Title: Semi-multifractal optimization algorithm

Author: Ireneusz Gościniak

CrossMark
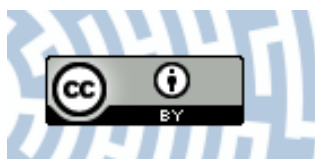
Semi-multifractal optimization algorithm

Ireneusz Gosciniak[1]

Abstract Observations on living organism systems are the inspiration for the creation of modern computational techniques. The article presents an algorithm implementing the division of a solution space in the optimization process. A method for the algorithm operation controlling shows the wide range of its use possibilities. The article presents properties of fractal dimensions of subareas created in the process of optimization. The paper also presents the possibilities of using this method to determine function extremes. The approach proposed in the paper gives more opportunities for its use.

Keywords Evolutionary algorithms · Immune algorithm · Multifractal algorithm · Multifractal analysis · Optimization

1 Introduction

Many tasks can be solved or simplified by means of subdividing a solution space. The approach can be used in a natural way to solve various two-dimensional tasks, such as:

– Searching areas in the geographical database (Belussi et al. 2007),
– Structural geometry of plane figures (Marinov and Kobbelt 2005),
– Algorithms for determining the visibility (Urrutia and Sack 2000),

✉ Ireneusz Gosciniak
   ireneusz.gosciniak@gmail.com

1 Institute of Computer Science, University of Silesia, Będzińska 39, 41-200 Sosnowiec, Poland

– Images transmission (Hwang and Derin 1995),
– Collision detection in plane motion (Weller 2013),
– Image segmentation (Alrawi et al. 2012; Vujovic 2014),
– Images identification (Rejaur Rahman and Saha 2009),
– Wireless sensor networks (Iwanicki and van Steen 2009),
– Path planning (Lu et al. 2012).

The multiresolution analysis is most frequently used in these task implementations. In general, the approach can be described as follows:

Divisions can be performed according to requirements, and the method of division can also be chosen according to the optimization issue. Every element (a subarea) is divided into proper elements (squares or rectangles). The process creates a tree. Each leaf of the tree is a subarea. Every leaf of the tree is associated with attribute:

– 1—if the interior of the element represented by this leaf is within the object;
– 0—if the interior of the element is separable from the object,
– [0,1]—if the intersection of the element interior with the object is not empty—it contains the part of the object.

The term object is abstract in the nature—it is a part of the solution space that meets predefined criteria. There is no reason to divide leaves with the attributes of 0 and 1. Other leaves are processed. The processing is limited by the height of tree which defines the precision.

Many approaches that are used to solve various two-dimensional issues apply evolutionary algorithms for the realized optimization tasks. They can effectively replace the hitherto used solutions. There are many unique solutions, such as a fractal algorithm, for finding global optimal solution described in Song and Qian (2006). A method of the

solution space division proposed in Song and Qian (2006) creates a fractal. Different ways of space division are realized in the study presented in Pereira et al. (2005). This division has not an evolutionary character and creates a deterministic multifractal. The evolutionary algorithm proposed in this article uses a similar method of division. Fractals and multifractals are also used to solve many tasks, e.g. Kotowski et al. (2008), Masayoshi et al. (1996). They are used to evaluate the given process or to identify objects. This article is just such an attempt—the proposed algorithm realizes multifractal division of solution space what allows an identification using fractal dimension.

The remaining part of the paper is organized as follows: Section 2 introduces the proposed algorithm. Next, Sect. 3 introduces algorithm used to divide area described by functions. Section 4 presents the algorithm used for the image division. Then, in Sect. 5 a discussion on the characteristic spectrum and the multifractal spectrum of proposed method of image divisions is conducted. Finally, Sect. 6 gives a short concluding remarks.

2 Optimization algorithm

Evolutionary algorithms are strictly specified by heuristics (mostly metaheuristics) used to solve some optimization problems, for which deterministic approaches are unknown or have an unacceptably long computation time (Weise 2009). The principle of their operation is based on simulation of mechanisms known from the natural world, i.e. natural selection and heredity. These algorithms start their operation by comparing the random (initial) solutions with earlier predetermined set of criteria for evaluation. Solutions that have these criteria are selected and are subject to reproduction and mutations. They are re-evaluated by the determined criteria, and the best ones are selected for the further evolution. The algorithm repeats this scheme until a solution is not optimal. The optimal solution is determined on the basis of criteria evaluation—fitness function. So, it is a specific adaptation process.

Modifications of evolutionary algorithms take ideas from other algorithms what makes the direct classification difficult. With such a case I deal in this paper. A comprehensive approach to the problem opens many ways to implement the algorithm. One of them is an immune algorithm (Mo 2008). A solved problem is represented by an antigen. A population of antibodies is developed under the influence of antigen creating a set of solutions. The development of the antibody population is controlled by the process of cloning, mutation and selection. The best of clones replace the antibodies which produced them.

The operation of algorithm 1—the semi-multifractal optimization algorithm—can be described as follows:

Algorithm 1: Semi-Multifractal Optimization Algorithm

Data: $\mathbf{Ab}$ – set of antibodies; Ab – an antibody ($Ab \in \mathbf{Ab}$); Ab^* – a selected antibody ($Ab^* \in \mathbf{Ab}$); $\mathbf{C}$ – set of clones; C – a clone ($C \in \mathbf{C}$); $\mathbf{C}^m$ – set of mutated clones; C^m – a mutated clone ($C^m \in \mathbf{C}^m$, $C^m \equiv \{Ab_1, Ab_2\}$); C^{*m} – a selected mutated clone ($C^{*m} \in \mathbf{C}^m$)

Input: Ag – an antigen ($f()$ – a solving problem); n_{Ab} – number of antibodies; n_C – number of clones;

Output: the best solution.

