Heuristic Approach to Birthday Paradox Problem with Simulated Annealing and Cuckoo Search Algorithm

Adrianna Benna
Faculty of Applied Mathematics
Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
Email: ada.benna@gmail.com

***Abstract*—In this paper, the application of heuristic methods to solve birthday paradox problem is presented. Methods are compared to show which of them gives better and quicker solution. Benchmark tests have been performed and discussed to show the results.**

I. Introduction

Heuristic methods are used to find a solution or solve problems which for some reasons cannot be solved by traditional way. Sometimes finding the optimal or exact solution, using classical methods is just immposible or takes too much time. Then we can use heuristic solving to speed up time of work or find approximate solution. It can be called a shortcut but due to shortening the operation time it makes those methods very practical. Even if our solution is not exact, our approximated one can be only slightly different. Those methods do not guarantee us the exact solution but in some cases it is not necessary us necessary as for example saving time.

Cuckoo Search Algorithm (CSA) and Simulated Annealing (SA) are the examples of the heuristic methods. To create those alorithms, Computational Intelligence is used. Computational Intelligence (CI) is considered as a methodology which uses computer's ability to learn specified skills or learn how to behave in the new conditions. Created programs imitiate intelligent behaviour of animals' or humans' organism. Despite the fact that they are inspired by nature, the reason why they are created is to solve real-world problems which for some reasons, described in the prior paragraph, cannot be solved by traditional way.

II. Related Works

CI methods find their applications in many different fields of science. We can use them to perform some optimalisation processes [1], for example optimalisation semantic web services [2]. They can be useful to simulate the process of making decision [3]. Reconstruction or retriving of missing data is also possible thanks to CI [4]. Further applications, presented in [5] and [6], are image processing and positioning the mass service system [7].

The very first application of Simulated Annealing algorithm was presented in [8] as the method of optimalisation. Later, this algorithm find applications in solving other problems so it has been developed and improved to make the calculations more precise [9]. In the course of time, biological algorithms like SA was perceived as precise and efficient ones and useful in the minimalization of the continues functions described in [10] and [11]. SA algoritm allow us also to work on complex structures of various populations [12] and users veryfication of the cloud - based systems [13].

Cuckoo Search Algorithm allow us to describe changes in genes while adapting to the new environment. It can be used to sizing thermal electricity panels [14]. We can also find CSA application for intelligent gathering video frame [15] or optimal synthesis six - bar double dwell linkage problem [16]. There were also presented multitasking planning problem in [17]. CSA gives are tools to create scenerios and control the plots of computer games described in [18] and [19]. Optimalisation and stabilization of methods like this is not a easy thing [20] but we can adequately change the implementation code to achieve satysfying accuracy in calculations [21].

In this article Simulated Annealing and Cuckoo Search Algorithm are used to solve the Birthday Paradox Problem. Algorithms are implemented in such way that they can help us with searching probability to find similar dates.

III. Birthday Paradox Problem

Presented in this article famous probability problem can be solved both traditionally and using CI methods. Traditional approach and all details about paradox are described in [22]. Our problem can be stated by question: what is the minimal number n of randomly chosen people in the group that the probability that there are two people with the same birthday date is bigger that probability that there is not a pair like that? After mathematical assessments or checking probability of for example 1000 samples of randomly chosen groups of $n = 1, 2, 3, \ldots$ we find out that the answer is $n = 23$. By using CI methods, we can implement a program which will check for determined parameters in which iteration we will get first birthday pair and get approxiamate solution which after rounding will give us exactly 23.

IV. Simulated Annealing

Simulated Annealing (SA) is a method that is based on the metallurgical process. The whole process consists of three stages: heating the metal up to the high temperature, keep it in those conditions and slowly cooling down. It is important to keep thermodynamic equilibrium during the whole process and that is why we can describe it by mathematical equations. In [22] we can find all important details about SA. In that atricle, Birthday Paradox Problem is solved using SA algorithm. Now we want to do the same for CSA and compare the results to find out which of those two methods is better for solving this problem.

V. Cuckoo Search Algorithm

Cuckooo Search Algorithm (CSA) is a method of optimalization, based on the Gauss distribution and simulate the behaviour of some species of cuckoos which use others' birds nests to lay their own eggs. Those birds try to choose nests where eggs have been recently laid to minimize probability that hosts will drop them out. They can even imitiate the colour or texture of those eggs to stay unnoticed. When hosts find out the cuckoo egg, they can get rid of it or just ignore.

