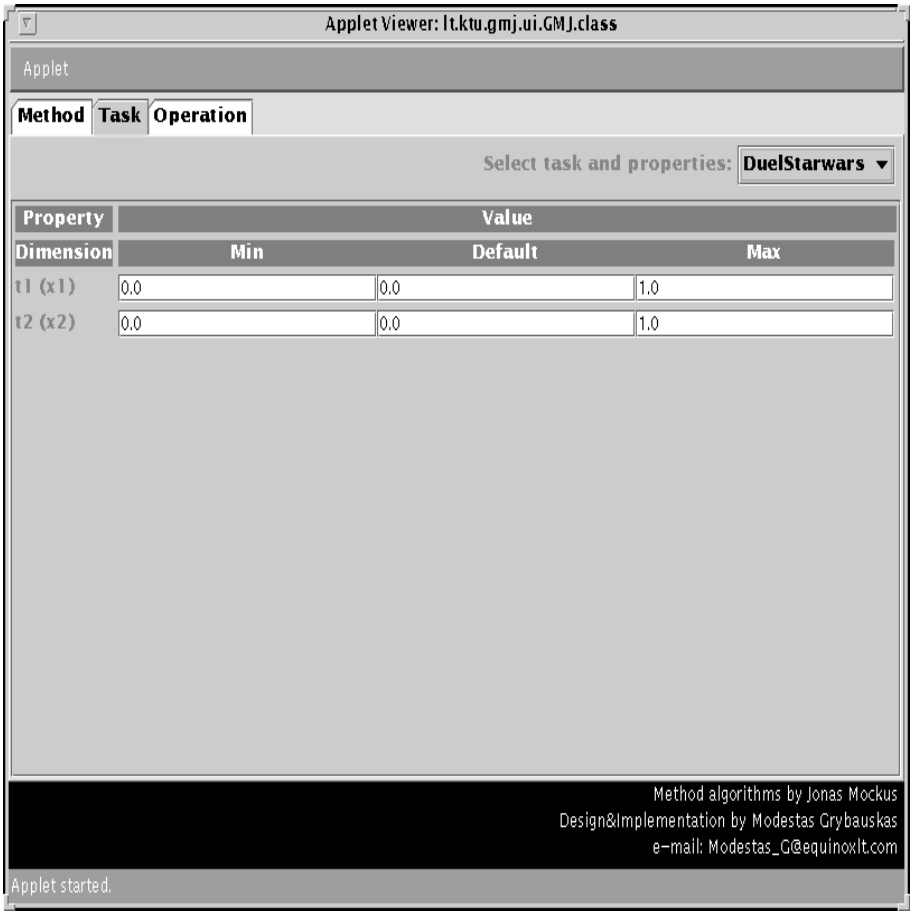

Figure 1: The duel task window.

Probabilities of parameters w_0, b of the second object and its firing time t_2 are defined by equilibrium conditions (35),(34), and (38). Parameters w_0, b of the second object, and the actual hitting, are defined by probabilities. Therefore, the outcome is random.

Figure 8 shows the input of the *Person vs Person* game. In this game parameters of both objects are fixed by persons using the input window. The outcome of this game is shown in a form similar to the *Person vs Computer* game (see the bottom Figure 7).

One sets the speed of moving objects in the *Speed* window (see Figureduelout1).

Figures 10 show *help* windows.

7.2 Future Developments

In the example, one searches for equilibrium in mixed strategies using the Irrelevant Fraud Algorithm (IF) which is described in section **??**. This algorithm solves linear programming problems (33) and (34). If no solution is found, one searches for pure equilibrium strategies using the Direct Search Algorithm (DS) (**??**). The Strategy Elimination Algorithm (SE) (see section **??**) helps, if both IF and DS fail.

The main future task is to implement flexible economic models. Another task is the implementation of second-order economic duel models. These tasks need additional theoretical investigations and software developments. Creation of entertaining graphical interfaces is important, too.

