An Idea to Apply Firefly Algorithm in 2D Image Key-Points Search

Marcin Woźniak and Zbigniew Marszałek

Institute of Mathematics, Silesian University of Technology,
ul. Kaszubska 23, 44-100 Gliwice, Poland
{Marcin.Wozniak,Zbigniew.Marszalek}@polsl.pl

Abstract. In this paper we discuss the idea to apply evolutionary computation method, in particular firefly algorithm, to search for key-points in 2D images. In the research, classic firefly algorithm is used to search for special areas in test images. Research results are presented and discussed to show potential efficiency of applied method.

Keywords: evolutionary computation, firefly algorithm, key-points search, 2D image classifier, 2D image processing.

1 Introduction

Evolutionary computation (EC) is one of most important fields in computer science. One may present it's applications in different sciences, projects and industry. EC represents power of computational intelligence (CI), where dedicated mechanisms solve complex problems. Among these methods one can name genetic algorithms (GA), evolutionary strategies (ES) and heuristic algorithms (HA). These methods are mapping behavior of real life to solve given task. They are efficient in positioning, simulating different objects and complex optimization. One can also find their efficiency in search for optimal solutions and easy implementation with high precision. Let us give some examples.

In [8] was presented application of ES to create learning sets for artificial intelligence (AI) control systems. LAN models positioning and optimization by the use of GA or ES is presented in [9], [29], [30] and [31]. GA are also efficient in optimization of medical diagnostic classifiers (see [20]) and positioning of ultrasound surgery (see [21]). They also play crucial role in manufacturing systems (see [14]). HA may be applied in industrial processes simulations, i.e. iron cast as shown in [15]. HA are adaptive to conditions of diverse fitness functions as presented in [18]. Efficiency of GA versus analytical approach for optimization of solar thermal electricity plants is discussed in [4]. While stability and convergence of EC is discussed in [2], [5], [10], [23] and [34]. Some hybrid EC methods are applied in model of airport gate scheduling as presented in [11]. EC are efficient in FUZZY-PID controllers positioning (see [28]), job scheduling problems (see [38]) and image processing (see [33]). As You see EC may be applied in various and complicated models which describe sophisticated co-working subsystems

G. Dregvaite and R. Damasevicius (Eds.): ICIST 2014, CCIS 465, pp. 312–323, 2014.
© Springer International Publishing Switzerland 2014

and complex problems. These convinced us to assess it's efficiency in 2D image processing.

In this paper we aim to show possibility of using EC, in particular firefly algorithm (FA), to search for key-points in 2D images. For the research were taken sample images from open test images databases *www.imageprocessingplace.com* and *www.ece.utk.edu/gonzalez/ipweb2e/*. In following sections we present classic attempt and FA application to search for key areas. Applied FA presents high precision, is easy to implement and solves given task very fast. This makes presented solution efficient.

2 Key-Points Search Methods

There are many aspects where EC can help to solve given problems and analyze 2D images. In [32] is discussed application of some algorithms in preprocessing human signatures for AI classifiers. In [12] EC methods are used to design shape imitation. Here we would like to discuss potential application of EC in the process of 2D image classification.

Computer image is composed of points, which have special position and properties. Among them are saturation, sharpness, brightness and more. All these features compose objects visible to our eyes and give information about objects in the picture. This information can help to identify somebody or something. Therefore position of each pixel (each one has measurable coordinates $X = (x, y) = (x_{i,1}, x_{i,2})$) and it's properties (i.e. brightness and saturation) are crucial for classification. We implemented EC method to find areas that contain many pixels of the same kind, which can be classified by AI system.

2.1 Classic Attempt

One of classic methods used for key-points recognition is SURF (Speeded-Up Robust Features) algorithm. This method gives description of the image by selecting characteristic key-points. Here is given only a short presentation, for more details please see [1], [3], [7], [13] and [22].