1   $\mathbf{Ab} \longleftarrow$ initialise set of antibodies;
2   **repeat**
3     $Ab^* \longleftarrow$ selection of $Ab \in \mathbf{Ab}$ using roulette's well;
4     $\mathbf{C} \longleftarrow$ produce n_C clones of Ab^*;
5     $\mathbf{C}^m \longleftarrow \{\forall C \in \mathbf{C} \ C$ is mutated and evaluated depending on Ag\};
6     $C^{*m} \longleftarrow$ select the best $C^m \in \mathbf{C}^m$;
7     $\mathbf{Ab} \longleftarrow (\mathbf{Ab} \setminus \{ Ab^*\}) \cup \{ C^{*m}\}$;
8   **until** stop criteria are not reached;
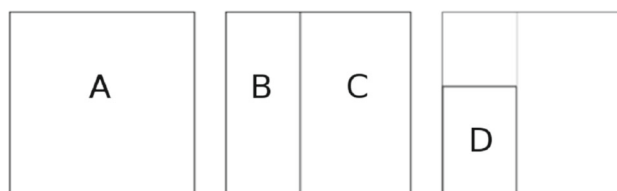9   **return** $\mathbf{Ab}$;

Fig. 1 Subarea divisions

1. The creation of an antibody or a set of antibodies (initial deterministic divisions of the solution space) and evaluation.
2. The selection of an antibody by the roulette's wheel.
3. Clones production of selected antibody.
4. Clones mutation—clones division by a rule and their evaluation.
5. Selection of the best clone.
6. The selected antibody is replaced by the best clone.
7. Repeating from step 2 up to the stop criterion is obtained.

The clones are a copy of an antibody. The number of produced clones depends on the antibody type. A way of clone mutation also depends on the type of antibody. The clone mutation realizes its division into two antibodies. The part of a solution space represented by a clone is divided into two areas in the proportions $\frac{1}{3}$, $\frac{1}{2}$ and $\frac{2}{3}$ in the vertical or horizontal directions (Fig. 1). The proposed rule of clones production and mutation allows to obtain four collections of figures differing in size (scale). The rule of clones production and mutation can be described as follows:

$$A \rightarrow (B, C)_{(4)},$$
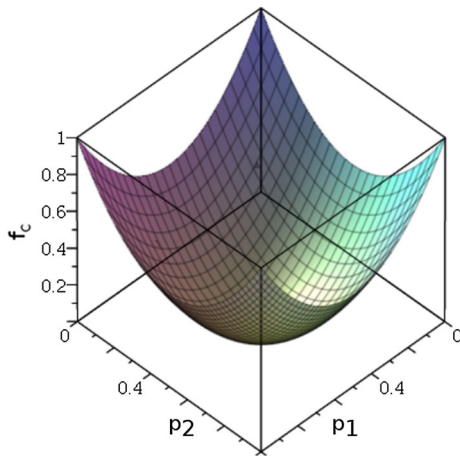$$B \rightarrow (D, A)_{(2)},$$

Fig. 2 Fitness function for clones

$$C \rightarrow \{(B, B)_{(1)}, (A, D)_{(2)}\},$$
$$D \rightarrow (A, A)_{(1)},$$

(the subscript gives the number of produced clones).

The antibody of type A produces four different clones containing antibodies B and C, while the antibody of type B produces two different clones containing antibodies A and D. The antibody of type C produces three different clones: one clone containing two antibodies of type B and two clones containing antibodies A and D. The antibody of type D produces one clone containing two antibodies of type A. Similar divisions are presented in studies (Corso et al. 2004; Pereira et al. 2005; Corso and Lucena 2005) as a deterministic and a random multifractal tilling.

Thus, the mutation of clone produces two antibodies. Each of such created antibodies is evaluated. Basing on this, the evaluation of clone is carried out. The best evaluated clone is selected—but how can the clone be evaluated? The easiest way is to select a clone containing the antibody with the best adaptation. However, this choice may be implemented

as a choice of a clone containing antibodies with possibly the worst or the best adaptation. This approach allows to identify areas that should be omitted—it is a kind of negative selection. The function of such a choice can be described by Eq. (1), and it is shown in Fig. 2. The selection of clones is responsible for the population development.

$$f_c(p_1, p_2) = 1 - \frac{f(p_1) + f(p_2)}{2}, \tag{1}$$

where

$$f(p) = p \cdot (1 - p), \tag{2}$$

and p_1, p_2—coefficients of adaptation of produced antibody (as a part of a clone).

Selection of antibodies is responsible for the direction of development and improvement in the population. It is realized by means of roulette's wheel that prevents from stagnation and ensures the correct selection of antibodies (Michalewicz 1996). The evaluation of antibodies is an important factor influencing the algorithm operation. In the evaluation of the antibody, the size of subarea and adaptation should be considered. The evaluation can be based on Eq. (1), and taking into account the factor of the surface (in a linear way)—Eq. (3) of function evaluation takes the form:

$$f_{sa}(p, a) = f(p)^\theta \cdot l(a), \tag{3}$$

where

$$l(a) = 1 - a \cdot \sigma. \tag{4}$$

and σ—coefficient of subarea impact (in the analysed examples $\sigma = 0.5$), a—coefficient of subarea (defined as the ratio of subarea to the solution space area), p—coefficient
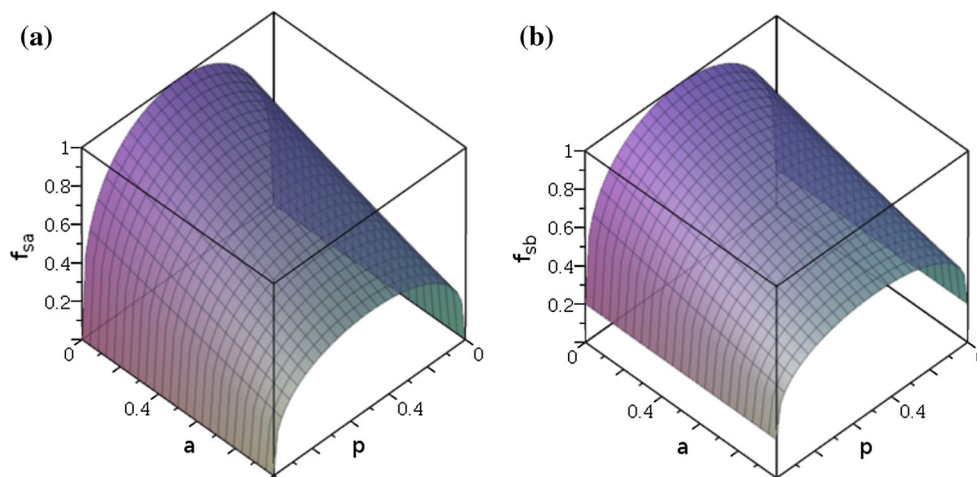