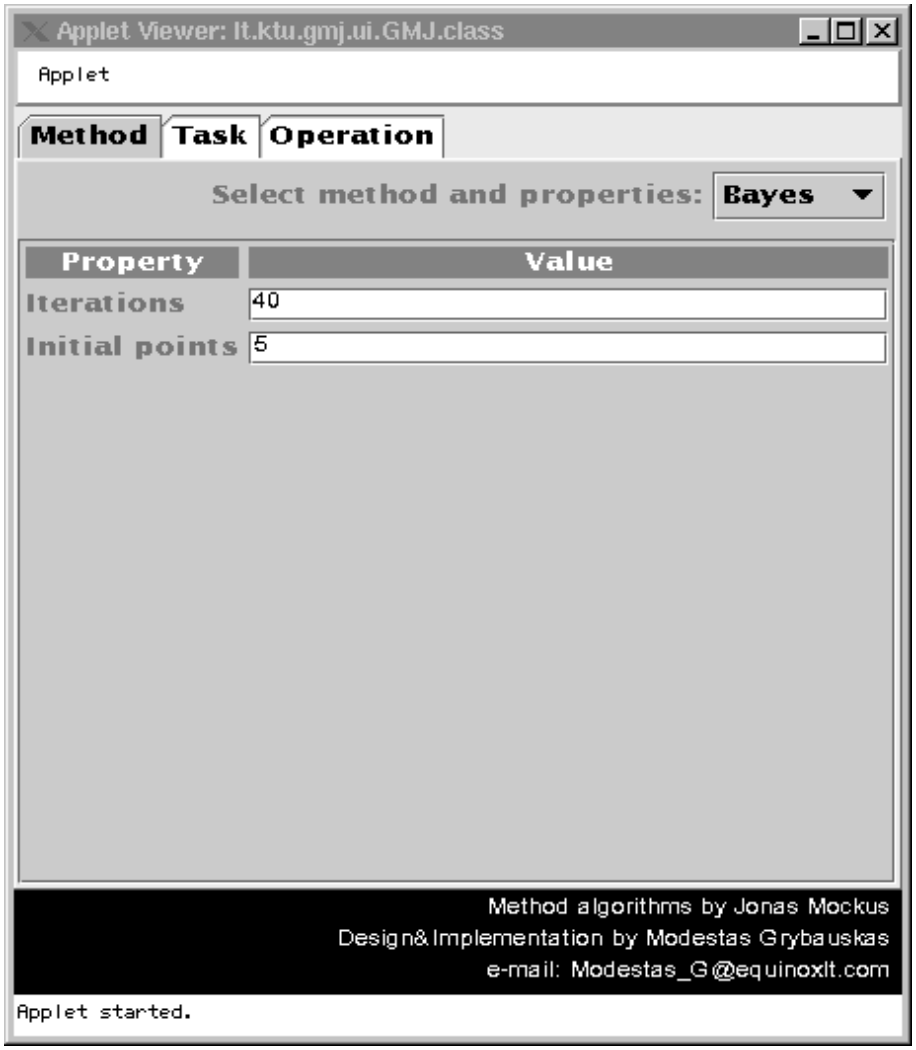

Figure 2: The Bayes method selection.

