Heuristic approach to the game of darts by using Genetic Algorithm and Ant Colony Optimization

Kamil Książek*, Wojciech Masarczyk†, Iwona Nowak ‡
Faculty of Applied Mathematics
Silesian University of Technology
Gliwice, Poland
*Email: kamiksi862@student.polsl.pl
†Email: wojcmas042@student.polsl.pl
‡Email: Iwona.Nowak@polsl.pl

***Abstract*—A paper illustrates the use of two metaheuristics: Genetic Algorithm and Ant Colony Optimization in darts play. The goal of the game is to hit on the centre of the dartboard. Both the creation of physical model and the optimization of the problem (based on heuristic algorithms) are presented in details. Two approaches are discussed and compared with respect to the results.**

***Index Terms*—metaheuristics, genetic algorithm, Ant Colony Optimization, darts, physical model**

I. Introduction

Heuristic algorithms are versatile method to solve many problems in which, for various reasons it is difficult to find a solution. These methods do not guarantee obtaining the optimal solution, but found with their help solutions are usually accurate enough and sufficient to deal with analyzed problems.

The word 'heuristic' comes from the Greek '*heurisko*' (it means: *I find*) [18] These are rules which help to find (discover) best approximation of solution. Various applications of these methods show that evolutionary computation can help in decision support systems. Heuristic methods are efficient in image processing [13], [15] but also voice recognition [10], while devoted combinations of artificial intelligence approaches are implemented together with other solutions into complex systems [8] and robotics [17]. The paper presents use of two population heuristics in darts play. Darts is a game in which darts are thrown at a dartboard fixed to a wall [1]. Generally, the accuracy of hitting into the appropriate pole of the dartboard is the main goal of the game. Appropriate speed and angle of throw is necessary to obtain success. In the paper the main goal is to demonstrate how to use heuristics in order to choose the parameters mentioned. It will be also analyzed a situation in which the setting of the player is not optimal, ie. when he stays not directly in front of the dartboard. To solve the problem, the Genetic Algorithm and Ant Colony Optimization (ACO) is applied. Evolutionary algorithm is treated as the classical metaheuristic using genetic operators like mutation or crossover. ACO is a type of Swarm Intelligence based on real ants behaviour. The results obtained in both methods and approptiate comparison is also presented at the end of work.

The article is divided into several sections. In section 2. it will be explained physical model playing darts. Genetic Algorithm, genetic operators like mutation and crossover, selection and other details connected with this topic will be analyzed in section 3. In section 4. there is described second metaheuristic: Ant Colony Optimization. In the last section, the results of both methods will be compared.

II. Physical Model

In dart game player is obliged to collect fixed sum of points that are signed to his account only if a thrown dart hits an appropriate part of the dartboard. Considering this short description of rules it seems obvious that accuracy of throw plays main role in dart game. In the presented approach the model simulates trajectory of a throw based on three variables: *speed*, α and β angles; *speed* is initial speed of a dart, α *angle* lies between vertical and horizontal components of velocity and β describes angle between horizontal and side velocity as shown on Fig. 1.

Friction force was skipped because its role is negligible - it is assumed that competitions take place in closed spaces. Dart is treated as a point. General formula to determine trajectory of a thrown dart is:

$$y = x \cdot \tan(\alpha) - \frac{g}{2 \cdot v_x^2} \cdot x^2, \tag{1}$$

where
$x -$ distance,
$g -$ standard gravity (about 9.80665 $\frac{m}{s}$),
$v_x -$ horizontal speed.

Distance obtained by a dart may be expressed as:

