

Can we preprocess 2D images using Artificial Bee Colony?

Marcin Woźniak¹, Dawid Połap¹, Marcin Gabryel², Robert K. Nowicki²,
Christian Napoli³, Emiliano Tramontana³ .

¹Institute of Mathematics, Silesian University of Technology,
Kaszubska 23, 44-101 Gliwice, Poland
`Marcin.Wozniak@polsl.pl`

²Institute of Computational Intelligence, Czestochowa University of Technology,
Al. Armii Krajowej 36, 42-200 Czestochowa, Poland
`Marcin.Gabryel@iisi.pcz.pl`, `Robert.Nowicki@iisi.pcz.pl`

³Department of Mathematics and Informatics, University of Catania,
Viale A. Doria 6, 95125 Catania, Italy
`napoli@dmi.unict.it`, `tramontana@dmi.unict.it`

Abstract. This paper is to discuss a matter of preprocessing 2D input images by selected methods of Evolutionary Computation. In the following sections we try to analyze possibility of using Artificial Bee Colony algorithm to preprocess input images for classification purposes. Experiments have been performed with the examined method applied on a set of test images, to present and discuss efficacy and precision of recognition.

Keywords: image processing, evolutionary computation, artificial bee colony algorithm, object classifier

1 Introduction

Image recognition is one of the processes, where dedicated computer systems may help to classify input objects. However, this is non-trivial operation and the efficacy is of a paramount importance. Therefore in this paper we try to analyze an application of selected Evolutionary Computation (EC) method, in particular Artificial Bee Colony Algorithm (ABCA), combined with a simplified sobel filter version what creates a 2D input image recognition system.

EC gives interesting solutions that are imitating the behavior of real organisms to efficiently assist in various problems solving. They find application in positioning, simulation, optimization, control and management. Dynamic systems can be positioned by using EC methods: [19], [18] or [22]. Evolutionary Strategies (ES) efficiently help to create learning sets for Artificial Intelligence (AI) control systems as presented in [6] and [3]. Moreover, EC powers the resolution of complicated differential and integral equations like in positioning queueing systems: [5], [25], [16], [24] and [23]. CI can be applied in the optimization of ultrasound surgery (see [11]), FUZZY-PID controllers (see [20]) or simulation of

iron cast and heat transfer (see [7] and [8]). As You see CI can provide many methods that are efficient in various operations. In this paper we will describes the application of Artificial Bee Colony Algorithm (ABCA) in image processing.

1.1 Related Works

EC is also efficient in image preprocessing. Firefly Algorithm was used in image compression [9], gray-scale image watermarking [13] and Key-Point classification [26]. Cuckoo Search Algorithm has efficient application in intelligent video target racking [21], satellite image segmentation [2] and image recognition [28].

Such notable examples convinced us to examine ABCA as a part of 2D image classifier. Therefore, we want discuss a novel approach to object classification based on the application of ABCA with a sobel filter that preprocesses 2D input images, so as to simplify object recognition. Research results for sample images from open test images databases¹² are presented and discussed in the following sections.

2 Sobel Filter

Sobel operator evaluates directional gradient of the luminosity. Then, it reveals the edges of the depicted objects as the pixels with high gradient value. These points are characterized by sharp-cut variations of luminosity (e.g. the sudden variation of luminosity for a well-lit object on a dark background provides the bright edge).

In the examined solution, a sobel based recognition in gray scale images is obtained by an edge detection approach using derivative preprocessing. In order to recognize shapes, the edges are detected by applying a sobel filter based on a differential operator, see [4] and [1]. Applied sobel operator approximates the two dimensional gradient of a luminance function by a convolution with an integer filter applied along the axial directions. In order to detect the edges contained in a gray scale image \mathcal{I} , the sobel operator makes use of the second order differential operator. Therefore it is a kind of an orthogonal gradient operator in local form. For a continuous function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ and a given image point $\mathbf{x}_i = (x_{i,1}, x_{i,2})$, the gradient can be expressed as ∇f so that:

$$df = \nabla f \cdot d\mathbf{x}_i = [\partial_1 f, \partial_2 f] \cdot [dx_{i,1}, dx_{i,2}], \quad (1)$$

where partial derivatives $\partial_1 f$, $\partial_2 f$ are computed for each pixel location. In the applied method, an approximation is achieved as a convolution of kernels for a small area of neighbor pixels. $\partial_1 f$ and $\partial_2 f$ use a separate kernel each, so there are two applied kernels combined into a simplified gradient operator:

$$S1 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \quad S2 = \begin{pmatrix} -1 & 0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}. \quad (2)$$