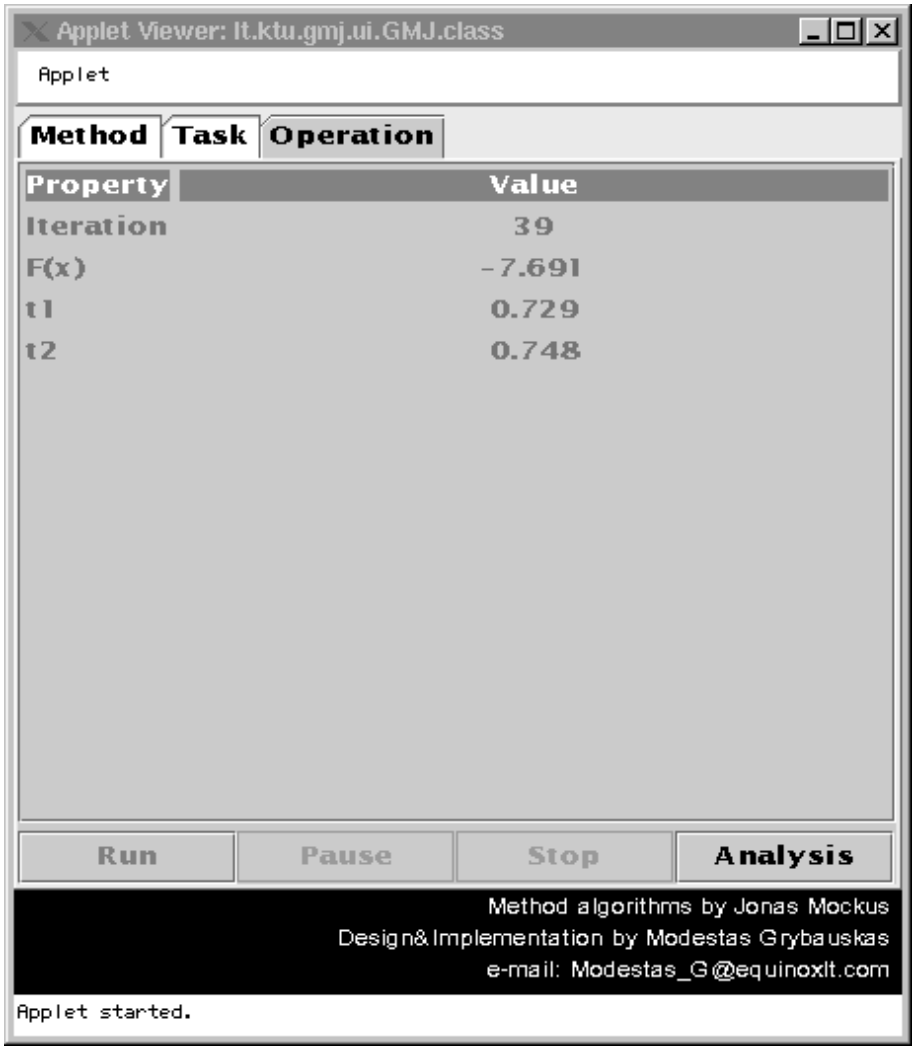

Figure 3: The results.

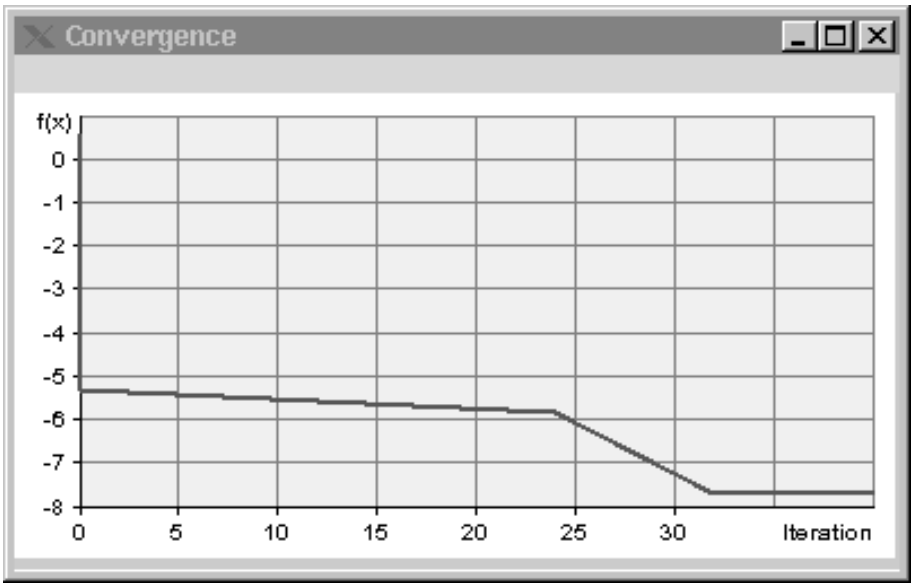

Figure 4: The convergence window.

