

## Research Article

# Stochastic Fractal Search Algorithm for Template Matching with Lateral Inhibition

Qifang Luo,<sup>1,2</sup> Sen Zhang,<sup>1</sup> and Yongquan Zhou<sup>1,2</sup>

<sup>1</sup>College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

<sup>2</sup>Guangxi High School Key Laboratory of Complex System and Computational Intelligence, Nanning 530006, China

Correspondence should be addressed to Qifang Luo; [l.qf@163.com](mailto:l.qf@163.com)

Received 24 November 2016; Revised 6 June 2017; Accepted 13 June 2017; Published 23 August 2017

Academic Editor: Tomàs Margalef

Copyright © 2017 Qifang Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Template matching is a basic and crucial process for image processing. In this paper, a hybrid method of stochastic fractal search (SFS) and lateral inhibition (LI) is proposed to solve complicated template matching problems. The proposed template matching technique is called LI-SFS. SFS is a new metaheuristic algorithm inspired by random fractals. Furthermore, lateral inhibition mechanism has been verified to have good effects on image edge extraction and image enhancement. In this work, lateral inhibition is employed for image preprocessing. LI-SFS takes both the advantages of SFS and lateral inhibition which leads to better performance. Our simulation results show that LI-SFS is more effective and robust for this template matching mission than other algorithms based on LI.

## 1. Introduction

Template matching is a way of getting a measure of similarity between two image sets that are superimposed on one another. It is a hot issue in the field of face recognition [1], pulmonary nodules detection [2], handwriting identification [3], and road detection [4]. Template matching techniques involve the translation of the template to every possible position in the original image and the evaluation of the match between the template and the source image at that position [5, 6].

The existing template matching techniques can be divided into two categories: the intensity-based approach and the feature-based approach [7]. The intensity-based method can be regarded as an optimization process of finding the maximum similar degree between the template and the original image. On the other hand, feature-based method matches the basis of image feature such as border, unique point, texture, entropy, and energy [8]. Compared with the feature-based method, the intensity-based method provides better performance and is widely used as it is independent of extensive feature extractions and has superior ability to restrain noise [9].

In recent years, researchers have gradually shifted their interests towards the utilization of intelligent algorithms, such as artificial bee colony (ABC) [10], internal-feedback artificial bee colony (IFABC) [11], balance-evolution artificial bee colony (BEABC) [12], states of matter search (SMS) algorithm [13], and imperialist competitive algorithm (ICA) [7] which have been proposed for this template matching problem. Although these algorithms aim to reduce the computation load of the global optimum searching, none of them can completely guarantee the derivation of suboptimal matching results. It is important to note that most intelligent algorithms are generally suitable for the optimization of convex or nearly convex functions [11, 12]. However, the template matching field contains a finite number of feasible matching locations. Such a discrete function is not smooth, but extremely ill-conditioned actually. In other words, the solution surface may drastically oscillate along the domain, which critically restricts the advantages of such intelligent algorithms. Therefore, it needs a new way to modify these existing intelligent algorithms in order to accommodate the discreteness and oscillation in the objective function.

In our paper, we utilize a recently proposed SFS, which was inspired by random fractals. SFS was firstly proposed by

Salimi [14] who used such algorithm to solve the continuous optimization problems. By using the diffusion phenomenon that reveals fractals, SFS offers a new insight to solve complex optimization problems with fast convergence rate and high search accuracy. Because of these merits, SFS is attracting more and more attention and it has been successfully used for frame structure optimization problems [15, 16], target recognition [17], precise trajectory optimization [18], distributed database queries optimization [19], and so on.

Moreover, LI approach is utilized to preprocess the original and template images before using SFS for matching. The hybrid method is named LI-SFS. Lateral inhibition mechanism is firstly discovered and verified by Hartline and his research group when they conducted an electrophysiology experiment on the limulus' vision [20]. This mechanism, which provides great help to detect object in cluttered background, can enhance the contrast of sensory information and reduce low-frequency noises [10]. Experimental results confirm that SFS is more capable than several intelligent algorithms such as IFABC [11], BEABC [12], SMS [13], and ICA [7] in this LI-based template matching scheme. The rest of the paper is structured as follows.

