

UDC 519.832.3

V. V. ROMANUKE  
Khmelnytskyy National University

## FAST SOLUTION OF THE DISCRETE NOISELESS DUEL WITH THE NONLINEAR SCALE ON THE LINEAR ACCURACY FUNCTIONS

*There has been defined the discrete noiseless duel with the skewsymmetric kernel on the finite nonlinearly scaled subset of the unit square. The program for fast getting the defined duel solution has been embodied.*

*Определено дискретную бесшумную дуэль с кососимметрическим ядром на конечном нелинейно масштабированном подмножестве единичного квадрата. Воплощено программу для быстрого получения решения определённой дуэли.*

*Key words:* discrete noiseless duel, skewsymmetric kernel, unit square, MATLAB, accuracy functions.

### The problem in a general view

The simplest math model of resolving the noncooperative conflict is the antagonistic game. The time selection antagonistic games are really versatile to use them in decision making on social-economic and bio-ecological events [1]. Those duels usually are defined as infinite games, though in practice a player mostly has the finite number of its alternatives. However, the finite alternatives duel games had been investigated in less, than the class of the infinite ones. Thus the actuality of the discrete duel investigation is the highest. The noiseless duels as a class of the time selection antagonistic games are crucial because of their fundamentality, and their sampling action gives the decision making models, allowing to practice the found optimal alternatives. Besides, this sampling should regard to that the duel end is closer the importance of the strategical alternative is higher.

### The near and last papers on the problem direction

Some narrowed classes of the noiseless duels had been solved or, at least, considered in [2, 3]. The corresponding discrete noiseless duels, having appeared in [4], are not highlighted. That might be caused with the reference to their easy sampling from the infinite or continuous duels, although the sampling of the continuous solution does not drive to the discrete noiseless duel solution. But naturally, that the sampled infinite duel is a matrix game, and its solution may be obtained just with some math software, working operatively in the linear programming way. And the importance of the longer time strategies must be built in there also.

### The task assignments and the paper aim formulation

Firstly the kernel

$$K(x, y) = x - y + xy \operatorname{sign}(y - x) \quad (1)$$

of the noiseless duel on the unit square

$$X \times Y = [0; 1] \times [0; 1] \quad (2)$$

should be sampled generally in such a way, that the importance of the more lingered strategies would be greater, where  $x \in X$  is a pure strategy of the first player, and  $y \in Y$  is a pure strategy of the second. Not incidentally, but here the accuracy functions are taken linear, as the sampled strategy importance will be only in its specified neighborhood with the other sampled strategies. That is the density of the pure strategies on  $X$  and  $Y$ , when the time is going on, must be not decreasing. Then the obtained discrete noiseless duel, being symmetric, will be solved within the MATLAB [5]. Thus there will be obtained the fast solution of the discrete noiseless duel (1) with the nonlinear scale on the linear accuracy functions, reflecting the significance of the more lingered alternatives, and the aim of this paper will be achieved.

### The noiseless duel (1) nonlinear scale sampling

The skew-symmetry of the game with the kernel (1) is obvious:

$$K(x, y) = -K(y, x).$$

This means that the optimal strategies of the players in the noiseless duel (1) are identical, and, as corollary, the

game value is zero. The sampling on the unit segment  $[0; 1]$  scale will be the same for both players, letting them shoot in the very beginning and in the end of the duel, that is the pure strategies 0 and 1 are included into the sampled set of the pure strategies of the player.

May the probability  $d_j$  be the distance between the left adjacent (previous) pure strategy and the present, where  $d_j \notin \{0, 1\}$  and  $j = \overline{1, N-2}$  by the  $N$  available pure strategies at the player. Then the corresponding vector

$$\mathbf{D} = [d_1 \ d_2 \ \dots \ d_{N-3} \ d_{N-2}] \quad (3)$$

of distances should satisfy the conditions in the following:

$$\begin{aligned} & [d_1 \ d_2 \ \dots \ d_{N-3} \ d_{N-2}] \in \\ & \in \left\{ \mathbf{D} \in \mathbb{R}^{N-2} \mid d_j \in (0; 1) \forall j = \overline{1, N-2}, d_j \geq d_{j+1} \forall j = \overline{1, N-3}, 1 - \sum_{j=1}^{N-2} d_j \leq d_{N-2} \right\}. \end{aligned} \quad (4)$$

