INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2017, VOL. 64, NO. 2, PP. 115–121
Manuscript received July 27, 2017; revised March, 2018. DOI: 10.24425/119358

Benchmark Tests on Heuristic Methods in the Darts Game

Kamil Książek, Dawid Połap, Marcin Woźniak

***Abstract*—Games are among problems that can be reduced to optimization, for which one of the most universal and productive solving method is a heuristic approach. In this article we present results of benchmark tests on using 5 heuristic methods to solve a physical model of the darts game. Discussion of the scores and conclusions from the research have shown that application of heuristic methods can simulate artificial intelligence as a regular player with very good results.**

***Keywords*—metaheuristics, optimization, computational intelligence**

Artificial Intelligence gives possibilities to simulate and model solutions for objects from various domains. Applications of systems composed for control and positioning are using intelligent implementations based on some phenomena from nature. We can find models using neural networks, swarm intelligence and some derivatives from these. Wlodarczyk-Sielicka discussed neural networks modeled to control traffic on water for touristic and professional purposes [1], [2]. Zhang et al. proposed collaborative strategies for intelligent systems [3]. Mandziuk and Swiechowski presented intelligent routing for traffic problems [4]. Ezma and Ani proposed neural networks for localization of objects [5]. Sometimes intelligent solutions operate on a model space in which information is constantly changing due to new incomes, new users, etc. For these solutions it is necessary to compose systems that can predict some situations. Therefore simulation in the model domain must be done by a group of intelligent agents. Esmaeili et al. presented research on multi-agent systems with self-organizing actions to adopt to the new situations [6], [7]. Marszalek proposed intelligent solutions for data storage systems [8], [9].

In our times optimization is one of the most important problems. A lot of engineering issues reduce the solution subspace to find a compromise between model variables. One of possible approaches to solving that problems are heuristic algorithms. They are useful in situations hard to optimize, where we do not know all the constraints or if the mathematical model is very hard to solve. We can apply swarm intelligence to solving systems of equations [10] or develop heuristic methods to simulate and position metallurgy processes [11], [12]. But it also turns out that it is possible to use these methods in finding strategies for the darts game, where a model of the throw can be efficiently optimized by application of heuristic methods.

Authors acknowledge contribution to this project of the "Diamond Grant 2016" No. 0080/DIA/2016/45 from the Polish Ministry of Science and Higher Education.

Kamil Książek, Dawid Połap and Marcin Woźniak are with Institute of Mathematics, Silesian University of Technology, Poland, (e-mail: kamilksiazek95@gmail.com, Dawid.Polap@polsl.pl, Marcin.Wozniak@polsl.pl).

A. Related works

Particle Swarm Optimization [13] is based on swarm-intelligence. The method simulates points called particles in a search for the new territory while taking into account the best localizations of the swarm and "craziness" (random generation of new places). Ant Colony Optimization [14] is the algorithm which uses intelligence of ant colony. In that idea ants leave chemical substance called pheromone during seeking food. Next followers rather exploit the route which has more pheromones. The main idea relies on narrowing territory around hopeful places (more chance to find the optimum). Clonal Selection Algorithm [15], [16] is based on human immune system which has to localize antigens and produce antibodies. Grey Wolf Optimizer [17] models wolf hunting tactics, in which all points called wolves find new solutions around the three most important wolves in the herd: α, β and γ wolf. Modified Ant Lion Optimizer is a modification of Mirjajili's idea [18] which is based on antlions ways of hunting ants. They prepare traps for prey and they wait for them on the bottom of snares. In this algorithm ants search for points around the nearest antlions.

Darts is a simple game in which missiles (called darts) are thrown at the dartboard hanged on the wall. It is essentially a technical discipline - angle and speed of each throw are main factors of the final success. Therefore it is necessary to estimate the appropriate parameters for the model of a throw. In this article we present our research on the physical model for it and the use of heuristic algorithms for finding optimal values of the model variables. Our idea was to verify how efficient in this game can be a model of the artificial intelligence composed by application of heuristic methods to optimize the result of the throws. For the research we have used five heuristic algorithms: Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) [19], Clonal Selection Algorithm (CLONALG), Grey Wolf Optimizer (GWO) and Modified Ant Lion Optimizer (MALO).

I. Simulation of the darts game

The game of the darts requires high precision. Players start a game with fixed number of points (usually 501) and they should reduce this score to zero. The main goal of all the competitors is hitting the dartboard in desirable place.

