Optik 127 (2016) 11775–11785

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Original research article

Image retrieval via balance-evolution artificial bee colony algorithm and lateral inhibition

Bai Li[a,b], Changjun Zhou[a,*], Hong Liu[c], Ya Li[d], Hongxin Cao[e]

[a] Key Laboratory of Advanced Design and Intelligent Computing, Dalian University, Dalian, 116622, PR China
[b] College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, PR China
[c] School of Information and Electrical Engineering, Zhejiang University City College, Hangzhou, 310015, PR China
[d] School of Mathematics and Systems Science & LMIB, Beihang University, Beijing, 100191, PR China
[e] College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, PR China

ARTICLE INFO

Article history:
Received 22 July 2016
Accepted 20 September 2016

Keywords:
Image retrieval
Metaheuristics
Lateral inhibition
Artificial bee colony
Overall degradation

ABSTRACT

Image retrieval is a fundamental issue in pattern recognition. In this work, lateral inhibition (LI) model is adopted as a pre-processing step, which widens the gray level gradients so as to facilitate the image retrieval scheme. In searching for a perfect match between a predefined template and a reference image, we adopt metaheuristic algorithms for good seach capability. Artificial bee colony (ABC) algorithm is a bio-inspired optimization technique, which imitates the foraging behavior of honey bee swarms. It is well known that the algorithm is good at exploration but poor at exploitation. We present balance-evolution artificial bee colony (BE-ABC) algorithm that aims to strike a balance between exploration and exploitation rather than just focusing on improving the latter. BE-ABC algorithm adaptively manipulates the search intensity at the exploration and exploitation stages during the iterations. Besides that, it incorporates an overall degradation procedure to prevent premature convergence. Simulation results confirm that BE-ABC algorithm is more capable than several state-of-the-art metaheuristic algorithms in this image retrieval scheme.

1. Introduction

Image retrieval is a way of recognizing pre-defined template patterns in the reference images [1]. Broadly speaking, similarity measurement and similarity search are two major procedures in image retrieval [2]. Specifically, similarity measurement refers to the action to evaluate how similar a reference image is with a pre-defined template image, when the template image masks over some specific region of the testing image. In the community of image retrieval, sum of absolute differences (SAD), sum of squared differences (SSD), and normalized cross correlation (NCC) are well-known ones among all the similarity measurement criterions [3–5]. Compared with SAD and SSD criteria, NCC is characterized by the robust matching ability, especially when illumination changes occur in the reference images [6]. This study considers measuring image retrieval similarity using NCC criterion. With regard to searching for the best-matching location, Ref. [7] applies a thorough search algorithm, in which all the pixel-candidate positions are checked until the one with the maximum similarity is located. Applications of thorough search algorithm are limited due to the heavy computational burden. In the past decade,

* Corresponding author.
E-mail addresses: libaioutstanding@163.com, libai@zju.edu.cn (B. Li), zhouchangjun@dlu.edu.cn (C. Zhou), liuhong@zucc.edu.cn (H. Liu), yli@buaa.edu.cn (Y. Li), chxin@zju.edu.cn (H. Cao).

http://dx.doi.org/10.1016/j.ijleo.2016.09.085
0030-4026/

intelligent search strategies have been largely developed, e.g., states-of-matter search algorithm [2], species-based genetic algorithm [8], bat algorithm [9], and chaotic quantum-behaved particle swarm optimization (CQ-PSO) [10]. In general, intelligent searchers aim to facilitate the best-matching solution finding process, but seldom do them guarantee obtaining best matching solutions within a certain period of time. Besides that, typical intelligent optimizers are designed for continuous optimization problems, whereas the 2-dimensional solution space is discrete and ill-conditioned in this image retrieval scheme, thus bringing about challenges. In other words, the solution surface would drastically oscillate in the domain, further restricting the efficient application of the typical intelligent optimizers. Therefore, there might still be room of improvement for those prevailing intelligent optimization methodologies to accomplish this concerned image retrieval scheme. This study provides a preliminary attempt in tackling the computational difficulties behind the scheme by adopting a modified artificial bee colony (ABC) algorithm.

As a typical swarm intelligence algorithm, ABC algorithm simulates the foraging behavior of honey bees [11]. Specifically, an effective bee swarm is assumed to consist of three components, namely, the employed bees, onlooker bees, and scout bees. Scout bees are responsible for finding nectar sources in the entire space in a blind manner. In contrast, employed bees are able to exchange information with all other colleagues, thereby making their search behavior more intentional. If high-quality nectar sources are roughly detected (by either scout bees or employed bees), on-looker bees would conduct local searches around to get the most out of it [12,13]. The simple structure of ABC algorithm makes it possible to acquire good results at a low computational cost, giving rise to applications spanning across diverse areas such as neural network training [14,15], structure design [16–19], scheduling [20], image processing [21,22], dynamic programming [23], among many others.

Broadly speaking, there have been three typical ways to improve the original ABC algorithm [24]. The first way is to adopt some strategies or theories from the outside world, such as the Rosenbrock's rotational direction strategy [25], idle time reduction techniques [26] and quantum theories [27]. Hybridized proposals that combine ABC with other algorithms fall into this category as well [28]. The second way concerns about modifying the solution search equations [29,30]. The third way is about utilizing convergence information from the interior algorithm framework. Refs. [31] and [32] are two typical examples that aim to make full use of interior states to accelerate the optimization process. Our balance-evolution artificial bee colony (BE-ABC) algorithm [24,33] pursues to use interior indications to adaptively control the global/local search intensity as well. Ideally, we hope to see no explicit gaps between the exploration and exploitation stages. Specifically, when the global search scale is small, the search process is more likely a local search; and when the local search scale is large, it is more likely a global search; thus it is fine if the search intensity can adaptively conform to true needs in each iteration. BE-ABC algorithm is applied for the image retrieval optimization scheme in the present study.