$$x = v_x \cdot t, \tag{2}$$

hence

$$y(t) = v_x \cdot t \cdot \tan(\alpha) - \frac{g \cdot t^2}{2}. \tag{3}$$

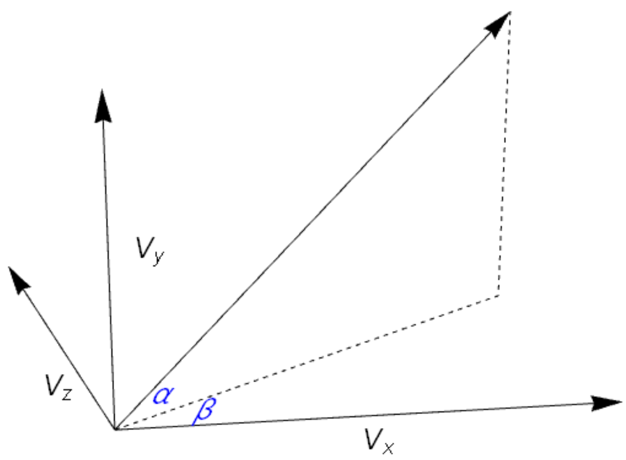

Figure 1. The Physical Model of thrown

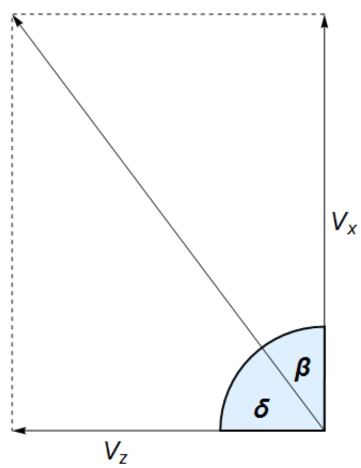

Figure 2. Curvature of dart's trajectory

The formula (3) describes height of a dart dependent on time while the rest of unknowns are treated as parameters. In order to describe the curvature of dart's trajectory the following formula can be used:

$$c = v_z \cdot t, \tag{4}$$

where
v_z – initial side speed of the dart (shown on Fig. 1),
t – time.

Fig. 2 presents view of Fig. 1 on plane $y = 0$. There are shown dependencies between v_x and v_z.

To estimate the accuracy of a throw we need to know when exactly a dart hits the wall. According to official rules of a dart game the distance between dartboard and players has to be exactly 2.37m. Once it is assumed that player may stand not perfectly in front of the dartboard the whole horizontal distance will be as followed:

$$x = \sqrt{d^2 + r^2}, \tag{5}$$

where
d – vertical distance from a perfect position (2.37m),
r – difference between optimum position and actual position of a player.

The last step is introduction of (5) into equation (1) and on the basis of it, to determine the vertical position of the dart. Both vertical and horizontal distance from center of the dartboard will be used to judge the value of fitness function.

III. Genetic Algorithm

A. History

Genetic Algorithm is a multi-agent algorithm based on idea of evolution that leads to survival only the best genotypes in whole population. Genetic algorithms have grown in popularity through the work of John Holland, especially by his book [2]. Until Genetic Algorithm Conference in Pittsburgh (Pennsylvania), the research was mainly theoretical, however after that more and more applications has been introduced. Evolutionary methods are one of the most popular and they begin to play significant role of classical heuristics. This makes that the action of heuristics of different types are compared with the work of GA.

B. Description

The main body of the Genetic Algorithm consists of modules that might be modified up to needs of user. Three basic modules of an algorithm are:

- Selection
- Genetic Operators
  - Mutation
  - Crossover
- Succession

General structure of GA is presented below in a form of pseudocode.

h] Pseudocode of Genetic Algorithm

Input: number of genotypes in population: *m*, number of iteration: *I*, boundary of the domain, coefficients: probability of crossover p_c, probability of mutation p_m
Output: coordinates of minimum, value of fitness function

Initialisation:
Creating the initial population $P_1 = \{x_1, x_2, ..., x_m\}$
Searching x_{best} in initial population P_1; $x_{opt} = x_{best}$.
Calculations:
$i = 1$
while $i < I$ **do**
  P_i = Selection(P_i)
  O_i = Genetic Operators(P_i)
  P_{i+1} = Succcession(O_i, P_i)
  Searching x_j^{best} in P_{i+1}.
  if x_j^{best} is better than x_{opt} **then**
    $x_{opt} = x_j^{best}$
  end if
  i++
end while
end

At the beginning, the algorithm randomly generates genotypes. In presented case each genotype consists of three genes: *speed*, α and β. Every genotype is separately graded by fitness function:

$$\begin{aligned}\Phi(v_x,\alpha,\beta) = |\sqrt{d^2+(d\cdot\tan\beta)^2}\cdot\tfrac{1}{\tan(90^\circ-\beta)}-s| + \\ +|\sqrt{d^2+(d\cdot\tan\beta)^2}\cdot\tan\alpha - \\ +0.5\cdot g\cdot\tfrac{d^2+(d\cdot\tan\beta)^2}{vx^2}-(p-h)|,\end{aligned} \tag{6}$$

where
s − distance from perfect position,
v_x − speed of dart,
h − height of throw.
g − standard gravity
d − vertical distance from a perfect position (2.37m)
p − height of a center of a dartboard (1.73m)
perfect position − a thrower is standing exactly on the center in front of a dartboard

Selection module picks two genotypes selected randomly and compares their fitness functions. The better one is picked to temporary population. Whole procedure is repeated until new population has m genotypes in it. This classical approach is called Tournament Selection.

