# On Application of Annealing Algorithm to Birthday Paradox Problem Solving

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 idea of solving birthday paradox problem is proposed. Presented method is based on the application of Computational Intelligence. For different parameters the proposed solution has been performed. Research results has been gathered and presented to show possible advantages.**

## I. Introduction

Computational Intelligence (CI) is a methodology which involves computing to obtain systems with the ability to learn specific behavior and act like intelligent one. There are three main pillars of CI - fuzzy logic, neural networks and evolutionary computation. These methodologies are usually inspired by nature but we can find their application in the real - world problems in which mathematical or traditional modeling are impossible to employ for a few reasons:

1) process is to complex for traditional modeling or simply there is no mathematical algorithm available
2) imprecise or incomplete data
3) process might have stochastic nature and the optimal solution is unknown

CI provides solutions for such problems by creating tools or systems which can imitate intelligent behavior and have some human - like abilities, i.e. learning, dealing with new situations or decision making.

## II. Related Works

CI methods take inspiration from our natural environment. They are based on observations of human organism - i.e. nervous or immune system and animals' behavior - their lifestyle, adaptation to new conditions and scrabbling.
They find their applications in many areas like optimization [1], simulation of human decision processes [2], mass service systems positioning [3], image processing [4]; [5], optimization of semantic web services [6], reconstruction of missing data [7], and more.

Algorithm simulating cuckoo search for nests in the forest was applied to intelligent video frames targeting [8]. Similarly, this approach was also implemented for optimal synthesis of six-bar double dwell linkage problem [9]. Cuckoos motion model was also applied to multi objective scheduling problem [10]. Solving dynamic multidimensional knapsack problem was implemented using developed model of fireflies behavior in the summer [11]. There are also methods simulating changes in genes while adaptation to new environment. These can be used for sizing of solar thermal electricity panels [12]. Similarly CI methods can serve in games, to compose scenarios and control plot [13] and [14]. Stability and optimization of these methods is not a trivial problem [15], however it is possible to modem adequately to the implementation to achieve sufficient precision in the calculations [16].

The first version of simulated annealing algorithm was presented in [17], where the authors proposed its implementation for optimization purposes. With time computer scientists used it for various purposes and therefore some improvements and developments were proposed to simulated annealing to increase precision of calculations [18]. Later, this method, and other bio-inspired algorithms, were reported for efficiency and precision in widely used minimization of various continues functions [19] and [20]. Moreover we can find comments on restoration of low resolution structures of macromolecules by application of annealing algorithm [21]. Simulated annealing approach can be used to compose structures of various populations [22] and cloud-based users verification systems [23].

In this article simulated annealing algorithm was used to solve a birthday paradox, where implemented procedure was made to calculate possibility of similarity in dates.

## III. Birthday Paradox Problem

Common probability problem, proposed by Richard von Mises in 1939 can be stated: what is the minimal number n of people in the randomly chosen group for whom the probability that some pair of them will share the same birthday is greater than there is no pair like that? In other words, probability that there are two people with the same birthday date must be greater than $50\%$. The answer is that there must be at least 23 people in the random group. Many people says that it is surprisingly little number and that is why problem is called paradox. Pigeonhole principle says that probability reaches $100\%$ when there are at least 366 (or 367 at leap years) people in the group, so for $50\%$ likelihood there should be 183 (184) people. The thing is that in the group of 23 people there is more than 22 comparisons. We have to compare everyone to everyone, not only one person. This way, for 23 people we have $\frac{23 \cdot 22}{2} = 253$ comparisons. Another counter - intuitive thing is that growth of the probabilities, which depends of number of people, is not linear. To simplify the problem we can make a few assumptions:

1) no leap years, every year has 365 days
2) two people have the same birthday when month and day are the same, year is ignored
3) all dates are equally likely (in fact, more babies are born in Spring then in other seasons)
4) multiple births are considered as one birthday

## IV. Traditional Approach

Sometimes it is easier to calculate probability of the opposite event to ours. In our case, instead of calculating probability that two people have birthday at the same day, we will find probability that they don't. For showing it, we can use inequality that follows from probability:

$$\begin{aligned} \overline{p}(k,n) &= p_k = \\ &= 1 \cdot (1-\tfrac{1}{n}) \cdot (1-\tfrac{2}{n}) \cdot \ldots \cdot (1-\tfrac{k-1}{n}) = \\ &= \textstyle\prod_{i=1}^{k-1} (1-\tfrac{i}{n}) \end{aligned} \tag{1}$$

where $\overline{p}(k,n)$ is the probability that sequence of $k$ elements (number of people) chosen from $n$ elements (365 days of the year) will be injective. Let's note it $p_k$ using 1.
This event is opposite to ours. Because we want our event to be more probable than 50%, $0 \leq p_k \leq 0,5$. We have to find minimal $k$ for which $p_k \leq 0,5$.