Fig. 3 Fitness function for antibodies. **a** Base function. **b** $b = 0.2$
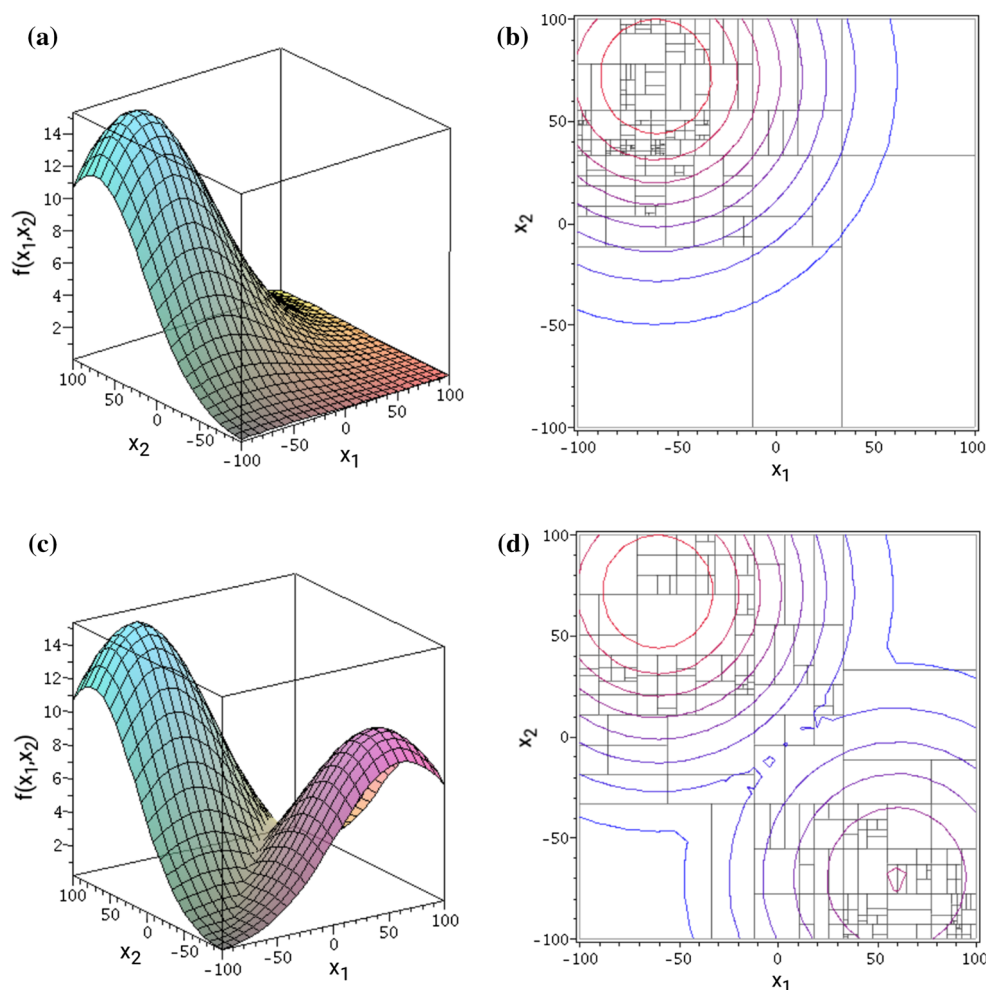
Fig. 4 Extreme designation of functions. **a** Function with one extreme. **b** One extreme designation. **c** Function with two extremes. **d** Two extremes designation

of adaptation of produced antibody, θ—aspect ratio (in analysed example $\theta = \frac{1}{3}$).

Parameters σ and θ are responsible for the shape of the function, and thus, they have a significant influence on the algorithm operation—these parameters were selected in an experimental way during the tuning of the algorithm.

Equation (3) can be supplemented by a factor of exploitation—b; it can be described by the following equation (Fig. 3b):

$$f_{sb}(p, a, b) = f(p)^{\theta} \cdot l(a) \cdot (1 - b) + b, \tag{5}$$

where b—coefficient of exploration pressure.

This function gives great opportunities to control the algorithm operation. This seemingly simple algorithm hides enormous potential.

3 Application of the algorithm for functions

The test function (antigen) is a combination of unimodal functions, which can be described as follows:

Table 1 Function parameters

| i	h	t_1	t_2	w_1	w_2
1	15.34	−60.05	71.68	175.0	175.0
2	10.34	60.0	−70.0	175.0	175.0

i—function index

$$f(x_1, x_2) = \max_{i=1,\dots,n} f_i(x_1, x_2) \quad \text{and} \quad f_i(x_1, x_2)$$

$$= h \cdot 100^{-\left(\frac{(x_1-t_1)^2}{w_1^2} + \frac{(x_2-t_2)^2}{w_2^2}\right)}, \tag{6}$$

where h—height, t_1, t_2—shift x_1 and x_2, w_1, w_2—slope inclination and n—the number of peaks.

The test functions are rather simple—Fig. 4a, c. Parameters of functions are given in Table 1. This approach may be useful in terrain modelling.

The application allows to illustrate the algorithm operation in a clear way. The value of the fitness function is calculated as the maximum value of the function of corner points.

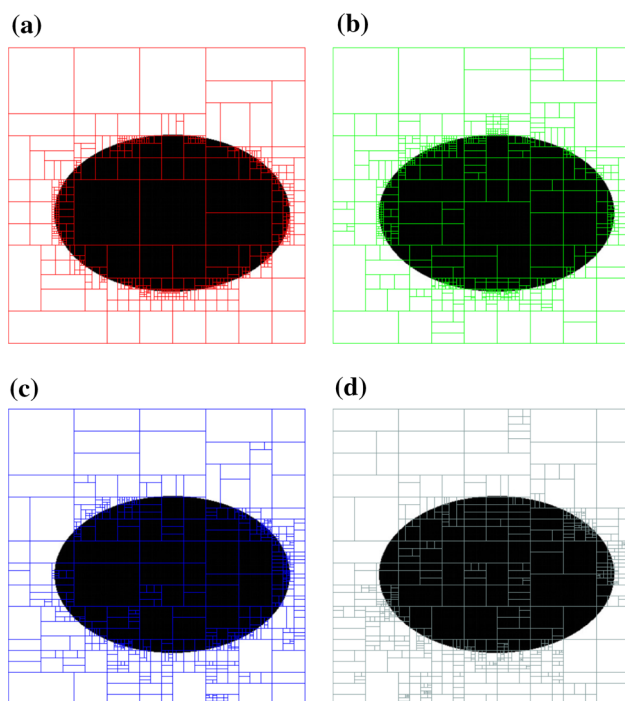**Fig. 5** Extreme designation of 2D benchmark functions

(a)

(b)

(c)