Mutation is an genetic operation which works on single genotype and probability of its occurrence is usually smaller than 10%. The following formula presents mechanism of averaging mutation:

$$y = x_i^k + \xi_r \tag{7}$$

where
x_i^k − the $i-th$ genotype of $k-th$ population,
ξ_r − vector random generated with normal distribution $N(0,1)$

Crossover process takes two parental genotypes from temporary population to evolve them into one new genotype by mixing their genes. There also exist modifications where e.g. each gene is mixed separately with different genotypes, however to keep this procedure simple and transparent the first approach was implemented:

$$y = x_i^k + \xi_{U(0,1)}(x_j^k - x_i^k) \tag{8}$$

where
x_i^k − the $i-th$ genotype of $k-th$ population,
$\xi_{U(0,1)}$ − number randomly generated from uniform distribution $U(0,1)$,

Elitism Succession relies that the population P_{i+1} is formed by picking m best solutions from set $P_i \cup O_i$, where O_i is a set of offsprings − solutions coming from genetic operators.

IV. Ant Colony Optimization

A. History

Ant Colony Optimization is biologically inspired multi-agent algorithm. The base of this algorithm was invented by Marco Dorigo [4] - he used it to solve combinatorial problem (more specifically, travelling salesman problem). The approach presented in the paper was developed by M. Duran Toksari [7] - he applied Ant Colony Algorithm in continuous problem. It is not the only proposition of ACO for solving continuous problem. Other modification of this algorithm based on different approach was analyzed by K. Socha and M. Dorigo [11]. The idea of Ant System Algorithm is based on behaviour of real ant colony. In the presented case ants are solutions (optimum of a function). Ants during search can communicate with each other by using chemical substance called pheromone. On more attractive roads ants leave more quantity of pheromone so next ants know which path is more promising. Furthermore, pheromone is evaporating from non-used paths. It means that pheromone is impulse to search. On the following iterations search area is narrowed so ants try to find more accurate solution and they abandon unpromising roads.

B. Description

h] Pseudocode of Ant Colony Optimization

Input: number of ants: *m*, number of iteration inside: *I*, number of iteration outside: *n*, boundary of the domain, initial parameters: α, β, coefficients: λ, ω
Output: coordinates of minimum, value of fitness function
Initialisation:
Creating the initial colony of ants. $C_1 = \{x_1^1, x_1^2, ..., x_1^m\}$
Searching x_{best} in initial colony; $x_{opt} = x_{best}$.
Calculations:
$i = 1$
while $i < n$ **do**
  $j = 1$
  while $j < I$ **do**
    Moving the nest of ants - defining new territory of ant colony.
    Searching x_j^{best} in present colony.
    if x_j^{best} is better than x_{opt} **then**
      $x_{opt} = x_j^{best}$
    end if
  end while
  Defining new search area (narrowing of the territory).
end while
end

After fixing input parameters, the algorithm is generating *m* ants in form:

$$x_i^k = \left(x_i^{k,1}, x_i^{k,2}, ..., x_i^{k,S}\right), \tag{9}$$

where
k − number of ant, $k = 1, ..., m$,
S − number of dimensions.
i − number of iteration

Then is necessary to find the best ant in population in current iteration: x_{best} (it is best solution until this iteration). Additionaly, it is assumed $x_{opt} = x_{best}$. The next step is attempt to find a better place for the nest, what is realized by formula:

$$\forall k \quad x_j^k = x_{opt} + dx, \tag{10}$$

where
$dx = \left(dx^1, dx^2, ..., dx^S\right)$ $-$ vector of pseudorandom value,
$dx^i \in [-\alpha_j, \alpha_j]$ in the case of angles,
$dx^i \in [-\beta_j, \beta_j]$ in the case of speed,
α_j, β_j - parameters of ACO procedure,
j $-$ number of iteration.

This part of program is intended to exploration the domain. Now it should be checked if colony found better place to the nest. If at least one point has better value of fitness function (x_j^{best} is better than x_{opt}), then the nest is moving ($x_{opt} = x_j^{best}$). Last step is decreasing the length of jump (searched domain will be smaller).

$$\alpha_j = \lambda \cdot \alpha_{j-1}, \qquad \lambda \in (0; 1) \tag{11}$$

$$\beta_j = \omega \cdot \beta_{j-1}, \qquad \beta \in (0; 1) \tag{12}$$

In this case λ is responsible for reduction of domain during finding angles: α and β. ω is in charge of narrowing neighbourhood in search of speed. Then calculations are continued in smaller domain. The parameters λ and ω should be chosen carefully. If the domain is wide, λ and ω could be greater than 0.5. If a territory for ants is relatively narrow, λ or ω may be close to 0.1. In multi-dimensional problems it may define other parameter for every coordinate.