Our SURF combines selection of key-points with calculating 64-element vector (descriptor). In SURF is applied integrated image and filter approximation of block Hessian determinant. To detect interesting points is used particular Hessian-matrix approximation (see [1] and [3]). For point $X = (x, y)$ in the image is defined Hessian matrix $H(X, \sigma)$ in X at scale σ using formula

$$H(X,\sigma) = \begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix}, \tag{1}$$

where the symbols are: $L_{xx}(X, \sigma)$ – convolution of Gaussian second order derivative $\frac{\partial^2}{\partial x^2}$. We define approximation D_{xx}, D_{yy} and D_{xy} using formula

$$det(H_{ap}) = D_{xx}D_{yy} - \left(\frac{|L_{xy}(\sigma)|_F |D_{xx}(\sigma)|_F}{|L_{xx}(\sigma)|_F |D_{xy}(\sigma)|_F} D_{xy} \right)^2 . \tag{2}$$

Then, image is blurred to get DoG (Difference of Gaussian) images, which helps to find edges. To localize interesting points is used non maximum suppression in $3 \times 3 \times 3$ neighborhood. Maximum determinant of Hessian matrix is interpolated at scale σ, which helps to differ between first level and each octave. In [3] or [7] SURF descriptor is based on similar properties. First is fixed reproducible orientation based on information from circular region around pixel. Then, is constructed square region aligned to selected orientation and SURF descriptor is extracted from it. Please see [1] and [3] for more details.

2.2 Firefly Algorithm

Firefly algorithm (FA) maps behavior of flying insects that we all can see in the summer - fireflies. This method maps them while searching for best partner. In the research we have implemented this process for 2D image key-points search.

One of first versions of FA was presented in [34], [35] or [36]. Since then it was efficiently applied in many fields. In [6] chaotic FA is applied to reliability-redundancy optimization. In [16] FA is applied to minimum cross entropy threshold selection. In [19] this method is applied to solve traveling salesman problem. It is efficient in continuous optimization (see [37]) and multi-modal optimization (see [34]). FA is also efficient in image compression, please see [17]. All these convinced us to apply classic FA in the process of 2D image classification. Let us now present mathematical model of classic FA algorithm applied to search for image key-points.

The method describes behavior of fireflies in natural conditions, characterized by several biological traits:

- Specific way of flashing.
- Specific way of moving.
- Specific perception of other individuals.

These features are modeled with numerical values. In this way we translate natural characteristics of biological organisms on mathematical model used to develop specific EC method. Thus, in implementation of classic FA we assume:

- I_{pop}–light intensity factor for given species.
- γ–absorption coefficient of light in given circumstances.
- β–factor for attractiveness of firefly species.
- μ–factor for random motion of individual.

In description of FA we also use the following assumptions:

- All fireflies are unisex, therefore one individual can be attracted to any other firefly regardless of gender.
- Attractiveness is proportional to brightness. Thus, for every two fireflies less clear flashing one will move toward brighter one.
- Attractiveness is proportional to brightness and decreases with increasing distance between individuals.

– If there is no clearer and more visible firefly within the range, then each one will move randomly.
– Firefly and pixel (2D image point) are equal in FA algorithm.

Distance between any two fireflies i and j situated at points (pixels) X_i and X_j in the picture is defined as Cartesian metric

$$r_{ij}^t = \|X_i^t - X_j^t\| = \sqrt{\sum_{k=1}^{2}(x_{i,k}^t - x_{j,k}^t)^2}, \tag{3}$$

where the symbols are: X_i^t, X_j^t–pixels in the picture in t iteration, $x_{i,k}^t$, $x_{k,j}^t$–k-th components of spatial coordinates X_i^t and X_j^t that describe each firefly position (pixel in the image) measured in t iteration.

Light intensity I_{ij}^t from firefly i that is received by firefly j decreases with increasing distance r_{ij}^t between them. Natural light is absorbed by media, so attractiveness also vary according to absorption and distance between them. In the model light intensity varies according to

$$I_{ij}^t(r_{ij}^t) = I_{pop} \cdot e^{-\gamma \cdot (r_{ij}^t)^2}, \tag{4}$$

where the symbols are: $I_{ij}^t(r_{ij}^t)$–intensity of light from firefly i that is received by firefly j in t iteration, r_{ij}^t–distance between firefly i and firefly j defined in (3), γ–light absorption coefficient mapping natural conditions.