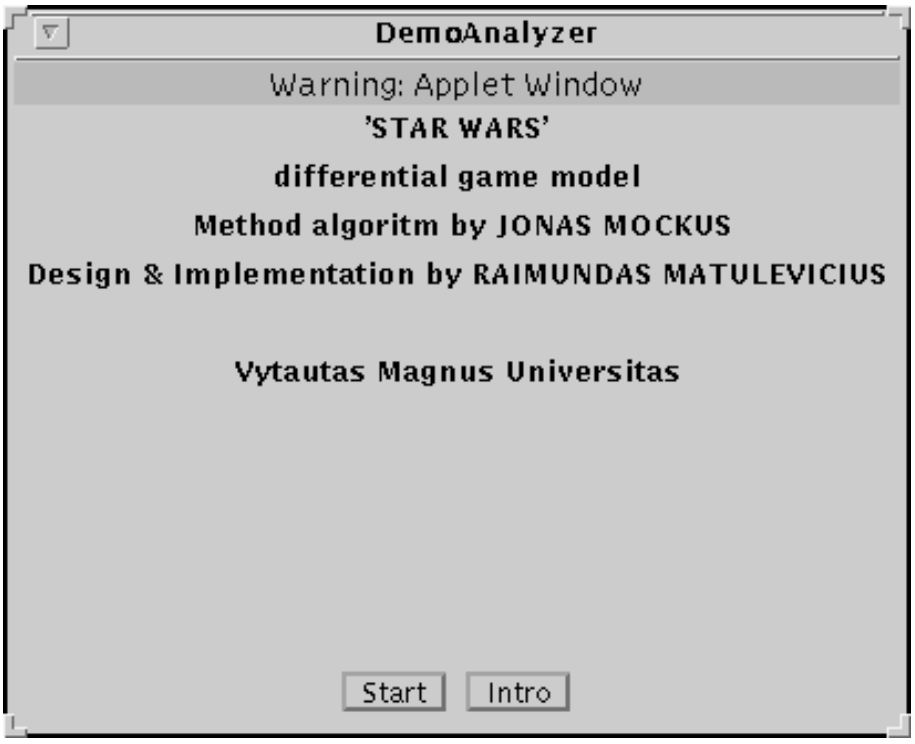

Figure 5: The starting window.

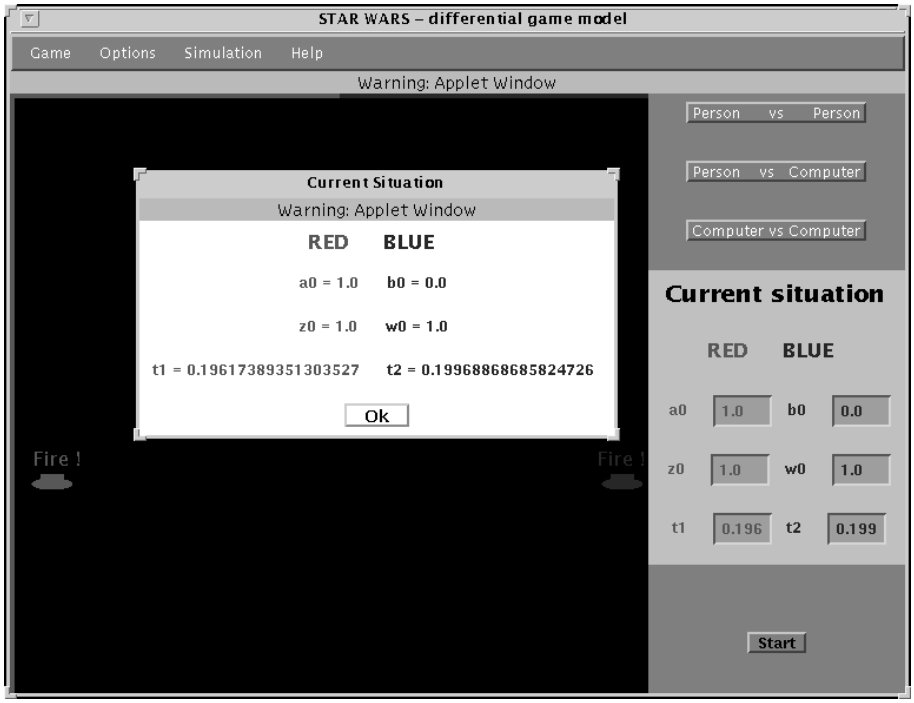

Figure 6: The operation window in the *Computer vs Computer* game.