For the research on application of heuristic methods to simulate throws we used a physical model, in which we assume that the best option is the center of the dartboard. The model is extended to situations, where player is not standing exactly in front of the dartboard. Distance from the dartboard

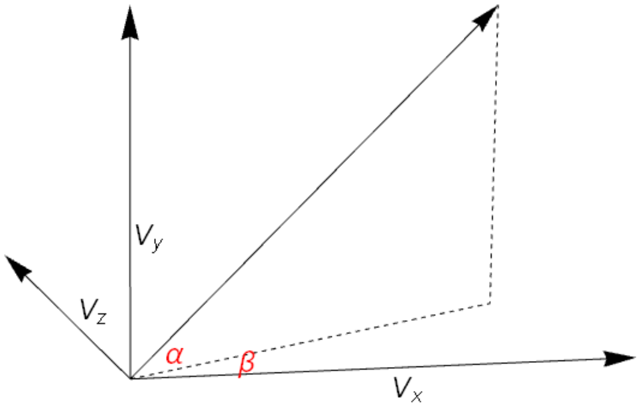

Fig. 1. Basic components of discussed model in the 3D space of the dart throw.

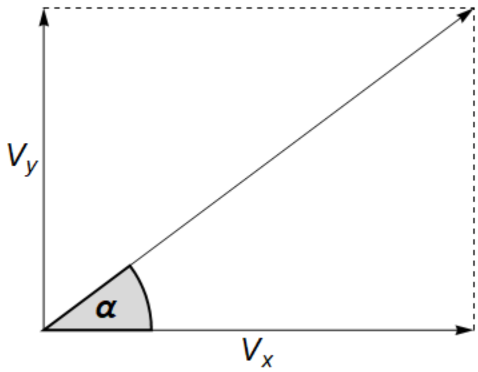

Fig. 2. Geometric interpretation of the α angle.

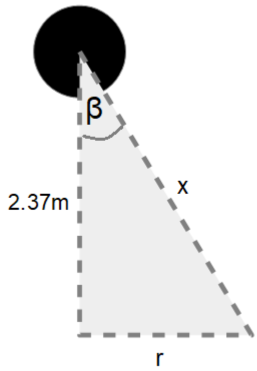

Fig. 3. Geometric interpretation of the β angle.

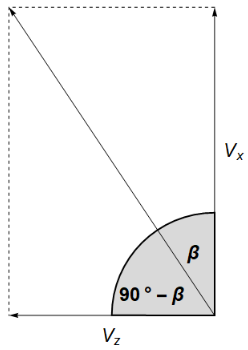

Fig. 4. Relation between v_x and v_z, for the model constraint $y = 0$.

in a straight line is equal to 2.37m (according to the rules of the game).

A. Description of the model

The model is based on three variables – angles: α, β and *speed* (initial speed of a dart). α is a vertical angle between initial position of a dart and the center of the dartboard – angle between horizontal and vertical components of velocity (Fig. 2). β is an angle between horizontal position of a thrower and an ideal position – exactly in front of the dartboard (Fig. 3). The vector components are shown on the Fig. 4. Friction force is omitted because its role is insignificant.

By the Pythagorean theorem we can define

$$x = \sqrt{(2.37)^2 + r^2} = \sqrt{(2.37)^2 + (2.37 \cdot \tan\beta)^2}, \quad (1)$$

where r is a distance from the optimal position. For the purpose of the model we assume that the dart is just a point, which makes the model simpler without assumptions of mechanical properties and dynamics of the motion. General equation describing the dart flight trajectory can be expressed as

$$y = x \cdot \tan\alpha - \frac{gx^2}{2v_x^2}, \quad (2)$$

where g is standard gravity ($\frac{m}{s^2}$), x is horizontal distance from the dartboard (m), v_x is horizontal speed ($\frac{m}{s}$).

Because of $x = v_x \cdot t$, after substitution of this dependence in (2) we have the following equation

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

where (3) describes position of the dart (height) in the dependence of the time.

B. Evaluation function

In order to evaluate the quality of solutions there was applied function

$$\begin{aligned}\Phi(\alpha, \beta, v_x) = {} & |\sqrt{(2.37)^2 + (2.37\tan\beta)^2} \cdot \frac{1}{\tan(90^\circ - \beta)} - r| \\ & + |\sqrt{(2.37)^2 + (2.37\tan\beta)^2} \cdot \tan\alpha \\ & - 0.5 \cdot g \cdot \frac{(2.37)^2 + (2.37\tan\beta)^2}{v_x^2} - (1.73 - h)|, \quad (4)\end{aligned}$$

where r is a distance from the optimal position (m), g is standard gravity ($\frac{m}{s^2}$), and v_x is horizontal speed ($\frac{m}{s}$).

