jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2015, VOL. 61, NO. 4, PP. 333–338
Manuscript received October 12, 2015; revised November, 2015. DOI: 10.1515/eletel-2015-0043

Real-Time Cloud-based Game Management System via Cuckoo Search Algorithm

Dawid Połap, Marcin Woźniak, Christian Napoli and Emiliano Tramontana

***Abstract*—This paper analyses the idea of applying Swarm Intelligence in the process of managing the entire 2D board game in a real-time environment. For the proposed solution Game Management System is used as a cloud resource with a dedicated intelligent control agent. The described approach has been analysed on the basis of board games like mazes. The model and the control algorithm of the system is described and examined. The results of the experiments are presented and discussed to show possible advantages and disadvantages of the proposed method.**

***Keywords*—Computational Intelligence, Heuristic Algorithm**

I. Introduction

In recent years, Computational Intelligence methods have been adopted in numerous applications. In the 50s of the twentieth century, the first model of artificial neural network was presented in [1]. Since then, scientists have created newer and newer models of learning systems to improve automatic control. One of the most famous is the back propagation algorithm [2] with dedicated applications, i.e. [3]–[5] present the use of Radial Basis Probabilistic Neural Network, e.g. as an automatic classifier processing employee profiles to find out a non-explicit custom-created groups. Moreover, supporting tools have been created to optimise the creation of learning systems by a specific engineering using design patterns [6]. As a learning system, neural networks find their applications also in chess games [7]. In [8], the authors showed the method for evolving complex systems in real time while the game is being played. Applied model allows control agents to change and improve during the game. Automatic control and management can be also implemented by the use of dedicated implementations of Computational Intelligence methods. In [9] it was presented a dedicate incomplete data handling, while in [10] and [11] it was presented another attempt to misclassification in data sets. Intelligent algorithms can find important areas for 2D images [12], [13], [14], [15] and [16]. Moreover, Swarm Intelligence algorithms are efficient in optimization and solving of complex systems like these applied in modeling of solidification processes [17], [18], [19]. Bio-inspired methods make text data clustering more efficient [20] and also improve control systems [21]. In [22] Swarm Intelligence was applied to improve medical decision support systems, while in [23] an ant colony algorithm was adapted for path finding in games, and in [24] authors presented a firefly algorithm for solving the Knights Tour Problem.

In this article we present an application of control agent based on Swarm Intelligence to manage games. Computer games in common opinion are considered to be entertainment where the implemented high-end technologies are used to develop skills of the players like spatial or hand-eye coordination. Computer games allow players to move to another virtual world and provide an opportunity for abreaction from daily life. Each game can be measured in terms of playability, which is a set of principles and impressions received by the player. Playability consists of nonlinearity and complexity for the plot, numerous subplots and connections between smaller elements of the game. In order to increase playability of a game, it is necessary to create a very extensive game or develop a dedicated management system which will increase the difficulty of the game [25], [26]. In the following sections we would like to discuss a model for an intelligent system based on a Swarm Intelligence method for game management in cloud architectures. As an example of game, we have implemented it for mazes. In the presented solution, over the complex structure of the board, a passage path through the maze with numerous alleys must be found.

A. General Idea of Cloud-Computing

Cloud Computing is a model of data processing which is based on the use of external services without the purchase of a specific software or licenses. The principle of clouds is to use the server on which specific services will be prepared and users can connect to them and use them. The biggest advantage is that the client computer is unladen (the whole computing power, which requires the customer is the power of the server). Rising popularity of cloud computing helps to create various types of clouds. One of the most popular models are IaaS and PaaS. Infrastructure as a Service (IaaS) is a model in which a client receives a specific infrastructure eg.: a specific amount of power or space. The other form, Platform as a Service (PaaS), is a model that assumes providing specific applications for customers the client connects to the server by a computer or web browser and can use a prepared platform. This solution allows people to connect to their data from any place on earth if there is access to the Internet. Moreover, many IT companies use this type of methods to allow all their employees work remotely. These types of systems can be also optimized by application of bio-inspired methods. In [27] and

This work has been partially supported by project PRIME funded by the Italian Ministry of University and Research within POR FESR Sicilia 2007-2013 framework.

