



Cuckoo Search Algorithm for Border Reconstruction of Medical Images with Rational Curves

Akemi Gálvez^{1,2}, Iztok Fister³, Iztok Fister Jr.³, Eneko Osaba⁴,
Javier Del Ser^{4,5}, and Andrés Iglesias^{1,2}(✉)

¹ University of Cantabria, Avenida de los Castros s/n, 39005 Santander, Spain
iglesias@unican.es

² Toho University, 2-2-1 Miyama, 274-8510 Funabashi, Japan

³ University of Maribor, Smetanova, Maribor, Slovenia

⁴ TECNALIA, Derio, Spain

⁵ University of the Basque Country (UPV/EHU), Bilbao, Spain
<http://personales.unican.es/iglesias>

Abstract. Border reconstruction is a key technology in medical image processing, where it is applied to identify and separate different tissues, organs, and tumors in diagnostic procedures. The classical approaches for this problem are based on either linear or polynomial functions to describe the border of the region of interest. However, little effort has been devoted to the more powerful case of rational functions, which extend the polynomial case by including extra degrees of freedom (the weights). As a consequence, rational functions are more difficult to compute. In this paper, we solve the problem by applying a nature-inspired swarm intelligence method called cuckoo search algorithm. The method is applied to two illustrative examples of medical images with satisfactory results.

Keywords: Swarm intelligence · Cuckoo search algorithm · Medical imaging · Border reconstruction · Rational curves

1 Introduction

The automatic detection and reconstruction of the borders of objects and areas in images has been a major topic of research in several areas for decades. This problem arises very often in fields such as image processing, pattern recognition, artificial vision, and virtual and augmented reality, to mention just a few. It is also a key technology for medical applications, as it helps to identify and discriminate different tissues and organs for medical visualization in popular non-invasive diagnostic procedures such as computer tomography, magnetic resonance imaging, infrared imaging, dermoscopy, ultrasonography, magnetic resonance spectroscopy, and many others. In fact, medical image processing is one of the most relevant application fields of border detection and reconstruction.

The classical graphical pipeline in automatic medical image processing consists of three stages: (1) image segmentation; (2) feature extraction and feature selection; and (3) classification. Typically, the process starts with a medical image obtained by either of the many clinical procedures based on this technology. While in the past the image was directly used by the medical specialist for visual analysis for diagnostic and therapeutic purposes, the current trend today is to automate the process as much as possible [7,10,11]. However, in many cases the first step of the process is still highly manual: after visual inspection of the image, the medical specialist selects a set of points of the image by clicking with the mouse on the computer screen or a display device. Such points, usually referred to as feature points, correspond to the border between regions of interest (ROIs), usually corresponding to different tissues or organs, or enclosing a possible lesion or tumor. This process is called border detection [2]. After this initial selection of feature points, a computer software applies graphical routines to determine the border of the ROI under analysis, a process called border reconstruction.

The classical procedure in border reconstruction of medical images is to consider a simple polyline connecting consecutive feature points with straight segments. The output of this process is a polygonal enclosing the ROI. Such a polyline gives a rough approximation of the boundary between the ROI and the surrounding background, which is usually enough in many cases, provided that the number of feature points is large enough to describe all the geometry of the border with certain accuracy. However, this procedure can be improved by considering free-form parametric curves, which take advantage of a higher number of degrees of freedom to add extra flexibility to this process. For instance, the polyline does not represent the real process well, as the border of medical images is not generally piecewise linear. In addition, this linear procedure relies on interpolation schemes that enforce the border to pass through *all* feature points. This is often troublesome, as medical data are typically affected by artifacts inherent to manual processes, such as outliers and noise in data. Under such conditions, approximation schemes (that only require the border to pass *near* the feature points) are better suited for this problem [1]. Furthermore, approximation schemes are more advantageous in terms of accuracy, computer memory, and data storage capacity, as the border can be accurately described by a few tens of parameters even for very large collections of feature points. Because of these reasons, in this paper we will consider approximation schemes for this border reconstruction step.

Classical approximation techniques for border reconstruction with free-form parametric curves are primarily based on polynomial schemes, such as the popular Bézier and B-spline curves [6]. Such schemes can be improved by considering some extra real parameters called weights, which allow the user to modify the shape of the curve locally by simply changing the weight of one or several poles of the curve without changing the location of the poles. This is an interesting and valuable feature, as it makes it possible to reduce the degree of the curve significantly without penalizing the approximation accuracy. The resulting parametric curve is no longer a polynomial function but a rational one.