The first component of the sum is responsible for horizontal distance from the optimal position and the second one is in charge of vertical distance from the center of the dartboard. Applied heuristic algorithms have to minimize this function because Φ expresses the sum of horizontal and vertical distances from the center of the dartboard. Therefore optimal value of Φ is 0.

TABLE I
MEAN RESULTS - 0.3M ON THE RIGHT OF THE DARTBOARD, HEIGHT OF A THROWER: 1.73M

number	algorithm	α	β	speed	Φ
1)	ACO	0.0231256361	0.124941	39.0490	$\mathbf{1.80666 \cdot 10^{-8}}$
2)	PSO	0.0083848393	0.124941	62.6451	**0.00043713**
3)	GWO	0.0014047948	0.124941	74.7890	**0.00208054**
4)	MALO	0.0223126743	0.124986	41.6353	**0.00222569**
5)	CLONALG	0.0149736186	0.124937	45.7715	**0.01289477**

TABLE II
BEST RESULTS - 0.3M ON THE RIGHT OF THE DARTBOARD, HEIGHT OF A THROWER: 1.73M

number	algorithm	α	β	speed	Φ
1)	ACO	0.00336677	0.124941	58.9809	$\mathbf{3.27184 \cdot 10^{-13}}$
2)	MALO	0.00417995	0.124941	52.9337	$\mathbf{3.35562 \cdot 10^{-11}}$
3)	PSO	0.00183003	0.124941	80.0000	$\mathbf{1.07957 \cdot 10^{-10}}$
4)	CLONALG	0.00360365	0.124941	56.9845	**0.0000076469**
5)	GWO	0.00207221	0.124955	75.2497	**0.0000418714**

TABLE III
MEAN RESULTS - 1M ON THE LEFT OF THE DARTBOARD, HEIGHT OF A THROWER: 1.8M

number	algorithm	α	β	speed	Φ
1)	PSO	0.0000715253	-0.374230	21.28329	$\mathbf{2.73755 \cdot 10^{-8}}$
2)	GWO	0.0009580619	-0.374164	20.97313	**0.000883946**
3)	MALO	0.0114732090	-0.374623	36.28673	**0.034601092**
4)	ACO	0.0051993067	-0.374230	44.59441	**0.038603302**
5)	CLONALG	0.0061366539	-0.374188	46.14489	**0.050274650**

TABLE IV
BEST RESULTS - 1M ON THE LEFT OF THE DARTBOARD, HEIGHT OF A THROWER: 1.8M

number	algorithm	α	β	speed	Φ
1)	ACO	0.02210443	-0.374230	15.8655	$\mathbf{4.90806 \cdot 10^{-11}}$
2)	PSO	0	-0.374230	21.3105	$\mathbf{2.71742 \cdot 10^{-10}}$
3)	GWO	0.00036421	-0.374152	21.1506	**0.00039354**
4)	MALO	0.09072624	-0.374230	10.2845	**0.00109297**
5)	CLONALG	0.00149611	-0.374243	22.5175	**0.0111526**

TABLE V
MEAN RESULTS - IN FRONT OF THE DARTBOARD, HEIGHT OF A THROWER: 1.85M

number	algorithm	α	β	speed	Φ
1)	PSO	0.01256227	0.000000060	15.11745	$\mathbf{6.3307611 \cdot 10^{-8}}$
2)	GWO	0.00696379	-0.000249463	14.57156	**0.00070932**
3)	CLONALG	0.00563713	0.002508136	29.28586	**0.07306828**
4)	ACO	0.00386170	0.000000005	42.43490	**0.08406682**
5)	MALO	0.00218455	0.003979974	37.28338	**0.09127863**

TABLE VI
BEST RESULTS - IN FRONT OF THE DARTBOARD, HEIGHT OF A THROWER: 1.85M

number	algorithm	α	β	speed	Φ
1)	PSO	0.00000423	$-4.48157 \cdot 10^{-10}$	15.1490	$\mathbf{1.25224 \cdot 10^{-9}}$
2)	ACO	0.00000071	0.000000006138176	15.1496	$\mathbf{1.46644 \cdot 10^{-8}}$
3)	GWO	0.00000636	-0.00006821	15.1492	**0.000169419**
4)	CLONALG	0.00954775	-0.00595235	13.8987	**0.0141573**
5)	MALO	0	$1.3 \cdot 10^{-12}$	17.4980	**0.0300479**

II. BENCHMARK RESULTS

Experiments were carried out by using following parameters

- **Particle Swarm Optimization:** 50 particles, 50 iterations, α – parameter connected with craziness (random search): 1, β – factor connected with best position of the particle: 1, γ – best position of the swarm: 1.