The remainder of this paper is organized as follows. In Sections 2 and 3, basic principles of LI-based image retrieval model and the conventional ABC algorithm are introduced. Section 4 holds a description of BE-ABC algorithm. Simulations results are shown in Section 5, wherein BE-ABC algorithm is compared to several state-of-the-art metaheuristics in several LI-based image retrieval schemes. Conclusions are drawn in the last section.

2. Principle of lateral-inhibition based image retrieval model

The phenomenon of LI was discovered by Hartline and Graham in 1932, during an electrophysiology experiment on limulus vision [34]. They noticed that each ommateum of a limulus was inhibited by its neighboring ommatea, and the nearer they were, the stronger the inhibited effect would be [35]. Such inhibited effect is mutual and spatially summable. In this way, the light and shade contrast in the sense of vision is enhanced. With regard to the mechanism of retinal imaging, some further research has confirmed that, excited retinal ganglion cells inhibit those non-excited ones more strongly than the non-excited retinal ganglion cells to the excited ones. In this work, LI model is applied to the image retrieval scheme as a pre-processing procedure in order to promote matching accuracy.

In the LI pre-processing procedure, the original reference image and pre-defined template image are changed according to Eq. (1):

$$\mathbf{R}(x,y) \triangleq \mathbf{I_0}(x,y) + \sum_{i=-M}^{M}\sum_{j=-N}^{N}\alpha_{ij}\cdot\mathbf{I_0}(x+i,y+j), \tag{1}$$

where $\mathbf{I_0}(x,y)$ represents the gray level of the pixel locating at (x,y) on the original reference/template image, α_{ij} stands for the lateral inhibition coefficient of the pixel locating at $(x+i,y+j)$ to the central pixel at (x,y), and $\mathbf{R}(x,y)$ denotes the pre-processed gray level. The selection of α_{ij} can be found in Ref. [35]. Fig. 1 schematically shows the structure under the condition that $M=N=2$.

When all the pixels on the template/reference image are pre-processed, the edge features of the template/reference image are extracted by means of calculating their corresponding binary images following Eq. (2), where $g_0 \in [0,255]\cap\mathbb{Z}$ is a user-specified threshold [10].

$$\boldsymbol{B}(x,y) = \begin{cases} 1, \text{when} \boldsymbol{R}(x,y) > g_0 \\ 0, \text{when} \boldsymbol{R}(x,y) \le g_0 \end{cases}. \tag{2}$$

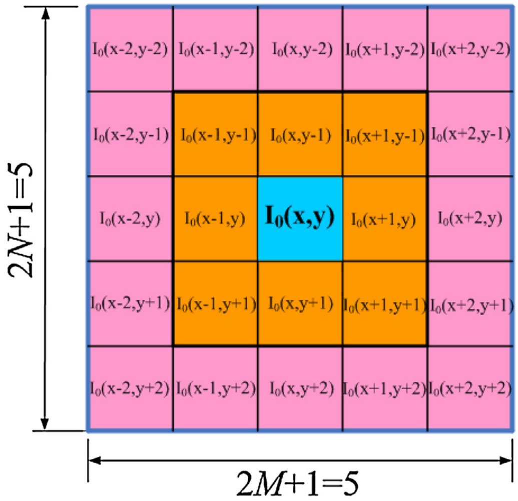

Fig. 1. Schematic diagram of lateral inhibition model under the condition that $M = N = 2$. $\mathbf{I_0}(x, y)$ stands for gray value of central pixel (x, y). Note that all these presented surrounding pixels have influences on the computation of central pixel.

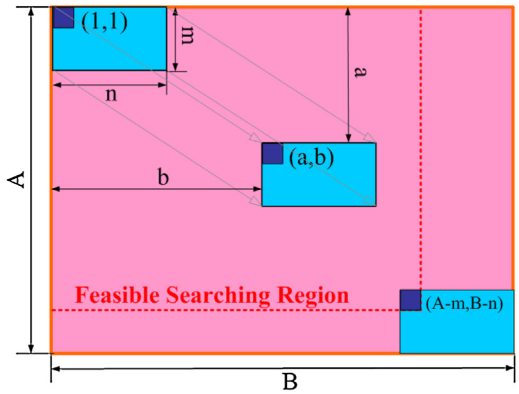

Fig. 2. Schematic diagram of image retrieval procedure. Note that large pink box represents binary reference image (A × B in size), a small (blue) box represents binary template image ($m \times n$ in size). Red dotted lines enclose the searching region within which all feasible solutions exist. This diagram visually expresses a candidate searching location at (a, b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Thereafter, template matching criteria are used to evaluate the matching similarities in different overlapping locations between the binary reference image and the binary template image. Given that the original NCC criterion is complicated, this study slightly simplifies the criterion definition as follows, following the customs in Refs. [6,10,36]. I.e., the similarity *Simi* is defined by Eqs. (3) and (4).

$$Simi([x, y]) = \frac{1}{m \cdot n}\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\phi_{ij}\right), \tag{3}$$

$$\phi_{ij} = \begin{cases} 1, \text{when}\mathbf{B_1}(i, j) = \mathbf{B_2}(x + i, y + j) \\ 0, \text{when}\mathbf{B_1}(i, j) \neq \mathbf{B_2}(x + i, y + j) \end{cases}. \tag{4}$$

Here, $\mathbf{B}_1$ represents the derived binary template image (which is $m \times n$ in size), and $\mathbf{B}_2$ represents the binary reference image. Fig. 2 shows how a template is overlapped onto a reference image at a candidate location (a, b) for similarity measurement *Simi* $([a, b])$. When $Simi = 1$, the corresponding solution $[x^*, y^*]$ refers to the global optimal one. In this study, *Simi* $([\bullet, \bullet])$ is chosen as the maximization objective when searching for satisfactory match configuration. The following two sections concern about the methodologies we use to solve the maximization problem.