Unfortunately, using rational curves is by far much more difficult than the polynomial case, because some extra variables (the weights) have also to be computed. In addition, the different variables (data parameters, poles, and weights) are strongly related to each other in a highly nonlinear way [3]. As a result, we have to solve a difficult continuous multivariate nonlinear optimization problem that cannot be properly solved in the general case through traditional mathematical optimization techniques.

Our method to solve this problem is based on a nature-inspired metaheuristics called cuckoo search algorithm and introduced by Prof. X.S. Yang in 2009 to solve difficult optimization problems [14]. The algorithm is inspired by the obligate interspecific brood-parasitism of some cuckoo species that lay their eggs in the nests of host birds of other species. Since its inception, the cuckoo search (specially its variant that uses Lévy flights) has been successfully applied in several papers reported recently in the literature to difficult optimization problems from different domains [12, 13, 15], including data fitting with polynomial curves [5]. However, to the best of our knowledge, the method has never been used so far for border reconstruction of medical images with rational curves.

The structure of this paper is as follows: the fundamentals and main features of the cuckoo search algorithm are discussed in Sect. 2. The problem of data fitting with rational Bézier curves is discussed in Sect. 3. The proposed method to solve the border reconstruction problem is presented in Sect. 4. To illustrate the performance of our method, it is applied in Sect. 5 to perform border reconstruction of two medical images. The paper closes in Sect. 6 with the main conclusions of this contribution and our plans for future work in the field.

2 The Cuckoo Search Algorithm

Cuckoo search (CS) is a nature-inspired population-based metaheuristic algorithm originally proposed by Yang and Deb in 2009 to solve optimization problems [14]. The algorithm is inspired by the brood-parasitism of some cuckoo species that lay their eggs in the nests of host birds of other species with the aim of escaping from the parental investment in raising their offspring. This strategy is also useful to minimize the risk of egg loss to other species, as the cuckoos can distributed their eggs amongst a number of different nests. Of course, sometimes it happens that the host birds discover the alien eggs in their nests. In such cases, the host bird can take different responsive actions varying from throwing such eggs away to simply leaving the nest and build a new one elsewhere.

This interesting and surprising breeding behavioral pattern is the metaphor of the cuckoo search metaheuristic approach for solving optimization problems. In the cuckoo search algorithm, the eggs in the nest are interpreted as a pool of candidate solutions of an optimization problem while the cuckoo egg represents a new coming solution. The ultimate goal of the method is to use these new (and potentially better) solutions associated with the parasitic cuckoo eggs to replace the current solution associated with the eggs in the nest. This replacement, carried out iteratively, will eventually lead to a very good solution of the problem.

In addition to this representation scheme, the CS algorithm is also based on three idealized rules [14,15]:

Table 1. Cuckoo Search Algorithm via Lévy flights as originally proposed in [14,15].

Algorithm: Cuckoo Search via Lévy Flights

```

begin
  Objective function  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_D)^T$ 
  Generate initial population of  $n$  host nests  $\mathbf{x}_i$  ( $i = 1, 2, \dots, n$ )
  while ( $t < MaxGeneration$ ) or (stop criterion)
    Get a cuckoo (say,  $i$ ) randomly by Lévy flights
    Evaluate its fitness  $F_i$ 
    Choose a nest among  $n$  (say,  $j$ ) randomly
    if ( $F_i > F_j$ )
      Replace  $j$  by the new solution
    end
    A fraction ( $p_a$ ) of worse nests are abandoned and new ones
      are built via Lévy flights
    Keep the best solutions (or nests with quality solutions)
    Rank the solutions and find the current best
  end while
  Postprocess results and visualization
end

```

1. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
2. The best nests with high quality of eggs (solutions) will be carried over to the next generations;
3. The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest and build a new nest in a new location.

For simplicity, the third assumption can be approximated by a fraction p_a of the n nests being replaced by new nests (with new random solutions at new locations). For a maximization problem, the quality or fitness of a solution can simply be proportional to the objective function. However, other (more sophisticated) expressions for the fitness function can also be defined.