A. Mathematical Model

To use CSA to solve Birthday Paradox Problem we need a few assumptions:

1) We have 365 nests which represent 365 days in the year
2) Number of cuckoos is constant
3) Each cuckoo has one egg to lay
4) Chance that the egg will be detected by hosts is $chance \in (0, 1)$

We can describe the process of finding the new solution by equation

$$x_i^{t+1} = x_i^t + L(\beta, \gamma, \delta) \tag{1}$$

where $x^{t+1} = (x_1, x_2, \ldots, x_k)^{t+1}$ is the $(t+1)-th$ CSA solution, n - number of cuckoos in the population and $L(\beta, \gamma, \delta)$ is a Lévy flight determined for the given parameters: β - step lenght, δ - minimum step lengh, γ - Lévy flight scalling parameter. We can obtain the value of the Lévy flight by using formula

$$L(\beta, \gamma, \delta) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \dfrac{exp[-\dfrac{\gamma}{2(\beta-\delta)}]}{(\beta-\delta)^{\dfrac{3}{2}}}, & 0 < \beta < \delta < \infty \\ 0, & \text{other} \end{cases} \tag{2}$$

Where parameters β, γ, δ means the same as in (1). Hosts' decision is determined by equation

$$H(x_i^{t+1}) = \begin{cases} chance < p_\alpha, & \text{drop the egg} \\ chance \geq p_\alpha, & \text{the egg stays} \end{cases} \tag{3}$$

where $H(x_i^{t+1})$ is a hosts' decision about cuckoo egg, $chance$ is a value generated randomly during every decision and p_α is defined at the beggining of the whole process, $chance, p_\alpha \in (0, 1)$.

B. Implemented Algorithm

The aplication of the CSA that was implemented for solving Birthday Paradox Problem is presented in Algorithm 1. Instead of generating full dates, we can use numbers 1 - 365 which represent 365 days of the year (we don't consider lap years). At the beggining, we establish all the initial values and generate the random population on the list *population*. Cuckoos are flying according to (1), (2) and (3) and then, the lacking cuckoos are replaced randomly at once. After that, we check if there are two cuckoos with the same number in one generation. If so, we write down the number of generation - *generations* in the list *result*. When the list *result* is completed, we take the average of all summands and that way we get the final outcome for given parameters.

VI. Benchmark Tests

For all sets of parameters we obtain 30 results each and after taking the average of them, the received values have been compared to find solution which is the closest to 23.

A. Fitness Function

To find the optimal solution, two different fitness functions have been performed. First of them is a simple linear function $f(x) = x$. Received results are placed in the Tab. I. As we can see, this function does not allow as to obtain the exact wanted value - 23. What is more, the particular outcomes are not similar to each other in many cases. After many trials, it has been stated that the most important impact to the value of outcome has p_α and number of $cuckoos$. If they are lower - we obtain too high values and when they are higher - received values are too low. We can also notice that when we take 500 or more $samples$, our results are more similar and close to each other. What is surprising, the parameters β, γ and δ does not have any significant influence to the final result. The only noticed difference is that if those parameters become high, the score is albo slightly higher. In the Tab. I we can find two sets of parameters for each we got the value very close to 23: $p_\alpha = 0,105$, $\beta = 2$, $\gamma = 6$, $\delta = 7$, $cuckoos = 12$, $samples = 100$ and $p_\alpha = 0,105$, $\beta = 2$, $\gamma = 6$, $\delta = 7$, $cuckoos = 12$, $samples = 500$ - the only difference is in number of $samples$. In the Tab. I we can see that particular results form the the first set are more divergent than those form the second one.

Algorithm 1 Cuckoo Search Algorithm to obtain Birthday Paradox Problem Solution

```
Define the value of the probability p_α, fitness function
f(x), parameters β, γ, δ, number of cuckoos in each
generation and number of samples in each iteration
Set counter := 0, generations := 0 and decision := 0,
Declare lists: population, result,
while counter ≤ samples do
    Generate a random population (on the list population),
    while decision == 0 do
        Cuckoos fly according to (1) and (2),
        Hosts decide on eggs by (3),
        Lacking cuckoos replaced randomly,
        for i = 0 to cuckoos − 1 do
            for j = i + 1 to cuckoos do
                if f(population[i]) == f(population[j]) then
                    decision = 1,
                end if
            end for
        end for
        if decision == 0 then
            generations + +,
        end if
    end while
    Add generations to the list result'
    decision = 0, generations = 0,
    counter + +,
    Clear the list population,
end while
counter = 0,
Set sum = 0,
for i = 0 to samples do
    Sum = Sum + result[i],
end for
Outcome = round(sum/samples),
Return Outcome.
```