Dawid Połap and Marcin Woźniak are with Institute of Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland, (e-mail: Dawid.Polap@gmail.com, Marcin.Wozniak@polsl.pl)

C. Napoli and E. Tramontana are with Department of Mathematics and Informatics, University of Catania, Viale A. Doria 6, 95125 Catania, Italy, (e-mail: napoli@dmi.unict.it, tramontana@dmi.unict.it)

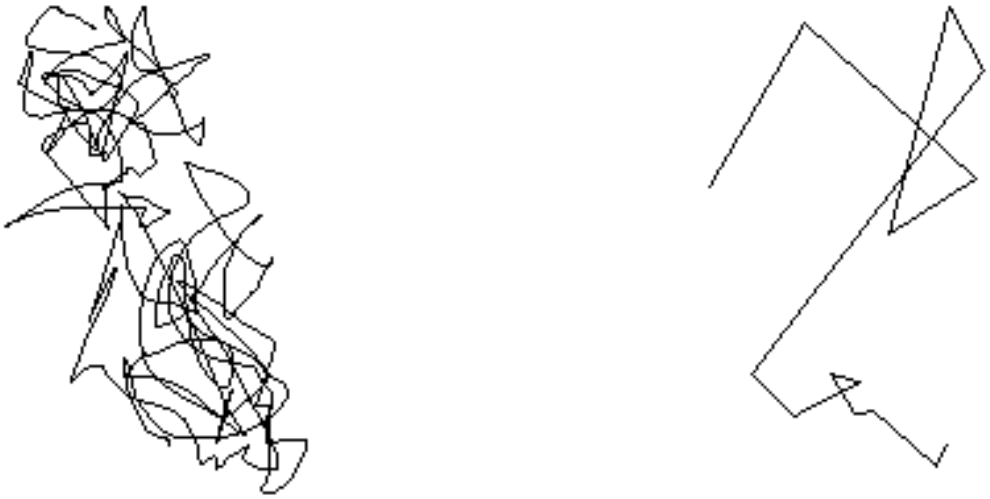


Fig. 1: An example of a random walk on the left and the Levy flight on the right.

[28] it was discussed how to optimize cloud resources for the optimal quality of service.

II. Computational Intelligence - Cuckoo Search Algorithm

In 2009, Cuckoo Search Algorithm (CSA) was presented for the first time in [29], where the authors showed the use of an algorithm for optimization problem. This algorithm belongs to a group of heuristic algorithms inspired by nature. The model of the algorithm is inspired by the behavior of cuckoos while tossing their eggs in nests of other birds. At the core of the algorithm, several assumptions are applied not only because of the simplification of the model but also to increase coherence. These assumptions are as follows

- In each iteration of the algorithm, the cuckoo can drop only one egg in one nest at random;
- The number of nests is constant;
- the host detect someone else's eggs with the probability $p \in \langle 0, 1\rangle$. In the case when detected egg is thrown out by the host, the new cuckoo is generated randomly.

The algorithm assumes that the cuckoos, eggs and nest hosts are similar, i.e. all of them are points in space that are changing their position (take a flight). Next, the egg is tossed and the hosts decision is imitated. Moreover, according to the two assumptions, the number of cuckoos is constant in each iteration due to the simplicity of numerical calculations. Each point in every epoch is assessed by fitness function f. The best points are transferred to the next iteration, the worst are replaced by new, random individuals.

Flights of some animals (including cuckoos) can be described by equation of Levy flight, which is a specific case of random walk, see some examples in Fig. 1. It is a stochastic process in which the steps are performed in a random directions. The equation for flights is based on continuous probability distribution and is named after french mathematician Paul Levy, whose equation is described as

$$L(\mathbf{x}; \mu, c) = \sqrt{\frac{c}{2\pi}} \frac{e^{-c/(2(\mathbf{x}-\mu))}}{(\mathbf{x}-\mu)^{3/2}} \tag{1}$$

where μ, c are specified parameters. The difference between the random walk and Levy flight is illustrated in Fig. 1.

The movement of cuckoos in the presented algorithm uses the Levy flight (1) and is described by the following equation

$$\mathbf{x_i}^{t+1} = \mathbf{x_i}^{t} + \mu * L(\mathbf{x}; \mu, c), \tag{2}$$

where μ is the length of stride for a random walk and t is the number of iteration.

