

M.K. BAGHINYAN

**A SOLUTION METHOD FOR NONE COOPERATIVE MULTI-PARAMETRIC GAME MODELS BASED ON DIFFERENTIAL TRANSFORMS**

A new approach for solving none cooperative game models with multi-parametric coefficient dependencies is proposed. The Solution is based on the differential transform method of Pukhov. The proposed algorithm is tested for multistep ship collision determination game.

**Keywords:** multistep game, multi-parametric game model, game theory, simplex transform, differential transform.

The Consumption of limited resources, minimization of costs, maximization of profit, as well as the market demand leads to the necessity of researching multiple variations of different task parameters for the optimal solution. In practice, these changes mostly appear as multi-parametric dependencies. During the recent years, interest in multi-parametric game models rises too. Unfortunately, there are not many methods and algorithms devoted to the solution of these specific type of game models. In this research, we will try to fill this gap and apply the differential transform method to solve the game models with parametric coefficients. As an example, we have a ship navigation system, where a collision risk is described as a game model. The game is presented by the  $R = [r_j(v_0, v_j)]$  risk matrix, containing the same number of columns as the number of the ship strategies.  $v_0$  is the speed of the own ship,  $v_j$  is a moving speed of the j-th encountered object [3].

$$R = [r_j(v_0, v_j)] = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1v_0-1} & r_{1v_0} \\ r_{21} & r_{22} & \dots & r_{2v_0-1} & r_{2v_0} \\ \dots & \dots & \dots & \dots & \dots \\ r_{v_1,1} & r_{v_1,2} & \dots & r_{v_1,v_0-1} & r_{v_1v_0} \\ \dots & \dots & \dots & \dots & \dots \\ r_{v_j,1} & r_{v_j,2} & \dots & r_{v_j,v_0-1} & r_{v_jv_0} \\ \dots & \dots & \dots & \dots & \dots \\ r_{v_m,1} & r_{v_m,2} & \dots & r_{v_m,v_0-1} & r_{v_mv_0} \end{pmatrix} \quad (1)$$

$$r_j = [a_1 \left( \frac{D_{min}^j}{D_s} \right)^2 + a_2 \left( \frac{T_{min}^j}{T_s} \right)^2 + a_3 \left( \frac{D_j}{D_s} \right)^2] J^{-\frac{1}{2}} \quad (2)$$

where  $r_j$  is the value of the risk of collision between the ship and the encountering j-th object,  $D_s$  - the safe distance of approach,  $T_s$  - the safe time which is necessary to execute a manoeuvre avoiding a collision with consideration of the actual distance,  $D_j$  - the actual distance between the ship and the encountering j-th ship,  $a_1, a_2, a_3$  - the weight coefficients appropriately depend on the state visibility at sea, the dynamic length and dynamic beam of the ship, the kind of the water region,  $D_{min}^j$  - the distance of the closest point of approach,  $T_{min}^j$  - the time to the closest point of approach.  $D_s \in [0.5-2.5]$  nm, where 0.5 nm is the considered good visibility, and the 2.5 nm restricted visibility,  $a_1 = a_2 = a_3 = 0.4$  (taken from [5]).

At each step, the most dangerous object is determined with regard to the value of the collision risk  $r_j$ . If, at a given step, no solution can be found at a speed of the ship  $V$ , the calculations are repeated at the speed reduced by 25% until the game has been solved. The calculations are repeated step by step until the moment when all the elements of matrix  $R$  become equal to zero.