As the second fitness function, the power function has been performed, $f(x) = x^6$. Results are shown in the Tab. II. It turns out that it gives better average outcomes. We can determine parameters for which results are closer to 23 than in the prior testing. While changing parameters, we can notice similar actions to the function $f(x) = x$. When p_α and number of *cuckoos* get higher, the result is lower and conversly. The more *samples* we take, differences between particular outcomes are lower. What is new, parameters β, γ and δ are more important - while β and γ are singinificantly higher than δ, the results get lower so in order to keep it in the neighbouring of 23, we have to decrease p_α or number of *cuckoos*. For parameters $p_\alpha = 0,083$, $\beta = 10$, $\gamma = 200$, $\delta = 300$, $cuckoos = 10$, $samples = 100$ the excact value 23 has been received but because there were only 100 *samples* in each iteration, the divergence in results is very serious so we can say that it just happend accidentally. Anyway, there is another set: $p_\alpha = 0,210$, $\beta = 1$, $\gamma = 2$, $\delta = 3$, $cuckoos = 8$, $samples = 500$ for which divergence is not that high and final result is still very close to 23.

B. Conclusions

Firstly, we have to choose which of two presented fitness functions is better for Cuckoo Search Algotithm. Secondly, we have to compare numerical results form both CSA and SA algorithms to find the best one. For SA, 26 different sets of parameters have been prestented for which average value was the closest to 23. For CSA it was 13 sets of parameters for first fitness function and 13 for second. 13 samples form every kind have been taken to further calculations. While taking the average of those averages we receive 22,87 for SA, 23,26 for CSA $f(x) = x$, 22,92 for CSA $f(x) = x^6$. We can see that for SA and CSA $f(x) = x^6$ the average is closer to 23 than in case of CSA $f(x) = x$. However, by counting average we cannot say how large are divergences between particular samples or averages. Of course, we want them to be as little as possible to make our result more stable. On the Fig. 1 the divergence between particular samples for single set of parameters is presented - purple from SA, red form CSA, $f(x) = x^6$ and green from CSA, $f(x) = x$. As we can see, the smallest differences we have for SA, the most incompatible are results from CSA, $f(x) = x$. Similar conclusions we have after studying the standard deviation of those numbers. We want to know how wide those results are interspersed around 23. The lower standard deviation is, the lower is dissipation of averages. Indeed, we get standard deviation 0,215 for CSA $f(x) = x$, 0,027 for CSA $f(x) = x^6$ and 0,026 for SA. It only confirms prior findings. On the Fig. 2, 3 and 4 we can see that with the same values on the vertical axis, the highest amplitude is for the CSA, $f(x) = x$ and the most stable is chart for SA samples. This explains values of standard deviation.

To sum up, there is no doubt that application the second fitness function gives us better results but if we compare Cuckoo Search Algorithm and Simulated Annealing it turns out that for this problem SA is unbeatably better. In our comparisons, samples for CSA, $f(x) = x^6$ gave us results very similar to samples from SA but we have to remember that in the SA we could set parameters in such a way that almost every particular result which is taken to average equals 23. In CSA it is immposible to establish parameters to obtain such exact value in the final result. We have numbers from the range 16-29 and that is why Simulated Annealing is better to use than Cuckoo Search Algirithm in this problem.

VII. Final Remarks

In this article Cuckoo Search Algorithm has been implemented to obtain the solution for the Birthday Paradox Problem. Benchmark tests have been performed to establish the best parameters which gives us the solution. The results have been compared to the prior results for the same problem but solved using Simulate Annealing Algorithm. Algorithm with the best solution for this problem has been chosen.