Section 2 discusses the principles of SFS. Section 3 describes lateral inhibition mechanism. Section 4 details the implementation procedures of the hybrid method LI-SFS. Then, several cases of comparative experiments have been conducted in Section 5. Finally, Section 6 concludes the study and advises some directions for future studies.

## 2. The Basic Principle of SFS

SFS [14] algorithm is a new metaheuristic optimization method, which is proposed by Salimi and inspired by random fractals. Two main processes to perform the SFS are the diffusion process and the update process. In the first process, each particle diffuses around to current position so as to ensure exploitation ability. With this process the chances of achieving global optimal increase and being trapped in local optimal is prevented. For SFS we consider a static diffusion process that means that only the best particle generated from diffusion process is considered, and the rest of the particles are discarded. Moreover, SFS utilizes some random update processes that results in exploration properties. To generate new particles from the diffusion procedure, a statistical method called Gaussian walk is applied, which is more promising in finding global optimal solution. A series of Gaussian walks proceeding in diffusion process is as follows:

$$GW_1 = \text{Gaussian}(\mu_{BP}, \sigma) + (\varepsilon \cdot BP - \varepsilon' \cdot P_i), \quad (1)$$

$$GW_2 = \text{Gaussian}(\mu_P, \sigma), \quad (2)$$

where  $\varepsilon$  and  $\varepsilon'$  are uniformly distributed random numbers restricted to  $[0, 1]$ ,  $BP$  is the position of the best point, and  $P_i$  is the  $i$ th point in the group.  $\mu_{BP}, \mu_P, \sigma$  are Gaussian

parameters where  $\mu_{BP}$  is exactly equal to  $BP$ ; also  $\mu_P$  is equal to  $P_i$ . The standard deviation is shown as follows:

$$\sigma = \left| \frac{\log(g)}{g} \cdot (P_i - BP) \right|. \quad (3)$$

In order to encourage a more localized search as individuals and get closer to the solution,  $(\log(g))/g$  in (3) is used for decreasing the size of Gaussian jumps with the increase of the number of generations.

Assume that our optimization problem is  $D$  dimensional problem; thus each individual particle is based on  $D$  dimensional vector. In time of initialization, each particle is initialized randomly on the basis of problem constraints. In our problem each individual particle is a vector whose length is equal to 2; namely,  $D = 2$ . Initialization equation of  $j$ th point,  $P_j$ , is addressed as follows:

$$P_j = LB + \varepsilon \cdot (UB - LB), \quad (4)$$

where  $LB$  and  $UB$  are the lower and the upper bounds of problem constrained vectors, respectively.

After initialization, the fitness function of each point is calculated to obtain the best point  $BP$ . In accordance with exploitation ability in the diffusion process, all the points roam around their current position to exploit problem search space. In addition, two statistical procedures are considered to increase the better space exploration. The first statistical method is applied to each individual vector index, while the second method performs on all points. In the first statistical procedure, at first, we sort all the points based on the fitness value of each individual. Each point  $i$  in the group is then assigned a probability through the following formula:

$$Pa_i = \frac{\text{rank}(P_i)}{N}, \quad (5)$$

where  $\text{rank}(P_i)$  is regarded as the rank of point  $P_i$  among the other points in the group and  $N$  is the number of all points in the group. According to this approach described in (5), the individual with a higher rank will have a larger probability to be selected.

Equation (5) serves two main functions. The first is used to increase the chance of changing the position of points that have not obtained a good solution. In the second generation, the good solutions will increase. For each point  $P_i$  in group, if  $Pa_i < \varepsilon$ , the  $j$ th component of  $P_i$  is updated using (6); otherwise it remains invariant.

$$P'_i(j) = P_r(j) - \varepsilon \cdot (P_t(j) - P_i(j)), \quad (6)$$

where  $P'_i$  is the new modified position of  $P_i$ ,  $P_r$  and  $P_t$  are random selected points in the group, and  $\varepsilon$  is a random number in  $[0, 1]$ .