Accordingly to the distances vector (4), in the discrete noiseless duel with the nonlinear scale on the linear accuracy functions, the first player has its set

$$X_{\mathbf{D}} = \{x_k\}_{k=1}^N = \{0, 1\} \cup \left\{ \sum_{j=1}^{k-1} d_j \right\}_{k=2}^{N-1} \quad (5)$$

of the pure strategies, and the second player has the set

$$Y_{\mathbf{D}} = \{y_k\}_{k=1}^N = \{0, 1\} \cup \left\{ \sum_{j=1}^{k-1} d_j \right\}_{k=2}^{N-1} \quad (6)$$

of its pure strategies. Thus such a duel is defined on the subset

$$\begin{aligned} X_{\mathbf{D}} \times Y_{\mathbf{D}} &= \{x_k\}_{k=1}^N \times \{y_l\}_{l=1}^N = \\ &= \left\{ \begin{bmatrix} x_k & y_l \end{bmatrix} \right\}_{k=1}^N = \\ &= \left\{ \{0, 1\} \cup \left\{ \sum_{j=1}^{k-1} d_j \right\}_{k=2}^{N-1} \right\} \times \left\{ \{0, 1\} \cup \left\{ \sum_{i=1}^{l-1} d_i \right\}_{l=2}^{N-1} \right\} \subset \\ &\subset X \times Y = [0; 1] \times [0; 1] \end{aligned} \quad (7)$$

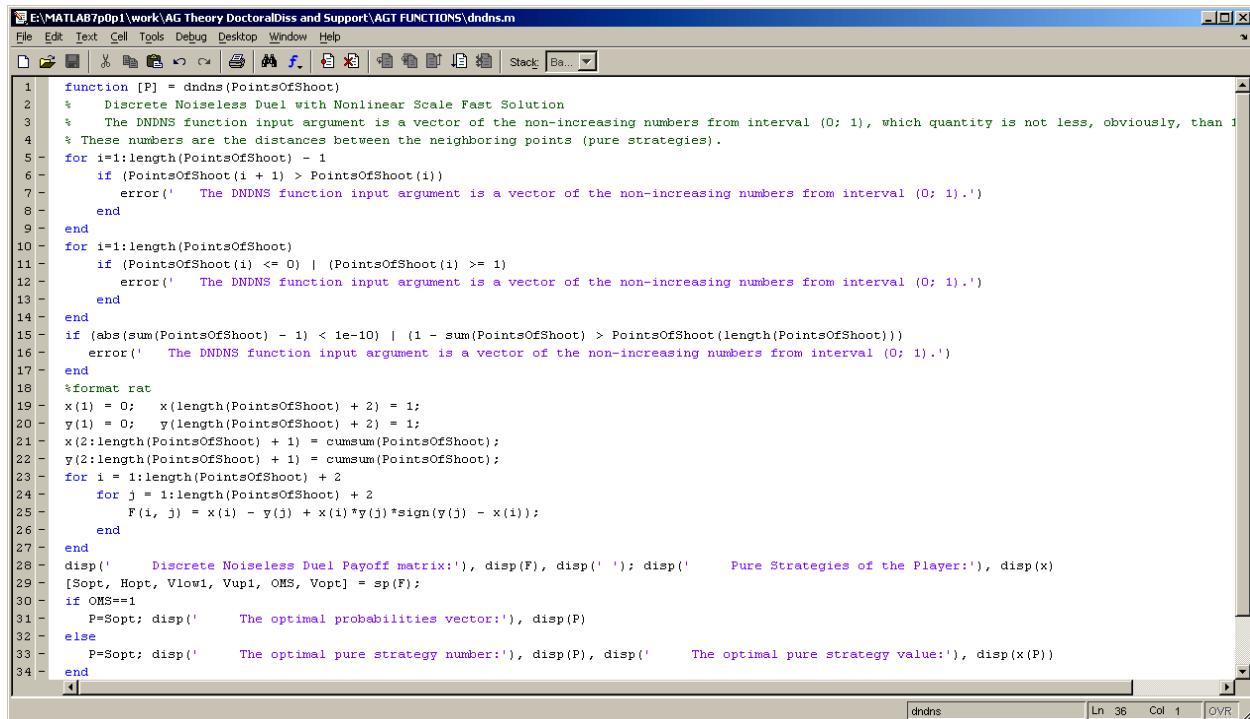
of the unit square (2), satisfying the conditions in (4). And here, for the not decreasing density of the sorted pure strategies in (5) and (6), the importance of the more lingered strategies is greater, as it has been assigned.