TABLE I: Results of numerical experiments for first fitness function $f(x) = x$

probability	0,105	0,105	0,105	0,104	0,105	0,089	0,089	0,105	0,105	0,105	0,105	0,105	0,105
β	2	2	10	2	2	100	1	1	20	20	20	200	50
γ	8	6	40	4	6	200	2	100	50	50	50	600	60
δ	9	7	600	6	7	3	3	500	70	70	70	7	7
cuckoos	12	12	12	12	12	13	13	12	12	12	12	12	12
samples	100	100	100	100	500	250	250	100	100	250	500	500	500
	26	23	22	22	23	20	22	21	25	24	22	26	22
	19	19	21	24	25	24	22	21	24	26	23	23	23
	27	26	23	22	23	21	22	31	21	25	24	23	21
	20	23	21	22	23	23	21	25	27	23	25	23	23
	23	25	20	26	23	21	20	29	23	25	23	22	22
	20	25	24	24	23	24	22	25	27	22	24	26	22
	25	27	26	26	23	20	26	24	21	21	23	24	23
	25	23	26	27	23	23	22	24	20	24	22	22	24
	27	20	24	20	24	21	22	23	23	22	23	23	23
	22	20	27	26	22	26	21	23	22	23	23	26	24
	14	26	21	25	22	24	25	20	23	24	23	25	24
	25	24	26	22	25	23	22	26	20	23	23	24	23
	26	21	23	24	24	19	25	22	23	24	24	25	24
	30	24	24	22	22	23	21	23	22	26	24	23	24
	20	21	24	19	25	24	21	22	22	23	24	24	22
	27	22	26	24	24	21	20	24	24	23	24	22	25
	22	25	24	20	22	22	24	21	19	23	24	22	24
	23	22	27	26	23	24	22	21	27	22	25	25	25
	25	26	24	23	20	22	21	20	20	22	25	22	24
	28	25	22	25	26	23	24	23	27	24	25	25	25
	17	21	19	22	24	21	22	27	26	25	27	24	24
	28	21	24	22	20	24	25	22	25	22	24	22	25
	22	25	21	22	25	23	22	26	23	24	24	23	22
	24	24	17	24	22	24	23	23	18	22	24	25	24
	25	22	24	24	23	24	24	21	24	22	21	21	24
	24	23	23	21	22	25	22	24	25	23	22	25	23
	24	21	25	24	21	23	23	26	26	23	23	23	24
	28	20	26	29	24	22	22	20	22	28	24	24	24
	19	23	25	25	23	23	23	20	2 4	23	26	24	21
	18	24	20	22	23	22	22	25	24	24	25	26	23
average	23,43	23,03	23,3	23,47	23,07	22,63	22,43	23,4	23,23	23,5	23,77	23,8	23,37

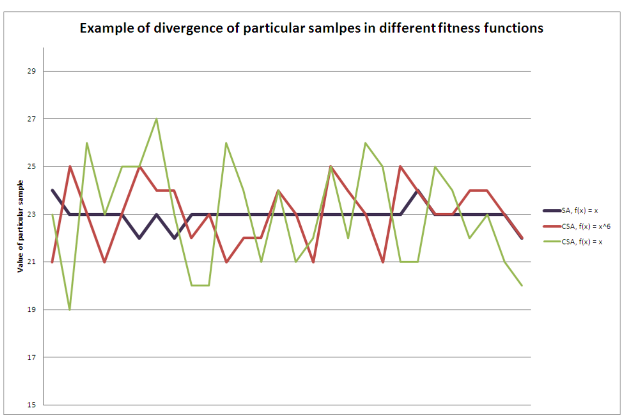

Fig. 1: Chart of divergence among particular samples taken to average with application of SA and CSA with all tested fitness functions.

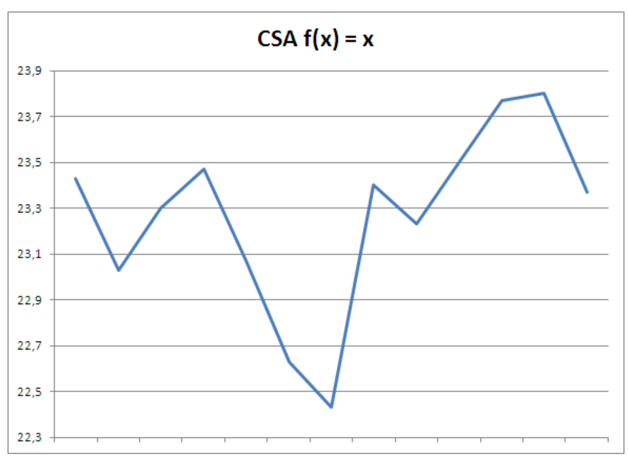

Fig. 2: Chart of distribution of averages for samples from CSA, $f(x) = x$.

REFERENCES

[1] M. Woźniak, "Fitness function for evolutionary computation applied in dynamic object simulation and positioning," in *IEEE SSCI 2014: 2014 IEEE Symposium Series on Computational Intelligence - CIVTS 2014: 2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems, Proceedings*. 9-12 December, Orlando, Florida, USA: IEEE, 2014, pp. 108–114, DOI: 10.1109/CIVTS.2014.7009485.