The first update statistical procedure is carried out on the components of the points, while the second statistical update is applied in which the position of a point is changed considering the position of other points in the group. Through this property, the quality of exploration is improved and diversification property is satisfied. Before performing the

```

Initialize a population of  $N$  points
While  $g < \text{maximum generation}$  Do
  For each Point  $P_i$  in the system Do
    Call Diffusion Process with the following process:
       $q = (\text{maximum considered number of diffusion}).$ 
      For  $j = 1$  to  $q$  Do
        If (user uses Gaussian walk to solve the problem)
          Create a new point based on Eq. (1).
        Else (user uses another Gaussian Walks to solve the problem)
          Create a new point based on Eq. (2).
    Call Updating Process with the following process
    First Updating Process:
    First, all points are ranked based on Eq. (5).
    For each Point  $P_i$  in the system Do
      For each component  $j$  in  $P_i$  Do
        If ( $\text{rand}[0, 1] \geq Pa_i$ )
          Update the component  $j$  in  $P_i$  based on Eq. (6).
        Else
          Do nothing.
      Second Updating Process:
      Once again, all points obtained by the first update Process are ranked based on Eq. (5).
      For each new point  $P'_i$  in the system Do
        If ( $\text{rand}[0, 1] \geq Pa'_i$ )
          Update the position based on Eqs. (7).
        Else
          Do nothing.

```

ALGORITHM 1: SFS pseudo algorithm.

second procedure, all points attained from the first update process are ranked according to (5). In the same manner as the first statistical procedure, if  $Pa_i < \varepsilon$  for a new point  $P'_i$ , the present position of  $P'_i$  is changed according to (7) and (8) as follows; otherwise there is no update.

$$\begin{aligned}
 P''_i &= P'_i - \hat{\varepsilon} \cdot (P'_i - BP) \mid \varepsilon' \leq 0.5, \\
 P''_i &= P'_i + \hat{\varepsilon} \cdot (P'_i - P'_r) \mid \varepsilon' > 0.5,
 \end{aligned} \tag{7}$$

where  $P'_i$  and  $P'_r$  are random points selected from the first procedure and  $\hat{\varepsilon}$  are generated by the Gaussian normal distribution. The new point  $P''_i$  is replaced by  $P'_i$  if its fitness function value is better than  $P'_i$ . A more detailed description of SFS method can be found in [14].

The standard SFS can be shown in Algorithm 1.

### 3. Lateral Inhibition Mechanism

Hartline et al. found that every microphthalmia of limulus' ommateum is regarded as a receptor that is restricted by its adjacent receptors, and the inhibited effect is mutual and spatially summed [9]. The nearer the receptors are close to each other, the stronger the inhibited effect is.

In retinal images, intensively excited receptors in illuminatingly light area inhibit those in dark area more strongly than the latter to the former [21]. Therefore, lateral inhibition effectively increases the contrast and the distortion of sensory

information. By this means, the important information of the image, namely, the edge of image including the intensity gradient in retinal image and the important characters of vision scene, is strongly strengthened. In this paper, we apply the mechanism to preprocess the template and the original images in order to improve the spatial resolution and increase the efficiency and accuracy of the template matching.

The classical lateral inhibition model is shown as follows:

$$r_p = e_p + \sum_{j=1}^n k_{p,j} \cdot (r_j - r_{p,j}), \quad p = 1, 2, \dots, n; \quad j \neq p. \tag{8}$$

In order to apply this mechanism to image processing, the model is modified into a two-dimensional and gray form. Gray value of the pixel  $(m, n)$  in image is expressed with the following equation:

$$R(m, n) = I_0(m, n) + \sum_{i=-M}^M \sum_{j=-N}^N \alpha_{i,j} \cdot I_0(m+i, n+j), \tag{9}$$

where  $\alpha_{i,j}$  is the lateral inhibition coefficient of the pixel  $(i, j)$  to the central pixel,  $I_0(m, n)$  represents the original gray value of pixel  $(m, n)$ , and  $R(m, n)$  is the gray value of pixel  $(m, n)$  after being processed by lateral inhibition. A schematic instance of such a receptive field under the condition that  $M = N = 2$  is shown in Figure 1.