### The MATLAB fast solution code

For knowing the sampled sets (5) and (6), the sampled kernel (1) may be analytically represented as the matrix  $\mathbf{F} = (f_{ij})_{N \times N}$  with the elements

$$\begin{aligned} r_{ij} &= K(x_i, y_j) = \\ &= x_i - y_j + x_i y_j \operatorname{sign}(y_j - x_i), \quad i = \overline{1, N}, \quad j = \overline{1, N}. \end{aligned} \quad (8)$$

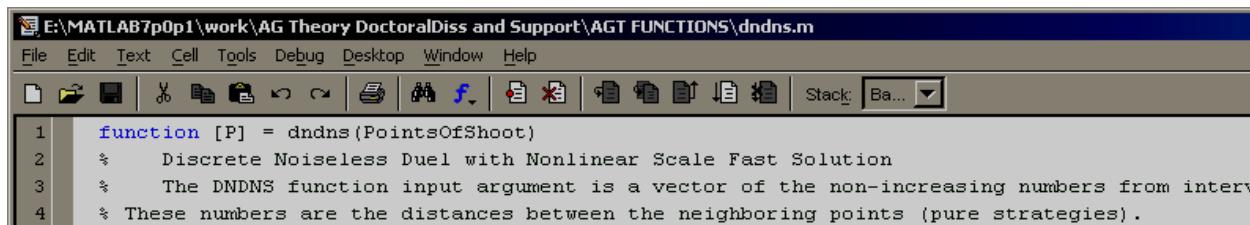
To solve this  $(f_{ij})_{N \times N}$ -game there has been constructed the MATLAB code `dndns` as the independent program module (figure 1), having the distances vector (3) as the input. Some examples, demonstrating the work of the module `dndns`, are in the figures 2 — 4.



```

1 function [P] = dndns(PointsOfShoot)
2 % Discrete Noiseless Duel with Nonlinear Scale Fast Solution
3 % The DNDNS function input argument is a vector of the non-increasing numbers from interval (0; 1), which quantity is not less, obviously, than 1
4 % These numbers are the distances between the neighboring points (pure strategies).
5 for i=1:length(PointsOfShoot) - 1
6     if (PointsOfShoot(i + 1) > PointsOfShoot(i))
7         error(' The DNDNS function input argument is a vector of the non-increasing numbers from interval (0; 1).')
8     end
9 end
10 for i=1:length(PointsOfShoot)
11     if (PointsOfShoot(i) <= 0) | (PointsOfShoot(i) >= 1)
12         error(' The DNDNS function input argument is a vector of the non-increasing numbers from interval (0; 1).')
13     end
14 end
15 if (abs(sum(PointsOfShoot) - 1) < 1e-10) | (1 - sum(PointsOfShoot) > PointsOfShoot(length(PointsOfShoot)))
16     error(' The DNDNS function input argument is a vector of the non-increasing numbers from interval (0; 1).')
17 end
18 %format rat
19 x(1) = 0; x(length(PointsOfShoot) + 2) = 1;
20 y(1) = 0; y(length(PointsOfShoot) + 2) = 1;
21 x(2:length(PointsOfShoot) + 1) = cumsum(PointsOfShoot);
22 y(2:length(PointsOfShoot) + 1) = cumsum(PointsOfShoot);
23 for i = 1:length(PointsOfShoot) + 2
24     for j = 1:length(PointsOfShoot) + 2
25         F(i, j) = x(i) - y(j) + x(i)*y(j)*sign(y(j)) - x(i);
26     end
27 end
28 disp(' Discrete Noiseless Duel Payoff matrix:'), disp(F), disp(' Pure Strategies of the Player:'), disp(x)
29 [Sopt, Hopt, Vlow1, Vup1, OMS, Vopt] = sp(F);
30 if OMS==1
31     P=Sopt; disp(' The optimal probabilities vector:'), disp(P)
32 else
33     P=Sopt; disp(' The optimal pure strategy number:'), disp(P), disp(' The optimal pure strategy value:'), disp(x(P))
34 end