(d)
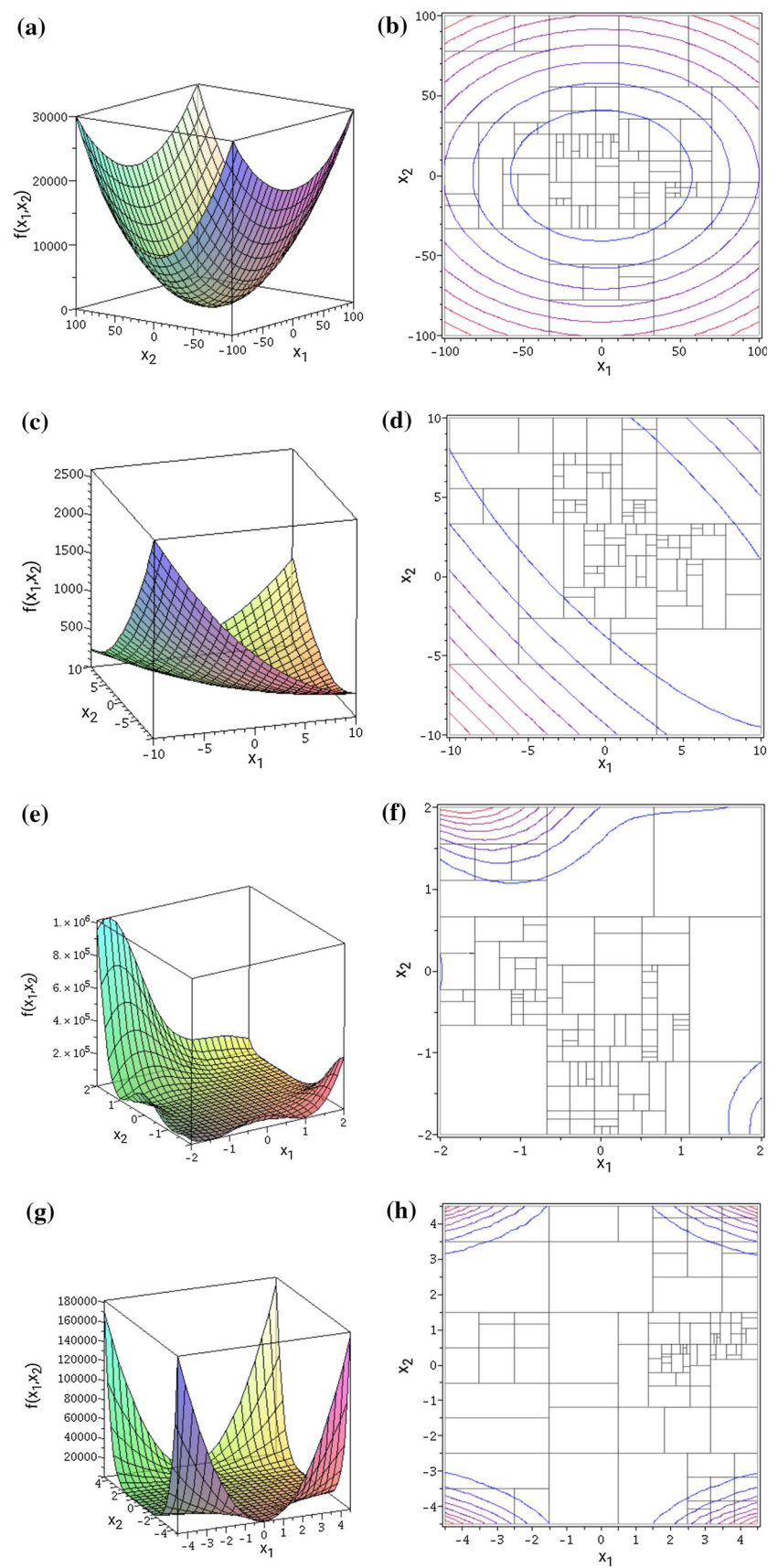
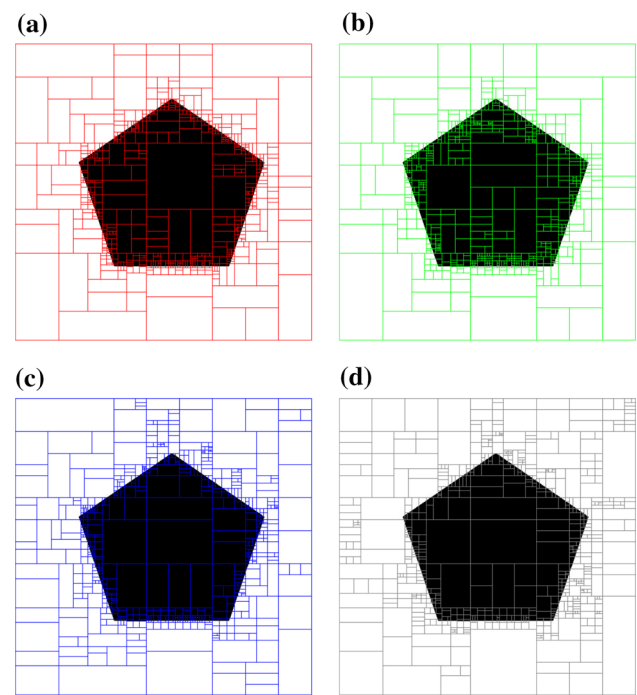**Fig. 6** An ellipse—600 divisions

(a)

(b)

(c)

(d)

Fig. 8 A pentagon: **a** and **b**—penetration towards the inside; **c**) and **d**—penetration towards the outside (600 divisions)

(a)

(b)
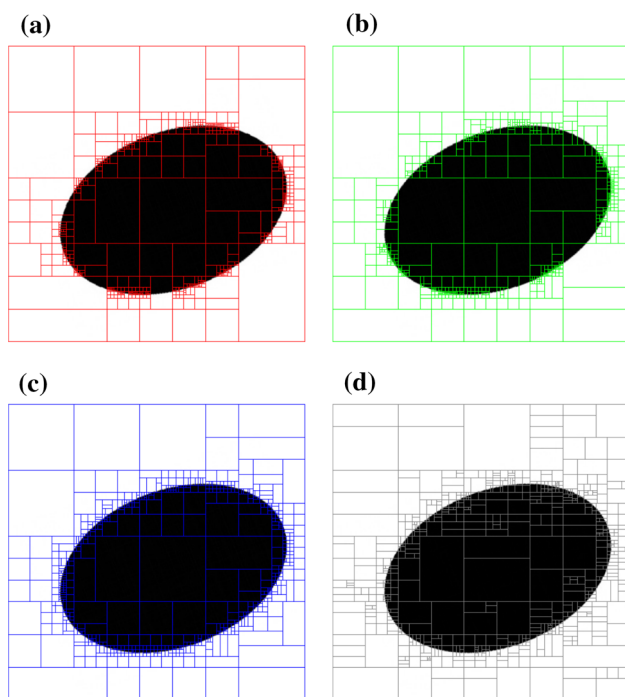
(c)