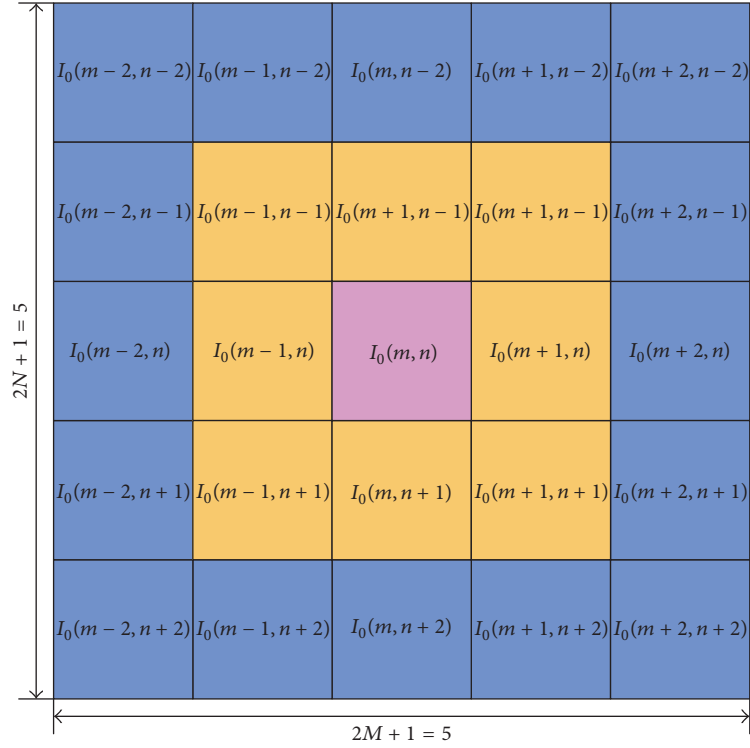


FIGURE 1: Schematic diagram of lateral inhibition model under the condition that  $M = N = 2$ .

In this paper, for the convenience of analysis, it is set to  $M = N = 2$ . Recalling (9) yields

$$R(m, n) = I_0(m, n) + \left[ \sum_{[i,j] \in S_2} \alpha_2 \cdot I_0(m, n) \right] + \left[ \sum_{[i,j] \in S_1} \alpha_1 \cdot I_0(m, n) \right] + \alpha_0 \cdot I_0(m, n), \quad (10)$$

where  $S_1$  and  $S_2$ , respectively, denote location sets of the surrounding 16 and 8 pixels as shown in Figure 1. Because there is no expectation whether  $I_0(m, n) > I_0(i, j)$  or not when  $[i, j] \in S_1 \cup S_2$ , it is unbiasedly expected that

$$E(R(m, n)) = E(I_0(m, n)) \quad (11)$$

which implies balanced inhibition energy. So,

$$E \left\{ \left[ \sum_{[i,j] \in S_2} \alpha_2 \cdot I_0(m, n) \right] + \left[ \sum_{[i,j] \in S_1} \alpha_1 \cdot I_0(m, n) \right] + \alpha_0 \cdot I_0(m, n) \right\} = (16\alpha_2 + 8\alpha_1 + \alpha_0) \cdot \text{const} \equiv 0. \quad (12)$$

Therefore, in order to ensure an unbiased preprocessing result, a natural constraint on lateral inhibition coefficients is satisfied as follows.

$$\alpha_0 + 8\alpha_1 + 16\alpha_2 \equiv 0. \quad (13)$$

Some parameters of LI model are set following the similar works reported in [7, 9, 10]. In detail,  $M = N = 2$  and the matrix  $U$  that contains lateral inhibition coefficients  $\alpha_{i,j}$  are set as follows:

$$U = \begin{bmatrix} -0.025 & -0.025 & -0.025 & -0.025 & -0.025 \\ -0.025 & -0.075 & -0.075 & -0.075 & -0.025 \\ -0.025 & -0.075 & 1 & -0.075 & -0.025 \\ -0.025 & -0.075 & -0.075 & -0.075 & -0.025 \\ -0.025 & -0.025 & -0.025 & -0.025 & -0.025 \end{bmatrix}. \quad (14)$$