Attractiveness of firefly i to firefly j decreases with increasing distance. Attractiveness is proportional to intensity of light seen by surrounding individuals and can be defined by formula

$$\beta_{ij}^t(r_{ij}^t) = \beta_{pop} \cdot \frac{1}{1 - e^{-\gamma \cdot (r_{ij}^t)^2}}, \tag{5}$$

where the symbols are: $\beta_{ij}^t(r_{ij}^t)$–attractiveness of firefly i to firefly j in t iteration, r_{ij}^t–distance between firefly i and firefly j defined in (3), γ–light absorption coefficient mapping natural conditions.

Movement of individual is based on conditioned distance to other individuals surrounding it. Firefly will go to most attractive one, measuring intensity of flicker over the distance between them. In given model, natural identification of individuals and their attractiveness defined in (5) depends on light intensity defined in (4) and distance separating them defined in (3). In nature fireflies that are closer not only see themselves better, but also are more attractive to each other. Using these features in the model, calculations remap natural behavior of fireflies. Firefly i motions toward more attractive and brighter (clearer flashing) individual j using information about other individuals denotes formula

$$X_i^{t+1} = X_i^t + (X_j^t - X_i^t) \cdot \beta_{ij}^t(r_{ij}^t) \cdot I_{ij}^t(r_{ij}^t) + \mu \cdot e_i, \tag{6}$$

where the symbols are: X_i^t, X_j^t–points in the picture, $\beta_{ij}^t(r_{ij}^t)$–attractiveness of firefly i to firefly j defined in (5), $I_{ij}^t(r_{ij}^t)$–intensity of light from firefly i

that is received by firefly j defined in (4), r_{ij}^t–distance between fireflies i and j defined in (3), γ–light absorption coefficient mapping natural conditions, μ–coefficient mapping natural random motion of individuals in population, e_i–vector randomly changing position of firefly. Using these facts we build CI to map behavior of fireflies in the process of computer algorithm. Let us now present FA implementation and discuss some examples.

Start,
Define all coefficients: I_{pop}–light intensity, γ–light absorption,
β_{pop}–attractiveness, μ–natural random motion, number of $fireflies$ and
$generation$–number of iterations in the algorithm,
Define fitness function for the algorithm using (7),
Create at random initial population P of $fireflies$ in 2D image,
$t = 0$,
while $t \leq generation$ **do**
    Calculate distance between individuals in population P using (3),
    Calculate light intensity for individuals in population P using (4),
    Calculate attractiveness for individuals in population P using (5),
    Evaluate individuals in population P using (7),
    Create population O: move individuals towards closest, brightest and most attractive individual using (6),
    Evaluate individuals in population O using (7),
    Replace δ worst individuals from P with δ best individuals from O,
    The rest of individuals take at random,
    Next generation $t = t + 1$,
end
Values from population P with best fitness are solution,
Stop.

Algorithm 1: FA to classify 2D images key-points

3 Research Results

Algorithm 1 presented in section 2.2 was applied to search for 2D image key-points. Each firefly is representing single pixel (point in image). Population move from pixel to pixel and search for specific areas. In the research we have used simplified fitness function. This function reflects quality of image points as

$$\Phi((x_{i,1}, x_{i,2})) = \varphi_i = \begin{cases} 0.1 \ldots 1 & \text{for points of defined saturation} \\ 0 & \text{other} \end{cases}, \tag{7}$$

where symbol φ_i denotes brightness and sharpness of evaluated pixel. This measure reflects value in scale from 0.0 to 1.0, where colors might change from black to white. When fireflies fly in iteration, they pick pixels with best fitness within the range of their flight. Then from all individuals we take δ of them, where fitness function is highest or lowest (depending on experiment). These points (fireflies) are taken to next iteration and the rest of population is taken at random from all image points. Random points selection in each iteration helps to search entire image for the best points of interest.