(d)

Fig. 7 A rotated ellipse (600 divisions)
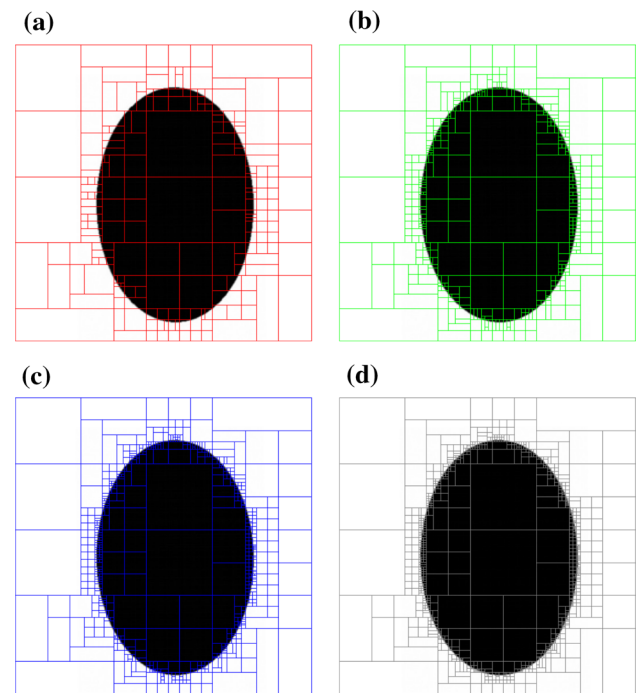
(a)

(b)

(c)

(d)

Fig. 9 A vertical ellipse—$f_{sa}(p^2, 1-a)$

The obtained results for algorithm operation are presented in Fig. 4b, d. The exploitation of extremes is clearly visible. Smaller areas that are not close to the maximum are created as a result of the roulette's wheel operation. Many test func-

tions are given in the literature—a description of the majority of these functions is available online (https://www.sfu.ca/ ssurjano/optimization.html). The operation of the algorithm can be presented using selected test functions for 2D space:

1. The Bohachevsky's function (Fig. 5a):

$$f(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7, \tag{7}$$

   and $x_i \in [-100, 100]$, $f(\mathbf{x}^*) = 0$ at $\mathbf{x}^* = (0, 0)$;
2. The Booth's function (Fig. 5c):

$$f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2, \tag{8}$$

   and $x_i \in [-10, 10]$, $f(\mathbf{x}^*) = 0$ at $\mathbf{x}^* = (1, 3)$;
3. The Goldstein–Price function (Fig. 5e):

$$f(x_1, x_2) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 \right.$$
$$- 14x_2 + 6x_1x_2 + 3x_2^2)\big]$$
$$\cdot \left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 \right.$$
$$\left. + 48x_2 - 36x_1x_2 + 27x_2^2)\right], \tag{9}$$

   and $x_i \in [-2, 2]$, $f(\mathbf{x}^*) = 3$ at $\mathbf{x}^* = (0, -1)$;
4. The Beale's function (Fig. 5g):

$$f(x_1, x_2) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2$$
$$+ (2.625 - x_1 + x_1x_2^3)^2, \tag{10}$$

   and $x_i \in [-4.5, 4.5]$, $f(\mathbf{x}^*) = 0$ at $\mathbf{x}^* = (3, 0.5)$;

where $i = 1, 2$; $\mathbf{x} = (x_1, x_2)$ and $\mathbf{x}^*$—coordinates of the global minimum.

The operation of the algorithm illustrated in Fig. 5 presents the exploratory divisions of the entire solution space and the concentrated exploration divisions of the promising area. However, the proposed algorithm is not dedicated to the global optimization issues. The proposed method can be useful for analysis of terrain and its models mathematically described by functions. The obtained results are satisfactory—but they can still be improved by changing the method for antibodies evaluation. Such modifications are not analysed in the paper because they lead to a completely new algorithm in the global optimization field.

4 Application of the algorithm for images

A simple way to present the possibilities of the algorithm operation is to use the test images. A given task is the exploitation of border areas (fault).

Figure 6 shows the influence of the exploration factor on the algorithm operation. Figure 6a shows the maximum
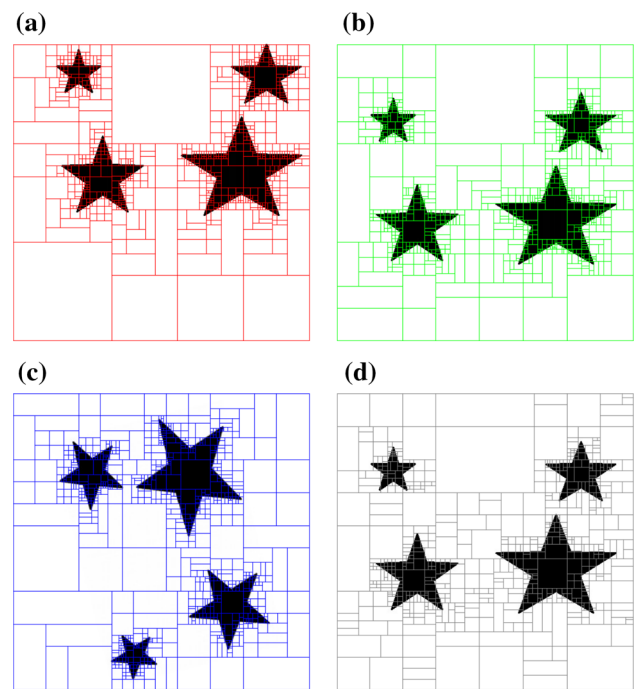