TABLE VII
MEAN RESULTS - IN FRONT OF THE DARTBOARD, HEIGHT OF A THROWER: 1.5M

number	algorithm	α	β	speed	Φ
1)	PSO	0.10674425	$1.62737 \cdot 10^{-10}$	43.91591	$\mathbf{1.078183 \cdot 10^{-8}}$
2)	ACO	0.11466644	$-1.15025 \cdot 10^{-7}$	31.45792	$\mathbf{2.728187 \cdot 10^{-7}}$
3)	GWO	0.10974264	-0.0001713632	36.85865	**0.000918796**
4)	MALO	0.11114653	0.0001629823	52.43210	**0.001060145**
5)	CLONALG	0.11017439	-0.0005032268	42.15859	**0.067002285**

TABLE VIII
BEST RESULTS - IN FRONT OF THE DARTBOARD, HEIGHT OF A THROWER: 1.5M

number	algorithm	α	β	speed	Φ
1)	ACO	0.100354	$5.227 \cdot 10^{-12}$	56.4595	$\mathbf{1.34682 \cdot 10^{-11}}$
2)	MALO	0.100179	$-1.053 \cdot 10^{-10}$	57.8807	$\mathbf{2.79078 \cdot 10^{-10}}$
3)	PSO	0.099590	$-9.954 \cdot 10^{-10}$	63.5848	$\mathbf{3.01553 \cdot 10^{-9}}$
4)	GWO	0.109273	-0.0000733866	30.3137	**0.00021599**
5)	CLONALG	0.112195	0.000277662	27.2744	**0.000658534**