3. Brief introduction of artificial bee colony algorithm

In ABC algorithm, an initial population is randomly generated using Eq. (5) first, which contains as many as $SN/2$ food sources (i.e., $SN/2$ feasible solutions). Each solution is denoted by a D-dimensional vector $\mathbf{X} = (X^1, X^2, \cdots, X^D)$.$\mathbf{X}_{\max}$ and $\mathbf{X}_{\min}$ are the pre-defined constraints of the optimization problem, and $rand(0, 1)$ denotes a random number in the range (0, 1) obeying the uniform distribution. Here, we convert our concerned maximization problem $\max(Simi([\bullet, \bullet]))$ into a minimization problem $\min(-Simi([\bullet, \bullet]))$, which is abstracted as $\min(obj)$ in this section and the next.

$$X_i^j = X_{\min}^j + rand(0, 1) \cdot (X_{\max}^j - X_{\min}^j), i = 1, 2, ..., {}^{SN}\!/\!_2, j = 1, 2, ..., D, \tag{5}$$

Then, the iteration process starts and as many as ${}^{SN}\!/\!_2$ employed bees search globally in each cycle of iteration. They utilize the position of one randomly chosen companion so as to generate a new searching position. Note that here only one (randomly chosen) element of the vector $\mathbf{X}$ is involved in such crossover and mutation procedure. For instance, when the ith employed bee utilizes position of the kth companion in the jth element, the involved elements is changed according to (6).

$$X{*}_i^j = X_i^j + rand(-1, 1) \cdot (X_k^j - X_i^j), k \in \left\{1, 2, \ldots, {}^{SN}\!/\!_2\right\}, j \in \left\{1, 2, \ldots, D\right\}, k \neq i. \tag{6}$$

Afterwards, the SN onlooker bees search locally around the “qualified” employed bees. The roulette selection strategy is employed as the principle to select the qualified employed bees. It is carried out as the follows: if $P_1 \geq rand(0, 1)$, the 1st employed bee is chosen for the specific onlooker bee; otherwise, comparison between P_2 and $rand(0, 1)$ is carried on. If all the P_i are smaller than $rand(0, 1)$, such process goes over again until one employed bee satisfies the condition. The employed bees that cannot make any progress within some certain cycles will be replaced by the scout bees.

Next, the greedy selection procedure is implemented. If the new position updated using Eq. (5) is better (i.e. the corresponding similarity degree is higher), the previous position is discarded; otherwise, the employed bee remains at the previous position. When all the ${}^{SN}\!/\!_2$ employed bees complete the searching procedure mentioned above, an index P is calculated as the qualification measurement for the employed bees using Eqs. (7) and (8).

$$P(i) = \frac{fitness(i)}{\sum_{j=1}^{SN} fitness(j)}, \tag{7}$$

$$fitness(i) = \begin{Bmatrix} \dfrac{1}{1 + obj(\boldsymbol{X}_i)} \text{if} obj(\boldsymbol{X}_i) \geq 0 \\ 1 + \text{abs}(obj(\boldsymbol{X}_i)) \text{if} obj(\boldsymbol{X}_i) < 0 \end{Bmatrix}. \tag{8}$$

The following Eq. (9) shows the location of the ith onlooker bee $\mathbf{Y}_i = \left(X_j^1, \ldots, X_j^{k-1}, Y_i^k, X_j^{k+1}, \ldots, X_j^D\right)$ that searches locally around the selected jth employed bee.

$$Y_i^k = X_j^k + rand(-1, 1) \cdot (X_m^k - X_j^k), m \in \left\{1, 2, \ldots, {}^{SN}\!/\!_2\right\}, k \in \left\{1, 2, \ldots, D\right\}, m \neq j. \tag{9}$$

Note that in this equation m and k are randomly selected integers as well. When all the ${}^{SN}\!/\!_2$ onlooker bees have determined their locations, a greedy selection strategy is implemented. This time, however, a comparison is made between $obj(\mathbf{X}_j)$ and $obj(\mathbf{Y}_i)$, $i = 1, 2, \ldots, {}^{SN}\!/\!_2$. If $obj(\mathbf{Y}_i)$ is smaller than $obj(\mathbf{X}_j)$, the jth employed bee will abandon the current location $\mathbf{X}_j$ and go to $\mathbf{Y}_i$, i.e., $\mathbf{X}_j \longleftarrow \mathbf{Y}_i$; otherwise, the jth employed bee remains at $\mathbf{X}_j$. Again, the greedy selection procedure is implemented here. In this way, each of the ${}^{SN}\!/\!_2$ onlooker bees determines which employed bee to follow respectively.

During each cycle of the iteration, once the ith employed bee or an onlooker bee (which searches around the ith employed bee) finds a better position in the crossover procedure, the parameter $trial(i)$ is directly reset to zero; otherwise, it is added by one. In this sense, $trial$ is regarded as a counter recording the invalid searching times around the ith employed bee. Before a new cycle of iteration starts, we need to check whether any $trial(i)$ exceeds a certain thresholdLimit. If $trail(i) > \text{Limit}$, the ith employed bee will be directly replaced by a scout bee, which simply stands for a randomly initialized position in the food source utilizing Eq. (5).

4. Principle of BE-ABC algorithm

The BE-ABC algorithm strives to overcome issues or limitations associated with the conventional ABC algorithm as discussed in the previous section. To do so, a number of improvements have been made in both the exploration and exploitation procedures.

During the exploration phase, when the ith employed bee $\mathbf{X}_i = (X_i^1, X_i^2, \cdots, X_i^D)$ shares information with a randomly chosen companion $\mathbf{X}_k$, the new search position is calculated by the following equation:

$$X{*}_i^j = X_g^j + rand(-1, 1) \cdot (X_k^j - X_i^j) \cdot \mu(i), k, g \in \left\{1, 2, \ldots, {}^{SN}\!/\!_2\right\}, j \in \left\{1, 2, \ldots, D\right\}, k \neq i. \tag{10}$$

where the value in the jth element X_i^j is changed to $X{*}_i^j$. A novel multiplier $\mu(i)$ is introduced, defined as

$$\mu(i) = \frac{trial(i)}{trial(i) + trial(k)}. \tag{11}$$

The operation previously described in Eq. (6) is also modified here in Eq. (10) by taking another randomly chosen companion $\mathbf{X}_g$ into account.