```

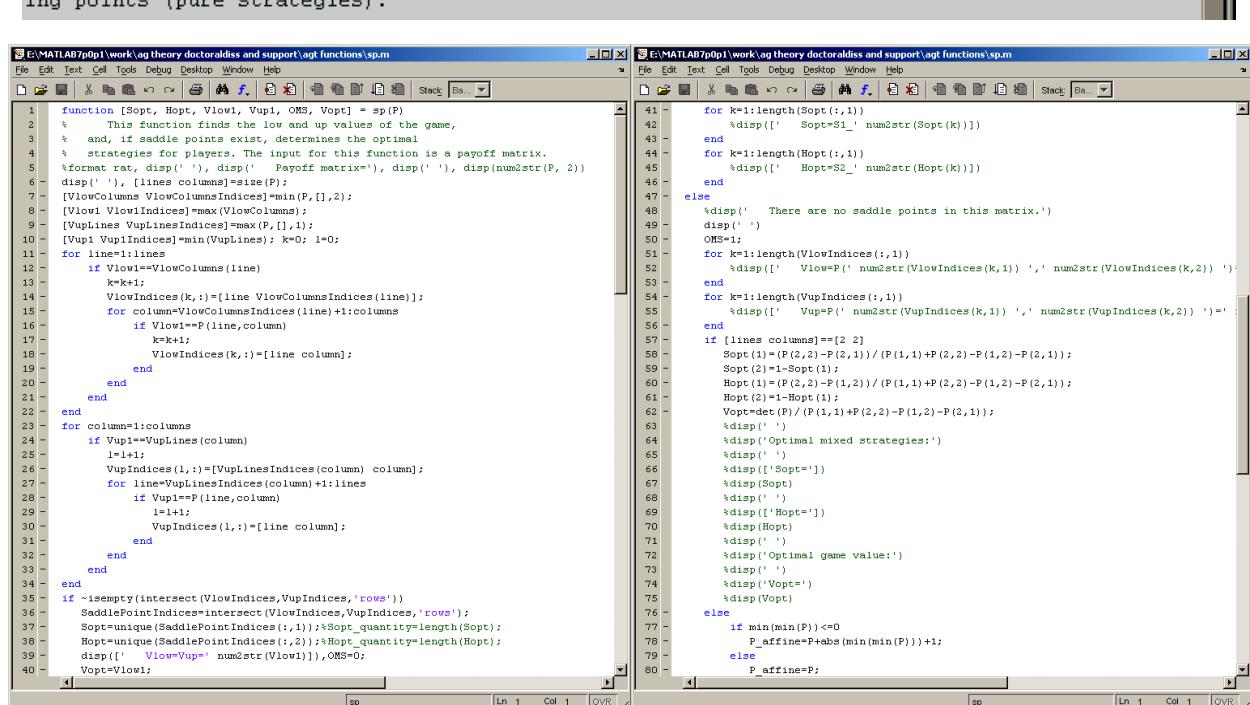


```

1 function [P] = dndns(PointsOfShoot)
2 % Discrete Noiseless Duel with Nonlinear Scale Fast Solution
3 % The DNDNS function input argument is a vector of the non-increasing numbers from interval (0; 1), which quantity is not less, obviously, than 1
4 % These numbers are the distances between the neighboring points (pure strategies).