Simulations were performed for 120 fireflies in 20 generations with set coefficients: $I_{pop} = 0.25$, $\beta_{pop} = 0.15$, $\gamma = 0.3$, $\mu = 0.25$, $\delta = 30\%$. As objects for examinations were taken standard test images, downloaded from open test image databases (see section 2). We have performed experiments on different types of pictures: sharp, blurred, landscapes, human postures and faces. In the research we were looking for special objects concerning brightness and saturation of pixels defined in standard way. Therefore, using FA from section 2.2 we searched for key areas. Each of resulting key-points (pixels) is marked in red. We have provided some close-ups of classified areas for better presentation. First we tried to find dark areas where $\varphi_i = 0$ or $\varphi_i = 0.1$. In second attempt we were looking for bright areas where $\varphi_i = 0.9$ or $\varphi_i = 1$.

3.1 Dark Areas in Images

Fig. 1. Dark key-areas in human posture images: on the left SURF, on the right FA

Dark objects are present in various images. They can represent objects in landscape (constructions, blocks, etc.), natural phenomena (tornadoes, shadows, etc.), human figures or human appearance (clothes, face features, hair, eyes, etc.). In Fig.1 - Fig.4 are presented research results for key-points representing dark areas. We can see that classic FA can easily find dark objects of different shapes like human hair, clothes and some appearance features. It is also efficient in finding some aspects in landscapes. In Fig.3 and Fig.4 are presented research results of searching for shades under constructions or dark parts of machinery. All these areas were found by FA using very small number of fireflies and iterations. Let us now present research results for bright objects localization.

Fig. 2. Dark key-areas in human face images: on the left SURF, on the right FA

Fig. 3. Dark key-areas in architecture images: on the left SURF, on the right FA

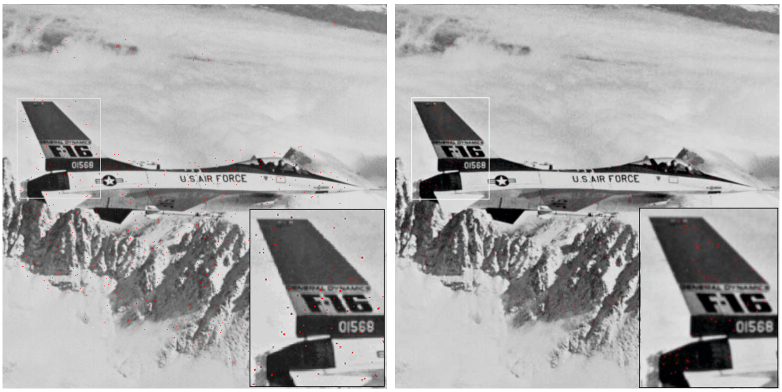

Fig. 4. Dark key-areas in machinery images: on the left SURF, on the right FA

3.2 Bright Areas in Images

Bright areas can represent objects in landscape (bright or lightened constructions, machinery, etc.), natural phenomena (clouds, sun, etc.), human figures or human appearance (gray hair, eyes, bright clothing, etc.). In Fig.5 - Fig.8 are presented research results for key-points representing bright areas in images. We can see that classic FA can easily find bright objects of different shapes. FA pointed human faces or bright clothes better than SURF (see Fig. 5 and Fig.6). It is also efficient in locating bright or lightened constructions like machinery or buildings present in Fig.7 and Fig.8.