For each solution, the decision to remove from the set is imitated. In nature, such decision is taken by the host which discovers tossed egg. Such a situation is described by

$$H(\mathbf{x_i}^{t+1}) = \begin{cases} 1-p & \text{remove the egg} \\ p & \text{leave the egg} \end{cases}, \tag{3}$$

where p is random value representing the chance to stay in the nest.

III. Adapting the Cuckoo Search Algorithm as an automatic control to manage the game

Described in the previous section CSA algorithm, must be subjected to several modifications for the purposes of adapting to modify and manage the maze. Cuckoos move in the solution space which is a two-dimensional array that represents a maze. The maze is interpreted as a set of fields where each field is composed of 5 numbers. Four numbers represent the walls of the field (the existence of the wall is marked as 1, and the

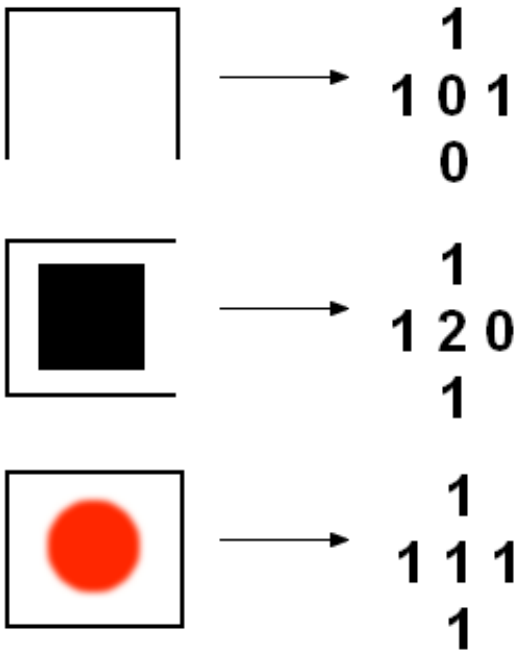


Fig. 2: Mathematical interpretation of a single field maze. From top - an empty field, the field occupied by the player and the field occupied by the cuckoo.

lack of wall as 0) and the fifth number indicates the position of the player or the cuckoo (and has a value of 2 for the players position, 3 if the space is occupied by the cuckoo, and 0 when the field is empty). This representation is shown in Fig. 2. On the other hand, cuckoo is understood as a point in two-dimensional space $\overline{x}$ whose coordinates indicate the row and column in the maze.

Cuckoo Search Algorithm is primarily responsible for modifying the labyrinth in real-time. Depending on the distance between the player and the virtual cuckoos, more and more of the modifications of the maze will occur to increase the effort and satisfaction of the person who will try to find a way out of the maze. For this purpose, cuckoos are evaluated by fitness function that depends on the distance from the player and also randomness. Fitness function describes the following equation

$$\Gamma(\mathbf{x_i}, \mathbf{x_j}) = \begin{cases} L_{i,j} + \nu & \text{if} \quad L_{i,j} < \sqrt{S} \\ \mathrm{e}^{-\alpha/\nu} L_{i,j} & \text{if} \quad L_{i,j} > \sqrt{S} \end{cases}, \tag{4}$$

where S is the value of a field maze, α is a constant predetermined parameter, ν is a random parameter in the range of $\langle 0, 1\rangle$, $L_{i,j}$ means the Cartesian distance between two points i and j (i means the cuckoo, j means the player). It is calculated by

$$L_{i,j} = ||\mathbf{x_i} - \mathbf{x_j}|| = \sqrt{\sum_{k=1}^{2} (x_{i,k} - x_{j,k})^2}. \tag{5}$$

Each cuckoo has the ability to modify the surrounding maze. It means that the cuckoo may remove or build walls around the neighboring fields. The assignment of this ability can be interpreted as a phenomenon of mimicry. Mimicry is a phenomenon conformed to something else. In the case of cuckoos, mimicry manifests itself in two ways. First, cuckoos try to imitate their eggs to others already in the nest (eg.: by using mud). Secondly known symptom of mimicry occurs in newly hatched cuckoos - they try not to distinguish themselves among others in the nest, but they try to imitate vocalizations of other birds, and sometimes even get rid of other eggs. In this model, removal/creation the walls is regarded as an attempt to camouflage eggs.