¹ www.imageprocessingplace.com

² <http://sipi.usc.edu/database/>

One of the kernels from (2) has a maximum response for the vertical edge and the other has a maximum response for the horizontal edge of the input object. Therefore, in the applied method, the maximum value of the two convolutions is used as the output bit of each input image point. As a result, we get an image of the edge amplitude. In order to obtain the edges of the image, starting by its associated luminance intensity matrix I , for every pixel $\mathbf{x}_i = (x_{i,1}, x_{i,2})$ of the image we compute the following functions:

$$\begin{aligned}
 g_1(\mathbf{x}_i) &= g_1(x_{i,1}, x_{i,2}) = \sum_{m=1}^3 \sum_{n=1}^3 S1_{mn} \cdot I(x_{i,1} + m - 2, x_{i,2} + n - 2), \\
 g_2(\mathbf{x}_i) &= g_2(x_{i,1}, x_{i,2}) = \sum_{m=1}^3 \sum_{n=1}^3 S2_{mn} \cdot I(x_{i,1} + m - 2, x_{i,2} + n - 2), \quad (3) \\
 g(\mathbf{x}_i) &= g(x_{i,1}, x_{i,2}) = g_1^2(\mathbf{x}_i) + g_2^2(\mathbf{x}_i),
 \end{aligned}$$

where $S1_{ij}$, $S2_{ij}$, g_1 and g_2 are taken from the classical Sobel approach. In classic version, once the values of the functions in (3) are obtained for each pixel, the edges are defined as the pixels in the subset of points $\mathcal{E} \subset \mathcal{I}$ so that:

$$\forall \mathbf{x}_i \in \mathcal{E} \Rightarrow \begin{cases} g(\mathbf{x}_i) > 4 \langle g^2 \rangle \\ g_1(\mathbf{x}_i) > g_2(\mathbf{x}_i) \\ g(\mathbf{x}_i) \geq g(x_{i,1}, x_{i,2} - 1) \\ g(\mathbf{x}_i) \geq g(x_{i,1}, x_{i,2} + 1) \end{cases} \vee \begin{cases} g(\mathbf{x}_i) > 4 \langle g^2 \rangle \\ g_1(\mathbf{x}_i) < g_2(\mathbf{x}_i) \\ g(\mathbf{x}_i) \leq g(x_{i,1} - 1, x_{i,2}) \\ g(\mathbf{x}_i) \leq g(x_{i,1} + 1, x_{i,2}) \end{cases}, \quad (4)$$

basing on the g_1 , g_2 and g computed in (3). When a sobel filter is applied to 2D input image \mathcal{I} , a new indexed image is depicted as a plain representation of the points of $\mathcal{E} \subset \mathcal{I}$, as defined in (4). The indexed image coordinate system is a representation of \mathcal{I} , whereby all the values are zeros, except for the coordinates of the points in \mathcal{E} which are ones. Such a representation is lossy since it does not carry the values given by the functions in (3) but only a logical representation of the set \mathcal{E} .

To obtain an advanced identification of the edges retaining partial information, such as the values of the functions in (3), Sobel filter must be modified. For each point of \mathcal{I} , instead of obtaining \mathcal{E} from (4), we used the value of $g(\mathbf{x}_i)$ and its square root:

$$\forall \mathbf{x}_i \in \mathcal{I} \Rightarrow \tilde{g}(\mathbf{x}_i) = \sqrt{g(\mathbf{x}_i)} = \sqrt{g_1^2(\mathbf{x}_i) + g_2^2(\mathbf{x}_i)}. \quad (5)$$

Using g and \tilde{g} functions defined in (3) and (5), edges become blurred. However now they are more suitable for EC recognition purpose, since higher values are obtained with respect to the other points of image \mathcal{I} . The bright patterns, become the shapes for applied EC methods which can now solve shape detection and recognition. Let us present the applied version of the sobel filter method, see Algorithm 1.