Fig. 5. Bright key-areas in human posture images: on the left SURF, on the right FA

Fig. 6. Bright key-areas in human face images: on the left SURF, on the right FA

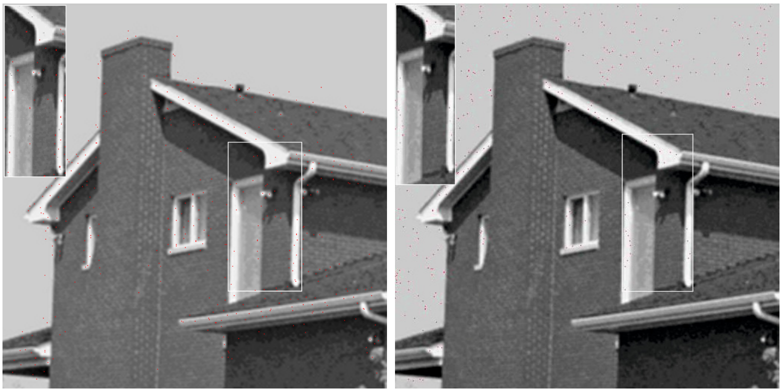

Fig. 7. Bright key-areas in architecture images: on the left SURF, on the right FA

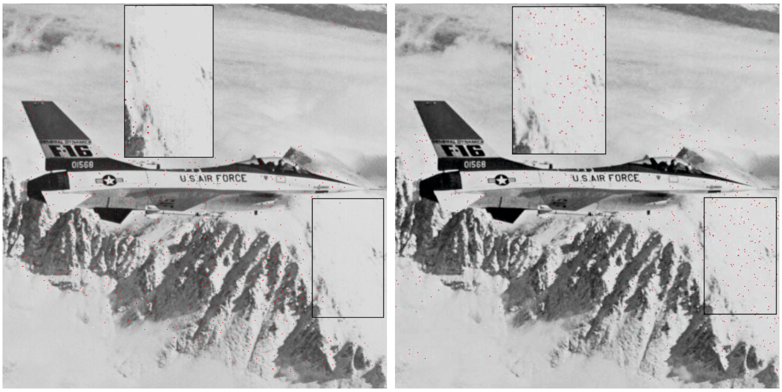

Fig. 8. Bright key-areas in machinery images: on the left SURF, on the right FA

3.3 Conclusions

Application of classic FA allows to easily and reliably find key-areas in examined images. Algorithm efficiency is increased if we are looking for key-points with high contrast in relation to surroundings (see Fig.1 and Fig.3). If we are processing images where many points are of the same valor, classification may be more complicated. There, algorithms must find bright points among many pixels of similar kind or brightest/darkest among many bright/dark ones. If photos are taken during day, all objects of bright properties are lightened in some way (see Fig.8). Similarly for night photos objects are darker. Therefore FA calculations are slightly more complex. For example in Fig.7 we were looking for bright constructions in highly sun-lighted photos. Moreover there are some clouds on the sky. All these made classification process complicated. However similarly to dark areas search, all bright areas were found by classic FA even using even small number of fireflies and iterations.

In conclusions we see that presented FA can find whole areas (objects) of interest, covering them with found key-points. This feature makes it promising tool for AI recognition systems and image classifiers. Moreover calculations performed by FA method are simple. We just use formulas (3) – (6) to calculate position and perform move of each point in examined images.

4 Final Remarks

Application of EC methods, in particular classic FA, to search for key-points allows to easily and reliably select areas of interest. At the same time, EC methods allow to easily explore entire 2D image in search for selected objects without many complicated mathematical operations. Presented idea to apply FA is efficient.

EC methods are efficient in positioning or optimization of different systems, like queuing models (see [29], [31] or [30]). One can find other EC methods efficiently applied in the process of data acquisition (see [8]) or dynamic object positioning (see [24], [25], [26] and [27]). Research presented in this paper show that EC methods, in particular FA, are also excellent to perform process of 2D image classification for AI systems. Therefore similar experiments may be performed using other EC methods, their modifications and other input objects.