```

**Solution**  
the non-increasing numbers from interval (0; 1), which quantity is not less, obviously, than 1  
ing points (pure strategies).



```

1 function [Sopt, Hopt, Vlow1, Vup1, OMS, Vopt] = sp(P)
2 % This function finds the low and up values of the game,
3 % and, if saddle points exist, determines the optimal
4 % strategies for players. The input for this function is a payoff matrix.
5 %format rat, disp(' Payoff matrix:'), disp(''), disp(num2str(P, 2))
6 disp(''), [lines columns]=size(P);
7 [VlowColumns VlowLinesIndices]=min(P,[],2);
8 [VupColumns VupLinesIndices]=max(P,[],1);
9 [Vup1 VupIndices]=min(VupLinesIndices); k=0; l=0;
10 for line=1:lines
11     if Vlow==VlowColumns(line)
12         k=k+1;
13         VlowIndices(k,:)=(line VlowLinesIndices(line));
14         for column=VlowColumnsIndices(line)+1:columns
15             if Vlow==P(line,column)
16                 k=k+1;
17                 VlowIndices(k,:)=line column;
18             end
19         end
20     end
21 end
22 for column=1:columns
23     if Vup1==VupLines(column)
24         l=l+1;
25         VupIndices(l,:)=VupLinesIndices(column);
26         for line=VupLinesIndices(column)+1:lines
27             if Vup1==P(line,column)
28                 l=l+1;
29                 VupIndices(l,:)=line column;
30             end
31         end
32     end
33 end
34 if ~isempty(intersect(VlowIndices,VupIndices,'rows'))
35     SaddlePointIndices=intersect(VlowIndices,VupIndices,'rows');
36     Sopt=unique(SaddlePointIndices(:,1));%Sopt_quantity=length(Sopt);
37     Hopt=unique(SaddlePointIndices(:,2));%Hopt_quantity=length(Hopt);
38     disp([' Vlow=Vup=' num2str(Vlow1)]), OMS=0;
39     Vopt=Vlow1;
40 end

```

```

41 for k=1:length(Sopt,:)
42     %disp([' Sopt=S1_ ' num2str(Sopt(k))])
43 end
44 for k=1:length(Hopt,:)
45     %disp([' Hopt=S2_ ' num2str(Hopt(k))])
46 end
47 else
48     %disp(' There are no saddle points in this matrix.')
49     %disp(' ')
50     OMS=1;
51     for k=1:length(VlowIndices(:,1))
52         %disp([' Vlow=P(' num2str(VlowIndices(k,1)) ', num2str(VlowIndices(k,2)) ')'])
53     end
54     for k=1:length(VupIndices(:,1))
55         %disp([' Vup=P(' num2str(VupIndices(k,1)) ', num2str(VupIndices(k,2)) ')'])
56     end
57     if [lines columns]==[2 2]
58         Sopt(1)=(P(2,2)-P(2,1))/(P(1,1)+P(2,2)-P(1,2)-P(2,1));
59         Sopt(2)=1-Sopt(1);
60         Hopt(1)=(P(2,2)-P(1,2))/(P(1,1)+P(2,2)-P(1,2)-P(2,1));
61         Hopt(2)=1-Hopt(1);
62         Vopt=det(P)/(P(1,1)+P(2,2)-P(1,2)-P(2,1));
63         %disp(' ')
64         %disp('Optimal mixed strategies:')
65         %disp(' ')
66         %disp(['Sopt='])
67         %disp(Sopt)
68         %disp(' ')
69         %disp(['Hopt='])
70         %disp(Hopt)
71         %disp(' ')
72         %disp(' ')
73         %disp(['Vopt='])
74         %disp(Vopt)
75     else
76         if min(min(P))<=0
77             P_affine=P+abs(min(min(P)))+1;
78         else
79             P_affine=P;
80         end
81     end

```

Figure 1. The MATLAB code of the module dndns for getting the fast solution [5] of discrete noiseless duel with the nonlinear scale on the linear accuracy functions

```

>> P = dndns([0.4 0.3 0.2])
Discrete Noiseless Duel Payoff matrix:
  0      -2/5      -7/10      -9/10      -1
  2/5      0      -1/50      -7/50      -1/5
  7/10      1/50      0      43/100      2/5
  9/10      7/50      -43/100      0      4/5
  1      1/5      -2/5      -4/5      0

Pure Strategies of the Player:
  0      2/5      7/10      9/10      1

Vlow=Vup=0
The optimal pure strategy number:
  3
The optimal pure strategy value:
  7/10