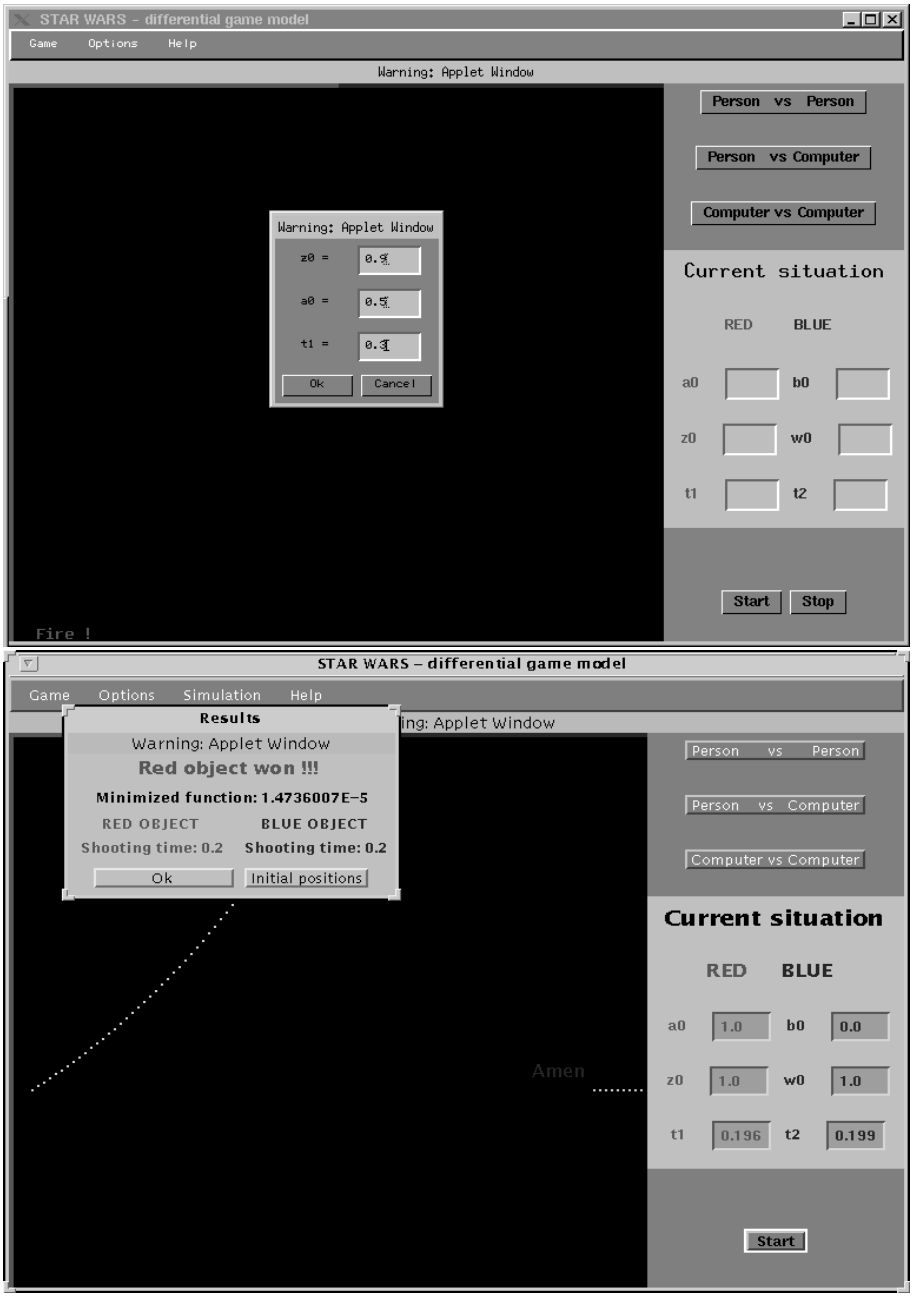

Figure 7: The input (top) and the outcome (bottom) of the *Person vs Computer* game.