Algorithm 1: Simplified sobel filter algorithm

```

Start,
Import image  $\mathcal{I}$  to  $Im$ ,
Calculate the number of pixels columns and rows in 2D input image  $Im$ ,
Create the  $3 \times 3$  filters  $S1$  and  $S2$  using (2),
while  $n \leq rows$  do
  while  $m \leq columns$  do
     $Grays[m][n] = ccvmean(Im[m][n])$ ,
    // ccvmean is the mean of colour channels values
    Compute  $g$ ,  $g_1$  and  $g_2$  on  $Grays[m][n]$  using (3),
    if  $f\_selection$  specifies it then
      |  $g\_tilde[m][n] = \text{sqrt}(g[m][n])$ ,
    end
    // Apply the filter to the 2D input image
  end
end
Save  $g$  as a bitmap grayscale image  $GIm$ ,
switch  $f\_selection$  do
  case 1 Save  $g$  as a grayscale image  $GIm$ ,

  case 2 Save  $g\_tilde$  as a grayscale image  $G\_tildeIm$ ,

  case 3
    | Save  $g$  as a grayscale image  $GIm$ ,
    | Save  $g\_tilde$  as a grayscale image  $G\_tildeIm$ ,
  end
endsw
Stop.

```

3 Artificial Bee Colony Algorithm

ABCA is one of the new EC methods. One of the first versions was presented in [10] and [29]. In the idea, ABCA is inspired by the behavior of honey bees.

During searching for food, bees have developed a variety of techniques to communicate with each other. This communication is related to the sources of nectar and helps to determine the location of best source. One of such techniques is a waggle dance. Performing it, the bees are providing the information about distance from the hive to the source of the nectar, it's direction and quality.

In implemented ABCA, bees are divided into groups. There are scouts, who are searching for food in a random way over the given space. Then there are onlookers and finally employed bees. After coming hive, scouts perform the waggle dance in which the information about the nectar is provided. Onlookers watch waggle dance of others bees, and then choose the best places to go for searching. The last group are employed bees. In the method we treat a bee as a point in the solution space of the model. The bee stays at each point until it finds (by contacting with other bees) a better point. If there is a new, better place of

nectar source in the space (the bee gets such information from others) then the source is abandoned and the employed bee goes for searching of it. Therefore it becomes a scout. The bee is flying and searching for nectar. At the end of each epoch we compare all the bees (the information is given to others in a waggle dance). This comparison gives an information, where the best nectar sources (according to the fitness function) are placed. Thus the bees from this part of the solution space are keeping their positions and the others are flying toward this direction to find more good nectar sources. In the hive, at the end of each epoch the bees are performing waggle dance. They watch each other to get information about best nectar source positions. An onlooker bee chooses the best food place according to the formula:

$$p(\mathbf{x}_i) = \frac{\Phi(\mathbf{x}_i)}{\sum_{i=1}^{i=n} \Phi(\mathbf{x}_i)}, \quad (6)$$

where $\Phi(\mathbf{x}_i)$ is a value of the fitness function calculated for each dancing bee. Having this information the population is sorted according to fitness, and in this way the set of best nectar sources (location of these employed bees) is evaluated. Therefore other onlookers can go to search for food in this direction. They fly in random way and their position over the input space is modified by the equation:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \alpha_k \cdot \Delta\mathbf{x}_{ik}, \quad (7)$$

where k is randomly chosen index of the bee among those placed in best nectar sources, α_k is a random number between $[-1, 1]$ and $\Delta\mathbf{x}_{ik}$ is calculated as:

$$\Delta\mathbf{x}_{ik} = (x_{ij} - x_{kj}), \quad (8)$$

where j is randomly chosen spatial coordinate of the chosen bee. In this way we model a waggle dance in the hive, selection of the best nectar sources (best fitness function points) and movement of all the bees toward this direction in each epoch.

An ABCA is quite simple. At the very beginning we create a random population of n bees. Each bee is a place over the classified image. We will be evaluating it by getting the brightness of the pixel (potential Key-Points). In the next step, we check fitness function for each of them and sort the entire population according to this value. After that we selected m best bees among them. The best bees are transferred to next round and the process starts from the beginning. An implementation of the method is presented in Algorithm 2.