### **Simplex method for solving multi-parametric linear programming tasks based on the differential transform.**

In order to apply the approach of a multi-parametric case, first let us consider Pukhov's differential transforms [8]:

- direct transformation:

$$U(K_1, \dots, K_n) = \frac{H_1^{K_1} \dots H_n^{K_n}}{K_1! \dots K_n!} \left[ \frac{\partial^{K_1 + \dots + K_n} u(t_1, \dots, t_n)}{\partial t_1^{K_1} \dots \partial t_n^{K_n}} \right]_{t=t_v} \quad (3)$$

$$K = (K_1, K_2, \dots, K_n) = \overline{0, \infty};$$

- reverse transformation:

$$u(t_1, \dots, t_n) = \sum_{f=0}^{\infty} \sum_{K_1 + \dots + K_n = f} \left( \frac{t_1 - t_{v1}}{H_1} \right)^{K_1} \dots \left( \frac{t_n - t_{vn}}{H_n} \right)^{K_n} U(K_1, \dots, K_n),$$

Where  $H = (H_1, \dots, H_n)$  is the vector of some positive scale coefficients;  $t = (t_1, \dots, t_n)$  - the vector of the approximation center coordinates;  $\mathcal{U}(K_1, \dots, K_n)$  - the discrete function of integer arguments; and  $K_1, \dots, K_n$  is the differential range of the  $\mathcal{U}(t_1, \dots, t_n)$  originals.

The mathematical model of the linear programming problem with multi-parametric coefficient dependencies in the left-hand parts of restrictions is presented:

$$\begin{cases} f(X) = c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n \rightarrow \underset{X \in D}{extr} \\ D: \begin{cases} A(t_1, t_2, \dots, t_l) X \leq b, \\ x_1 \geq 0, \dots, x_n \geq 0, \end{cases} \end{cases} \quad (4)$$

By applying direct differential transformations (3) against the elements of functional dependencies in (4), we receive relations [1-2] for calculating the D-T(differential transform) images, corresponding to simplex-conversions.

Parametric changes of the left-hand side constraint are presented:

$$a_{i_0 j}^{(q)}(K_1, K_2, \dots, K_l) = \left( a_{i_0 j}^{(q-1)}(K_1, K_2, \dots, K_l) - \sum_{p=1}^{K_1, K_2, \dots, K_l K} a_{i_0 j}^{(q)}(K-p) \cdot a_{i_0 j_0}^{(q-1)}(p) \right) / a_{i_0 j_0}^{(q-1)}(0),$$

$$a_{ij}^{(q-1)}(K_1, K_2, \dots, K_l) = \sum_{p=0}^{K_1, K_2, \dots, K_l K} a_{i_0 j}^{(q-1)}(K-p) \cdot a_{i_0 j_0}^{(q-1)}(p),$$

$$b_{ij}^{(q-1)}(K_1, K_2, \dots, K_l) = \left( a_{ij}^{(q-1)}(K_1, K_2, \dots, K_l) - \sum_{p=1}^{K_1, K_2, \dots, K_l K} b_{ij}^{(q-1)}(K-p) \cdot a_{i_0 j_0}^{(q-1)}(p) \right) / a_{i_0 j_0}^{(q-1)}(0),$$

$$a_{ij}^{(q)}(K_1, K_2, \dots, K_l) = a_{ij}^{(q-1)}(K_1, K_2, \dots, K_l) - b_{ij}^{(q-1)}(K_1, K_2, \dots, K_l),$$

where  $i = \overline{1, n}$ ,  $i \neq i_0$ ,  $j = \overline{1, n+m}$ ,  $K(K_1, K_2, \dots, K_l) = \overline{0, \infty}$ .

The following problem can be presented in the form of linear programming equations:

$$F = x_1 + x_2 + x_3 + \dots + x_{10} \rightarrow \underset{x_1 x_2 x_3 \dots x_{10}}{max}$$

$$\begin{cases} r_{1,1}x_1 + r_{1,2}x_2 + r_{1,3}x_3 + \dots + r_{1,10}x_{10} \leq 1, \\ r_{2,1}x_1 + r_{2,2}x_2 + r_{2,3}x_3 + \dots + r_{2,10}x_{10} \leq 1, \\ \dots \\ r_{10,1}x_1 + r_{10,2}x_2 + r_{10,3}x_3 + \dots + r_{10,10}x_{10} \leq 1, \\ x_1, x_2, x_3, \dots, x_{10} \geq 0 : \end{cases} \quad (5)$$

$$D_S : d, T_S : t, r_j = \left[ 0.4 \left( \frac{D_{\min}^j}{d} \right)^2 + 0.4 \left( \frac{T_{\min}^j}{t} \right)^2 + 0.4 \left( \frac{D_j}{d} \right)^2 \right]^{\frac{1}{2}}, d \in [0.5 - 2.5], t > 0.$$