[2] V. Chifu, C. Pop, I. Salomie, D. Suia, and A. Niculici, "Optimizing the semantic web service composition process using cuckoo search," in *IDC'2012 - Studies in Computational Intelligence*, no. 382. Springer, 2012, pp. 93–102.

[3] M. Woźniak, D. Połap, L. Kosmider, C. Napoli, and E. Tramontana, "A novel approach toward x-ray images classifier," in *IEEE SSCI 2015 - 2015 IEEE Symposium Series on Computational Intelligence, Proceedings*. 8-10 December, Cape Town, RSA: IEEE, 2015, pp. 1635–1641, DOI:

TABLE II: Results of numerical experiments for second fitness function $f(x) = x^6$

probability	0,083	0,084	0,084	0,083	0,084	0,083	0,083	0,083	0,083	0,210	0,210	0,210	0,075
β	100	1	1	1	1	10	1	1	10	1	1	1	400
γ	200	500	500	500	500	200	2	400	200	2	500	2	600
δ	300	3	3	3	3	300	3	900	300	500	3	3	3
cuckoos	10	10	10	10	10	10	10	10	10	8	8	8	13
samples	500	100	500	500	200	100	500	500	500	500	500	500	500
	22	27	24	23	22	24	23	22	24	23	23	21	24
	24	22	22	23	24	24	23	22	22	22	24	25	24
	22	24	23	25	23	22	21	24	23	23	24	23	21
	23	24	23	24	23	22	22	23	22	21	23	21	22
	24	19	24	22	22	22	25	24	22	21	23	23	22
	23	20	24	23	23	22	23	23	25	25	20	25	23
	22	22	23	23	25	23	22	22	26	24	23	24	25
	23	24	23	22	25	24	23	23	21	22	23	24	22
	24	24	22	23	24	25	22	24	26	24	22	22	24
	24	25	23	23	22	21	23	25	25	23	22	23	23
	22	23	23	23	26	17	24	24	22	23	21	21	23
	24	22	23	23	22	20	21	22	24	22	26	22	24
	24	24	24	24	25	23	22	24	21	24	22	22	21
	22	24	24	26	23	22	24	23	22	23	23	24	25
	22	23	23	25	24	22	24	21	25	23	24	23	23
	22	20	24	24	24	25	23	22	24	24	24	21	23
	23	23	22	24	21	25	24	22	22	21	23	25	21
	24	25	21	22	21	23	24	22	22	23	23	24	24
	22	19	23	22	24	16	23	24	22	22	22	23	23
	22	22	23	23	21	25	26	22	23	22	23	21	21
	23	23	22	25	22	22	22	23	25	22	23	25	23
	22	24	24	22	24	26	23	23	22	24	20	24	23
	23	21	24	23	24	22	23	21	22	21	23	23	24
	23	23	24	23	22	24	23	22	22	23	23	23	22
	23	20	20	22	22	25	24	22	23	24	22	24	21
	21	17	22	23	21	29	22	23	23	24	22	24	25
	23	25	22	22	22	25	22	24	22	23	22	23	26
	25	25	22	22	21	21	23	23	23	23	24	22	23
	21	24	22	24	24	25	24	23	23	22	22	23	21
	23	23	23	22	22	24	25	23	24	21	24	23	21
average	22,83	22,7	22,87	23,17	22,93	23	23,1	22,83	23,07	22,73	22,77	23,03	22,9

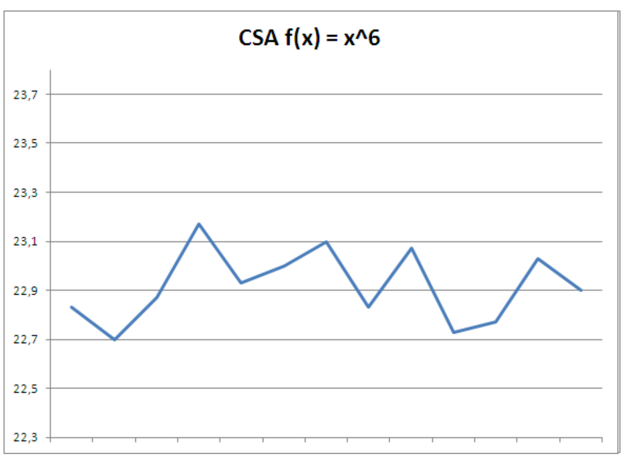

Fig. 3: Chart of distribution of averages for samples from CSA, $f(x) = x^6$.