The modulus template  $U$  is combined with  $R(m, n)$  and then a new gray scale of the image is obtained. Finally, the image's edge is extracted by the following equation.

$$F(m, n) = \begin{cases} 0 & R(m, n) \leq T \\ 255 & R(m, n) > T, \end{cases} \quad (15)$$

where  $T$  is a user-defined threshold value according to practical situations and  $F(m, n)$  is the final obtained gray value of pixel  $(m, n)$ .

## 4. Hybrid SFS and Lateral Inhibition

**4.1. Template Matching Principle and the Fitness Function of LI-SFS.** Image template matching aims to concern about locating a given template image over an original image, so that they can best match each other. In this section, the template matching is converted to a numerical optimization

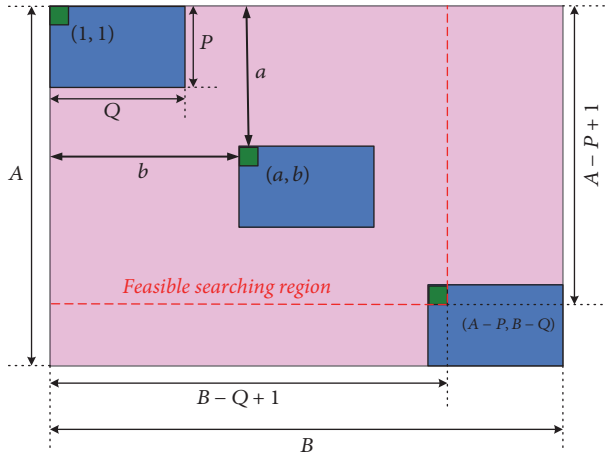


FIGURE 2: Schematic diagram of template process.

problem, during which the lateral inhibition mechanism is implemented to promote the subsequent matching process. It is generally assumed that the predefined template image and original image are given in RGB format. Figure 2 shows how a template image is overlapped onto an original image at a candidate location  $(a, b)$ . The template matching scheme can be expressed as finding an optimal location  $(a, b)$  for the gray template image so that the similarity between template and original image is maximized within the feasible searching region.

In Figure 2, the pink box denotes original image  $(A \times B)$ ; the small blue box represents template image  $(P \times Q)$ . The red dotted lines surround the searching region where all feasible solutions exist. The template image is shifted across the original image by an offset  $(a, b)$  by using the origins of the two images as reference points. The dimensions of the original image  $(A \times B)$  and the template image  $(P \times Q)$  determine the maximal feasible searching region for the comparison.

In this paper, the proposed LI-SFS is introduced to conduct the searching task. The fitness function is defined to calculate the fitness of every agent according to different situations and real practices. Template matching based on gray cross-correlation measurement has strong ability to suppress noise and the performance of simple calculation, while it is time-consuming [22]. Our work follows the custom of criterion selection as in [22, 23]; the fitness value of template matching is shown in Algorithm 2.

$fitness$  increases by 1 only when  $F(m+i, n+j) = F_t(i, j)$ , where  $fitness$  is the similar degree between the template and the pattern whose starting point is pixel  $(m, n)$  in the original image. So the maximum  $fitness$  denotes the best solution of the template matching problem. The similarity measurement can reduce the calculation time and is easy to program.  $(m+i, n+j)$  and  $(i, j)$ , respectively, represent the coordinates of the pixel in the original image and the template image.  $F(m+i, n+j)$  and  $F_t(i, j)$  are the gray values of pixel  $(m+i, n+j)$  and pixel  $(i, j)$  processed by (15). The scope of the coordinates in the original image for matching is  $1 \leq m \leq A-P+1$  and  $1 \leq n \leq B-Q+1$ .

```

Begin
  fitness = 0
  For i = 0 : P - 1
    For j = 0 : Q - 1
      If  $F(m+i, n+j) = F_t(i, j)$  then
        fitness = fitness + 1
      End If
    End For
  End For
End

```

ALGORITHM 2: Fitness value pseudocode.