$v_0 = 13, v_j = 25$ , we choose values for  $D_{\min}^j, D_j$  between [4-20] taking into account the calculating values in [5].

Considering the following values:  $K_1 = K_2 = 2, H_1 = H_2 = 1, d=0.5, t=0$ .

$$r_{i,j} = \begin{bmatrix} \left[ 0.4 \left( \frac{7}{d} \right)^2 + 0.4 \left( \frac{0.5}{t} \right)^2 + 0.4 \left( \frac{7.1}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{11}{d} \right)^2 + 0.4 \left( \frac{0.4}{t} \right)^2 + 0.4 \left( \frac{7.1}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{9}{d} \right)^2 + 0.4 \left( \frac{0.6}{t} \right)^2 + 0.4 \left( \frac{9.1}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{4}{d} \right)^2 + 0.4 \left( \frac{0.6}{t} \right)^2 + 0.4 \left( \frac{4.2}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{14}{d} \right)^2 + 0.4 \left( \frac{0.7}{t} \right)^2 + 0.4 \left( \frac{14.2}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{16}{d} \right)^2 + 0.4 \left( \frac{0.6}{t} \right)^2 + 0.4 \left( \frac{15.2}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{6}{d} \right)^2 + 0.4 \left( \frac{1.5}{t} \right)^2 + 0.4 \left( \frac{6.3}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{6}{d} \right)^2 + 0.4 \left( \frac{1.6}{t} \right)^2 + 0.4 \left( \frac{6.3}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{6}{d} \right)^2 + 0.4 \left( \frac{0.7}{t} \right)^2 + 0.4 \left( \frac{6.3}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{11}{d} \right)^2 + 0.4 \left( \frac{0.7}{t} \right)^2 + 0.4 \left( \frac{11.1}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{11}{d} \right)^2 + 0.4 \left( \frac{0.7}{t} \right)^2 + 0.4 \left( \frac{11.1}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{11}{d} \right)^2 + 0.4 \left( \frac{0.5}{t} \right)^2 + 0.4 \left( \frac{11.1}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{5}{d} \right)^2 + 0.4 \left( \frac{0.6}{t} \right)^2 + 0.4 \left( \frac{5.4}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{5}{d} \right)^2 + 0.4 \left( \frac{0.5}{t} \right)^2 + 0.4 \left( \frac{5.41}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{14}{d} \right)^2 + 0.4 \left( \frac{0.6}{t} \right)^2 + 0.4 \left( \frac{15.4}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{14}{d} \right)^2 + 0.4 \left( \frac{0.5}{t} \right)^2 + 0.4 \left( \frac{13.9}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{17}{d} \right)^2 + 0.4 \left( \frac{0.6}{t} \right)^2 + 0.4 \left( \frac{16.4}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{17}{d} \right)^2 + 0.4 \left( \frac{0.7}{t} \right)^2 + 0.4 \left( \frac{16.9}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{9}{d} \right)^2 + 0.4 \left( \frac{0.4}{t} \right)^2 + 0.4 \left( \frac{8.6}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{9}{d} \right)^2 + 0.4 \left( \frac{1.5}{t} \right)^2 + 0.4 \left( \frac{7.1}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{9}{d} \right)^2 + 0.4 \left( \frac{0.5}{t} \right)^2 + 0.4 \left( \frac{8.61}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{5}{d} \right)^2 + 0.4 \left( \frac{0.7}{t} \right)^2 + 0.4 \left( \frac{5.3}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{8}{d} \right)^2 + 0.4 \left( \frac{0.6}{t} \right)^2 + 0.4 \left( \frac{5.3}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{13}{d} \right)^2 + 0.4 \left( \frac{0.8}{t} \right)^2 + 0.4 \left( \frac{12.9}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{12}{d} \right)^2 + 0.4 \left( \frac{0.5}{t} \right)^2 + 0.4 \left( \frac{11.3}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{12}{d} \right)^2 + 0.4 \left( \frac{0.5}{t} \right)^2 + 0.4 \left( \frac{11.3}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{18}{d} \right)^2 + 0.4 \left( \frac{0.5}{t} \right)^2 + 0.4 \left( \frac{18.1}{d} \right)^2 \right]^{\frac{1}{2}} \\ \left[ 0.4 \left( \frac{6}{d} \right)^2 + 0.4 \left( \frac{1.6}{t} \right)^2 + 0.4 \left( \frac{7.2}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \left[ 0.4 \left( \frac{13}{d} \right)^2 + 0.4 \left( \frac{0.7}{t} \right)^2 + 0.4 \left( \frac{14.6}{d} \right)^2 \right]^{\frac{1}{2}} & \dots & \dots & \dots & \left[ 0.4 \left( \frac{8}{d} \right)^2 + 0.4 \left( \frac{0.3}{t} \right)^2 + 0.4 \left( \frac{8.3}{d} \right)^2 \right]^{\frac{1}{2}} \end{bmatrix}$$