Fig. 10 A collection of stars—$f_{\text{sb}}\left(p^2, 1 - a, (1 - w) \cdot 2e - 3\right)$
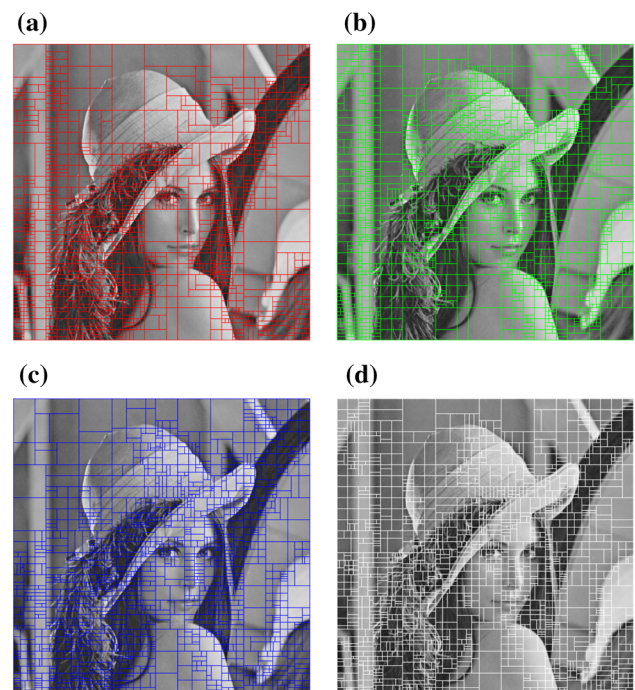

Fig. 11 Lena image—$f_{\text{sb}}(p^2, (1 - a) \cdot 0.5, (1 - w) \cdot 0.01)$ (1500 divisions); levels—level of shades

exploitation—p is a coefficient of antibody adaptation measured by its filling. The following Fig. 6b–d shows the growing influence of exploration.

Figure 7a illustrates the effect of scaling of antibody adaptation coefficient—it causes the strengthening of exploita-
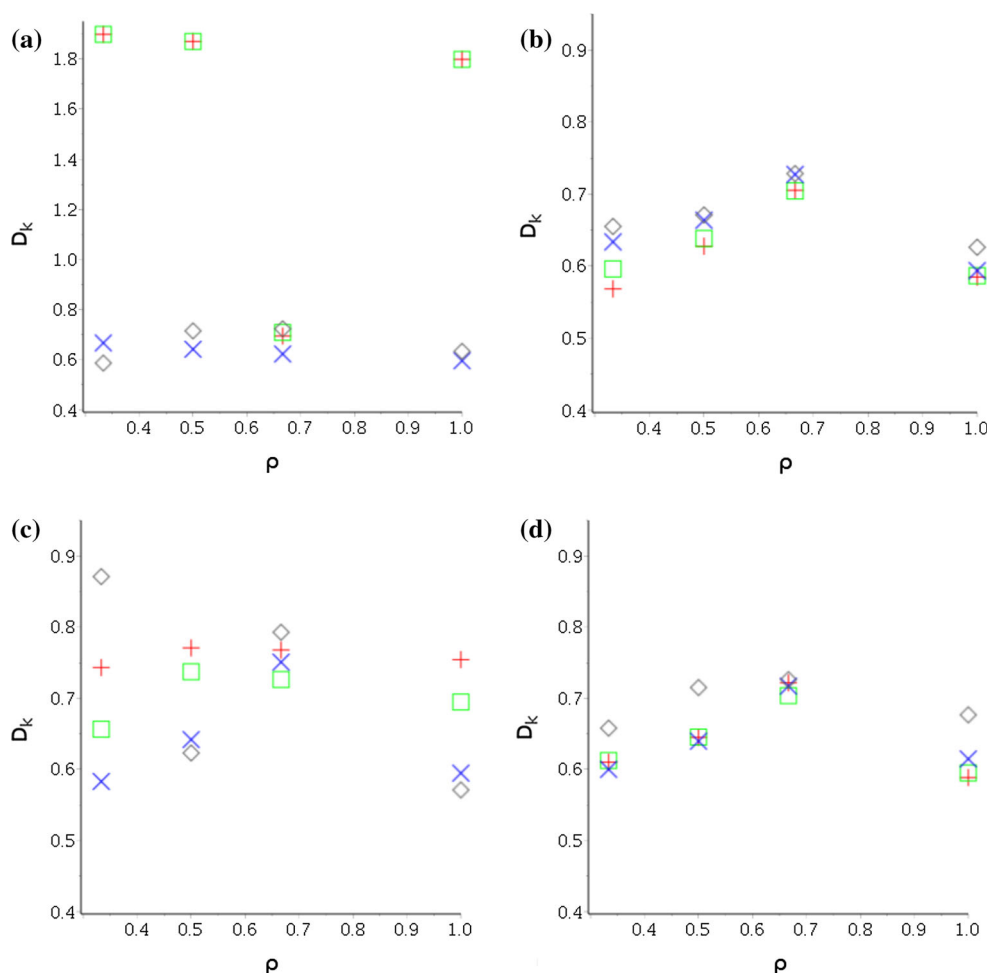
Fig. 12 Characteristic spectrum

tion. A similar effect can be achieved by inserting the coefficient of subarea into the formula (Fig. 7b). The best result is achieved in Fig. 7c through the increase in the impact of coefficient of adaptation and decreasing in the influence of the subarea size coefficient. Inserting the small coefficient of exploration pressure into f_{sb} formula (Fig. 7d) causes a widening of the exploitation scope (increase in exploration).

The antibody keeps the coefficient of adaptation w of neighbour—the second antibody created in the process of mutation. This information can be used to drive the process of exploitation. Figure 8a, b shows that the process of the exploitation is directed into the interior of object with different intensity, whereas Fig. 8c, d shows that the process of the exploitation is directed outside the object with different intensities.

Figure 9 shows the algorithm operation for different numbers of divisions of the image. The increase in the number of divisions clearly improves the exploitation—the exploitation process is uniform.

Figure 10 shows the algorithm operation for image containing several objects—it can be compared with a function

consisting of several extremes. All objects, regardless of their location, are exploited. The decrease in the number of iterations does not affect the deterioration of the algorithm operation (Fig. 10d).

A more complex function can be the image of Lena (Fig. 11). Figure 11a shows the result of exploitation of the border of darker shades (low levels of shades). In Fig. 11b, the border is displaced towards the brighter shades (high levels of shades). Figure 11c shows the algorithm operation for the complex criteria of the exploitation of the border of darker and lighter shades. Figure 11d shows the result for expanded range of shades. Even for such a complex example, the algorithm works very effectively.