By using inequality $1 + x \leq e^x$ that is true $\forall x \in R$, we can estimate $p_k$:

$$\begin{aligned} p_k &= (1-\tfrac{1}{365}) \cdot (1-\tfrac{2}{365}) \cdot \ldots \cdot (1-\tfrac{k-1}{365}) = \\ &\leq e^{-\frac{1}{365}} \cdot e^{-\frac{2}{365}} \cdot \ldots \cdot e^{-\frac{k-1}{365}} = \\ &= e^{-\frac{1+2+\ldots+(k-1)}{365}} = \\ &= e^{-\frac{k(k-1)}{730}} \end{aligned} \tag{2}$$

To satisfy $p_k \leq 0,5$, we need to find the minimal $k$, for which we have

$$k^2 - k - 730 \geq 0 \tag{3}$$

The least positive solution of this inequality is

$$\frac{1+\sqrt{1+4 \cdot 730 \cdot \ln 2}}{2} = 22.99994315 \tag{4}$$

Hence, the analytical solution of the problem is $k = 23$.

Due to shortening operation time of every program, we are looking for the fastest solutions. This is the main reason for using CI to solve birthday paradox problem. We want to know how many people must be in the randomly chosen group to make sure that probability that there is a pair of people that have birthday at the same day is greater than 50%. By doing it on traditional way, we have to take many samples of 1,2,3,... person group (randomly chosen) and count the probability. By using CI we can get the minimum number of people much quicker.

## V. Simulated Annealing Method

Annealing is a metallurgical process based on heating the metal up to a high temperature, then keeping it at given conditions and after that, slow cooling it down. The last stage of this process is the most important step to achieve final conditions. It has to be monitored in order to keep the metal in the state similar to thermodynamic equilibrium, i.e. the state in which parameters such as volume and pressure are constant in time. We have three main elements that thermodynamic equilibrium consists of:

1) thermal equilibrium - constant temperature as a result of no heat exchange with the environment
2) mechanical equilibrium - constant pressure at any point
3) chemical equilibrium - no chemical reactions, no changes in the structure of the metal

We can describe the thermodynamics of the whole process by equation below:

$$P(E) \approx e^{-\frac{E}{kT}} \tag{5}$$

where $E$ is the thermodynamic system, $T$ is the absolute temperature and $k$ is the Boltzmann's constant.

### A. Mathematical Model

Mathematical models are to describe processes and relations that are present in nature, science and technology by application of devoted sequences of equations describing modeled situation in a mathematical way, where we use these equations to calculate the state of the simulated objects in initial conditions and convert it after changes in the following operations. These operations are performed by application of various computer procedures where we use computational power to perform numerical experiments simulating the object. In the presented approach one of important CI methods was implemented to solve the birthday paradox in a way similar to annealing processes i.e. discussed for application in verification systems [23] or compose structures of various populations [22].

Simulated Annealing Method (SAM) assumes that temperature at the beginning of the process is high. It enables frequent changes in configurations. When the temperature is lower, there is less possibility for choosing the worst solution so it is the criterion of acceptation of the solution. Therefore, we use modified equation (5) in a simplified form:

$$P(E) \approx e^{-\frac{\delta}{T}} \tag{6}$$

where $\delta$ is the difference between the value of fitness function calculated at the new random solution chosen from neighborhood $x'$ and current solution $x$ according to:

$$\delta = f(x') - f(x) \tag{7}$$

For a benchmark tests a simplified fitness condition was chosen

$$f(x) = \frac{x}{2} \tag{8}$$

This equation is enough to perform experiments since linear function is enough to simulate controlled growth of numerical data in this experiment. For a new solution we have criterion of acceptance:

$$\gamma < e^{-\frac{\delta}{T}} \tag{9}$$

where $\gamma$ is chosen randomly, and $\gamma \in (0,1)$. Change of temperature we can denote as :

$$T_{k+1} = T_k \cdot r \tag{10}$$

where $T_k$ is the temperature in the $k$-th iteration and $r$ is constant given at the beginning, where $r \in (0,1)$. For the benchmark test stop criterion was adapted to the modeled object.