The final divert transformations after applying the simplex method with differential transform are presented below:

The parametric expression of the value of the game will be restored as [1-2]:

$$\begin{aligned}
V = & 0.0176(t - 0.2)^0 (d - 0.5)^0 + 0.0028(t - 0.2)^1 (d - 0.5)^1 + 0.0109(t - 0.2)^2 (d - 0.5)^2 + \\
& + 0.0419(t - 0.2)^0 (d - 0.5)^1 + 0.0097(t - 0.2)^0 (d - 0.5)^2 + 0.0243(t - 0.2)^1 (d - 0.5)^0 + \\
& + 0.0297(t - 0.2)^2 (d - 0.5)^0 + 0.0087(t - 0.2)^1 (d - 0.5)^2 + 0.0069(t - 0.2)^2 (d - 0.5)^1 = \\
& 0.0419d + 0.024t + (d - 0.5)^2(0.0087t - 0.0043) + 0.0109(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.097(d - 0.5)^2 + (d - 0.5)(0.0028t - 0.0014) + 0.0069(d - 0.5)(t - 0.2)^2 + \\
& + 0.0297(t - 0.5)^2 - 0.0155
\end{aligned}$$

Appropriately, the probabilities of the collision risk between the ship and the encountering j-th objects will be restored as:

$$\begin{aligned}
r_1 = & 0.6626d + 0.0032t + (d - 0.5)^2(0.0014t - 0.0028) + 0.0104(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.0007(d - 0.5)^2 + (d - 0.5)(0.1409t - 0.02818) + 0.0008(d - 0.5)(t - 0.2)^2 + \\
& + 0.154(t - 0.5)^2 - 0.314
\end{aligned}$$

$$\begin{aligned}
r_2 = & 0.1996d + 0.1657t + (d - 0.5)^2(0.0008t - 0.00016) + 0.0443(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.0008(d - 0.5)^2 + (d - 0.5)(0.2305t - 0.0461) + 0.0006(d - 0.5)(t - 0.2)^2 + \\
& + 0.2754(t - 0.5)^2 - 0.845
\end{aligned}$$

$$\begin{aligned}
r_3 = & 0.0179d + 0.0136t + (d - 0.5)^2(0.0136t - 0.00272) + 0.0443(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.0131(d - 0.5)^2 + (d - 0.5)(0.0149t - 0.0298) + 0.9006(d - 0.5)(t - 0.2)^2 + \\
& + 0.127(t - 0.5)^2 - 0.0019
\end{aligned}$$