In addition, in the case when the player met cuckoo - the player is occupied by a certain period of time. For a player, the game continues - it means that he came across a subplot game, which he must resolve to continue his journey through the maze/the world. In the meantime, the rest of the cuckoos perform extra move and modify the maze. The complete algorithm is shown in Algorithm 1.

Algorithm 1 Cuckoo Search Algorithm

```
Start,
Define all coefficients: c, μ, ν, p and number of cuckoos,
Create at random initial population,
while the game is not finished do
  if t > 1 then
    Generate two random numbers in the range ⟨0, α⟩
    (the number of walls to be removed (r)/build (b)),
    if value of adaptation < 5 then
      Remove r random walls within 5 fields.
      Create b random walls within 5 fields.
    else
      while i < r do
        Generate a random number rand between ⟨0, 1⟩,
        if rand > 0.5 then
          Remove random walls within 5 fields.
        end if
      end while
      while i < b do
        Generate a random number rand between ⟨0, 1⟩,
        if rand > 0.5 then
          Create random walls within 5 fields.
        end if
      end while
    end if
  end if
  Move cuckoos according to (2) and (1),
  Hosts decide if the eggs stay or no according to (3),
  Evaluate cuckoos by (4) and take the best of them to
  next iteration,
  Rest of cuckoos take at random,
  t++
end while
Stop.
```

IV. Game Management System

Game Management System (GMS) can not only manages the action of the game but the entire mechanism that supports the game. Presented in previous sections method requires an additional mechanism that will support described Cuckoo

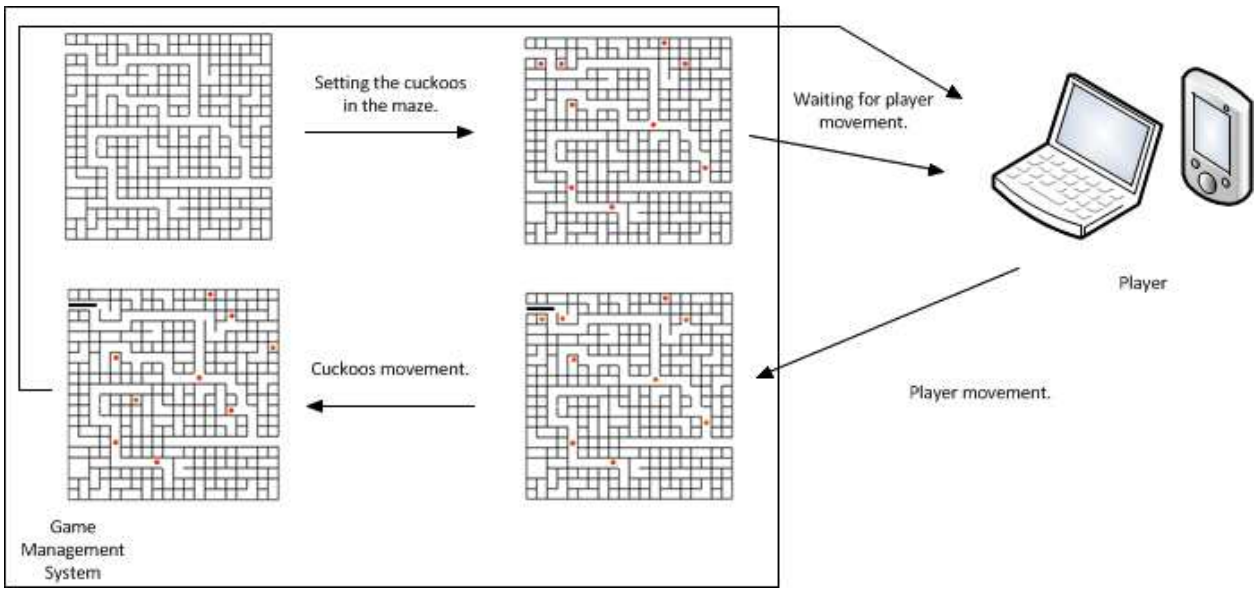


Fig. 3: Model of the Game Management System. In the first step a maze is loaded and cuckoos are set. Player makes a move, then the cuckoo modify the maze and change position. Next player makes a move, cuckoos change position and so on until the player finds a way out of the maze.