References

1. Abeles, P.: Speeding Up SURF. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Li, B., Porikli, F., Zordan, V., Klosowski, J., Coquillart, S., Luo, X., Chen, M., Gotz, D. (eds.) ISVC 2013, Part II. LNCS, vol. 8034, pp. 454–464. Springer, Heidelberg (2013)
2. Baonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press (1999)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding 110(3), 346–359 (2008)
4. Cabello, J.M., Cejudo, J.M., Luque, M., Ruiz, F., Deb, K., Tewari, R.: Optimization of the Sizing of a Solar Thermal Electricity Plant: Mathematical Programming Versus Genetic Algorithms. In: IEEE Congress on Evolutionary Computation (CEC 2009), pp. 1193–1200. IEEE (2009)
5. Clerc, M., Kennedy, J.: The Particle Swarm-Explosion, Stability and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
6. dos Santos Coelho, L., de Andrade Bernert, D.L., Mariani, V.C.: A chaotic firefly algorithm applied to reliability-redundancy optimization. In: IEEE Congress on Evolutionary Computation (CEC 2011), pp. 517–521. IEEE (2011)
7. Decker, P., Paulus, D.: Model Based Pose Estimation Using SURF. In: Koch, R., Huang, F. (eds.) ACCV 2010 Workshops, Part II. LNCS, vol. 6469, pp. 11–20. Springer, Heidelberg (2011)
8. Gabryel, M., Woźniak, M., Nowicki, R.K.: Creating learning sets for control systems using an evolutionary method. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) EC 2012 and SIDE 2012. LNCS, vol. 7269, pp. 206–213. Springer, Heidelberg (2012)

9. Gabryel, M., Nowicki, R.K., Woźniak, M., Kempa, W.M.: Genetic cost optimization of the GI/M/1/N finite-buffer queue with a single vacation policy. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part II. LNCS, vol. 7895, pp. 12–23. Springer, Heidelberg (2013)
10. Gazi, V., Passino, K.M.: Swarm stability and optimization. Springer, Heidelberg (2011)
11. Gómez de Silva Garza, A., Torres Campos Licastro, P.A., Ogando Justo, R.M.: A Hybrid Knowledge-Based And Evolutionary Process Model Of Airport Gate Scheduling. International Journal of Uncertainty, Fuzziness and Knowledge-Based System 12(2), 43–62 (2004)
12. Gómez de Silva Garza, A., Zamora Lores, A.: Evaluating an evolutionary method of design style imitation. Journal Artificial Intelligence for Engineering Design, Analysis and Manufacturing 25(1), 1–13 (2011)
13. Gossow, D., Decker, P., Paulus, D.: An Evaluation of Open Source SURF Implementations. In: Ruiz-del-Solar, J. (ed.) RoboCup 2010. LNCS, vol. 6556, pp. 169–179. Springer, Heidelberg (2010)
14. Guan, Z., Wang, C., Huang, J., Wan, L., Shao, X.: Optimization of Manufacturing Systems using Genetic Search and Muti-Resolution Simulation. In: IEEE International Conference on Control and Automation (ICCA 2010), pp. 1473–1480. IEEE (2010)
15. Hetmaniok, E., Słota, D., Zielonka, A.: Experimental verification of immune recruitment mechanism and clonal selection algorithm applied for solving the inverse problems of pure metal solidification. International Journal of Pure and Applied Mathematics 85(1), 171–178 (2013)
16. Horng, M.H., Liou, R.J.: Multilevel minimum cross entropy threshold selection based on the firefly algorithm. Expert Systems with Applications 38, 14805–14811 (2011)
17. Horng, M.H.: Vector quantization using the firefly algorithm for image compression. Expert Systems with Applications 39, 1078–1091 (2012)
18. Hu, M., Wu, T., Weir, J.D.: An Adaptive Particle Swarm Optimization With Multiple Adaptive Methods. IEEE Transactions on Evolutionary Computation 17(5), 705–720 (2013)
19. Jati, G.K., Suyanto: Evolutionary discrete firefly algorithm for travelling salesman problem. In: Bouchachia, A. (ed.) ICAIS 2011. LNCS (LNAI), vol. 6943, pp. 393–403. Springer, Heidelberg (2011)
20. Kupinski, M.A., Anastasioy, M.A.: Multiobjective Genetic Optimization of Diagnostic Classifiers with Implications for Generating Receiver Operating Characteristic Curves. IEEE Transactions on Medical Imaging 18(8), 675–685 (1999)
21. Lu, M., Wan, M., Xu, F., Wang, X., Zhong, H.: Focused Beam Control for Ultrasound Surgery with Spherical-Section Phased Array: Sound Field Calculation and Genetic Optimization Algorithm. IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control 52(8), 1270–1290 (2005)
22. Mehrotra, H., Majhi, B., Gupta, P.: Annular Iris Recognition Using SURF. In: Chaudhury, S., Mitra, S., Murthy, C.A., Sastry, P.S., Pal, S.K. (eds.) PReMI 2009. LNCS, vol. 5909, pp. 464–469. Springer, Heidelberg (2009)
23. Nandy, S., Sarkar, P.P., Das, A.: Analysis of nature-inspired firefly algorithm based back-propagation neural network training. International Journal of Computer Applications 43(22), 8–16 (2012)
24. Nowak, A., Woźniak, M.: Algorithm for optimization of the active module by the use of genetic algorithm. Acta Mechanica Slovaca 3C, 307–316 (2008)