$$\begin{aligned}
r_4 = & 0.6677d + 0.0008t + (d - 0.5)^2(0.0003t - 6.0e - 5) + 0.0092(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.001(d - 0.5)^2 + (d - 0.5)(0.1413t - 0.028) + 0.0002(d - 0.5)(t - 0.2)^2 + \\
& + 0.1509(t - 0.5)^2 - 0.333
\end{aligned}$$

$$\begin{aligned}
r_5 = & 0.6677d + 0.245t + (d - 0.5)^2(0.0003t - 6.0e - 5) + 0.0092(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.001(d - 0.5)^2 + (d - 0.5)(0.1413t - 0.028) + 0.1509(d - 0.5)(t - 0.2)^2 + \\
& + 0.1509(t - 0.5)^2 - 0.256
\end{aligned}$$

$$\begin{aligned}
r_6 = & 0.0007d + 0.245t + (d - 0.5)^2(0.0011t - 0.0002) + 0.001(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.256(d - 0.5)^2 + (d - 0.5)(0.0016t - 0.0003) + 0.1509(d - 0.5)(t - 0.2)^2 + \\
& + 0.1509(t - 0.5)^2 - 0.048
\end{aligned}$$

$$\begin{aligned}
r_7 = & 0.00041d + 0.245t + (d - 0.5)^2(0.0011t - 0.0002) + 0.00085(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.256(d - 0.5)^2 + (d - 0.5)(-1.256e^{-5}t - 2.5008e^{-6}) - 2.12e - 5(d - 0.5)(t - 0.2)^2 + \\
& + 0.1509(t - 0.5)^2 - 0.049
\end{aligned}$$

$$\begin{aligned}
r_8 = & 0.00035d + 0.0092t + (d - 0.5)^2(0.0011t - 0.0002) + 0.00045(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.0061(d - 0.5)^2 + (d - 0.5)(0.008t - 0.0016) + 0.0003(d - 0.5)(t - 0.2)^2 + \\
& + 0.0022(t - 0.5)^2 - 0.396
\end{aligned}$$

$$\begin{aligned}
r_9 = & 0.00061d + 0.0019t + (d - 0.5)^2(0.0012t - 0.00024) + 0.0009(d - 0.5)^2(t - 0.2)^2 + \\
& + 0.00621(d - 0.5)^2 + (d - 0.5)(0.001t - 0.0002) + 0.00045(d - 0.5)(t - 0.2)^2 + \\
& + 0.0038(t - 0.5)^2 - 0.00061
\end{aligned}$$

$$r_{10} = 0.00016d + 0.00024t + (d - 0.5)^2 (0.0034t - 0.00068) + 0.0002(d - 0.5)^2 (t - 0.2)^2 + \\ + 0.0001(d - 0.5)^2 + (d - 0.5)(0.00011t - 0.00022) + 0.00038(d - 0.5)(t - 0.2)^2 + \\ + 0.0038(t - 0.5)^2 - 0.0001$$

**Conclusions and results:** We applied a parametric simplex method to solve the linear programming problem of the game matrix. For the ship collision game, there are 2 essential points to calculate the risk values. These points are the given sea visibility boundaries with 0.5(good visibility) and 2.5(low visibility). Numerical values of the ship collision risk are presented in table:

Table

$D_S$	$\{r_j\}, j = \overline{1, n}, n=10$	V
0.5	{0.0136, 0.0427, 0.0102, 0.0198, 0.4010, 0.1036, 0.011, 0.513, 0.0020, 0.0004 }	0.157
2.5	{0.141, 0.009, 0.0102, 0.667, 0.150, 0.0307, 0.132, 0.120, 0.0075, 0.0039 }	6.57

We see that the highest probability has the strategy 8(for good visibility) and strategy 4(for low visibility) which means that the risk of the collision with the first object is dominantly high, while for the rest of strategies it is almost 0. So for the next step of the game, the risk matrix should be generated taking into account the direction of the highest risky object whose appropriate object number is 8(for good visibility) or 4(for low visibility).