Table 1
Parameter settings for six algorithms.

ABC	BE-ABC	Gbest-ABC [37]	IF-ABC [13]	ICA [38]	CQ-PSO [10]
$SN = 40$ $Limit = 10$	$SN = 40$ $Limit = 10$	$SN = 40$ $Limit = 10$ $C = 2$	$SN = 40$ $Limit = 10$ $\alpha = 0.1$	$SN = 40$ $Imper = 4$ $Colony = 36$	$SN = 40$ $c_1 = c_2 = 2.05$ $w_{\max} = 1$ $w_{\min} = 0.3$

As in the conventional ABC algorithm, $trial(i)$ of BE-ABC records the number of times an inefficient search is performed. However, instead of 0 the lower boundary of $trial(i)$ is set to 1. The upper boundary of $trial$, meanwhile, is set to D (i.e., $Limit = D$). Since $trial(i) \in \{1, 2, \ldots, D\}$, it is now feasible to have as many as $trial(i)$ (randomly chosen) elements in the vector $\mathbf{X}_i$ changed according to Eq. (10). That is, we randomly choose $trial(i)$ different integers from 1 to D, and then set each of them to j when performing Eq. (10).

In the exploitation phase, we add a multiplier $\mu(i)$ in a similar way as per the exploration phase. Specifically, the position of the ith onlooker bee $\mathbf{Y}_i = \left(X_j^1, \ldots, X_j^{k-1}, Y_i^k, X_j^{k+1}, \ldots, X_j^D\right)$ is calculated by the following equation when it chooses the jth employed bee $\mathbf{X}_j$ to follow:

$$Y_i^k = X_j^k + rand(-1, 1) \cdot (X_m^k - X_j^k) \cdot \mu(j), m \in \left\{1, 2, \ldots, {}^{SN}\!/_2\right\}, k \in \left\{1, 2, \ldots, D\right\}, m \neq j. \tag{12}$$

where m and k are randomly selected integers, and $\mu(j)$ is defined as follows:

$$\mu(j) = \frac{trial(j)}{trial(j) + trial(m)}. \tag{13}$$

During the re-initialization phase at the end of each iteration, any $trial(i)$ that has exceeded $Limit = D$ is set to D (rather than 0). Before proceeding to the next iteration, the average value of $trial$ (i.e., $\frac{2}{SN}\sum_{i=1}^{SN/2} trial(i)$) is compared to $\alpha_{odr} \cdot D$, where $\alpha_{odr} \in (0, 1)$ is a user-specified scalar. If $\alpha_{odr} \cdot D$ is smaller than $\frac{2}{SN}\sum_{i=1}^{SN/2} trial(i)$, the whole swarm is considered to be not working efficiently to a degree of α_{odr}. Then, as many as $round\left(\alpha_{odr} \cdot {}^{SN}\!/_2\right)$ (randomly selected) employed bees will be re-initialized according to Eq. (5). At the same time, their corresponding $trial$ indices should be reset to 1. If $\frac{2}{SN}\sum_{i=1}^{SN/2} trial(i)$ is smaller, the current iteration is terminated directly and a new iteration will begin. In this enhanced re-initialization procedure, we do not limit it to just a single scout bee at each iteration. Instead, we accumulate the necessary convergence information and make the required change at once. We call this the overall degradation strategy.

In this work, BE-ABC algorithm is applied to the LI-based image retrieval optimization scheme, aiming to find the optimal matching locations more efficiently than some state-of-the-art intelligent algorithms. The flow chart of whole LI-based image retrieval process is given in Fig. 3, where MCN represents the user-specific maximum cycle number.

Given that the candidate solutions in our concerned minimization problem consist of vertical and horizontal locations that are discretized, rather than continuous, extra efforts are needed to discretize (round) the continuous solutions.

5. Simulations

In this section, simulations are carried out on two cases to evaluate the efficiency of BE-ABC algorithm when searching for best-matching solutions. Competitive optimization algorithms are adopted as well. For validation, each single kind of experiment repeats itself for 500 times with different random initializations. Some user-specified parameters of the simulated intelligent optimization algorithms are listed in Table 1, where SN denotes the swarm population.

All simulations involved in this section were implemented in MATLAB R2010a and executed on an Intel Core 2 Due CPU with 2 GB RAM running at 2.53 GHz in the year of 2014. Reference images originate from the Google Earth® (www.google.com/earth/). In detail, $M = N = 2$, $g_0 = 105$ and the matrix $\mathbf{F}$ that contains lateral inhibition coefficients α_{ij} is set as [10,35]:

$$\mathbf{F} = \begin{bmatrix} -0.025 & -0.025 & -0.025 & -0.025 & -0.025 \\ -0.025 & -0.075 & -0.075 & -0.075 & -0.025 \\ -0.025 & -0.075 & 1 & -0.075 & -0.025 \\ -0.025 & -0.075 & -0.075 & -0.075 & -0.025 \\ -0.025 & -0.025 & -0.025 & -0.025 & -0.025 \end{bmatrix}.$$

Figs. 4 and 5 respectively illustrate the matching results in the two cases. The performances of these involved intelligent algorithms are shown in Figs. 6 and 7 respectively. The surface plots for the objective functions in the two cases are illustrated in Figs. 8 and 9. For the convenience of evaluation, three indexes (i.e., the mean, the standard deviation S.D. and the convergence rate C.R.) are calculated, as listed in Table 2.