Therefore, we define  $low_1 = 1$ ,  $low_2 = 1$ ,  $up_1 = A - P + 1$ ,  $up_2 = B - Q + 1$  in our experiments.

**4.2. The Procedure of LI-SFS.** The proposed LI-SFS absorbs the efficiency of SFS and the accuracy of lateral inhibition mechanism. The main procedures of the LI-SFS are given as follows.

*Step 1* (image preprocessing). We obtain the original image and the template image and transform both of them into gray scale format. Then, we filter images to suppress noises and apply lateral inhibition mechanism to preprocess these images according to (10)–(15) and save the new matrixes of images.

*Step 2* (initialize the LI-SFS parameters). We initialize the size of the population  $N$ , the dimension of the problem  $D$ , and the maximum function evolution number (MaxFEs) which is defined to terminate the process. In this paper,  $D$  is defined as two, that is, the dimension of images.

*Step 3* (diffusion process). In this step the maximum diffusion number in SFS algorithm is set to 1.

*Step 4* (Gaussian walk method). Note here that in this step we apply the first Gaussian walk to solve the problem. By using this Gaussian walk method to solve the problem, the new position of search agent is defined.

*Step 5* (updating process). In this step the position of search agents is updated. In starting stage all search agents are ranked as per their optimum position. After that, using two updating processes, the best search agent is identified. These search agents are given the optimum location of particles.

## 5. Experimental Results

In this section, we verify the performance of our proposed algorithm through a series of comparative experiments with those produced by the IFABC method [11], BEABC method [12], SMS method [13], and ICA method [7]. Simulations have been executed across seven images 1–7 that are shown in Figures 3–9, respectively. All the simulations were carried