```

Figure 2. The solution in the pure strategies  $\{x_3, y_3\} = \left\{\frac{7}{10}, \frac{7}{10}\right\}$

for the  $(f_{ij})_{5 \times 5}$ -game,

consisting in the three distances between the neighboring points,

generating the pure strategies set  $\left\{0, \frac{2}{5}, \frac{7}{10}, \frac{9}{10}, 1\right\}$  of the player

```

>> P = dndns([0.4 0.3 0.2 0.06])
Discrete Noiseless Duel Payoff matrix:
  0      -2/5      -7/10      -9/10      -24/25      -1
  2/5      0      -1/50      -7/50      -22/125      -1/5
  7/10      1/50      0      43/100      103/250      2/5
  9/10      7/50      -43/100      0      201/250      4/5
  24/25      22/125      -103/250      -201/250      0      23/25
  1      1/5      -2/5      -4/5      -23/25      0

Pure Strategies of the Player:
  0      2/5      7/10      9/10      24/25      1

Vlow=Vup=0
The optimal pure strategy number:
  3
The optimal pure strategy value:
  7/10

```

Figure 3. The solution in the pure strategies  $\{x_3, y_3\} = \left\{\frac{7}{10}, \frac{7}{10}\right\}$

for the  $(f_{ij})_{6 \times 6}$ -game,

consisting in the four distances between the neighboring points,

generating the pure strategies set  $\left\{0, \frac{2}{5}, \frac{7}{10}, \frac{9}{10}, \frac{24}{25}, 1\right\}$  of the player

```

>> P = dndns([0.4 0.3 0.2 0.06 0.01])
?? Error using ==> dndns
    The DNDNS function input argument is a vector of the non-increasing numbers from interval (0; 1).

>> P = dndns([0.4 0.3 0.2 0.06 0.03])
    Discrete Noiseless Duel Payoff matrix:
      0          -2/5        -7/10       -9/10       -24/25      -99/100      -1
      2/5          0         -1/50       -7/50       -22/125     -97/500      -1/5
      7/10         1/50        0        43/100      103/250     403/1000      2/5
      9/10         7/50       -43/100       0        201/250     801/1000      4/5
      24/25        22/125     -103/250     -201/250       0        2301/2500     23/25
      99/100       97/500     -403/1000     -801/1000     -2301/2500       0        49/50
      1           1/5         -2/5        -4/5        -23/25      -49/50        0

    Pure Strategies of the Player:
      0          2/5         7/10       9/10       24/25      99/100      1

Vlow=Vup=0
    The optimal pure strategy number:
      3
    The optimal pure strategy value:
      7/10

```

Figure 4. The erroneous inputting of the vector (3),

and the solution in the pure strategies  $\{x_3, y_3\} = \left\{\frac{7}{10}, \frac{7}{10}\right\}$

for the  $(f_{ij})_{7 \times 7}$ -game

```

>> P = dndns([0.25 0.2 0.2 0.15 0.1])
    Discrete Noiseless Duel Payoff matrix:
      0          -1/4        -9/20       -13/20      -4/5       -9/10      -1
      1/4          0         -7/80       -19/80      -7/20      -17/40      -1/2
      9/20         7/80        0        37/400      1/100      -9/200      -1/10
      13/20        19/80      -37/400       0        37/100      67/200      3/10
      4/5           7/20      -1/100      -37/100       0        31/50      3/5
      9/10          17/40      9/200      -67/200     -31/50       0        4/5
      1            1/2         1/10      -3/10      -3/5       -4/5        0

    Pure Strategies of the Player:
      0          1/4         9/20       13/20      4/5       9/10      1

    The optimal probabilities vector:
      0            0        60/71        0        10/71        0        1/71
P =
      0            0        60/71        0        10/71        0        1/71
>>

```

Figure 5. The solution in the mixed strategies for another  $(f_{ij})_{7 \times 7}$ -game

By the way, in the module dndns the solution is implied to be only the optimal strategy of the player, and this strategy, being pure or mixed, is returned into the assigned variable for further processing.