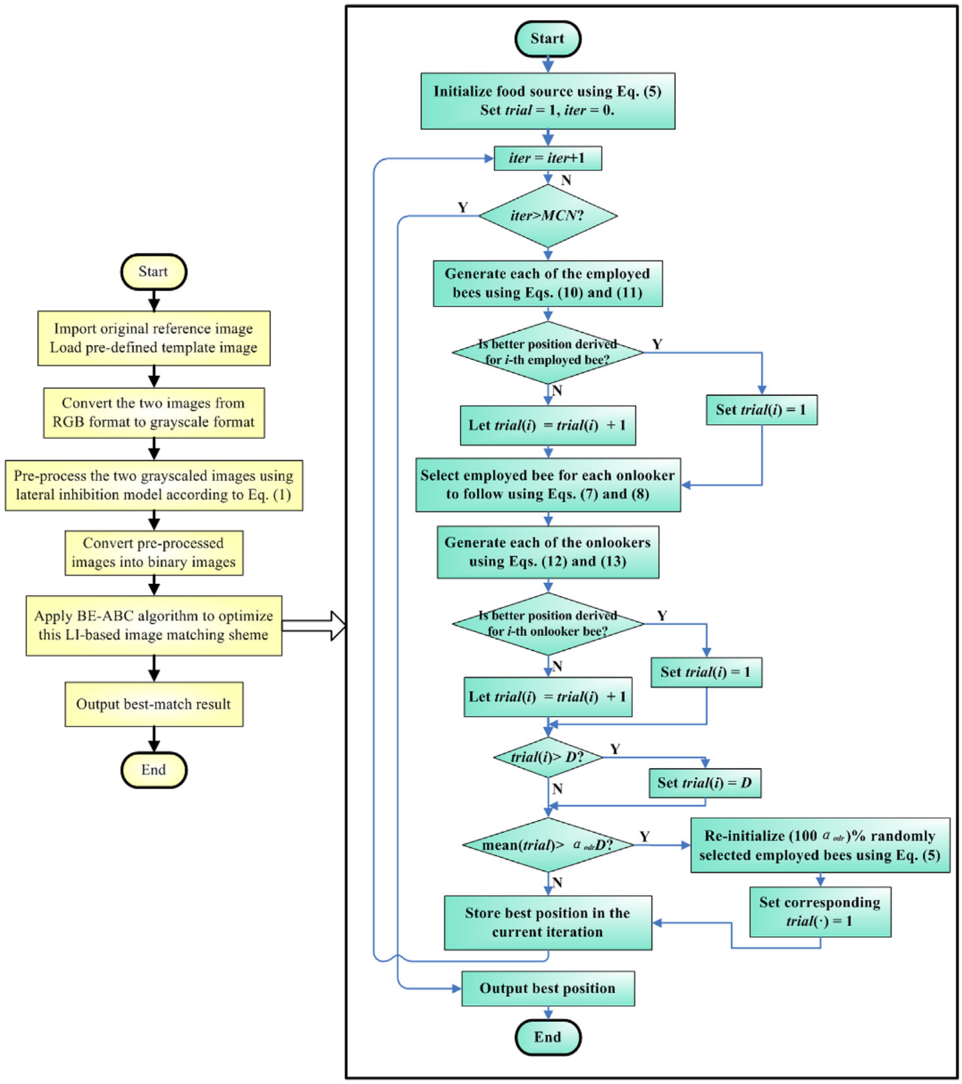

Fig. 3. Flow chart of overall LI-based image retrieval method.

Table 2
Comparative optimization statistics of different intelligent algorithms.

Case No.	*MCN*	ABC			Gbest-ABC			IF-ABC			BE-ABC			ICA			CQ-PSO		
		Mean	S. D.	C. R.	Mean	S. D.	C. R.	Mean	S. D.	C. R.	Mean	S. D.	C. R.	Mean	S. D.	C. R.	Mean	S. D.	C. R.
1	500	0.8670	0.0500	8.3%	0.8609	0.0590	12.8%	0.8802	0.0572	14.3%	**0.9516**	0.0380	**60.8%**	0.8276	**0.0206**	0.5%	0.8670	0.0746	13.9%
2	1000	0.6968	0.0636	3.5%	0.6865	0.0567	2.9%	0.7097	0.0832	6.7%	**0.7276**	0.0933	**10.3%**	0.6818	0.0426	1.0%	0.6763	**0.0323**	1.0%

As can be noticed from Figs. 6 and 7, BE-ABC algorithm contributes to converge more efficiently than the other five intelligent algorithms. In spite that its convergence speed is not the fastest among the six algorithms in some early cycles of iteration, BE-ABC algorithm is seldom involved in premature convergence, which may be the primary difference in the results between BE-ABC and ICA/CQ-PSO. In order to reveal the reasons behind the results, let us take a look at Figs. 8 and 9, i.e., the surface plots for the two cases. It is notable that the local region around the global optimum is not as convex as expected, which calls for some extra global searching abilities and efficient strategies fighting against premature convergence. When the feasible solution surface is oscillatory or rugged (i.e., anything but convex), balance-evolution strategy begins to take effect. Viewing the statistics listed in Table 2, we may find that BE-ABC algorithm is more robust.