5 Characteristic spectrum

The mutation of clones creates four sets of objects (antibodies) depending on the ratio of adjacent sides $\rho = \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}$. These sets can be grouped into subsets of elements of the same area—it creates the image coverage by objects of a cer-

tain scale. Each of the subsets has a certain scale, which can be determined as the sum of two adjacent sides. Knowing the coverage $A(\varepsilon)$ and the scale ε for each set, its fractal dimension can be calculated:

$$D_k = \lim_{\varepsilon \to 0} \frac{\log A(\varepsilon)}{\log \varepsilon}, \tag{11}$$

A similar approach is found in the work (Pereira et al. 2005). The difference consists in the fact that the number of objects created at a specific scale is unpredictable. Such a set of fractal dimensions can be treated as a characteristic spectrum. Determination of the characteristic spectrum for objects can be also found in other works, for example in Kotowski et al. (2008). The examples of fractal analysis used in reference to evolutionary algorithms are presented in works (Juliany and Vose 1994; Kies 2001). The characteristic spectrum is shown in Fig. 12.

In Fig. 7a, b, parameters of created divisions are similar which results in similar characteristic spectra—Fig. 12a. (The data are marked with red cross and green box.) For the same reason for divisions presented in Fig. 7c, d, the characteristic spectra are similar—Fig. 12a, and the data are marked with blue diagonal cross and grey diamond. Small differences in the creation of divisions in Fig. 8 insignificantly influence the characteristic spectrum presented in Fig. 12b. Large variation in characteristic spectra shown in Fig. 12c is related to the number of divisions visible in Fig. 9. The characteristic spectrum of a complex object (Fig. 12d) does not depend significantly on its position as shown in Fig. 10. The characteristic spectrum is influenced by the number of divisions—Fig. 12d, and the data are marked with grey diamond.

The discussion on the algorithm operation can be supplemented by the analysis of multifractal spectrum $f(\alpha)$ (α—the Hölder's exponent). This analysis (with calculation method) is presented in Gosciniak (2017), and it mainly shows a way of objects grouping. To achieve this for each object, a value equal to the sum of adjacent sides (scale) is assigned—it creates cuboids. Multifractal spectrum for the above-discussed example is shown in Fig. 13. Comparing the distribution of divisions in space in relation to multifractal
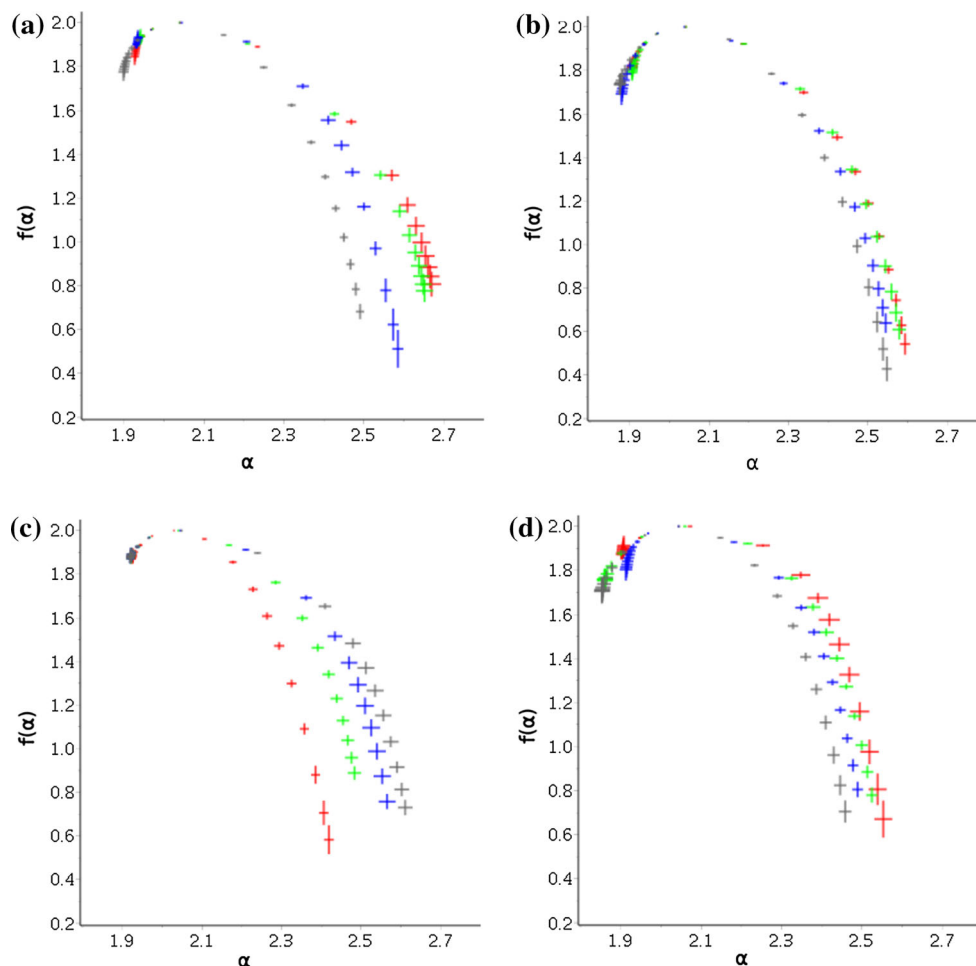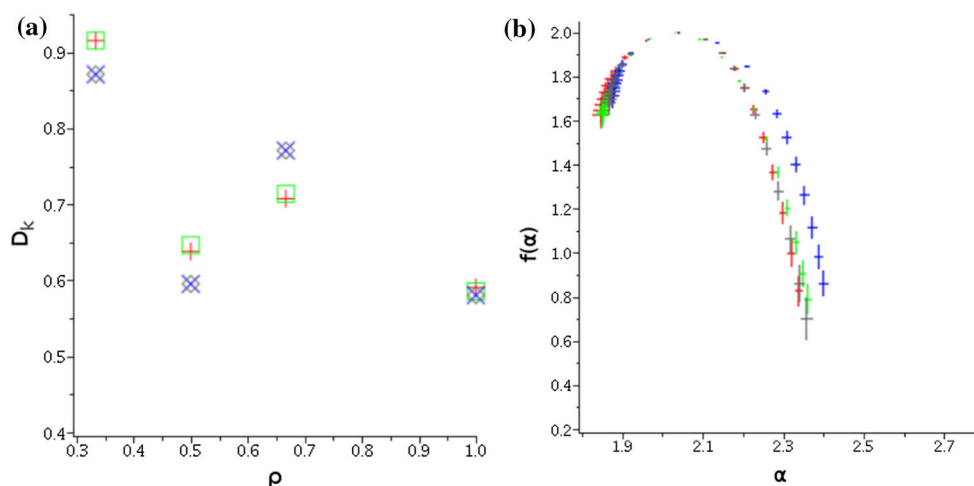

Fig. 13 Multifractal spectrum

Fig. 14 Characteristic (**a**) and multifractal (**b**) spectrum for Lena images

spectrum, the same regularity is observed. The shape of the spectrum is related to the uniformity of the distribution of objects.