Search Algorithm. Moreover, GMS except for management of the action game and its mechanism must have control over the end of the game i.e. it is required that the system does its work until the player pass a game. In the case of mazes, end of the game takes place in case of finding a way out of the maze.

This type of GMS is designed to operate in the cloud. The user can use applications installed on PC, smartphone, tablet or other device (with a connection to the Internet) to connect to the system in the cloud where all operations are performed and results are sent back to the user. This solution makes it possible to install the game on devices with low processing power because everything is done not on the side of the player but using server resources, similar solution was applied to improve user verification system [30] and position traffic in cloud architectures. Model of applied system is shown in Fig. 3.

Described GMS, in the first step loads or generates maps/mazes. Labyrinth in GMS must exist in two forms: graphically for the user and as an array of numbers representing the maze for the system. Then, the CSA process starts the initial population will be created on the board game. After that, GMS does not perform any action user response is expected to start the game (running time is defined depending on the needs of the game). After performing the movement by the player, the system makes a move cuckoo modify the maze and change their position. The game continues until the end on the basis of altering movements of the system and the player.

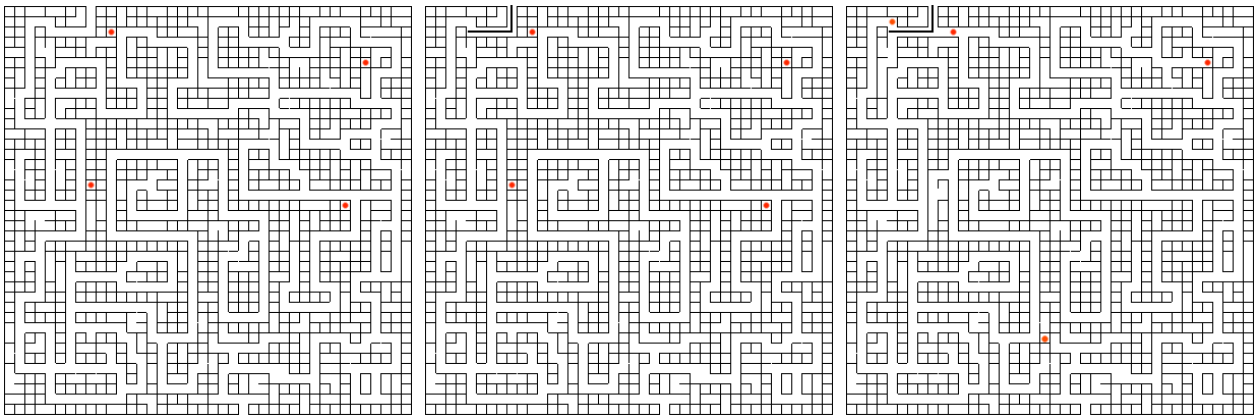
V. Experiments

For the purposes of testing described GMS, mazes were used as game maps and the parameters for Cuckoo Search Algorithm were chosen by the numerous tests. For different combinations, we studied the maze modifications due to player actions - it allowed us to achieve optimum parameters ($c = 0.7$, $\mu = 0.6$, $\nu = 0.4$, $p = 0.3$, 4 cuckoos). During the selection of these parameters, we considered the possibility of obtaining as much as possible modification of the maze because of minimized damage to the labyrinth and the distance from the player - the smaller the distance between cuckoo and the player, the more modifications are created. Usage of the GMS to the labyrinth is shown in Fig. 4.