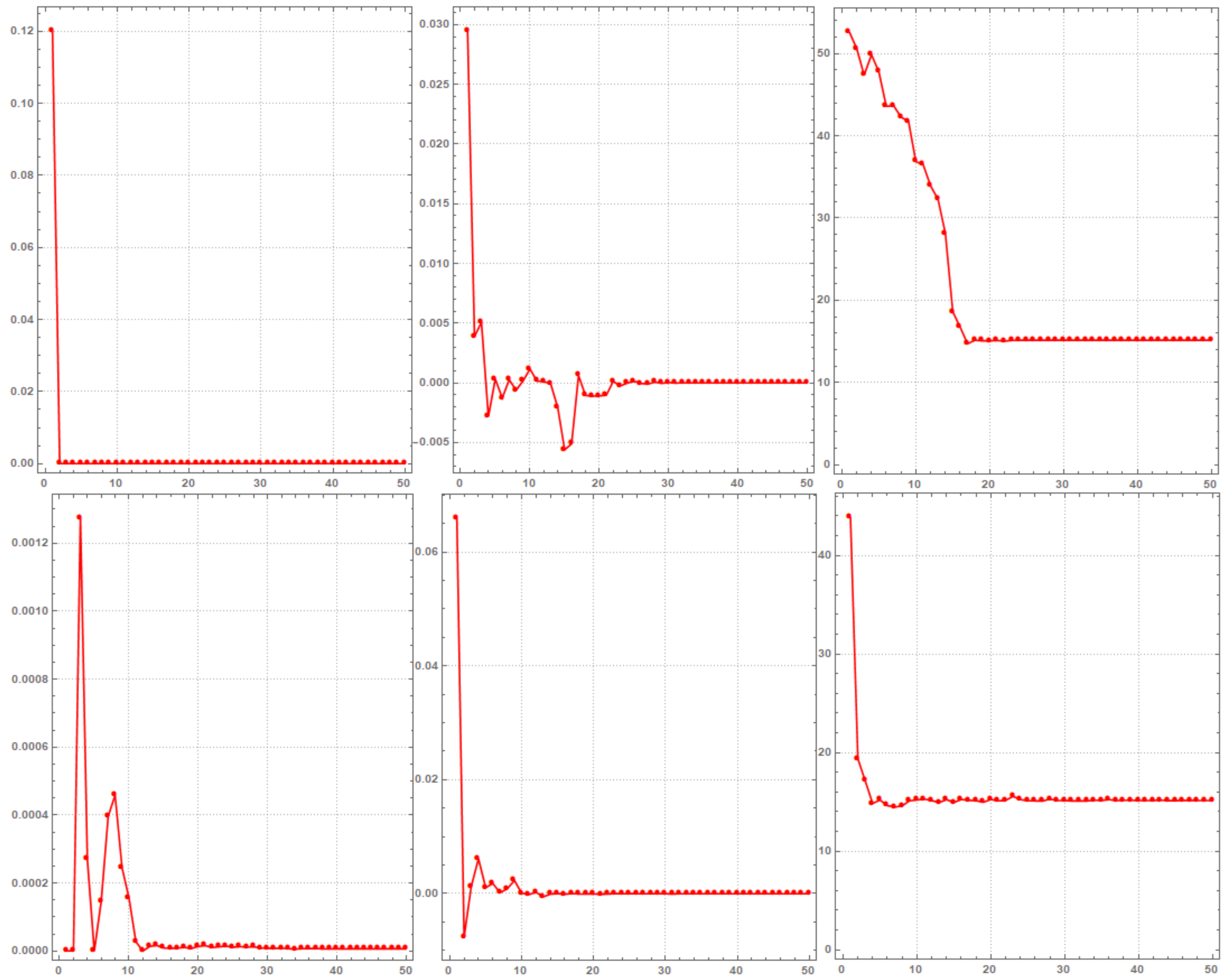

Fig. 5. Sample of PSO and GWO in third configuration: a thrower (1.8m tall) is in front of the dartboard. First row presents α, β and *speed* during one of the measurements by using PSO; second row shows the same variables during one of measurements by using GWO.