Based on these three rules, the basic steps of the CS algorithm can be summarized as shown in the pseudo-code reported in Table 1. Basically, the CS algorithm starts with an initial population of n host nests and it is performed iteratively. In the original proposal, the initial values of the j th component of the i th nest are determined by the expression $x_i^j(0) = rand.(up_i^j - low_i^j) + low_i^j$, where up_i^j and low_i^j represent the upper and lower bounds of that j th component, respectively, and $rand$ represents a standard uniform random number on

the open interval $(0, 1)$. Note that this choice ensures that the initial values of the variables are within the search space domain. These boundary conditions are also controlled in each iteration step.

For each iteration g , a cuckoo egg i is selected randomly and new solutions $\mathbf{x}_i(g + 1)$ are generated by using the Lévy flight, a kind of random walk in which the steps are defined in terms of the step-lengths, which have a certain probability distribution, with the directions of the steps being isotropic and random. According to the original creators of the method, the strategy of using Lévy flights is preferred over other simple random walks because it leads to better overall performance of the CS. The general equation for the Lévy flight is given by:

$$\mathbf{x}_i(g + 1) = \mathbf{x}_i(g) + \alpha \oplus \text{levy}(\lambda) \tag{1}$$

where g indicates the number of the current generation, and $\alpha > 0$ indicates the step size, which should be related to the scale of the particular problem under study. The symbol \oplus is used in Eq. (1) to indicate the entry-wise multiplication. Note that Eq. (1) is essentially a Markov chain, since next location at generation $g + 1$ only depends on the current location at generation g and a transition probability, modulated by the Lévy distribution as:

$$\text{levy}(\lambda) \sim g^{-\lambda}, \quad (1 < \lambda \leq 3) \tag{2}$$

which has an infinite variance with an infinite mean. The generation of random numbers with Lévy flights is comprised of two steps: firstly, a random direction according to a uniform distribution is chosen; then, the generation of steps following the chosen Lévy distribution is carried out. The authors suggested to use the so-called Mantegna’s algorithm for symmetric distributions, where “symmetric” means that both positive and negative steps are considered (see [12] for details). Their approach computes the factor:

$$\hat{\phi} = \left(\frac{\Gamma(1 + \hat{\beta}) \cdot \sin\left(\frac{\pi \cdot \hat{\beta}}{2}\right)}{\Gamma\left(\left(\frac{1 + \hat{\beta}}{2}\right)\right) \cdot \hat{\beta} \cdot 2^{\frac{\hat{\beta}-1}{2}}}\right)^{\frac{1}{\hat{\beta}}} \tag{3}$$

where Γ denotes the Gamma function and $\hat{\beta} = \frac{3}{2}$ in the original implementation by Yang and Deb [15]. This factor is used in Mantegna’s algorithm to compute the step length ς as: $\varsigma = \frac{u}{|v|^{\frac{1}{\hat{\beta}}}}$, where u and v follow the normal distribution of zero mean and deviation σ_u^2 and σ_v^2 , respectively, where σ_u obeys the Lévy distribution given by Eq. (3) and $\sigma_v = 1$. Then, the stepsize ζ is computed as:

$$\zeta = 0.01 \varsigma (\mathbf{x} - \mathbf{x}_{best}) \tag{4}$$

where ς is computed as above. Finally, \mathbf{x} is modified as: $\mathbf{x} \leftarrow \mathbf{x} + \zeta \cdot \Psi$ where Ψ is a random vector following the normal distribution $N(0, 1)$.

The CS method then evaluates the fitness of the new solution and compares it with the current one. In case the new solution brings better fitness, it replaces the current one. On the other hand, a fraction of the worse nests (according to the fitness) are abandoned and replaced by new solutions so as to increase the exploration of the search space looking for more promising solutions. The rate of replacement is given by the probability p_a , a parameter of the model that has to be tuned for better performance. Moreover, for each iteration step, all current solutions are ranked according to their fitness and the best solution reached so far is stored as the vector \mathbf{x}_{best} (used, for instance, in Eq. (4)). This algorithm is applied in an iterative fashion until a stopping criterion is met.

3 Problem to Be Solved

We assume that the reader is familiar with the free-form parametric curves [8,9]. Mathematically, a *free-form rational Bézier curve* $\Phi(\tau)$ of degree η is defined as:

$$\Phi(\tau) = \frac{\sum_{j=0}^{\eta} \omega_j \mathbf{\Lambda}_j \phi_j^{\eta}(\tau)}{\sum_{j=0}^{\eta} \omega_j \phi_j^{\eta}(\tau)} \tag{5}$$

where $\mathbf{\Lambda}_j$ are vector coefficients called the *poles*, ω_j are their scalar weights, $\phi_j^{\eta}(\tau)$ are the *Bernstein polynomials of index j and degree η* , given by:

$$\phi_j^{\eta}(\tau) = \binom{\eta}{j} \tau^j (1 - \tau)^{\eta-j}$$

and τ is the *curve parameter*, defined on the finite interval $[0, 1]$. By convention, $0! = 1$. Note that in this paper vectors are denoted in bold.