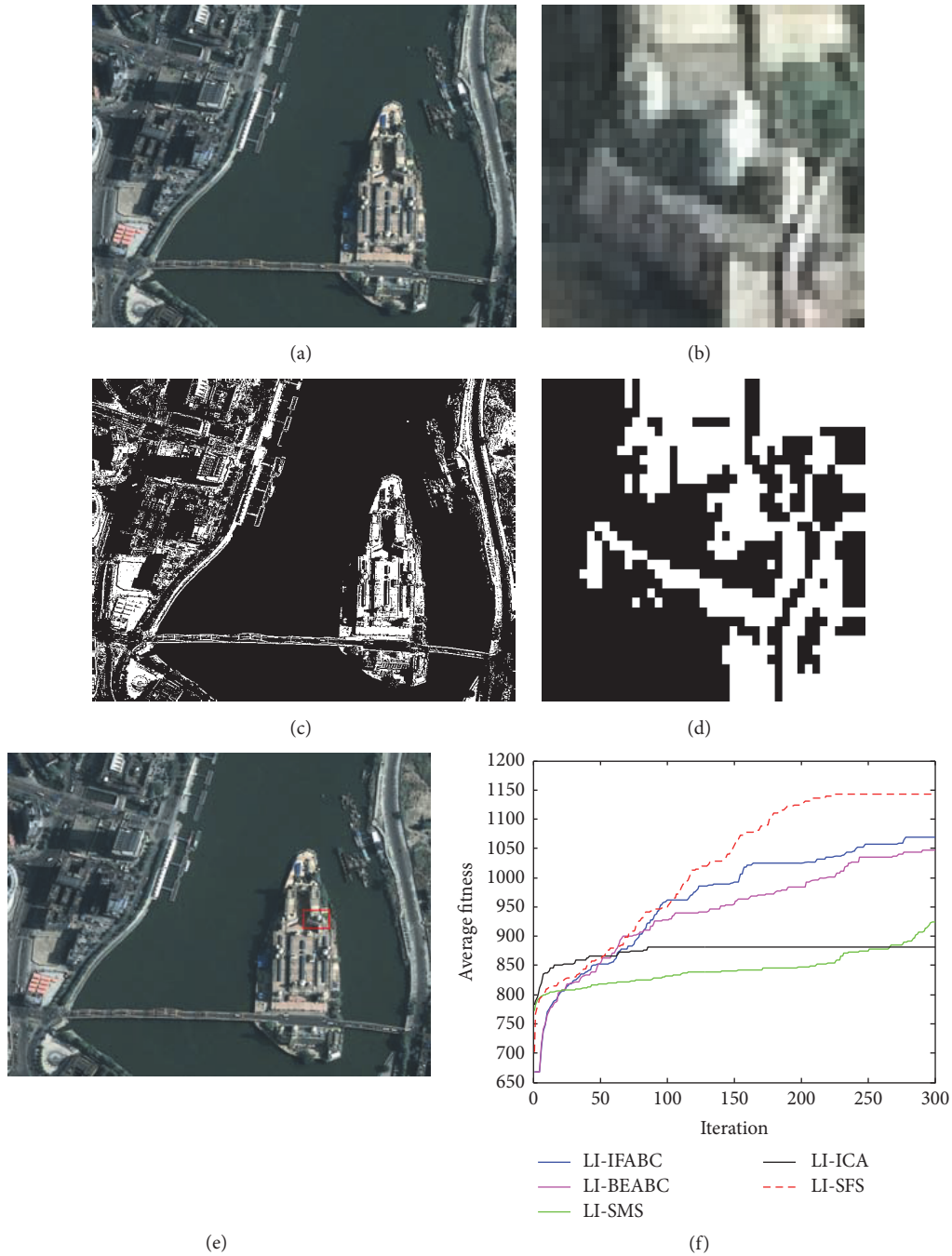


FIGURE 3: Experimental results for image 1. (a) Original image ( $684 \times 591$ ). (b) Template image ( $43 \times 34$ ). (c) Original image processed by the lateral inhibition. (d) Template image processed by the lateral inhibition. (e) The final template matching result. (f) Evolution curves of experimental results for image 1.

out in a Matlab R2012a environment and executed on a 4-core Intel Core i5-4590 CPU with 4 GB RAM running at  $4 \times 3.30$  GHz under Windows 7 operating system. In this work, all images are from Google Earth®.

In this test, the population size  $N$  is set to 100. The maximum function evolution numbers (MaxFEs) are set to 300. Such a standard of stopping has been carefully selected

to assure compatibility between similar works reported in the literature [8, 13].

Initial parameters for each algorithm that is used in the comparison are listed as follows.

(1) *LI-IFABC*. The parameters are set to  $\alpha = 0.1$  and Limit = 10.