3.1 Image preprocessing method

A digital 2D image \mathcal{I} consists of points, each having a position and special properties. The position of a pixel with given coordinates $\mathbf{x}_i = (x_{i,1}, x_{i,2})$ and its properties (i.e. brightness or saturation) are crucial for classification. This combination brings unique information about the objects in the picture. However, correct classification depends on the right decision (i.e. recognition), which is

Algorithm 2: ABCA to classify 2D images Key-Points

Start,
 Define all coefficients: n -size of population, m -number of chosen best bees,
 number of *generations* and number of *bees*,
 Define fitness function for the algorithm using (9),
 Create a random initial population of *bees* in 2D image,
 $t = 0$,
while $t \leq \textit{generation}$ **do**
 Evaluate *bees* population using (6),
 Sort *bees* according to the value of fitness function,
 Select m best locations among all *bees*,
 Using (7) move the *bees* toward nectar source (best locations) defined in (8),
 Next generation $t++$,
end
 Values from last population with best fitness are the solution,
 Stop.

based on some features. Thus, a *Key-Point* is a pixel in a 2D input image \mathcal{I} with peculiar properties making it important for object recognition. A *Key-Area* is containing many Key-Points that all together compose an object to recognize. In other words, to find the object of interest our eyes search for areas in the picture that contain many points of the same kind. This recognition process has some steps that may be implemented in a computer system. Sobel filter is applied for preprocessing 2D input images to be passed to an ABCA based object recognition. The filter is extracting the most important features in each input image \mathcal{I} .

Each bee is representing a single pixel (point in the image). A population of bees is then simulated in order to move from point to point and search for specific areas. Searching is based on a simplified fitness function that reflects brightness of each filtered image point

$$\Phi(\mathbf{x}_i) = \Phi((x_{i,1}, x_{i,2})) = \begin{cases} 0.1 \dots 1 & \text{saturation} \\ 0 & \text{other} \end{cases}, \quad (9)$$

where $\Phi(\mathbf{x}_i)$ denotes the quality of the evaluated pixel reflected in the scale from 0.0 to 1.0, where color saturation changes from black to white. Therefore using ABCA with fitness function (9) for sobel filtered input images we are to build a simplified 2D image classifier, see Figure 1. Using filtering we extract the borders of input objects, which will be marked in white on a dark background. Therefore, these filtered images are proper input objects for ABCA classifier. When bees fly in search for nectar, they pick points with the best fitness within the range of their flight. Then, from all points we take m best locations, where fitness function is highest. These points are taken to next round and the rest of the population is moved toward them, as defined in the algorithm. Finally, the last generation cover areas of interest. In the research, simulations were performed

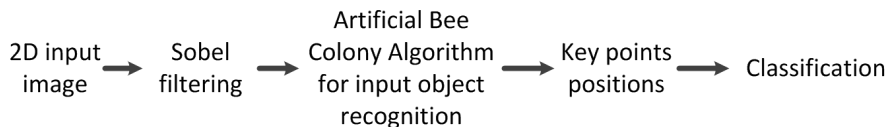


Fig. 1. 2D input image recognition process

for 400 bees in 20 generations with $\alpha = 0.3$ and $m = 30$. Let us see some sample research experiments results.

4 Research results

In the examinations sample test images were used (see Section 1). We tried to examine the behavior of the proposed classifier on various 2D images. Research results are presented to discuss potential efficiency. Fig. 2 to Fig. 6 present the classifier steps. Firstly, the leftmost top image in each Figure presents the original input image. Then, going towards the right on the first row, we can see the image after the first sobel filtering, then the filtered image with ABCA recognized Key-Points (in red). The second row shows (from left to right) the image after the second filtering operation, the filtered image with ABCA recognized Key-Points (in red), and, finally the main result: classified 2D input image, where Key-Points are presented on top of the original input image (each one in red).

We can see that proposed solution can find shapes of recognized objects. ABCA is able to recognize objects, however applied sobel filtering is helping a lot to improve classification. If we look at images closely, we see that in recognition without filtering ABCA finds only location. However after filtering the recognition is much improved. Depending on the input object, recognition over first filtering gives good quality. However sometimes second filtering is even more crucial to help in exact shape recognition (see Fig. 5 and Fig. 6).