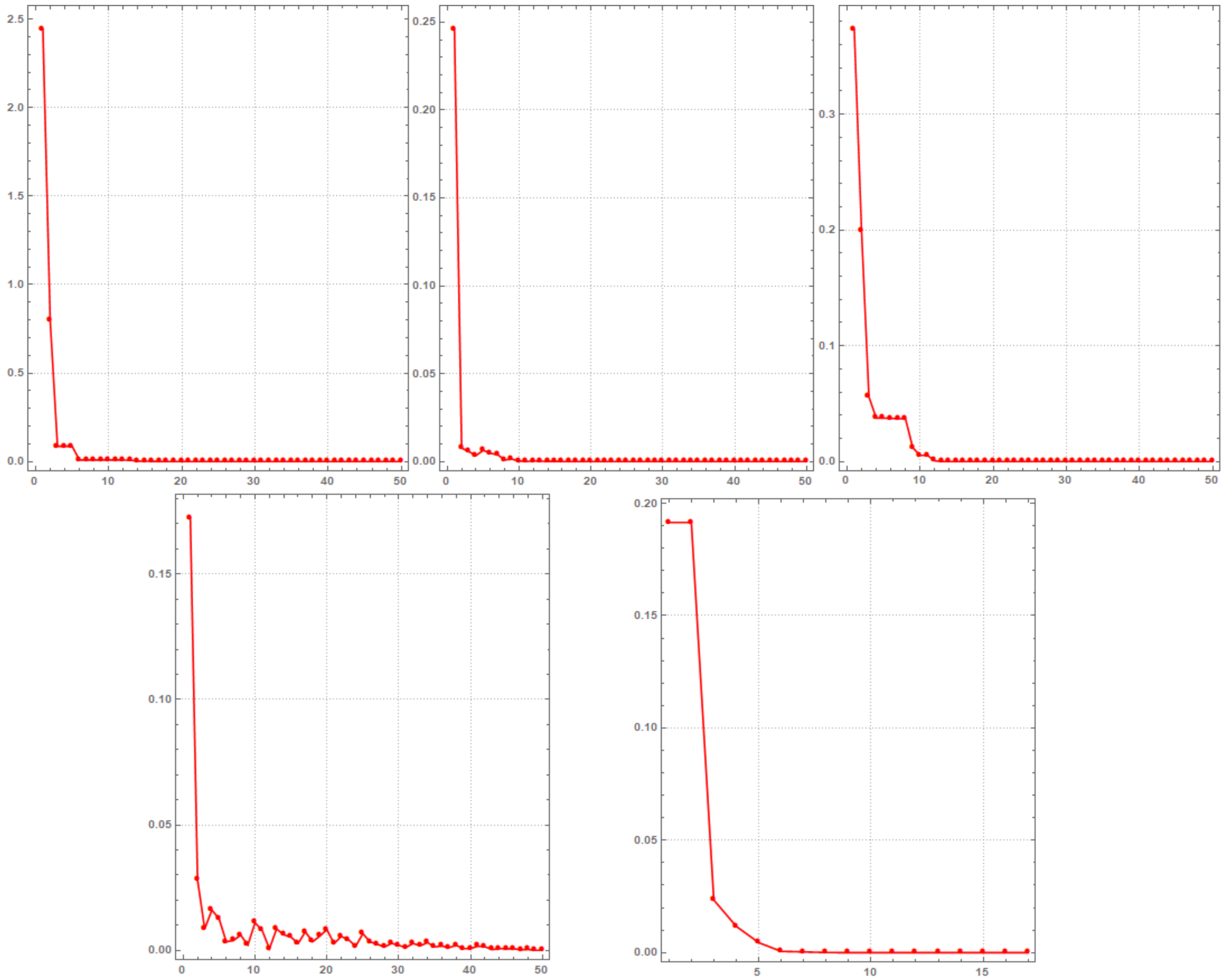

Fig. 6. Values of the evaluation function Φ during subsequent iterations in first configuration — a thrower (1.73m) is 0.3m on the right of the dartboard. Charts present best measurement coming from: MALO, PSO and CLONALG (first row), GWO and ACO (second row; in case of ACO only external iterations).

- **Ant Colony Optimization:** 50 ants, 3 inside iterations, 17 outside iterations; domain was narrowing during subsequent steps:

$$\delta_{i+1} := \lambda_1 \cdot \delta_i;$$
$$\rho_{i+1} := \lambda_2 \cdot \rho_i;$$
$$\gamma_{i+1} := \lambda_3 \cdot \gamma_i,$$

  where i is number of iteration, δ is size of domain for α angle, ρ for β angle and γ for $speed$. It was assumed that $\delta_1 = \rho_1 = 10, \gamma_1 = 10, \lambda_1 = \lambda_2 = 0.2, \lambda_3 = 0.35$.
- **Modified Ant Lion Optimizer:** 30 ants, 20 antlions, 50 iterations, $p = 10$ (number of ants captured by the best antlion of the population), initial standard deviation of the normal distribution: 3 in case of α and β angle, 20 in case of $speed$, narrowing during consecutive iterations: 0.6 ($par_{i+1} = 0.6 \cdot par_i$, where i is number of iteration).
- **Grey Wolf Optimizer:** 50 wolfs, 50 iterations.
- **CLONALG (Clonal Selection Algorithm):** 50 antibodies, 50 iterations, number of exchanged antibodies at the end of each iteration: 7, $\beta = 1 \;\; (N_j = \frac{\beta \cdot n1}{j})$, where N_j is number of clones $j-th$ antibody, $n1$ is total number of antibodies designed for cloning.

Four locations of a thrower were tested:

1) a thrower is 1.73m tall, 0.3m on the right of the dartboard (Table I-II),
2) a thrower is 1.8m tall, 1m on the left of the dartboard (Table III-IV),
3) a thrower is 1.85m tall, in front of the dartboard (Table V-VI),
4) a thrower is 1.5m tall, in front of the dartboard (Table VII-VIII).

There were carried out 10 independent measurements for each of the settings. Angles values are given in radians.

The results show that each of the tested algorithms gave sufficient approximation of variables to accurate throw. The worst score of evaluation function was about 0.03, while the best outcome was about 10^{-13}. In general, ACO and PSO