Despite it converges faster and more stable than the other algorithms, BE-ABC algorithm can not guarantee the derivation of an optimal solution every single time. Here comes the question: what is the advantage of such a heuristic algorithm while a full search algorithm has been proposed nearly 40 years ago that always guarantees the derivation of a perfect matching result? Fig. 10 gives a direct answer: the time consumption. Regarding case 1, it takes BE-ABC algorithm as long as 9.70 s on average to find the optimal location, while it takes 13.00 s for the full search algorithm. In case 2, BE-ABC algorithm spends

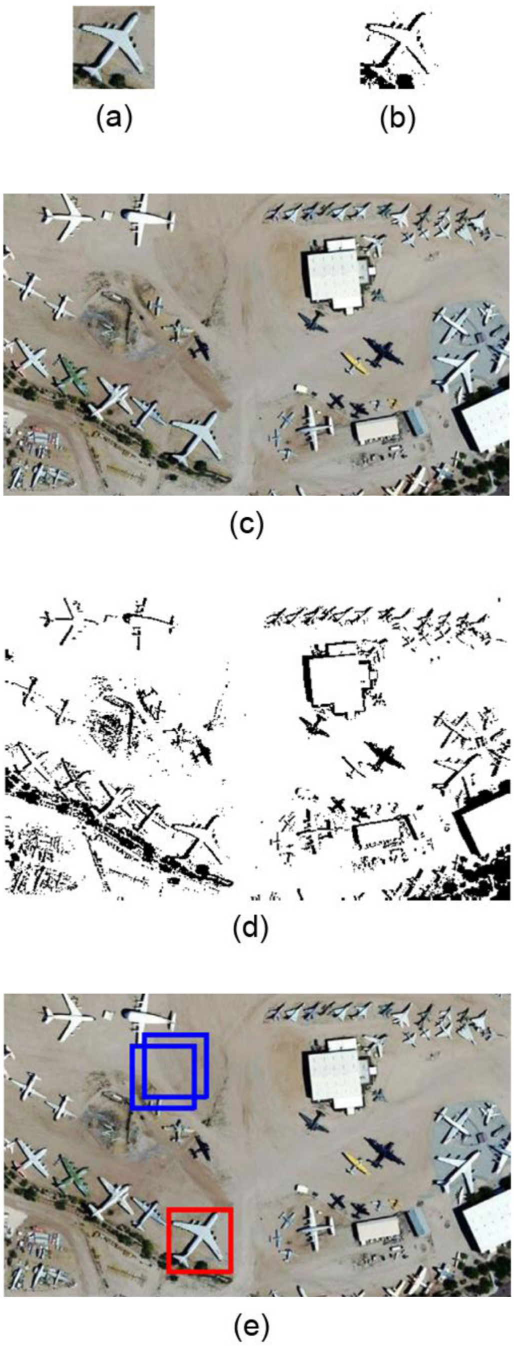

Fig. 4. Matching result of case 1. (a) original template image (65 × 66); (b) template image processed by lateral inhibition; (c) original reference image (500 × 300); (d) reference image processed by lateral inhibition; (e) matching results (red box denotes the optimal matching location, blue boxes denote most frequently derived sub-optimal ones). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

43.10 s on average, while the full search algorithm spends 194.87 s. These comparative results are derived by repeating each single experiment for 100 times so as to ensure the statistical significances. It is notable that BE-ABC algorithm helps to reduce the time by 25.3% in the first case, where the reference image size is 500 × 300. However, in case 2, where the reference image size is 820 × 820, the time-saving rate grows up to 77.9%. As the size of the reference image becomes larger, inefficiency of the full search algorithm emerges. This may be a typical reason why in many applications researchers gradually shifted their interests away from the enumeration methods towards the heuristic ones [6].

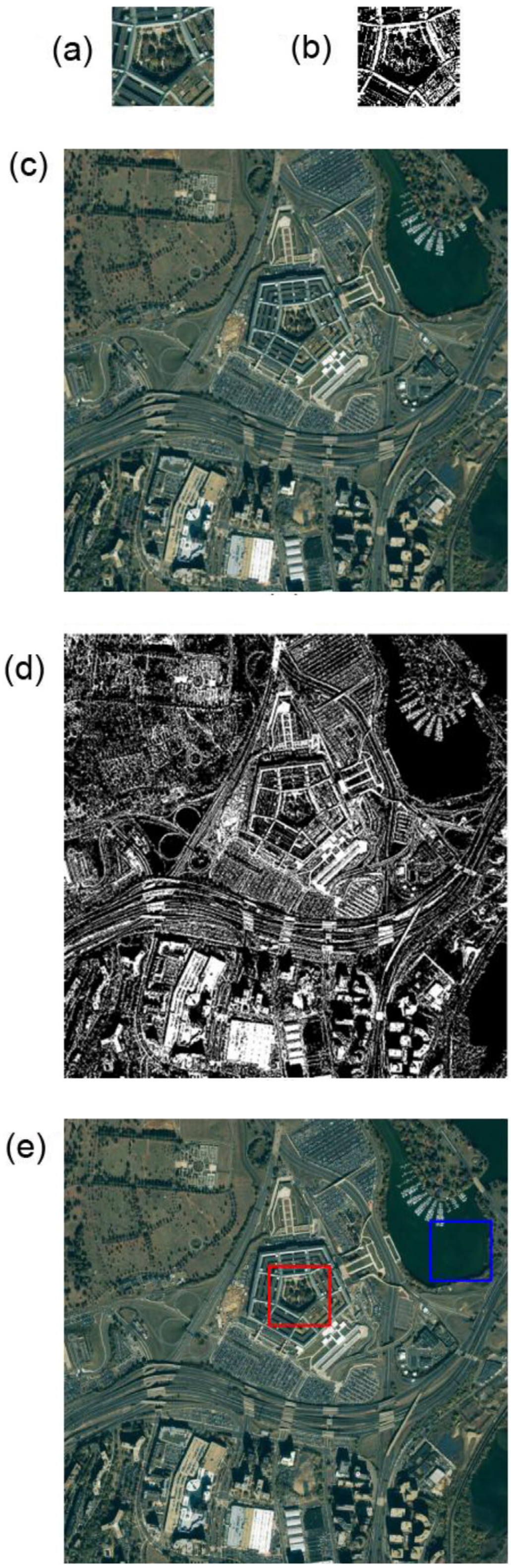

Fig. 5. Matching result of case 2. (a) original template image (116 × 114); (b) template image processed by lateral inhibition; (c) original reference image (820 × 820); (d) reference image processed by lateral inhibition; (e) matching results (red box denotes the optimal matching location, blue box denotes the most frequently derived sub-optimal one). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

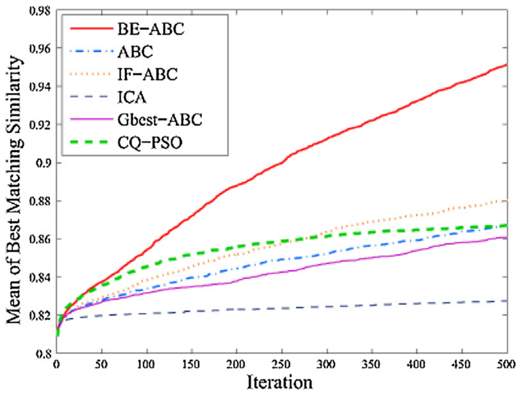

Fig. 6. Comparative convergence curves of different algorithms in case 1.