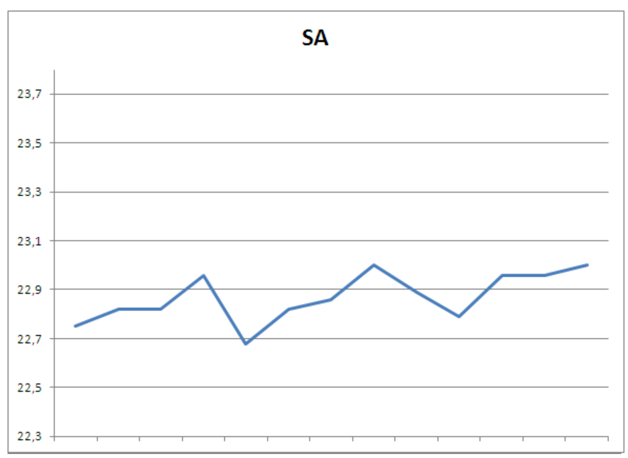

Fig. 4: Chart of distribution of averages for samples from SA.

10.1109/SSCI.2015.230.

[4] 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.

[5] 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.

[6] 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.

[7] M. Woźniak, M. Gabryel, R. K. Nowicki, and B. Nowak, “An application of firefly algorithm to position traffic in nosql database systems,” *Advances in Intelligent Systems and Computing - KICSS’2014*, vol. 416,

pp. 259–272, 2016, DOI: 10.1007/978-3-319-27478-2_18.
[8] S. Kirkpatrick and M. Vecchi, "Optimization by simmulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[9] L. Ingber, "Very fast simulated re-annealing," *Mathematical and Computer Modelling*, vol. 12, no. 8, pp. 967–973, 1989.
[10] M. Woźniak and D. Połap, "On some aspects of genetic and evolutionary methods for optimization purposes," *International Journal of Electronics and Telecommunications*, vol. 61, no. 1, pp. 7–16, 2015, DOI: 10.1515/eletel-2015-0001.
[11] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the simulated annealing algorithm," *ACM Transactions on Mathematical Software*, vol. 13, no. 3, pp. 262–280, 1987.
[12] I. Dupanloup, S. Schneider, and L. Excoffier, "A simulated annealing approach to define the genetic structure of populations," *Molecular Ecology*, vol. 11, no. 12, pp. 2571–2581, 2002.
[13] 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.
[14] J. Cabello, J. Cejudo, M. Luque, F. Ruiz, K. Deb, and R. Tewari, "Optimization of the sizing of a solar thermal electricity plant: Mathematical programming versus genetic algorithms," in *CEC'2009 Proceedings*. IEEE, 2009, pp. 1193–1200.
[15] G. S. Walia and R. Kapoor, "Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6315–6326, 2014.
[16] R. Bulatović, S. Bordević, and V. Dordević, "Cuckoo search algorithm: a metaheuristic approach to solving the problem of optimum synthesis of a six-bar double dwell linkage," *Mechanism and Machine Theory*, vol. 61, pp. 1–13, 2013.
[17] K. Chandrasekaran and S. Simon, "Multi-objective scheduling problem: hybrid appraoch using fuzzy assisted cuckoo search algorithm," *Swarm and Evolutionary Computation*, vol. 5, no. 1, pp. 1–16, 2012.
[18] D. Połap, M. Woźniak, C. Napoli, and E. Tramontana, "Is swarm intelligence able to create mazes?" *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 305–310, 2015, DOI: 10.1515/eletel-2015-0039.
[19] D. Połap, M. Woźniak, C. Napoli, and E. Tramontana, "Real-time cloud-based game management system via cuckoo search algorithm," *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 333–338, 2015, DOI: 10.1515/eletel-2015-0043.
[20] M. Clerc and J. Kennedy, "The particle swarmexplosion, stability and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
[21] I. Fister, X. S. Yang, J. Brest, and I. F. Jr., "Modified firefly algorithm using quaternion representation," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7220–7230, 2013.
[22] A. Benna "On Application of Anealing Algorithm to Birthday Paradox Problem", in *International Conference for Young Researchers in Informatics, Mathematics, and Engineering ICYRIME'2016*. 27-29 June, Catania, Italy, 2016.