### *B. Implemented Algorithm*

In the test method presented in Algorithm 1 was implemented, for which we can also present a block diagram show in Fig. 1. Firstly we establish initial values and random initial solution x. The list *date* was created to remember x and next solutions which satisfy our algorithm. When new solution, y, gratify condition (9), we add it to the list *date*. After every iteration we check if there are two the same numbers in the list *date*. If so, we break our loop and save the length of the *list* in the next list *average*. After clearing *date*, we are doing the same as long as length of *average* is less then or equal to number of samples in each iteration which was given at the beginning. When *average* is complete, we take the average value of all elements from *average* and that is our result for given parameters.

**Algorithm 1** The contrast enhancement algorithm with a threshold value

Define the value of the initial temperature $T$, the fitness function $f(\cdot)$, the radius of neighbourhood $p$, the temperature change $r$ and number of samples in each iteration $pr$,
Set $counter := 0$ and $k := 0$,
Declare lists: $date$, $average$,
**while** $counter \leq pr$ **do**
    $decision = 0$,
    Generate random initial solution $x$,
    Add $x$ to the list $date$,
    $k := 1$,
    **while** $decision = 0$ **do**
        Generate a random neighboring solution $y$,
        Calculate the difference delta using (7),
        **if** $delta < 0$ **then**
            Add $y$ to the list $date$,
            $x = y$,
            Increase the iterator variable $k++$,
        **else**
            Generate a random value $gamma$,
            **if** $gamma$ satisfy equation (9) **then**
                Add $y$ to the list $date$,
                $y = x$,
                Increase the iterator variable $k++$,
            **end if**
        **end if**
        **for** $h = 0$ to $k - 1$ **do**
            **if** $date[h] = x$ **then**
                Add $k$ to the list $average$,
                Increase the iterator variable $counter++$,
                $decision = 1$,
                Clear the list $date$,
                Break the loop,
            **end if**
        **end for**
        Reduce the temperature using (10),
    **end while**
**end while**
**for** $i = 0$ to $pr$ **do**
    $Sum = Sum + average[i]$,
**end for**
$Result = round(sum/pr)$,
Return $Result$.

## VI. Benchmark Tests

Results of numerical experiments are shown in Tab. II - Tab. III. For these results changes of probability that among selected population are people for whom birthday paradox can be encountered are presented in Tab. I and depicted in Fig. 2. For each set of parameters, 28 results has been received and then by taking the average of them, we get out final result - number of people. Outcomes presented in Tab. II - Tab. III are very close to expected 23. In two cases we get exactly this number - presented in Tab. III for parameters $T = 1050$, $p = 103$, $r = 0,84$, $pr = 850$ and $T = 1049$, $p = 103$, $r = 0,83$, $pr = 850$, as we can see, very similar to each other.

### *A. Conclusions*

We can state parameters for which, by rounding out we get the expected value for our paradox - 23. As a fitness function linear one was chosen but it turn out that for power and exponential function we obtain similar results. The most important was choosing optimal parameters. The greatest impact for value of result have initial temperature and radius of neighbourhood - along with their decrease, values of results are also lower. Drop of number of samples in each iteration causes bigger range of received solutions. What is surprising, different values of the temperature change, don't change the results.

## VII. Final Remarks

In this article simulated annealing method was used to solve a problem of birthday paradox. This is definitely better way to obtain solution than traditional counting probability by taking many samples of 1,2,3,... person groups. By using CI methods,we get the solution easier and what is very important, faster. Benchmark tests have been performed to indicate the best paramaters for our algorithm.