Suppose now that we are given a set of data points $\{\Delta_i\}_{i=1, \dots, \kappa}$ in \mathbb{R}^{ν} (usually $\nu = 2$ or $\nu = 3$). Our goal is to obtain the rational Bézier curve $\Phi(\tau)$ performing discrete approximation of the data points $\{\Delta_i\}_i$. To do so, we have to compute all parameters (i.e. poles $\mathbf{\Lambda}_j$, weights ω_j , and parameters τ_i associated with data points Δ_i , for $i = 1, \dots, \kappa, j = 0, \dots, \eta$) of the approximating curve $\Phi(\tau)$ by minimizing the least-squares error, \mathcal{Y} , defined as the sum of squares of the residuals:

$$\mathcal{Y} = \underset{\substack{\{\tau_i\}_i \\ \{\mathbf{\Lambda}_j\}_j \\ \{\omega_j\}_j}}{\text{minimize}} \left[\sum_{i=1}^{\kappa} \left(\Delta_i - \frac{\sum_{j=0}^{\eta} \omega_j \mathbf{\Lambda}_j \phi_j^{\eta}(\tau_i)}{\sum_{j=0}^{\eta} \omega_j \phi_j^{\eta}(\tau_i)} \right)^2 \right]. \tag{6}$$

Now, taking:

$$\varphi_j^{\eta}(\tau) = \frac{\omega_j \phi_j^{\eta}(\tau)}{\sum_{k=0}^{\eta} \omega_k \phi_k^{\eta}(\tau)} \tag{7}$$



Fig. 1. Original medical images and their corresponding selected border feature points.

Eq. (6) becomes:

$$\mathcal{Y} = \underset{\substack{\{\tau_i\}_i \\ \{\Lambda_j\}_j \\ \{\omega_j\}_j}}{\text{minimize}} \left[\sum_{i=1}^{\kappa} \left(\Delta_i - \sum_{j=0}^{\eta} \Lambda_j \varphi_j^{\eta}(\tau) \right)^2 \right], \quad (8)$$

which can be rewritten in matrix form as: $\mathbf{\Omega} \cdot \mathbf{\Lambda} = \mathbf{\Xi}$, where: $\mathbf{\Omega} = [\Omega_{i,j}] = \left[\left(\sum_{k=1}^{\kappa} \varphi_i^{\eta}(\tau_k) \varphi_j^{\eta}(\tau_k) \right)_{i,j} \right]$, $\mathbf{\Xi} = [\Xi_j] = \left[\left(\sum_{k=1}^{\kappa} \Delta_k \varphi_j^{\eta}(\tau_k) \right)_j \right]$, $\mathbf{\Lambda} = (\Lambda_0, \dots, \Lambda_{\eta})^T$, for $i, j = 0, \dots, \eta$, and $(\cdot)^T$ means the transposition of a vector or a matrix. In general, $\kappa \gg \eta$ meaning that the system of equations $\mathbf{\Omega} \cdot \mathbf{\Lambda} = \mathbf{\Xi}$ is over-determined. If values are assigned to the τ_i , our problem can be solved as a classical linear least-squares minimization, with the coefficients $\{\Lambda_i\}_{i=0, \dots, \eta}$ as unknowns. This problem can readily be solved by standard numerical techniques. On the contrary, if the values of τ_i are treated as unknowns, the problem becomes much more difficult. Indeed, since the polynomial blending functions $\phi_j^{\eta}(\tau)$ are nonlinear in τ and so are the rational blending functions $\varphi_j^{\eta}(\tau)$, the least-squares minimization of the errors is a nonlinear continuous optimization problem. Note also that in many practical cases the number of data points can be extremely large, meaning that we have to deal with a large number of unknowns. It is also a multimodal problem, since there might be arguably more than one set of parameter values leading to the optimal solution.

In conclusion, the complex interplay among all sets of unknowns (data parameters, poles, and weights) leads to a very difficult over-determined, multimodal, multivariate, continuous, nonlinear optimization problem. In this work, we are interested to solve this general problem. Instead of making assumption about the values of some free parameters, we include all of them in our computations.