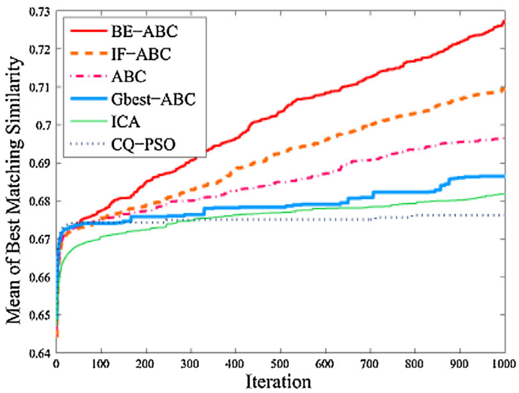

Fig. 7. Comparative convergence curves of different algorithms in case 2.

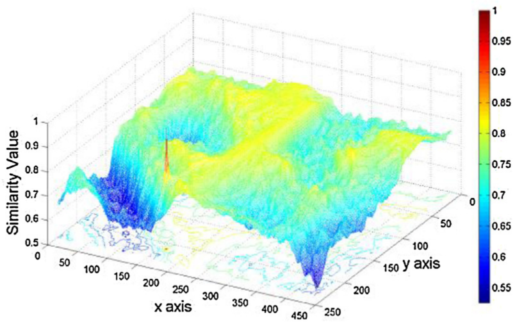

Fig. 8. Surface plot of objective similarity function in case 1.

6. Conclusions

Underlying contributions of this study lie in the following few aspects. First, we apply BE-ABC algorithm to handle the LI-based image retrieval optimization problem. Experimental results confirm the efficiency of BE-ABC algorithm in comparison with all the comparative state-of-the-art intelligent algorithms involved in this work. Second, theoretical analyses on the necessity of using LI model in image retrieval schemes are given, together with some simulations. Our future work focuses on the utilization of this proposed approach to handle more complicated image retrieval schemes in fields of aeronautics and astronautics. In addition, analyses are still needed to thoroughly reveal the advantages/disadvantages of LI model in the image retrieval scheme.

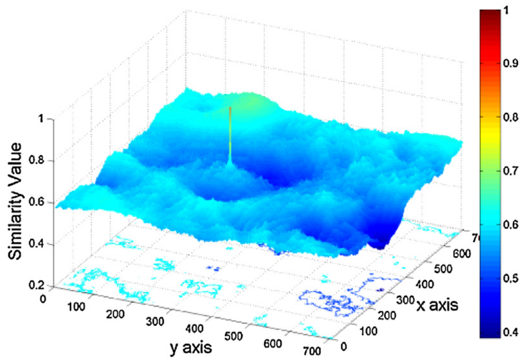

Fig. 9. Surface plot of objective similarity function in case 2.

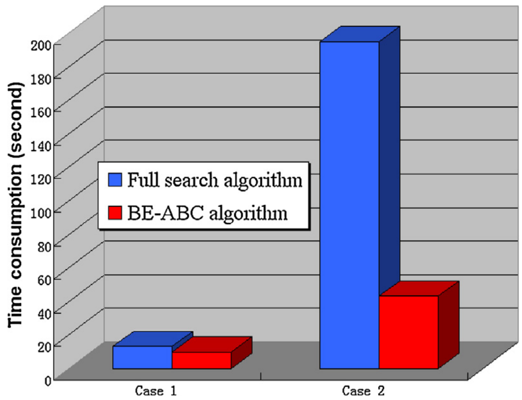

Fig. 10. Comparative average time consumptions between using BE-ABC algorithms and full searching algorithm.

Acknowledgments

Bai Li thanks Ms. Jia'en Li for the inspiration during this research study. This work is supported by the National Natural Science Foundation of China under Grant 61370005 and College Students' Science and Technology Innovation Program of Zhejiang Province (Xinmiao Talent Project) under Grant 2016R401239.