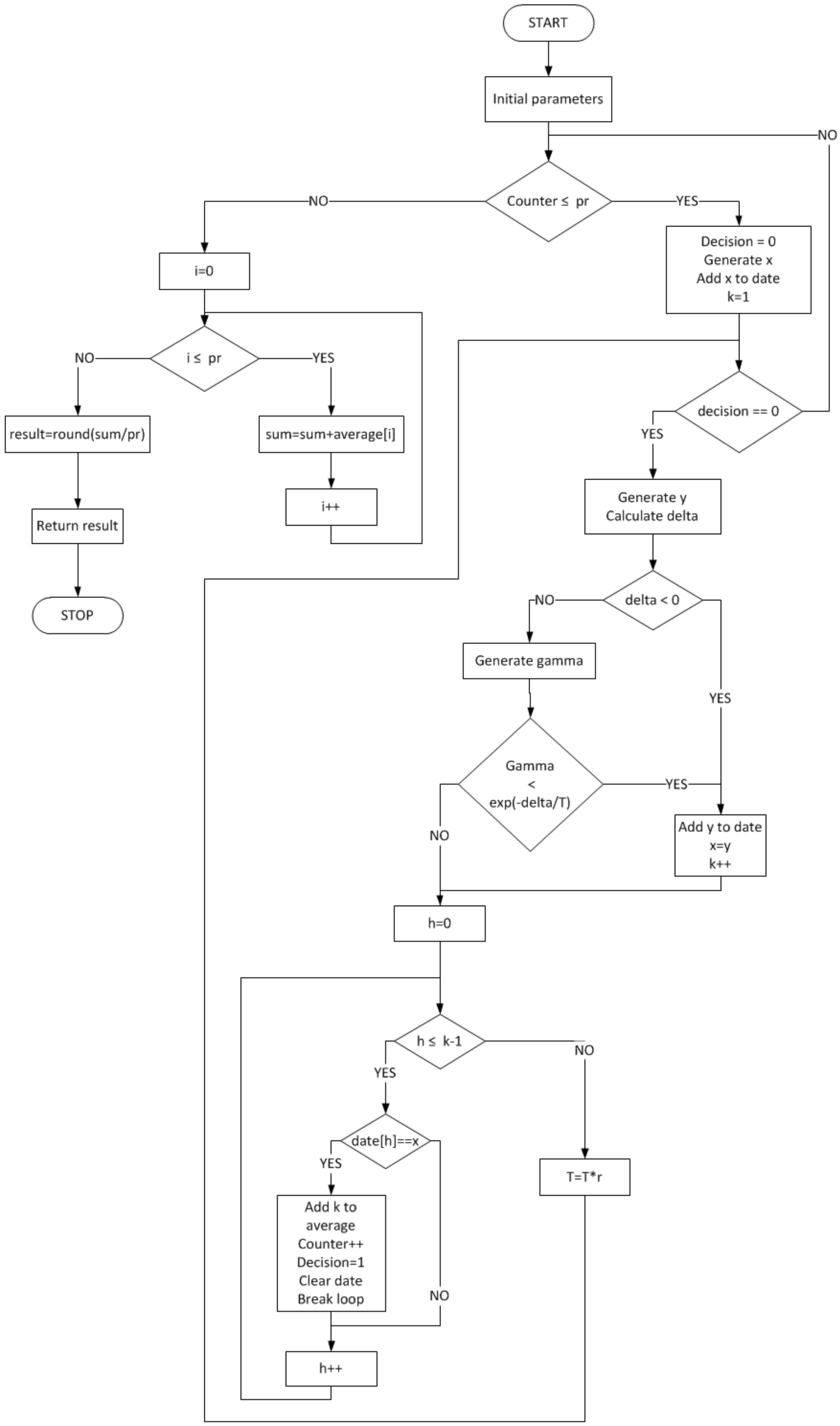


Fig. 1: Block diagram of the implemented processing software.