V. Results

The purpose of this paper is presenting heuristic approach to the game of darts. Two algorithms have been tested: Genetic Algorithm and And Colony Optimization. It was done for the same input data and results were compared in Tab I, obviously both algorithms had exactly the same fitness function (6). The stopping criterium was number of iterations. In the case of Genetic Algorithm it was 50 iterations for all measurements while researches by using of Ant Colony Optimization was done by 20 internal iterations and 30 external iterations. Generally Ant Colony Optimization algorithm provides higher precision results (in two tested positions of thrower results was better by using Genetic Algorithm $-$ font is bold in better scores). It was explored different distances from center position and height of thrower. In every case it was obtained great approximation of perfect throw. Graphic interpretation of our results is presented on the Fig. 6-7, what can be treated as warranty that given results are right data.

In the physical model it was applied Manhattan metrics [12] in order to describe the distance between center of a dartboard and the point where dart hits. Due to the fact that heuristic algorithms give only approximate results it is unlikely to obtain fitness function equal to 0, however presented results, especially from Ant Colony Optimization are close enough to claim that presented algorithms solve this problem really well.

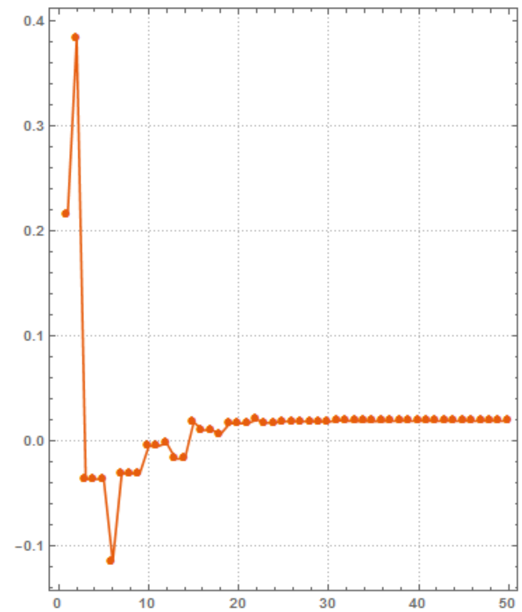

Figure 3. Result no. 3 (ACO): the values of the alpha angle during successive iterations

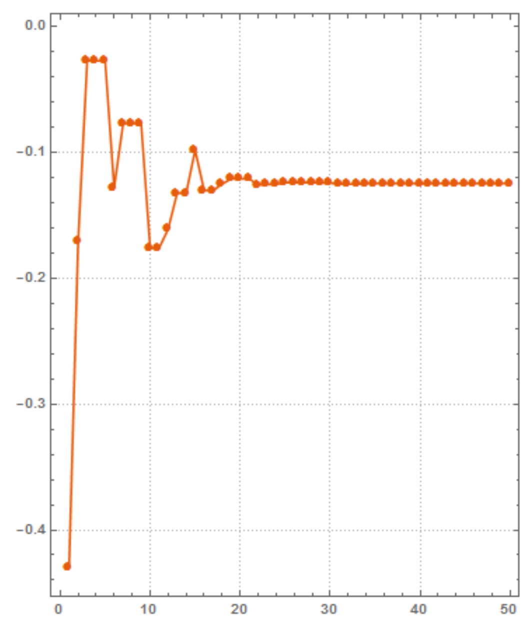

Figure 4. Result no. 3 (ACO): the values of the beta angle during successive iterations

VI. Final remarks

Heuristic algorithms are one of the best option in problems hard to optimization. The simple idea and in consequence implementation makes that these methods are often used by researchers in many fields. It is possible to find results with high precision by using these methods.