25. Nowak, A., Woźniak, M.: Multiresolution derives analysis of module mechatronical systems. Journal Mechanika 6(74), 45–51 (2008)
26. Nowak, A., Woźniak, M.: Analysis of the active module mechatronical systems. In: International Conference Mechanika 2008, pp. 371–376. Kaunas University of Technology Press (2008)
27. Nowak, A., Woźniak, M.: Optimization of the active vibroisolation system for operator's cabin with the hydropneumatical element. Transactions of the Universities of Košice 3, 113–116 (2009)
28. Rubaai, A., Castro-Sitiriche, M.J., Ofoli, A.R.: DSP-Based Laboratory Implementation of Hybrid Fuzzy-PID Controller Using Genetic Optimization for High-Performance Motor Drives. IEEE Transactions on Industry Applications 44(6), 1977–1986 (2008)
29. Woźniak, M.: On applying cuckoo search algorithm to positioning GI/M/1/N finite-buffer queue with a single vacation policy. In: Mexican International Conference on Artificial Intelligence (MICAI 2013), pp. 59–64. IEEE (2013)
30. Woźniak, M., Kempa, W.M., Gabryel, M., Nowicki, R.K.: A finite-buffer queue with single vacation policy - analytical study with evolutionary positioning. International Journal of Applied Mathematics and Computer Science 24(4) (in-press, 2014)
31. Woźniak, M., Kempa, W.M., Gabryel, M., Nowicki, R.K., Shao, Z.: On Applying Evolutionary Computation Methods to Optimization of Vacation Cycle Costs in Finite-buffer Queue. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2014, Part I. LNCS (LNAI), vol. 8467, pp. 480–491. Springer, Heidelberg (2014)
32. Woźniak, M., Połap, D., Marszałek, M.: On handwriting preprocessing for 2D object recognition systems. In: Proceedings of International Conference on Advances in Information Processing and Communication Technology (IPCT 2014). The IRED Digital Seek Library, pp. 46–53 (2014)
33. Woźniak, M., Połap, D.: Basic concept of Cuckoo Search Algorithm for 2D images processing with some research results. In: Proceedings of the 11th International Conference on Signal Processing and Multimedia Applications (SIGMAP 2014). SciTePress - INSTICC (accepted, in press, 2014)
34. Yang, X.-S., Deb, S.: Eagle strategy using Lévy walks and firefly algorithm for stochastic optimization. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) NICSO 2010. SCI, vol. 284, pp. 101–111. Springer, Heidelberg (2010)
35. Yang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley & Sons, USA (2010)
36. Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)
37. Yang, X.S., Cui, Z.H., Xiao, R.B., Gandomi, A.H., Karamanoglu, M.: Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Elsevier, Waltham (2013)
38. Yousif, A., Abdullah, A.H., Nor, S.M., Abdelaziz, A.A.: Scheduling jobs on grid computing using firefly algorithm. Journal of Theoretical and Applied Information Technology 33(2), 155–164 (2011)