References

[1] J. Dou, J. Li, Image matching based local Delaunay triangulation and affine invariant geometric constraint, Optik 125 (2014) 526–531.
[2] E. Cuevas, A. Echavarría, D. Zaldívar, M. Pérez-Cisneros, A novel evolutionary algorithm inspired by the states of matter for template matching, Expert Syst. Appl. 40 (2013) 6359–6373.
[3] J. Luo, E.E. Konofagou, A fast normalized cross-correlation calculation method for motion estimation, IEEE Trans. Ultrason. Ferroelectr. Freq.Control 57 (6) (2010) 1347–1357.
[4] M.Z. Brown, D. Burschka, G.D. Hager, Advances in computational stereo, IEEE Trans. Pattern Anal. Mach. Intell. 25 (8) (2003) 993–1008.
[5] W. Ouyang, F. Tombari, S. Mattoccia, L.D. Stefano, W.K. Cham, Performance evaluation of full search equivalent pattern matching algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 127–143.
[6] B. Li, L. Gong, Y. Li, A novel artificial bee colony algorithm based on internal-Feedback strategy for image template matching, Sci. World J. 2014 (906861) (2014) 1–14, http://dx.doi.org/10.1155/2014/906861.
[7] A. Rosenfeld, Coarse-fine template matching, IEEE Trans. Systems, Man and Cybernet 7 (1977) 104–107.
[8] N. Dong, C. Wu, W. Ip, Z. Chen, C. Chan, K. Yung, An improved species based genetic algorithm and its application in multiple template matching for embroidered pattern inspection, Expert Syst. Appl. 38 (2011) 15172–15182.
[9] J. Zhang, G. Wang, Image matching using a bat algorithm with mutation, Appl. Mech. Mater. 203 (2012) 88–93.
[10] F. Liu, H. Duan, Y. Deng, A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching, Optik 123 (2012) 1955–1960.
[11] K.Z. Gao, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, A. Sadollah, Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion, Knowledge-Based Syst. 109 (2016) 1–16.
[12] B. Li, R. Chiong, L. Gong, Search-evasion path planning for submarines using the artificial bee colony algorithm, In 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE (2014) 528–535.
[13] Y. Shi, C.M. Pun, H. Hu, H. Gao, An improved artificial bee colony and its application, Knowledge-Based Syst. 107 (2016) 14–31.
[14] G. Li, P. Niu, Y. Ma, H. Wang, W. Zhang, Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency, Knowledge-Based Syst. 67 (2014) 278–289.
[15] B. Li, Research on WNN modeling for gold price forecasting based on improved artificial bee colony algorithm, Comput. Intell. Neurosci. 2014 (270658) (2014) 1–10, http://dx.doi.org/10.1155/2014/270658.

[16] A.M. Mohsin, G.B. Jasmon, H. Mokhlis, Optimum tie switches allocation and DG placement based on maximisation of system loadability using discrete artificial bee colony algorithm, IET Gener. Transm. Distrib. (2016).
[17] A. Hashim, B.O. Ayinde, M.A. Abido, Optimal placement of relay nodes in wireless sensor network using artificial bee colony algorithm, J. Network Comput. Appl. 64 (2016) 239–248.
[18] İ. Aydoğdu, A. Akın, M.P. Saka, Design optimization of real world steel space frames using artificial bee colony algorithm with Levy flight distribution, Adv. Eng. Software 92 (2016) 1–14.
[19] K. Roy, K.K. Mandal, A.C. Mandal, Modeling and managing of micro grid connected system using Improved Artificial Bee Colony algorithm, Int. J. Electr. Power Energy Syst. 75 (2016) 50–58.
[20] D.H. Tran, M.Y. Cheng, M.T. Cao, Hybrid multiple objective artificial bee colony with differential evolution for the time−cost−quality tradeoff problem, Knowledge-Based Syst. 74 (2015) 176–186.
[21] B. Li, Y. Yao, An edge-based optimization method for shape recognition using atomic potential function, Eng. Appl. Artif. Intell. 35 (2014) 14–25.
[22] A. Bose, K. Mali, Fuzzy-based artificial bee colony optimization for gray image segmentation, Signal Image Video Process. (2016) 1–8.
[23] S.K. Nseef, S. Abdullah, A. Turky, G. Kendall, An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems, Knowledge-Based Syst. 104 (2016) 14–23.
[24] B. Li, M. Lin, Q. Liu, Y. Li, C. Zhou, Protein folding optimization based on 3D off-lattice model via an improved artificial bee colony algorithm, J. Mol. Model. 21 (10) (2015) 1–15.
[25] F. Kang, J. Li, Z. Ma, Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions, Inf. Sci. 181 (16) (2011) 3508–3531.
[26] Q. Tang, Z. Li, L. Zhang, An effective discrete artificial bee colony algorithm with idle time reduction techniques for two-sided assembly line balancing problem of type-II, Comput. Ind. Eng. 97 (2016) 146–156.
[27] X. Yuan, P. Wang, Y. Yuan, Y. Huang, X. Zhang, A new quantum inspired chaotic artificial bee colony algorithm for optimal power flow problem, Energy Convers. Manage. 100 (2015) 1–9.
[28] C. Ozturk, E. Hancer, D. Karaboga, A novel binary artificial bee colony algorithm based on genetic operators, Inf. Sci. 297 (2015) 154–170.
[29] M.S. Kiran, H. Hakli, M. Gunduz, H. Uguz, Artificial bee colony algorithm with variable search strategy for continuous optimization, Inf. Sci. 300 (2015) 140–157.
[30] X. Zhou, Z. Wu, H. Wang, S. Rahnamayan, Gaussian bare-bones artificial bee colony algorithm, Soft Comput. 20 (3) (2016) 907–924.
[31] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inf. Sci. 192 (2012) 120–142.
[32] M.S. Kıran, O. Fındık, A directed artificial bee colony algorithm, Appl. Soft Comput. 26 (2015) 454–462.
[33] B. Li, L. Gong, W. Yang, An improved artificial bee colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning, Sci. World J. 2014 (232704) (2014) 1–10, http://dx.doi.org/10.1155/2014/232704.
[34] B. Li, An evolutionary approach for image retrieval based on lateral inhibition, Optik 127 (13) (2016) 5430–5438.
[35] B. Li, Y. Li, H. Cao, H. Salimi, Image enhancement via lateral inhibition: an analysis under illumination changes, Optik 127 (12) (2016) 5078–5083.
[36] H.B. Wang, D.L. Duan, L. Cauchy, Cauchy biogeography-based optimization based on lateral inhibition for image matching, Optik 124 (2013) 5447–5453.
[37] F. Zhong, H. Li, S. Zhong, A modified ABC algorithm based on improved-global-best-guided approach and adaptive-limit strategy for global optimization, Appl. Soft Comput. 46 (2016) 469–486.
[38] L. Huang, H. Duan, Y. Wang, Hybrid bio-inspired lateral inhibition and imperialist competitive algorithm for complicated image matching, Optik 125 (1) (2014) 414–418.