The calculations, which results were presented, were designed to study the mechanism and capabilities of both heuris-

Table I
RESULTS - GENETIC ALGORITHM AND ANT COLONY OPTIMIZATION

Initial values				Genetic Algorithm				Ant Colony Optimization			
number	number of points	distance from center [m]	height [m]	Alpha	Beta	Speed	Fitness Function value	Alpha	Beta	Speed	Fitness function value
1)	20	0.3	1.72	1.54827	7.09627	20.7708	0.01295	**0.563096**	**7.15305**	**46.9301**	**0.00101282**
2)	20	0.3	1.42	9.03306	6.4393	20.0802	0.000873	**7.69221**	**7.15355**	**46.8414**	**0.000348181**
3)	30	-0.3	1.72	**1.8878**	**-7.1609**	**20.2996**	**0.000938**	1.87287	-7.14723	20.3986	0.00136371
4)	20	0	1.72	2.5794	0.8024	16.8526	0.033815	**0.502524**	**0.00657695**	**49.9034**	**0.000544269**
5)	20	-0.5	1.72	2.00376	-11.647	21.893	0.016154	**0.430806**	**-11.6467**	**59.7989**	**0.00139404**
6)	40	-0.5	1.72	2.11129	-11.411	19.7369	0.016178	**0.779071**	**-11.656**	**36.2781**	**0.00188963**
7)	40	-1	1.72	2.8478	-21.164	18.6461	0.041302	**1.12286**	**-21.4121**	**28.2772**	**0.00188559**
8)	40	-1	1.52	**6.2153**	**-21.425**	**21.7337**	**0.0000581**	6.12793	-11.687	24.0042	0.000631615

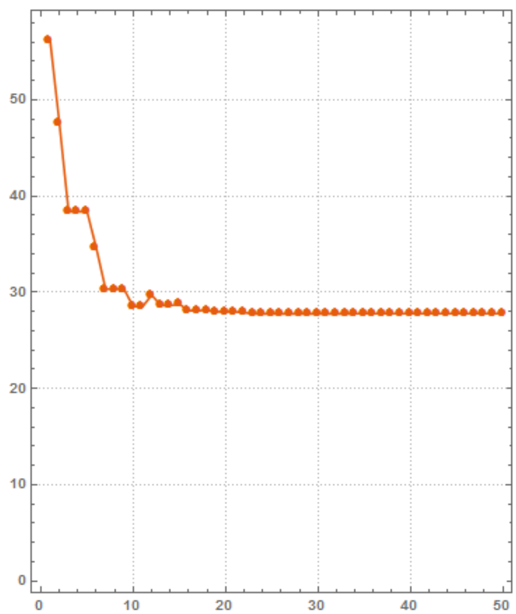

Figure 5. Result no. 3 (ACO): the speed during succesive iterations

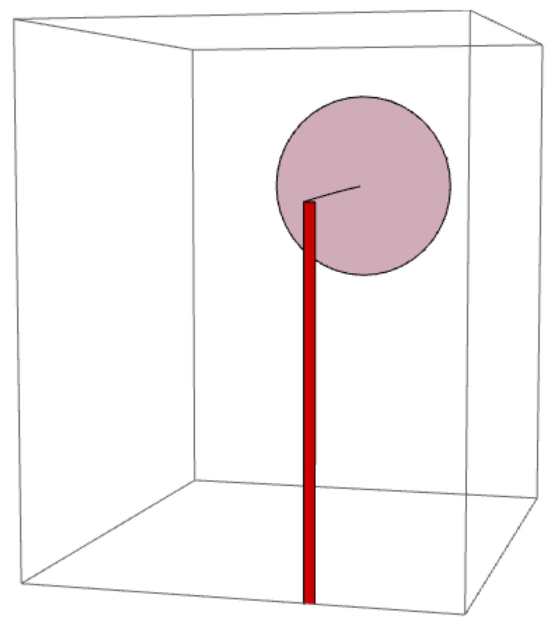

Figure 6. Result no. 1: graphic interpretation

tics. All solutions are on satisfactory level what proves that both the mathematical model and optimization procedure have been chosen correctly. One can use heuristic algorithms to solving other real problems, perhaps another game or some more complicated models.