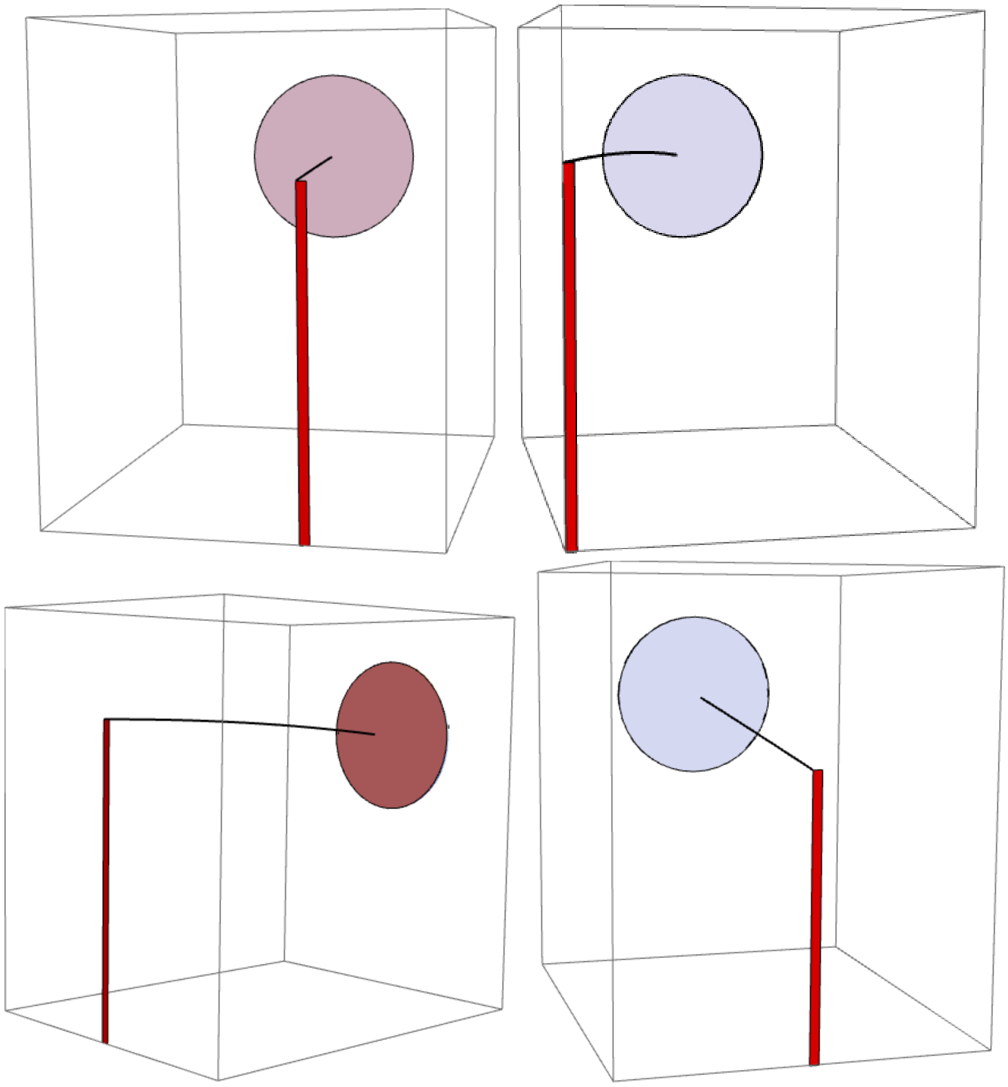

Fig. 7. Visualization of the simulation darts game. In every case has been used best scores coming from algorithms. In the first are shown row configuration 1 (0.3m on the right of the dartboard, height: 1.73m) and 2 (1m on the left of the dartboard, height: 1.8m), in the second row configuration 3 (center of the dartboard, height: 1.85m) and 4 (center of the dartboard, height: 1.5m).

proved to be the most effective during experiments. Otherwise, CLONALG was very good in exploration of the domain. One can see some differences in values of the speed. The reason for this is the fact that angles are definitely more important than speed. Fig. 5 and Fig. 6 present convergence coordinates and evaluation function for the sample coming from simulation. Fig. 7 embodies output data on the delineations.

A. *Conclusions*

As the results for numerical experiments have shown heuristic algorithms can play the darts game. It is necessary to create physical model which embodies a real situation. It should be specified which moves player can do and interrelations between model variables. Then heuristics can optimize the function and find the best strategy. During the simulation, implemented swarm is searching through the solution space for variables values to optimize the function.

Different examined methods gave the very similar results. all of them reached the goal of the darts throw with a very good precision. From presented values we can see that it was not a problem for heuristics to simulate the throw from various starting positions of the simulated players. In Fig. 7 we can see that initial position of the thrower could be in front of the board, from the left or from the right side. Also the height from which the throw was started was not giving any problems to the methods. PSO and ACO gave a slightly better result from other methods, therefore in the tables we can see that these two methods are ranked in first positions for all experiments. This situation may result from the nature of these heuristics. Both methods simulate a swarm moving over the space in the search for given criterion. Since the swarm is co-working on the final success, single particles exchange information about their founding. Therefore all the swarm is using knowledge even from a single particle, if this one is closer to the best solution. Since other examined methods a simulating behaviors from nature without information exchange, the optimal solution is reached slower. This observation gives a very interesting conclusion from our research. It seems that information exchange between agents in the search for optimum can be a crucial factor for the final success. Therefore in our future research we plan to examine various strategies for the information exchange between swarm agents. It would be also interesting to examine how much complexity of the information exchange model can influence the overall efficiency. These are the matters we plan to work on in our future research. One can also use other tools in finding optimal solution, for instance neural networks.