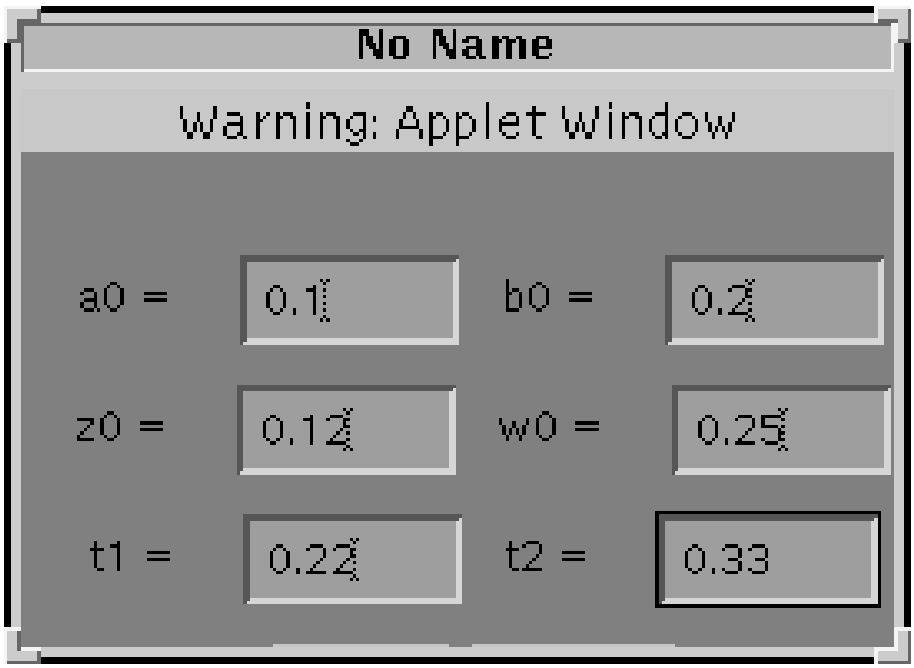

Figure 8: The input of the *Person vs Person* game.



Figure 9: The speed control window.