REFERENCES

[1] https://en.wikipedia.org/wiki/Darts (Retrieved January 31, 2017)
[2] J. H. Holland, “Adaptation in natural and artificial systems.“ (1975).
[3] Á. E. Eiben, R. Hinterding, Z. Michalewicz. “Parameter control in evolutionary algorithms.“ IEEE Transactions on evolutionary computation 3.2 (1999): 124-141.
[4] M. Dorigo, T. Stützle, “Ant Colony Optimization“, *MIT Press, Cambridge, MA, 2004*
[5] D. Połap, M. Woźniak, C. Napoli, E. Tramontana, “Is Swarm Intelligence Able to Create Mazes?,“ in *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 305–310, 2015, DOI: 10.1515/eletel-2015-0039.
[6] D. Połap, M. Woźniak, C. Napoli, E. Tramontana, R. Damasevicius, “Is the colony of ants able to recognize graphic objects?,“ in *International Conference on Information and Software Technologies*. pp. 376–387, Springer International Publishing, DOI: 10.1007/978-3-319-24770-0_33.
[7] Toksarı, M. Duran, "A heuristic approach to find the global optimum of function." Journal of Computational and Applied Mathematics 209.2 (2007): 160-166.
[8] D. Połap, M. Woźniak, “Introduction to the Model of the Active Assistance System for Elder and Disabled People,“ in *Communications in Computer and Information Science*, vol. 639, pp. 392–403, 2016, DOI: 10.1007/978-3-319-46254-7_31.
[9] S. Cicciarella, C. Napoli, E. Tramontana, “Searching Design Patterns Fast by Using Tree Traversals,“ in *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 321–326, 2015, DOI: 10.1515/eletel-2015-0041.
[10] D. Połap, “Neuro-heuristic voice recognition,“ in *2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Proceedings*. 11-14 September, Gdańsk, Poland, IEEE 2016, pp. 487–490, DOI: 10.15439/2016F128.
[11] K. Socha, M. Dorigo, "Ant colony optimization for continuous domains." European journal of operational research 185.3 (2008): 1155-

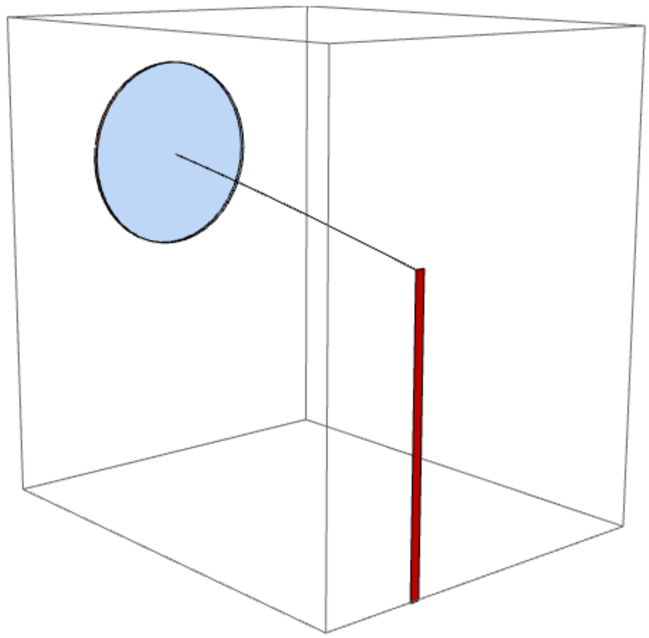

Figure 7. Result no. 8: graphic interpretation

1173.

[12] M. Barile, "Taxicab Metric." From MathWorld A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/TaxicabMetric.html (Retrieved January 31, 2017)

[13] M. Woźniak, “Novel Image Correction Method Based on Swarm Intelligence Approach,“ in *Communications in Computer and Information Science*, vol. 639, pp. 404–413, 2016, DOI: 10.1007/978-3-319-46254-7_32.

[14] C. Napoli, G. Pappalardo, E. Tramontana, R.K. Nowicki, J.T. Starczewski, M. Wozniak, “Toward Work Groups Classification Based on Probabilistic Neural Network Approach,“ in *Artificial Intelligence and Soft Computing (ICAISC), Lecture Notes in Computer Science*, vol. 9119, pp. 79–89, 2015, DOI: 10.1007/978-3-319-19324-3_8.

[15] M. Woźniak, D. Połap, C. Napoli and E. Tramontana, “Graphic object feature extraction system based on Cuckoo Search Algorithm“ *Expert Systems with Applications*, vol. 60, pp. 20–31, 2016, DOI: 10.1016/j.eswa.2016.08.068.

[16] D. Połap, M. Woźniak, C. Napoli, E. Tramontana, “Real-time cloud-based game management system via cuckoo search algorithm,“ in *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 333–338, 2015, DOI: 10.1515/eletel-2015-0043.

[17] L. Zmudzinski, A. Augustyniak, P. Artiemjew, Control of Mindstorms NXT robot using Xtion Pro camera skeletal tracking, Technical Sciences, vol. 19, no.1, pp. 71-81, Olsztyn, UWM Publisher, 2016.

[18] K. Trojanowski, “Metaheurystyki praktycznie“, Warsaw School of Information Technology, p.7, Warsaw 2008.