Fig. 2. Graphical results for the first example: image and rational border curve: (left) without and (right) with the feature points.

4 The Cuckoo Search Method

Our approach is based on the application of the cuckoo search method with Lévy flights described in Sect. 2 to our initial input, i.e., the set of feature points $\{\Delta_i\}_{i=1,\dots,\kappa}$. We consider an initial parameterization and set of weights, assumed to be random in their respective domains, and apply the cuckoo search on a collection of individuals consisting of the vector $S^g = \{\mathcal{P}^g, \mathcal{W}^g\}$, where $\mathcal{P}^g = \{\tau_1^g, \dots, \tau_\kappa^g\}$, $\mathcal{W}^g = \{\omega_0^g, \dots, \omega_\eta^g\}$, and the superscript g denotes the generation index. The procedure computes the final values of data parameters and weights. Then, inserting them into Eq. (7), we apply least-squares minimization to compute the values of $\{\Lambda_i\}_{i=0,\dots,\eta}$ according to Eq. (8).

As it is very well-known, the parameter tuning of metaheuristic methods is largely a problem-dependent issue [4]. Fortunately, the cuckoo search is specially advantageous in this regard. In clear contrast to other metaheuristics that typically depend on several parameters, cuckoo search depends on only two parameters (the population size, n and the probability p_a) so the parameter tuning becomes a much simpler task. In this paper they are set to $n = 100$ and $p_a = 0.25$, as these values have already been used in previous papers with good results [5]. Regarding our stopping criterion, the method was executed for n_{iter} iterations, with n_{iter} set to 50,000 generations in our examples.

5 Experimental Results

Our method has been applied to two medical images of skin cancer obtained from the digital image archive of the Medical Center, University of Groningen (The Netherlands). They are shown in Fig. 1. The figure shows the original images with a collection of superimposed feature points identified by a trained dermatologist as belonging to the boundary of the skin lesion. Our results for both examples

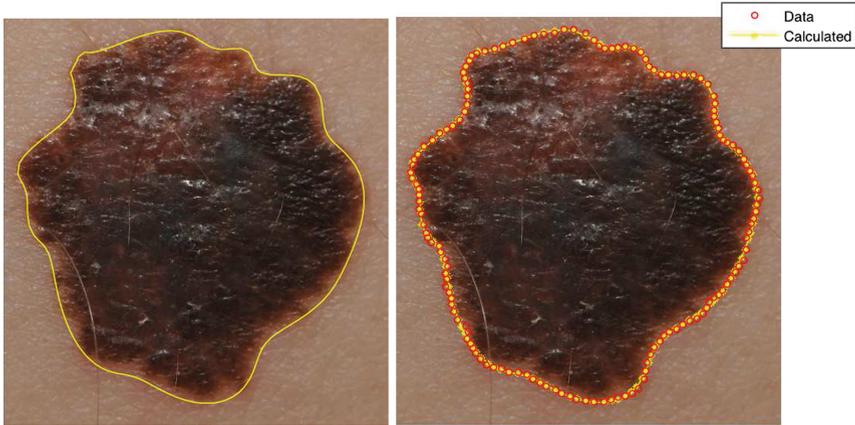


Fig. 3. Graphical results for the second example: image and rational border curve: (left) without and (right) with the feature points.

are shown in Figs. 2 and 3, respectively. The figures show the original medical image and the best reconstructed rational border curve without (left) and with (right) the feature points. From the figures we can see that the method obtains a very good fitting of the data points for both examples. The corresponding convergence diagrams for both examples are shown in Fig. 4.

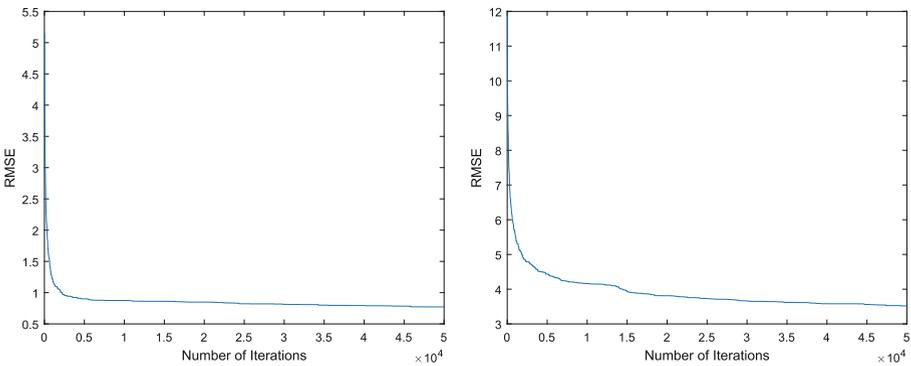


Fig. 4. Convergence diagram for the first (left) and the second example (right).