## The conclusion and outlook for further programming investigation

The formulated discrete noiseless duel with the nonlinear scale on the linear accuracy functions has been conceived for modeling the conflict events, where in the course of time the player has more chances to shoot, that is the longer wait the more important pure strategy. The fast solution of this antagonistic game has been realized [6 — 11] thanking to the authorized MATLAB code dnddns, using within also the authorized MATLAB code sp [5, 12]. The duel solution may be saved if needed and transferred to other math applications or databases for processing. The further programming investigation should be directed to the discrete noiseless duel sophistication, where there will be more than just the single bullet for the shot. Moreover, there must be explored the case with nonlinear accuracy functions  $a_1(x)$  and  $a_2(y)$ , being, in general, the monotonous nondecreasing functions with the corresponding edge conditions, under which in the duel beginning the accuracy functions  $a_1(0)=a_2(0)=0$ , and in the duel end the accuracy functions  $a_1(1)=a_2(1)=1$ .

## References

1. Вороб'єв Н. Н. Теория игр для экономистов-кибернетиков / Вороб'єв Н. Н. — М. : Наука, Главная редакция физико-математической литературы, 1985. — 272 с.
2. Теория игр : [учеб. пособие для ун-тов] / Петросян Л. А., Зенкевич Н. А., Семина Е. А. — М. : Высшая школа, Книжный дом “Университет”, 1998. — 304 с. : ил.
3. Оуэн Г. Теория игр / Оуэн Г. ; [пер. с англ.]. — 2-е изд. — М. : Едиториал УРСС, 2004. — 216 с.
4. Романюк В. В. Дискретна прогресуюча безшумна дуель з кососиметричним ядром на кінцевій решітці одиничного квадрату з лінійними функціями влучності / В. В. Романюк // Інформаціонно-вичислювальні технології та їх застосування : збірник статей XI Міжнародної науково-техніческої конференції. — Пенза : РІО ПГСХА, 2009. — С. 8 — 16.
5. Романюк В. В. Разрешение системы преследователь — добыча для экспоненциальной вероятности поражения добычи преследователем / В. В. Романюк // Вестник НТУ “ХПИ”. Тематический выпуск: Информатика и моделирование. — Харьков : НТУ “ХПИ”, 2009. — № 13. — С. 138 — 149.
6. Романюк В. В. Моделювання реалізації оптимальних змішаних стратегій в антагоністичній грі з двома чистими стратегіями в кожного з гравців / В. В. Романюк // Наукові вісті НТУУ “КПІ”. — 2007. — № 3. — С. 74 — 77.
7. Романюк В. В. Метод реалізації принципу оптимальності у матричних іграх без сідової точки / В. В. Романюк // Вісник НТУ “ХПІ”. Тематичний випуск: Інформатика та моделювання. — Харків : НТУ “ХПІ”, 2008. — № 49. — С. 146 — 154.
8. Романюк В. В. Метод реалізації оптимальних змішаних стратегій у матричній грі з пустою множиною сідлових точок у чистих стратегіях з невідомою кількістю партій гри / В. В. Романюк // Вісник Хмельницького національного університету. Технічні науки. — 2009. — № 2. — С. 224 — 229.
9. Романюк В. В. Метод реалізації оптимальних змішаних стратегій у матричній грі з порожньою множиною сідлових точок у чистих стратегіях з відомою кількістю партій гри / В. В. Романюк // Наукові вісті НТУУ “КПІ”. — 2009. — № 2. — С. 45 — 52.
10. Romanuke V. V. Method of practicing the optimal mixed strategy with innumerable set in its spectrum by unknown number of plays / V. V. Romanuke // Measuring and Computing Devices in Technological Processes. — 2008. — № 2. — P. 196 — 203.
11. Романюк В. В. Метод реалізації оптимальних змішаних стратегій в антагоністичній грі, де гравець володіє незліченною множиною чистих стратегій, при відомій кількості партій гри / В. В. Романюк // Вісник Хмельницького національного університету. Технічні науки. — 2009. — № 5. — С. 130 — 142.
12. Романюк В. В. Моделирование выхода на рынок двух конкурирующих предприятий с помощью игровой бесшумной дуэли в MATLAB 7.0.1 / В. В. Романюк // Вісник Хмельницького національного університету. Економічні науки. — 2009. — № 3. — Т. 2. — С. 233 — 238.

Надійшла 16.10.2010