TABLE I: Probability of sharing birthday

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| number of people | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| probability of sharing birthday | 0,26% | 0,83% | 1,56% | 2,72% | 4,17% | 5,42% | 7,16% | 9,40% | 11,59% |
| number of people | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| probability of sharing birthday | 14,41% | 16,60% | 20,03% | 22,00% | 25,39% | 28,54% | 31,66% | 34,75% | 37,94% |
| number of people | 20 | 21 | 22 | 23 | 24 | 25 | 30 | 35 | 50 |
| probability of sharing birthday | 40,89% | 43,66% | 47,51% | 50,48% | 53,83% | 56,87% | 70,63% | 81,44% | 97,00% |

TABLE II: Results of numerical experiments

| T | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1100 | 1001 | 1000 | 1000 | 1000 | 1000 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| p | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 110 | 105 | 104 | 102 | 100 |
| r | 0,9 | 0,9 | 0,9 | 0,9 | 0,8 | 0,85 | 0,85 | 0,85 | 0,85 | 0,85 | 0,85 | 0,85 | 0,84 |
| pr | 700 | 800 | 900 | 750 | 800 | 800 | 800 | 800 | 800 | 800 | 800 | 800 | 800 |
| | 23 | 22 | 23 | 24 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 24 | 22 | 23 | 22 | 22 | 23 | 24 | 23 | 23 | 22 | 24 | 24 | 23 |
| | 23 | 23 | 22 | 22 | 23 | 22 | 24 | 23 | 24 | 23 | 23 | 23 | 22 |
| | 22 | 23 | 23 | 24 | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 22 |
| | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 24 | 23 | 23 | 23 | 23 |
| | 24 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 24 |
| | 23 | 23 | 23 | 23 | 23 | 24 | 23 | 23 | 23 | 23 | 22 | 22 | 22 |
| | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 24 | 22 | 24 | 24 | 23 |
| | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 24 | 23 | 23 | 23 | 22 |
| | 24 | 23 | 23 | 22 | 23 | 23 | 22 | 23 | 23 | 22 | 23 | 23 | 22 |
| | 22 | 23 | 22 | 23 | 22 | 23 | 22 | 22 | 23 | 23 | 23 | 23 | 22 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 24 | 23 | 23 | 23 |
| | 23 | 23 | 23 | 23 | 22 | 23 | 22 | 22 | 23 | 23 | 23 | 23 | 23 |
| | 22 | 23 | 23 | 22 | 22 | 23 | 23 | 23 | 24 | 23 | 23 | 23 | 23 |
| | 22 | 22 | 22 | 23 | 23 | 23 | 23 | 22 | 23 | 24 | 23 | 23 | 23 |
| | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 24 | 24 | 23 | 23 | 23 |
| | 22 | 24 | 22 | 23 | 23 | 23 | 23 | 23 | 24 | 23 | 22 | 22 | 23 |
| | 23 | 24 | 23 | 23 | 22 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 22 | 23 | 22 | 23 | 23 | 22 | 22 | 23 | 23 | 24 | 24 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 22 | 23 | 23 | 23 | 23 | 23 |
| | 22 | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 22 | 23 | 23 | 23 | 23 | 24 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 22 | 22 | 22 | 23 | 22 | 23 | 22 | 23 | 24 | 23 | 23 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 |
| | 22 | 23 | 23 | 22 | 23 | 24 | 22 | 24 | 23 | 23 | 24 | 24 | 23 |
| | 23 | 24 | 22 | 22 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| average | 22,79 | 22,89 | 22,79 | 22,75 | 22,75 | 22,96 | 22,89 | 22,75 | 23,18 | 23,04 | 23,07 | 23,07 | 22,82 |

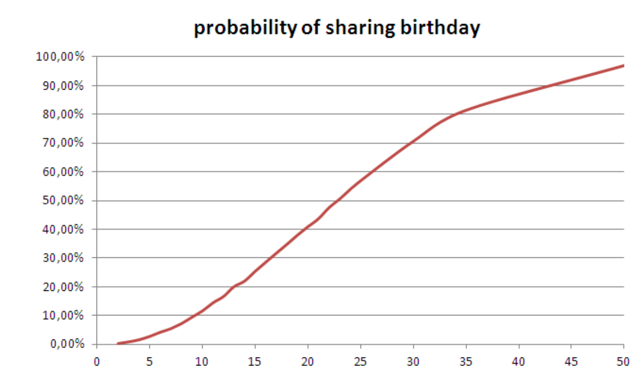


Fig. 2: Chart of changes of probability that among selected population we can encounter a birthday paradox.