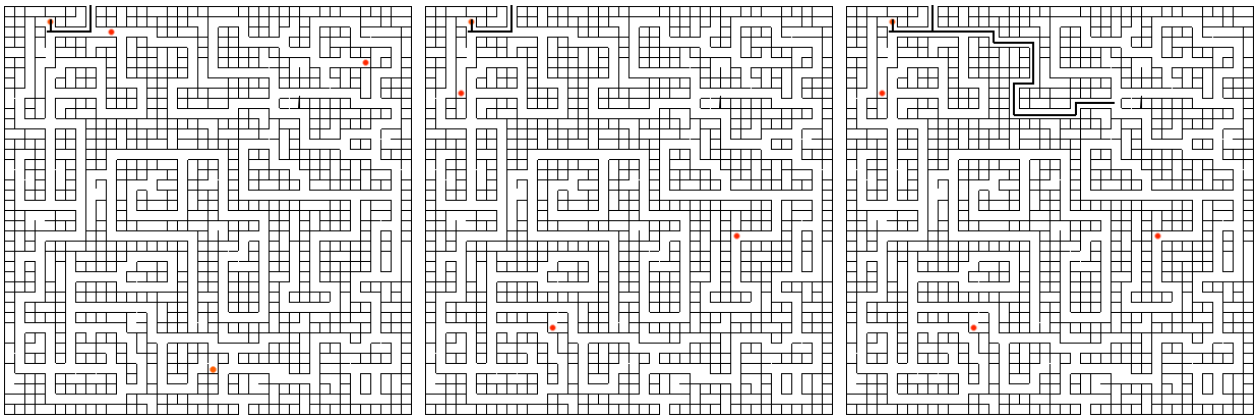
VI. Final Remarks

The presented method allows to create a system that can be implemented in any 2D and 3D game. Adapting Cuckoo Search Algorithm for managing the entire game gives the opportunity to remedy a problem of linearity of the storyline. It should be noted, that GMS has many advantages such as the ability to create dynamic (via cuckoos) numerous side quests in the main story and the frequent modification of the board. The player can repeatedly play the same game and each time will have the impression that it is something different than last time.

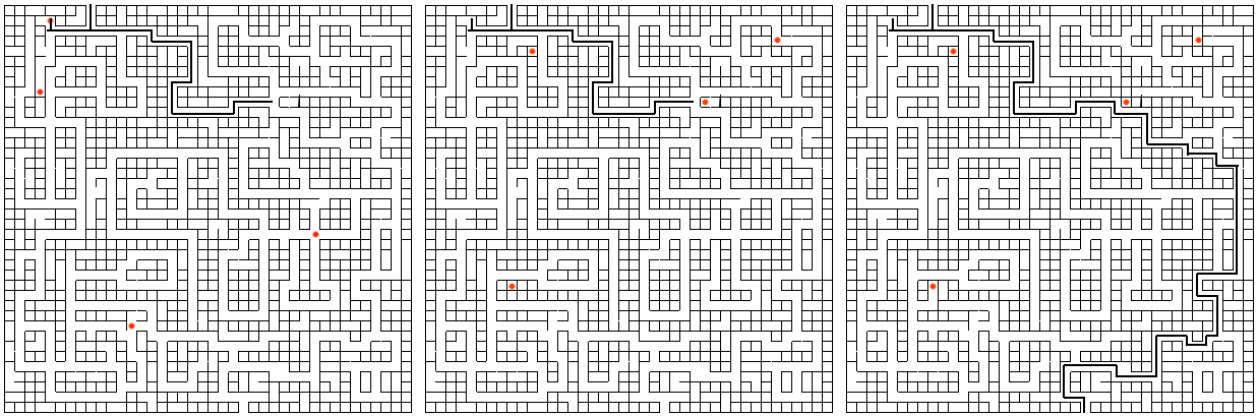
The biggest advantages of the system is real-time operation, which in combination with simple calculations due to the Cuckoo Search Algorithm result in a fast Game Management System. Nowadays, it is important for 3D games, where detailed graphic is as important as the story. For these games, the proposed system would require additional computing power, which is a disadvantage, however the finer the desired graphics the better the graphic card has to be. The example given illustrates mentioned drawback of this system. Another advantage is the possibility of action in the cloud, which would allow to eliminate this problem to some extent.



(a) Setting the cuckoos in random places. (b) Player movement. (c) Modification of the maze and the flight of cuckoos.



(d) Player movement and encounters an obstacle. Other cuckoo modify the maze and change position. (e) Cuckoos change position. (f) The player makes a move.



(g) Cuckoos modify the maze. (h) Cuckoos change position. (i) Cuckoos modify the maze and the player performs final move.

Fig. 4: An example of a modification of the maze by Cuckoo Search Algorithm due to the movement performed by the player.

REFERENCES

[1] B.G. Farley, W.A. Clark, "Simulation of Self-Organizing Systems by Digital Computer," *IRE Transactions on Information Theory 4 (4)*, 1954, pp. 76–84.

[2] D.R.G.H.R. Willimas, G.E. Hilton, "Learning representations by back-propagating errors," *Nature*, 1986, pp. 323–533.