## REFERENCES

1. **Аветисян А.Г., Гюльзаян Л.С.** Метод решения задач параметрического линейного программирования, основанный на дифференциальных преобразованиях // Известия Томского политехнического университета. Серия "Математика и механика. Физика". – Томск, 2014. – Т. 324, № 2. – С. 25-30.
2. **Аветисян А.Г., Гюльзаян Л.С.** Метод решения многопараметрических задач линейного программирования, основанный на дифференциальных преобразованиях. Сер. ТН. – 2014. - Т. LXVII, №2. – С. 235-243.
3. **Lisowski J.** Multi-step matrix game with the risk of ship collision. – Gdynia: 2004. –P. 4-7.
4. **Lisowski J.** The sensitivity of state differential game vessel traffic model // Polish Maritime Research. –Gdansk, 2016. - 2(90), -23. –P. 14-18.
5. **Lazarowa A.** Decision support system for collision avoidance at sea // Polish Maritime Research. –Gdansk, October, 2012. -19. –P. 19-24.

6. **Hamed Y. Kourosh E.** Sensitivity analysis of Matching Pennies game // Mathematical and Computer Modelling. - 2010. -51. –P. 722-735.
7. **Takha H.** Vvedenie v issledovanie operatsiy [Introduction into operations research].- Moscow: Mir, 2005. – 912 p.
8. **Pukhov G.E.** Differentsialnye preobrazovaniya funktsiy i uravneniy [Differential transforms of functions and equations]. - Kiev: Naukova Dumka, 1980. –419 p.
9. **Simonyan S.O., Avetisyan A.G.** Prikladnaya teoriya differentsialnykh preobrazovaniy [Applied theory of differential transforms]. - Erevan: Chartaraget, 2010. –361 p.

#### **Մ.Կ. ԲԱԴԻՆՅԱՆ**

### **ՈՉ ԿՈՈՊԵՐԱՏԻՎ ԲԱԶՄԱՊԱՐԱՄԵՏՐԱԿԱՆ ՏԵՍՔԻ ԽԱՂԱՅԻՆ ՄՈԴԵԼՆԵՐԻ ԼՈՒԾՈՒՄԸ ԴԻՖԵՐԵՆՑԻԱԼ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐԻ ՀԻՄԱՆ ՎՐԱ**

Առաջարկվել է բազմապարամետրական տարրեր պարունակող ոչ կոոպերատիվ խաղային մոդելների լուծման նոր մոտեցում՝ հիմնված Պուխովի կողմից առաջարկված դիֆերենցիալ ձևափոխությունների վրա: Առաջարկվող ալգորիթմը փորձարկվել է նավերի բախման որոշման բազմապարամետրական խաղի օրինակով:

**Առանցքային բաներ.** բազմաբայլ խաղ, բազմապարամետրական խաղային մոդել, խաղերի տեսություն, սիմպլեքս ձևափոխություն, դիֆերենցիալ ձևափոխություն:

#### **М.К. БАГИНЯН**

### **РЕШЕНИЕ НЕКООПЕРАТИВНЫХ МУЛЬТИПАРАМЕТРИЧЕСКИХ ИГРОВЫХ МОДЕЛЕЙ НА ОСНОВЕ ДИФФЕРЕНЦИАЛЬНЫХ ПРЕОБРАЗОВАНИЙ**

Предложен новый подход к решению некооперативных мультипараметрических игровых моделей на основе дифференциальных преобразований. Метод основан на дифференциальных преобразованиях Пухова. Предложенный алгоритм был тестирован на примере мультипараметрической игры для определения столкновения судов.

**Ключевые слова:** многоэтапная игра, мультипараметрическая игровая модель, теория игр, симплекс-преобразование, дифференциальное преобразование.