## 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] M. Woźniak, D. Połap, L. Kosmider, C. Napoli, and E. Tramontana, "A novel approach toward x-ray images classifier," in *Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI)*. 8-10 December, Cape Town, RSA: IEEE, 2015, pp. 1635–1641, DOI: 10.1109/SSCI.2015.230.

[3] 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.

[4] D. Połap, M. Woźniak, C. Napoli, E. Tramontana, and R. Damaševičius, "Is the colony of ants able to recognize graphic objects?" *Proceedings of International Conference on Information and Software Technologies (ICIST)*, vol. 538, pp. 376–387, 2015, DOI: 10.1007/978-3-319-24770-0_33.

[5] M. Woźniak, D. Połap, M. Gabryel, R. K. Nowicki, C. Napoli, and E. Tramontana, "Can we preprocess 2d images using artificial bee colony?" *Proceedings of International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, vol. 9119, pp. 660–671, 2015, DOI: 10.1007/978-3-319-19324-3_59.

[6] 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.

TABLE III: Results of numerical experiments

| T | 1000 | 1000 | 1000 | 1000 | 1100 | 1000 | 1100 | 1050 | 1040 | 1045 | 1048 | 1049 | 1049 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| p | 100 | 100 | 100 | 102 | 100 | 103 | 103 | 103 | 103 | 103 | 103 | 103 | 103 |
| r | 0,83 | 0,83 | 0,84 | 0,84 | 0,84 | 0,84 | 0,84 | 0,84 | 0,84 | 0,84 | 0,84 | 0,84 | 0,83 |
| pr | 800 | 850 | 850 | 850 | 850 | 850 | 850 | 850 | 850 | 850 | 850 | 850 | 850 |
| | 22 | 23 | 23 | 24 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 24 | 23 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 24 | 23 |
| | 23 | 23 | 22 | 22 | 23 | 22 | 23 | 23 | 23 | 22 | 23 | 24 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 23 |
| | 23 | 23 | 24 | 22 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 23 |
| | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 22 | 23 | 22 | 23 | 22 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 22 | 23 | 24 | 23 | 23 | 22 | 23 | 23 |
| | 23 | 23 | 22 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 22 |
| | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 24 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 24 | 23 | 22 | 23 | 23 | 23 | 23 |
| | 22 | 23 | 23 | 23 | 22 | 22 | 23 | 23 | 23 | 22 | 23 | 23 | 22 |
| | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 22 | 22 | 23 |
| | 23 | 23 | 23 | 23 | 22 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 23 |
| | 22 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 23 |
| | 23 | 23 | 23 | 23 | 22 | 23 | 22 | 23 | 23 | 23 | 23 | 24 | 23 |
| | 23 | 23 | 22 | 23 | 23 | 22 | 23 | 23 | 23 | 22 | 23 | 23 | 23 |
| | 23 | 23 | 23 | 24 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 22 | 22 | 23 | 22 | 23 | 22 |
| | 23 | 23 | 22 | 23 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 24 |
| | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| | 22 | 22 | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 22 | 24 | 23 | 23 |
| | 23 | 22 | 23 | 23 | 22 | 23 | 22 | 24 | 23 | 23 | 23 | 23 | 23 |
| | 23 | 22 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 22 | 23 | 23 | 24 |
| average | 22,75 | 22,82 | 22,82 | 22,96 | 22,68 | 22,82 | 22,86 | 23 | 22,89 | 22,79 | 22,96 | 22,96 | 23 |


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

[8] 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.

[9] 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.

[10] K. Chandrasekaran and S. Simon, "Multi-objective scheduling problem: hybrid approach using fuzzy assisted cuckoo search algorithm," *Swarm and Evolutionary Computation*, vol. 5, no. 1, pp. 1–16, 2012.

[11] A. Baykasoğlu and F. B. Ozsoydan, "An improved firefly algorithm for solving dynamic multidimensional knapsack problems," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3712–3725, 2014.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] S. Kirkpatrick and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[18] L. Ingber, "Very fast simulated re-annealing," *Mathematical and Computer Modelling*, vol. 12, no. 8, pp. 967–973, 1989.

[19] 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.

[20] 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.

[21] D. Svergun, "Restoring low resolution structure of biological macromolecules from solution scattering using simulated annealing," *Biophysical Journal*, vol. 76, no. 6, pp. 2879–2886, 1999.

[22] 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.

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