4.1 Conclusions

It is possible to preprocess images by ad-hoc combination of sobel filter with ABCA as a simplified EC classifier. Performed calculations are very simple. We just use formulas (1) – (4) to filter input image and then (6) – (8) to calculate the position of Key-Areas in examined images. The classifier covers recognized objects with Key-Points. If the input images have many points with the same value, the classification process may be more complicated. On the other hand, the system efficacy is self-increased if we are looking for Key-Points with high contrast in relation to surroundings. Moreover, it is possible to increase the classifier efficiency in some ways. The first step would be introducing more precise sobel kernels or filtering functions. The second step is to have more efficient ABCA motion or information processing. Finally, we can introduce better fitness function. Better filtering or ABCA method models will be considered in

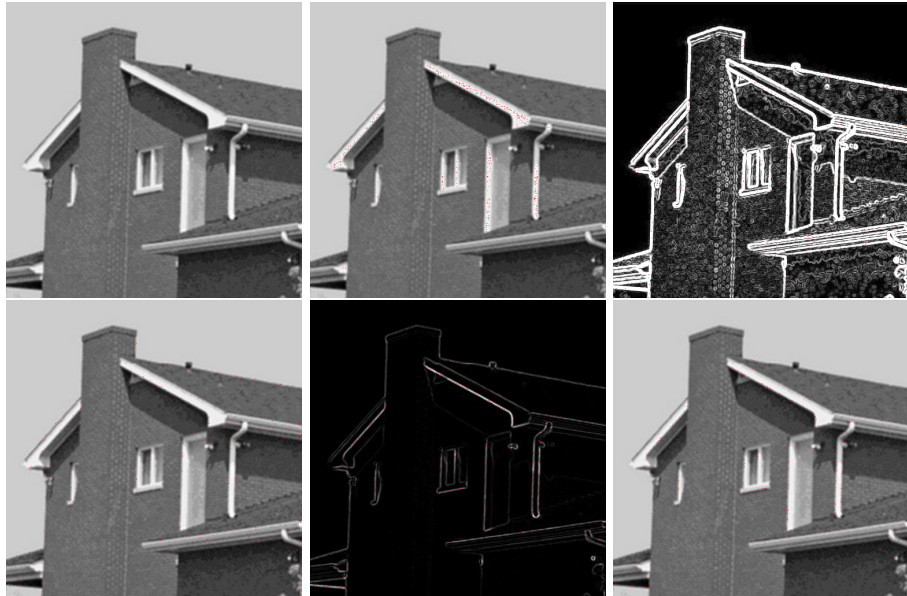


Fig. 2. 2D input image recognition process over construction images

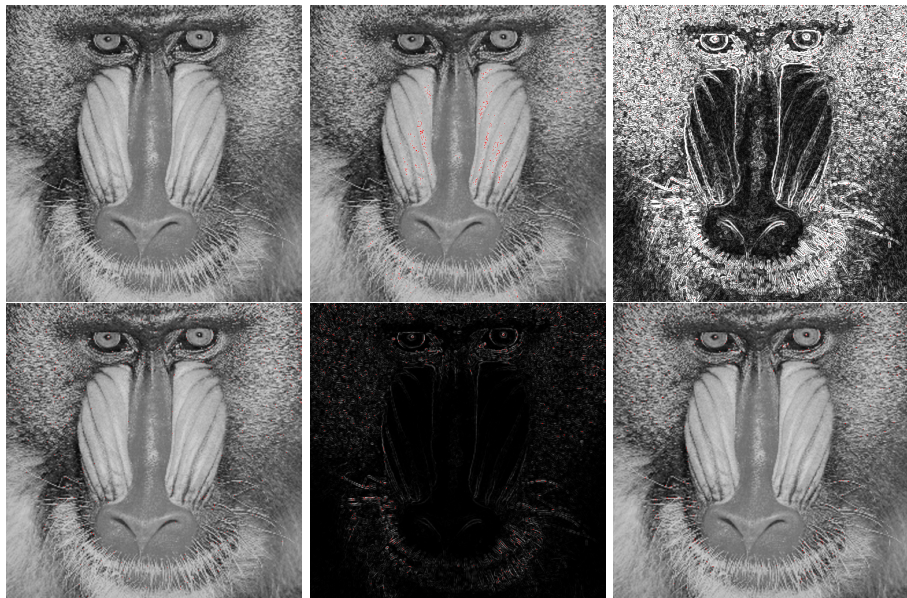


Fig. 3. 2D input image recognition process over facial images

the future research. However, introducing a special recognition function is non-trivial task. The fitness function is the most sensitive part of the classifier. To



Fig. 4. 2D input image recognition process over landscape images



Fig. 5. 2D input image recognition process over single objects

increase its efficiency in classification we must define a special function for the various classes of objects, i.e. different for blocks or buildings, for faces and other