6 Conclusions and Future Work

In this paper we focus on the problem of border reconstruction for medical images with rational curves. Given a collection of feature points assumed to lie on the boundary of a certain region of interest, the goal is to determine the rational

curve that best fits these data points in the least-squares sense. Although this problem has already been addressed in the literature for the polynomial case, it was still unsolved for rational curves, which are more general and flexible but also more challenging, as they include some extra degrees of freedom (the weights) that have also to be computed. The method presented in this paper is based on the cuckoo search algorithm. It has been applied to two examples of medical images with visually satisfactory results. The limiting factor of this method is that it requires many iterations to converge, making it unsuitable for real-time applications. Future work includes the extension of this approach to the case of piecewise rational functions, its generalization to the case of surfaces for volumetric reconstruction, and reducing the number of required iterations.

Acknowledgements. Work supported by projects: PDE-GIR #778035 (EU Horizon 2020 program), #TIN2017-89275-R (Spanish Research Agency, AEI/UE FEDER), P2-0057 & P2-0041 (Slovenian Research Agency) and EMAITEK (Basque Government).

References

1. Abbas, A.A., Guo, X., Tan, W.H., Jalab, H.A.: Combined spline and B-spline for an improved automatic skin lesion segmentation in dermoscopic images using optimal color channel. *J. Med. Syst.* **38**(80), 1–8 (2014)
2. Celebi, M.E., Iyatomi, H., Schaefer, G., Stoecker, W.V.: Lesion border detection in dermoscopy images. *Comput. Med. Imaging Graph.* **33**(2), 148–153 (2009)
3. Dierckx, P.: *Curve and Surface Fitting with Splines*. Oxford University Press, New York (1993)
4. Engelbrecht, A.P.: *Fundamentals of Computational Swarm Intelligence*. Wiley, Chichester (2005)
5. Gálvez A., Iglesias A.: Cuckoo search with Lévy flights for weighted Bayesian energy functional optimization in global-support curve data fitting. *Sci. World J.* **11** (2014). Article ID 138760
6. Gálvez A., Iglesias A.: Computational intelligence CSA-based approach for machine-driven calculation of outline curves of cutaneous melanoma. In: *Proceedings of Cyberworlds 2018*, pp. 270–275. IEEE CS Press (2018)
7. Garnavi, R., Aldeen, M., Celebi, M.E., Varigos, G., Finch, S.: Border detection in dermoscopy images using hybrid thresholding on optimized color channels. *Comput. Med. Imaging Graph.* **35**(2), 105–115 (2011)
8. Iglesias, A., Gálvez, A.: Memetic electromagnetism algorithm for finite approximation with rational bézier curves. In: Tan, Y., Shi, Y., Buarque, F., Gelbukh, A., Das, S., Engelbrecht, A. (eds.) *ICSI 2015*. LNCS, vol. 9140, pp. 30–40. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20466-6_3
9. Iglesias, A., Gálvez, A., Collantes, M.: Global-support rational curve method for data approximation with bat algorithm. In: Chbeir, R., Manolopoulos, Y., Maglogiannis, I., Alhadj, R. (eds.) *AIAI 2015*. IAICT, vol. 458, pp. 191–205. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23868-5_14
10. Machado, D.A., Giralaldi, G., Novotny, A.A.: Multi-object segmentation approach based on topological derivative and level set method. *Integr. Comput. Aided Eng.* **18**, 301–311 (2011)

11. Schmid, P.: Segmentation of digitized dermatoscopic images by two-dimensional color clustering. *IEEE Trans. Med. Imaging* **18**(2), 164–171 (1999)
12. Yang, X.-S.: *Nature-Inspired Metaheuristic Algorithms*, 2nd edn. Luniver Press, Frome (2010)
13. Yang, X.-S.: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, New Jersey (2010)
14. Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: *Proceedings World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214. IEEE (2009)
15. Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. *Int. J. Math. Model. Numer. Optim.* **1**(4), 330–343 (2010)