The characteristic spectrum shown in Fig. 14a determined for dividing the Lena's image (shown in Fig. 11) has a regularity in its distribution. For the images a and b shown in Fig. 11, the distribution of elements of obtained divisions is similar. This results in the fact that characteristic spectra overlap (Fig. 14a). The spectrum of a multifractal analysis of these divisions is also very similar. The results of more complex rules of image division are shown in Fig. 11c, d. The differences in the distribution of image divisions show the multifractal spectrum presented in Fig. 14b. However, the characteristic spectrum of these divisions overlaps (Fig. 14a).

6 Conclusions

So, simple algorithm gives wide possibilities to control its operation. Subareas of analysed images are divided in nearly the same way. Algorithm options allow to adjust the algorithm operation to solve problem. It allows to achieve various effects using the same input images. The applied method of the characteristic spectrum determination gives satisfactory results. This method can be helpful in identification of objects or set of objects. The selection plays a very important role. An equally important role in evolutionary algorithms is played by fitness function. Methods of their determination have a significant impact on the algorithm operation.

The future work will include the possibility of the characteristic spectrum applying to identify objects. The algorithm can be adapted to use beamlets. A possible application is the image indexing. The proposed algorithm can be used in the analysis of three-dimensional space. The attempts to use the algorithm in the field of UAV (unmanned aerial vehicle) route

planning and in the related field of sensor networks can be made. The algorithm should be helpful in the similar situations identifying. It seems possible to use fuzzy logic in the control of algorithm operation.

References

Alrawi A, Sagheer A, Ibrahim D (2012) Texture segmentation based on multifractal dimension. Int J Soft Comput (IJSC) 3(1):1–10

Belussi A, Catania B, Clementini E, Ferrari EE (eds) (2007) Spatial data on the web modeling and management. Springer, Berlin. doi:10. 1007/978-3-540-69878-4

Corso G, Freitas J, Lucena L (2004) A multifractal scale-free lattice. Phys A Stat Mech Appl 342(1–2):214–220. doi:10.1016/j.physa. 2004.04.081

Corso G, Lucena L (2005) Multifractal lattice and group theory. Phys A Stat Mech Appl 357(1):64–70. doi:10.1016/j.physa.2005.05.049

Gosciniak I (2017) Discussion on semi-immune algorithm behaviour based on fractal analysis. Soft Comput 21(14):3945–3956. doi:10. 1007/s00500-016-2044-y

Hwang WJ, Derin H (1995) Multiresolution multiresource progressive image transmission. IEEE Trans Image Process 4:1128–1140. doi:10.1109/83.403418

Iwanicki K, van Steen M (2009) Using area hierarchy for multiresolution storage and search in large wireless sensor networks. In: Communications, 2009. ICC '09. IEEE international conference on, pp 1–6. doi:10.1109/ICC.2009.5199556

Juliany J, Vose M (1994) The genetic algorithm fractal. Evol Comput 2(2):165–180. doi:10.1162/evco.1994.2.2.165

Kies P (2001) Information dimension of a population's attractor a binary genetic algorithm. In: Artificial neural nets and genetic algorithms: proceedings of the international conference in Prague, Czech Republic. Springer, pp 232–235. doi:10.1007/978-3-7091-6230-9_57

Kotowski S, Kosinski W, Michalewicz Z, Nowicki J, Przepiorkiewicz B (2008) Fractal dimension of trajectory as invariant of genetic algorithms. In: Artificial intelligence and soft computing (ICAISC 2008). Springer, pp 414–425. doi:10.1007/978-3-540-69731-2_41

Lu Y, Huo X, Tsiotras P (2012) A beamlet-based graph structure for path planning using multiscale information. IEEE Trans Autom Control 57(5):1166–1178. doi:10.1109/TAC.2012.2191836

Marinov M, Kobbelt L (2005) Automatic generation of structure preserving multiresolution models. In: Eurographics, pp 1–8

Masayoshi K, Masaru N, Yoshio S (1996) Identification of complicated shape objects by fractal characteristic variables categorizing dust particles on LSI wafer surface. Syst Comput Jpn 27(6):82–91. doi:10.1002/scj.4690270608

Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer, Berlin. doi:10.1007/978-3-662-03315-9

Mo H (2008) Handbook of research on artificial immune systems and natural computing: applying complex adaptive technologies. Information Science Reference - Imprint of: IGI Publishing. doi:10.4018/978-1-60566-310-4

Pereira M, Corso G, Lucena L, Freitas J (2005) A random multifractal tilling. Chaos Solitons Fractals 23:1105–1110. doi:10.1016/j.chaos.2004.06.045

Rejaur Rahman M, Saha SK (2009) Multi-resolution segmentation for object-based classification and accuracy assessment of land use/land cover classification using remotely sensed data. J Indian Soc Remote Sens 36:189–201. doi:10.1007/s12524-008-0020-4

Song J, Qian F (2006) Fractal algorithm for finding global optimal solution. In: International conference on computational intelligence for modelling control and automation, and international conference on intelligent agents, web technologies and internet commerce (CIMCA–IAWTIC'06). IEEE Computer Society, pp 149–153

Urrutia J, Sack JR (eds) (2000) Handbook of computational geometry. North-Holland, Amsterdam. doi:10.1016/B978-0-444-82537-7.50027-9

Weise T (2009) Global Optimization Algorithms—Theory and Applications, 2nd edn. University of Kassel, Distributed Systems Group. http://www.it-weise.de

Weller R (2013) New geometric data structures for collision detection and haptics. Springer, Cham. doi:10.1007/978-3-319-01020-5

Vujovic I (2014) Multiresolution approach to processing images for different applications: interaction of lower processing with higher vision. Springer, Cham. doi:10.1007/978-3-319-14457-3

Virtual library of simulation experiments: test functions and datasets, optimization test problems. https://www.sfu.ca/ssurjano/optimization.html. Accessed 28 July 2017