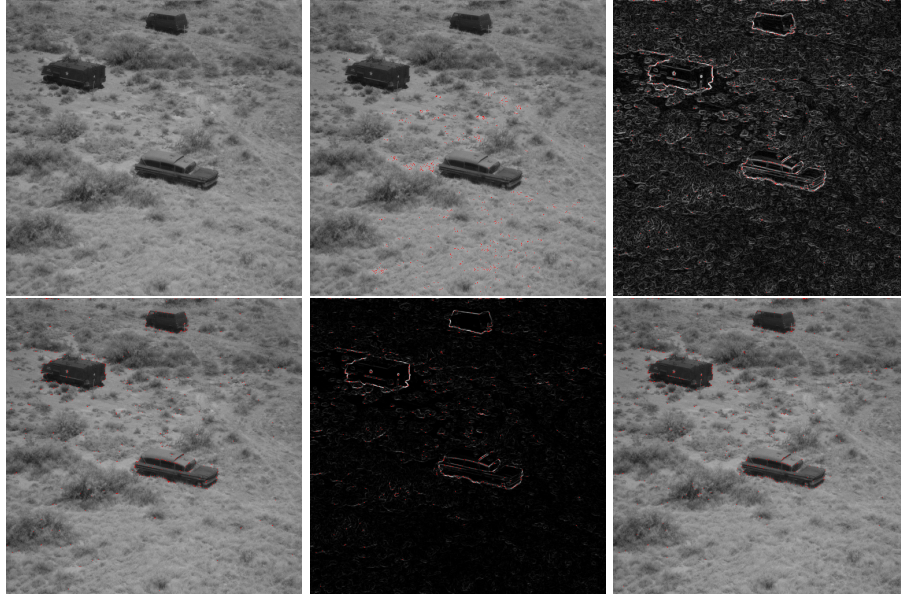


Fig. 6. 2D input image recognition process over many objects

appearance features, etc. From our research findings, we can state that it is very unlikely to have one function for all classified objects types. Therefore another attempt to improve it, would be to use some dedicated AI methods, like neural networks [14] or fuzzy systems [17].

5 Final Remarks

We can state that the presented classifier can find objects of interest, and therefore answer to the question given in the title is positive.

This feature makes it a promising tool for Artificial Intelligence recognition systems. The solution discussed in this article can work as a part of sophisticated image classifiers. We decided to apply sobel operator for image filtering for its low complexity and coarse precision, which both have importance for the problem of edge detection. Applied ABCA is easy to implement and allows easily exploration of the entire input image without complicated mathematical operations. However the precision in recognition is still not enough. It is possible to increase ABCA efficiency, if applied to filtered objects located among many other of lower brightness.

In further research, we plan to work on sobel filter precision, by applying other kernels and ABCA method by improving motion or decision. We will also lead further research on classifier fitness function. Furthermore, we will try to build a classifier for large collections of input images using fast methods like [15], [27] and [12].

References

1. G. Anusha, T. Prasad, and D. Narayana. Implementation of sobel edge detection on fpga. *International Journal of Computer Trends and Technology*, 3(3):472–475, 2012.
2. A.K. Bhandari, V.K. Singh, A.Kumar, and G.K. Singh. Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using kapurs entropy. *Expert Systems with Applications*, 41(7):3538–3560, 2014.
3. F. Bonanno, G. Capizzi, G. Lo Sciuto, C. Napoli, G. Pappalardo, and E. Tramontana. A cascade neural network architecture investigating surface plasmon polaritons propagation for thin metals in openmp. *Lecture Notes in Artificial Intelligence - ICAISC'2014*, 8468, PART I:22–33, 2014.
4. J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.
5. M. Gabryel, R. K. Nowicki, M. Woźniak, and W. M. Kempa. Genetic cost optimization of the $GI/M/1/N$ finite-buffer queue with a single vacation policy. *Lecture Notes in Artificial Intelligence - ICAISC'2013*, 7895(PART II):12–23, 2013.
6. M. Gabryel, M. Woźniak, and R. K. Nowicki. Creating learning sets for control systems using an evolutionary method. *Lecture Notes in Computer Science - ICAISC'2012*, 7269:206–213, 2012.
7. 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*, 62:462–478, 2012.
8. 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.*, 47:7–14, 2013.
9. M. H. Horng. Vector quantization using the firefly algorithm for image compression. *Expert Systems with Applications*, 39(1):1078–1091, 2012.
10. D. Karaboga and B. Basturk. Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. *Lecture Notes in Computer Science - IFSA'2007*, 4529:789–798, 2007.
11. M. Lu, M. Wan, F. Xu, X. Wang, and H. Zhong. 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.
12. Z. Marszałek, D. Połap, and M. Woźniak. On preprocessing large data sets by the use of triple merge sort algorithm. In *Proceedings of International Conference on Advances in Information Processing and Communication Technologies - IPCT'2014*, pages 65–72, Santa Barbara, California, USA, 2014. The IRED, Seek Digital Library.
13. A. Mishra, C. Agarwal, A. Sharma, and P. Bedi. Optimized gray-scale image watermarking using dwt svd and firefly algorithm. *Expert Systems with Applications*, 41(17):7858–7867, 2014.
14. C. Napoli, G. Pappalardo, and E. Tramontana. A hybrid neuro-wavelet predictor for qos control and stability. *Lecture Notes in Computer Science - AI*IA 2013*, 8249:527–538, 2013.