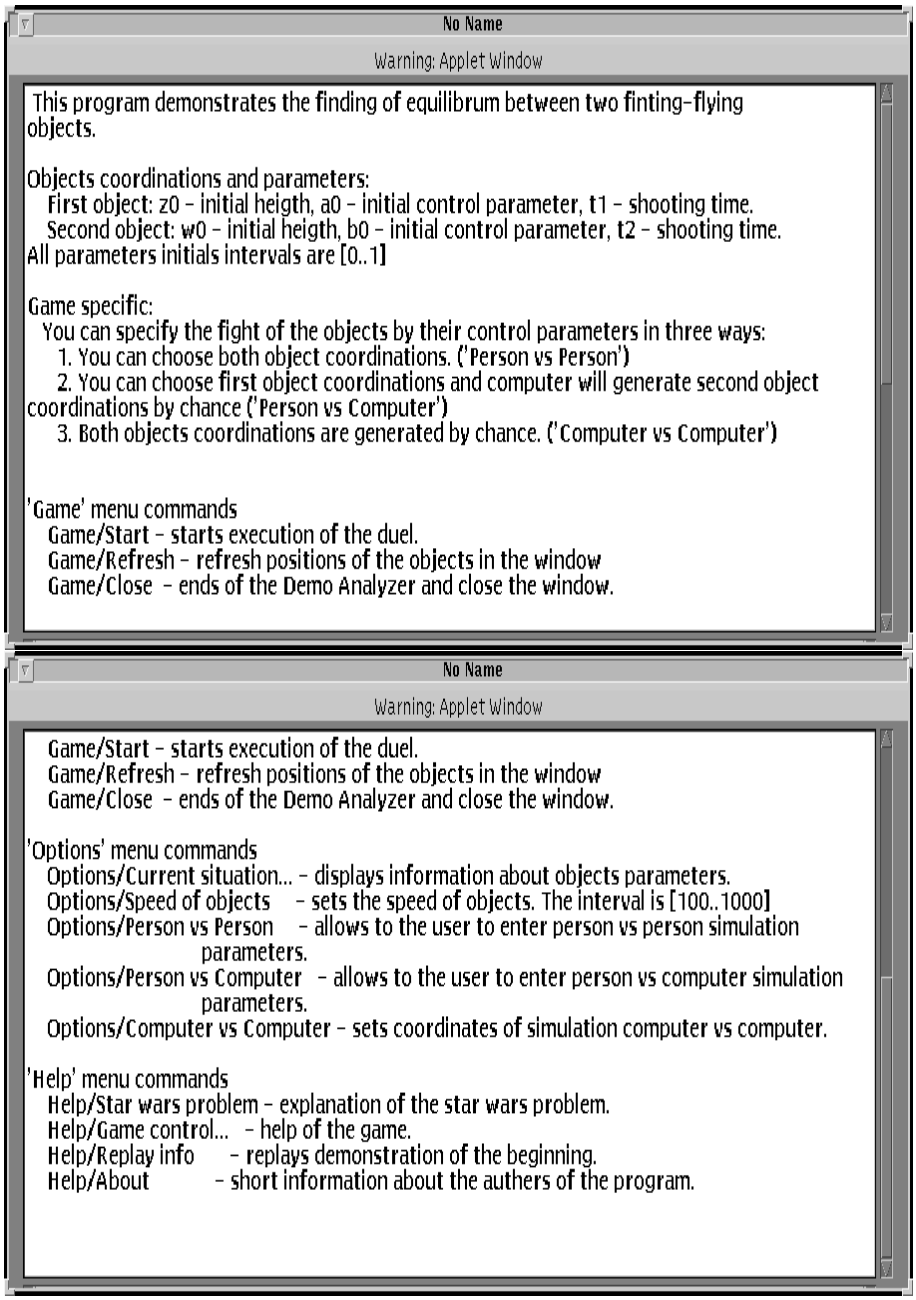

Figure 10: The *help* pages.

References

[1] F. Forgo, Jeno Szep, and Ferenc Szidarovszky. *Introduction to the Theory of Games*. Kluwer Academic Publishers, 1999.

[2] A. Friedman. *Dfferential Games*. Wiley-Interscience, New York, 1971.

[3] Efim Galperin and Quan Zheng. *Global Solutions in Optimal Control and Games*. Presses de l'Universite du Quebec a Montreal, Montreal, 1991.

[4] O. Hajek. *Pursuit Games*. Academic Press, New York, 1975.

[5] R. Isaacs. *Dfferential Games*. Wiley, New York, 1965.

[6] N. Krasovski. *Game Problems on the encounter motions*. Nauka, Moscow, 1970. (in Russian).

[7] R. Matulevicius. Differential game model. Technical report, Vytautas Magnus University, Faculty of Informatics, Kaunas, Lithuania, 1999.

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

[9] G. Owen. *Game Theory*. W.B. Saunders, Philadelphia, PA, 1968.

[10] L. Petrosyan. *Cooperative Dfferential Games*. Tomsk University Publishing, Tomsk, Russia, 1985. (in Russian).

[11] L. Petrosyan. *Dfferential Games of Pursuit*. Worls Scientific Publishing Co., River Edge, NJ, 1993.

[12] E. Y. Rodin. A pursuit-evasion bibliography. *International Journal of Computers and Mathematics with Applications*, 13:275–340, 1987.

[13] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, New Jersey, 1953.