III. Final remarks

In this article we have presented our research on the swarms composed as artificial intelligence tools to solve mathematical models and therefore simulate players in the darts game. It is possible in games like darts or basketball (a throw into the basket), golf or ski jumping (take-off angle) to model the optimal conditions for heuristic optimization. The results have shown efficiency of the proposed solution and gave very interesting conclusions. The exchange of the information about the search can influence the overall efficiency. For the methods, in which this exchange was simulated, the optimal solution was found with higher precision.

References

[1] M. Wlodarczyk-Sielicka, "Importance of neighborhood parameters during clustering of bathymetric data using neural network," in *Information and Software Technologies - 22nd International Conference, ICIST 2016, Druskininkai, Lithuania, October 13-15, 2016, Proceedings*, 2016, pp. 441–452.

[2] M. Wlodarczyk-Sielicka and A. Stateczny, "Clustering bathymetric data for electronic navigational charts," *The Journal of Navigation*, vol. 69, no. 5, pp. 1143–1153, 2016.

[3] L. Zhang, L. Wang, W. Lin, and S. Yan, "Geometric optimum experimental design for collaborative image retrieval," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 24, no. 2, pp. 346–359, 2014.

[4] J. Mandziuk and M. Swiechowski, "UCT in capacitated vehicle routing problem with traffic jams," *Inf. Sci.*, vol. 406, pp. 42–56, 2017.

[5] L. S. Ezema and C. I. Ani, "Artificial neural network approach to mobile location estimation in gsm network," *International Journal of Electronics and Telecommunications*, vol. 63, no. 1, pp. 39–44, 2017.

[6] A. Esmaeili, N. Mozayani, M. J. Motlagh, and E. T. Matson, "The impact of diversity on performance of holonic multi-agent systems," *Eng. Appl. of AI*, vol. 55, pp. 186–201, 2016.

[7] ——, "A socially-based distributed self-organizing algorithm for holonic multi-agent systems: Case study in a task environment," *Cognitive Systems Research*, vol. 43, pp. 21–44, 2017.

[8] Z. Marszalek, "Novel recursive fast sort algorithm," in *Information and Software Technologies - 22nd International Conference, ICIST 2016, Druskininkai, Lithuania, October 13-15, 2016, Proceedings*, 2016, pp. 344–355.

[9] Z. Marszałek, "Performance test on triple heap sort algorithm," *PUBLISHER UWM OLSZTYN 2017*, vol. 20, no. 1, pp. 49–61, 2017.

[10] K. Ksiazek, "First approach to solve linear system of equations by using ant colony optimization," in *Proceedings of the Symposium for Young Scientists in Technology, Engineering and Mathematics (SYSTEM 2017), CEUR Vol. 1853, Kaunas University of Technology, Lithuania 2017*, 2017, pp. 57–61.

[11] R. Brociek and D. Slota, "Application and comparison of intelligent algorithms to solve the fractional heat conduction inverse problem," *ITC*, vol. 45, no. 2, pp. 184–194, 2016.

[12] ——, "Application of real ant colony optimization algorithm to solve space and time fractional heat conduction inverse problem," *ITC*, vol. 46, no. 2, pp. 171–182, 2017.

[13] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[14] M. D. Toksari, "Ant colony optimization for finding the global minimum," *Applied Mathematics and Computation*, vol. 176, no. 1, pp. 308–316, 2006.

[15] J. Brownlee, "Clonal selection theory & clonalg-the clonal selection classification algorithm (csca)," *Swinburne University of Technology*, 2005.

[16] L. N. De Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE transactions on evolutionary computation*, vol. 6, no. 3, pp. 239–251, 2002.

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[18] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[19] K. Ksiazek, W. Masarczyk, and I. Nowak, "Heuristic approach to the game of darts by using genetic algorithm and ant colony optimization," in *Proceedings of the International Conference for Young Researchers in Informatics, Mathematics and Engineering (ICYRIME 2017), CEUR Vol. 1852, Kaunas University of Technology, Lithuania 2017*, 2017, pp. 33–38.