15. C. Napoli, G. Pappalardo, and E. Tramontana. Using modularity metrics to assist move method refactoring of large systems. In *Seventh International Conference on Complex, Intelligent, and Software Intensive Systems - CISIS 2013*, pages 529–534, July 2013.
16. C. Napoli, G. Pappalardo, and E. Tramontana. Improving files availability for bit-torrent using a diffusion model. In *IEEE 23rd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises - WETICE 2014*, pages 191–196, June 2014.
17. A. Niewiadomski. Imprecision Measures for Type-2 Fuzzy Sets. Applications to Linguistic Summarization of Databases. *Lecture Notes in Artificial Intelligence*, 5097:285–294, 2008.
18. A. Nowak and M. Woźniak. Analysis of the active module mechatronical systems. In *Proceedings of Mechanika 2008 - ICM'2008*, pages 371–376, Kaunas, Lietuva, 2008. Kaunas University of Technology Press.
19. A. Nowak and M. Woźniak. Multiresolution derives analysis of module mechatronical systems. *Mechanika*, 6(74):45–51, 2008.
20. A. Rubaai, M.J. Castro-Sitiriche, and A.R. Ofoli. 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.
21. G. S. Walia and R. Kapoor. Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search. *Expert Systems with Applications*, 41(14):6315–6326, 2014.
22. M. Woźniak. Fitness function for evolutionary computation applied in dynamic object simulation and positioning. In *Proceedings of the IEEE Symposium Series on Computational Intelligence - SSCI'2014*, pages accepted–in press. IEEE, 2014.
23. M. Woźniak. On positioning traffic in nosql database systems by the use of particle swarm algorithm. In *Proceedings of XV Workshop DAGLI OGGETTI AGLI AGENTI - WOA '2014*, page paper 5. University of Catania Press, 2014.
24. M. Woźniak, W. M. Kempa, M. Gabryel, and R. K. Nowicki. A finite-buffer queue with single vacation policy - analytical study with evolutionary positioning. *International Journal of Applied Mathematics and Computer Science*, 24(4):accepted–in press, 2014.
25. M. Woźniak, W. M. Kempa, M. Gabryel, R. K. Nowicki, and Z. Shao. On applying evolutionary computation methods to optimization of vacation cycle costs in finite-buffer queue. *Lecture Notes in Artificial Intelligence - ICAISC'2014*, 8467(PART I):480–491, 2014.
26. M. Woźniak and Z. Marszałek. An idea to apply firefly algorithm in 2D images key-points search. *Communications in Computer and Information Science - ICIST'2014*, 465:312–323, 2014.
27. M. Woźniak, Z. Marszałek, M. Gabryel, and R. K. Nowicki. Modified merge sort algorithm for large scale data sets. *Lecture Notes in Artificial Intelligence - ICAISC'2013*, 7895(PART II):612–622, 2013.
28. M. Woźniak and D. Połap. 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*, pages 164–173, Setubal, Portugal, 2014. SciTePress - INSTICC.
29. W. Zou, Y. Zhu, H. Chen, and Z. Zhu. Cooperative approaches to artificial bee colony algorithm. In *Proceedings of International Conference on Computer Application and System Modeling-ICCA SM'2010*, pages V9–44–V9–48. IEEE, 2010.