[3] C. Napoli, G. Pappalardo, E. Tramontana, R. K. Nowicki, J. T. Starczewski, and M. Woźniak, "Toward automatic work groups classification based on probabilistic neural network approach," *Lecture Notes in Artificial Intelligence - ICAISC'2015*, vol. 9119, pp. 79–89, 2015, DOI: 10.1007/978-3-319-19324-3_8.

[4] C. Napoli, G. Pappalardo, and E. Tramontana, "An agent-driven semantical identifier using radial basis neural networks and reinforcement learning, *Proceedings of XV Workshop From Objects to Agents (WOA)*, vol. 1260, CEUR-WS, September, 2014.

[5] F. Bonanno, G. Capizzi, G. Lo Sciuto, C. Napoli, G. Pappalardo, and E. Tramontana, "A novel cloud-distributed toolbox for optimal

energy dispatch management from renewables in igss by using wrnn predictors and gpu parallel solutions," *Proceedings of IEEE International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, pp. 1077–1084, June, 2014.

[6] C. Napoli and E. Tramontana, "An object-oriented neural network toolboox based on design patterns," *Communications in Computer and Information Science - ICIST'2015*, vol. 538, pp. 388–399, 2015, DOI: 10.1007/978-3-319-24770-0_34.

[7] S. Thrun, "Learning to play the game of chess," *Advances in neural information processing systems*, vol. 7, 1995.

[8] K.O. Stanley, B.D. Bryant, R. Miikkulainen, "Real-time neuroevolution inthe NERO video game," *Evolutionaty Computation, IEEE Transactions on,* vol. 9, issue 6, 2005, pp. 653–668.

[9] B. Nowak, R. K. Nowicki, M. Woźniak, and C. Napoli, "Multi-class nearest neighbor classifier for incomplete data handling," *Lecture Notes in Artificial Intelligence - ICAISC'2015*, vol. 9119, pp. 469–480, 2015, DOI: 10.1007/978-3-319-19324-3_42.

[10] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "A new method for data stream mining based on the misclassification error," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 5, pp. 1048–1059, 2015, DOI: 10.1109/TNNLS.2014.2333557.

[11] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "Decision trees for mining data streams based on the gaussian approximation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 108–119, 2014, DOI: 10.1109/TKDE.2013.34.

[12] M. Woźniak and D. Połap, "Basic concept of cuckoo search algorithm for 2D images processing with some research results : An idea to apply cuckoo search algorithm in 2d images key-points search," in *SIGMAP 2014 - Proceedings of the 11th International Conference on Signal Processing and Multimedia Applications, Part of ICETE 2014 - 11th International Joint Conference on e-Business and Telecommunications*. 28-30 August, Vienna, Austria: SciTePress, 2014, pp. 157–164, DOI: 10.5220/0005015801570164.

[13] C. Napoli, G. Pappalardo, E. Tramontana, Z. Marszałek, D. Połap, and M. Woźniak, "Simplified firefly algorithm for 2D image key-points search," in *IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CIHLI 2014: 2014 IEEE Symposium on Computational Intelligence for Human-Like Intelligence, Proceedings*. 9-12 December, Orlando, Florida, USA: IEEE, 2014, pp. 118–125, DOI: 10.1109/CIHLI.2014.7013395.

[14] M. Woźniak, D. Połap, M. Gabryel, R. K. Nowicki, C. Napoli, and E. Tramontana, "Can we preprocess 2d images using artificial bee colony?" *Lecture Notes in Artificial Intelligence - ICAISC'2015*, vol. 9119, pp. 660–671, 2015, DOI: 10.1007/978-3-319-19324-3_59.

[15] D. Połap, M. Woźniak, C. Napoli, E. Tramontana, and R. Damaševičius, "Is the colony of ants able to recognize graphic objects?" *Communications in Computer and Information Science - ICIST'2015*, vol. 538, pp. 376–387, 2015, DOI: 10.1007/978-3-319-24770-0_33.

[16] M. Woźniak, C. Napoli, E. Tramontana, G. Capizzi, G. Lo Sciuto, R. K. Nowicki, and J. T. Starczewski, "A multiscale image compressor with rbfnn and discrete wavelet decomposition," in *IEEE IJCNN 2015 - 2015 IEEE International Joint Conference on Neural Networks, Proceedings*. 12-17 July, Killarney, Ireland: IEEE, 2015, pp. 1219–1225, DOI: 10.1109/IJCNN.2015.7280461.

[17] R. Brociek and D. Słota, "Application of intelligent algorithm to solve the fractional heat conduction inverse problem," *Communications in Computer and Information Science - ICIST'2015*, vol. 538, pp. 356–365, 2015, DOI: 10.1007/978-3-319-24770-0_31.

[18] E. Hetmaniok, D. Słota, and A. Zielonka, "Experimental verification of immune recruitment mechanism and clonal selection algorithm applied for solving the inverse problems of pure metal solidification," *Int. Comm. Heat & Mass Transf.*, vol. 47, pp. 7–14, 2013.

[19] E. Hetmaniok, I. Nowak, D. Słota, and A. Zielonka, "Determination of optimal parameters for the immune algorithm used for solving inverse heat conduction problems with and without a phase change," *Numer. Heat Transfer B*, vol. 62, pp. 462–478, 2012.

[20] P. Dziwiński, L. Bartczuk, and J. T. Starczewski, "Fully controllable ant colony system for text data clustering," *Lecture Notes in Computer Science - ICAISC'2012*, vol. 7269, pp. 199–205, 2012, DOI: 10.1007/978-3-642-29353-5.

[21] P. Dziwiński, L. Bartczuk, A. Przybyl, and E. Avedyan, "A new algorithm for identification of significant operating points using swarm intelligence," *Lecture Notes in Artificial Intelligence - ICAISC'2014*, vol. 8468, pp. 349–362, 2014, DOI: 10.1007/978-3-319-07176-3_31.

[22] M. Woźniak, D. Połap, R. K. Nowicki, C. Napoli, G. Pappalardo, and E. Tramontana, "Novel approach toward medical signals classifier," in *IEEE IJCNN 2015 - 2015 IEEE International Joint Conference on Neural Networks, Proceedings*. 12-17 July, Killarney, Ireland: IEEE, 2015, pp. 1924–1930, DOI: 10.1109/IJCNN.2015.7280556.

[23] J.A. Mocholi, J. Jaen, A. Catala, E. Navarro, "An emotionally biased ant colony algorithm for pathfinding in games," *Expert Systems with Applications*, vol. 37, issue 7, 2010, pp. 4921–4927.

[24] M.M. Ismail, A.F.Z. Abidin, S. Widiyando, M.H. Misran, M. Alice, N.A. Nordin, M.S. Zainudin, "Solving Knight's Tour Problem using Firefly Algorithm".

[25] M. Swiechowski and J. Mandziuk, "Self-adaptation of playing strategies in general game playing," *IEEE Trans. Comput. Intellig. and AI in Games*, vol. 6, no. 4, pp. 367–381, 2014, DOI: 10.1109/TCIAIG.2013.2275163.

[26] J. Karkowski and J. Mandziuk, "A new approach to security games," *Lecture Notes in Artificial Intelligence - ICAISC'2015*, vol. 9120, pp. 402–411, 2014, DOI: 10.1007/978-3-319-19369-4_36.

[27] M. Gabryel, M. Woźniak, and R. Damaševičius, "An application of differential evolution to positioning queueing systems," *Lecture Notes in Artificial Intelligence - ICAISC'2015*, vol. 9120, pp. 379–390, 2015, DOI: 10.1007/978-3-319-19369-4_34.

[28] C. Napoli, G. Pappalardo, E. Tramontana, and Zappalà, "A cloud-distributed gpu architecture for pattern identification in segmented detectors big-data surveys," *The Compter Journal*, p. bxu147, 2014, DOI: 10.1093/comjnl/bxu147.

[29] X-S. Yang, S. Deb, "Cuckoo search via Lvy flights," *Nature & Biologically Inspired Computing (NaBIC 2009)*, World Congress on. IEEE, 2009, pp. 210–214.

[30] M. Woźniak, D. Połap, G. Borowik, and C. Napoli, "A first attempt to cloud-based user verification in distributed system," in *Asia-Pacific Conference on Computer Aided System Engineering APCASE'2015*. 14-16 July, Quito, Ecuador: IEEE, 2015, pp. 226–231, DOI: 10.1109/APCASE.2015.47.