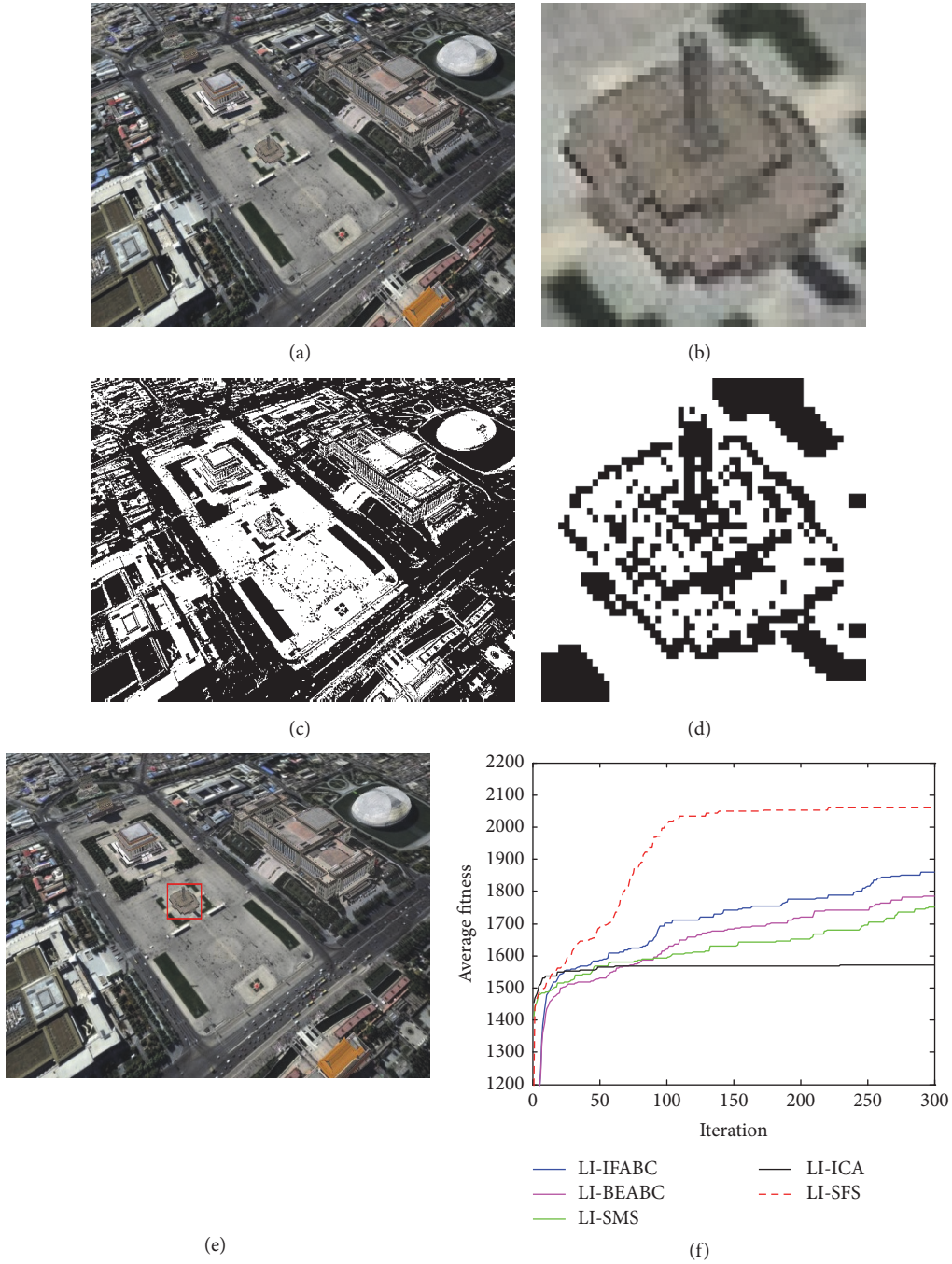


FIGURE 4: Experimental results for image 2. (a) Original image (690 × 405). (b) Template image (57 × 45). (c) Original image processed by the lateral inhibition. (d) Template image processed by the lateral inhibition. (e) The final template matching result. (f) Evolution curves of experimental results for image 2.

(2) *LI-BEABC*. The parameters are set to Limit = 10.

(3) *LI-SMS*. The parameters are set in gas  $\rho \in [0.8, 1]$ ,  $\beta = 0.8$ ,  $\alpha = 0.8$  and  $H = 0.9$ ; in liquid  $\rho \in [0.3, 0.6]$   $\beta = 0.4$ ,  $\alpha = 0.2$  and  $H = 0.2$ ; in solid  $\rho \in [0, 0.1]$ ,  $\beta = 0.1$ ,  $\alpha = 0$  and  $H = 0$ .

(4) *LI-ICA*. The parameters are set to *NumOfCountries* = 100, *NumOfImper* = 10, and *NumOfColony* = 90.

(5) *LI-SFS*. The algorithm has been configured by using maximum diffusion number MDN = 1.

The threshold for image edge extracting was set to  $T = 110$ . These experiments are designed to successfully match the template image to the original image. The coordinate of the template image fixed in the original image determines the success of matching. The experimental results are listed in Table 1. The results are averaged over 50 independent runs,

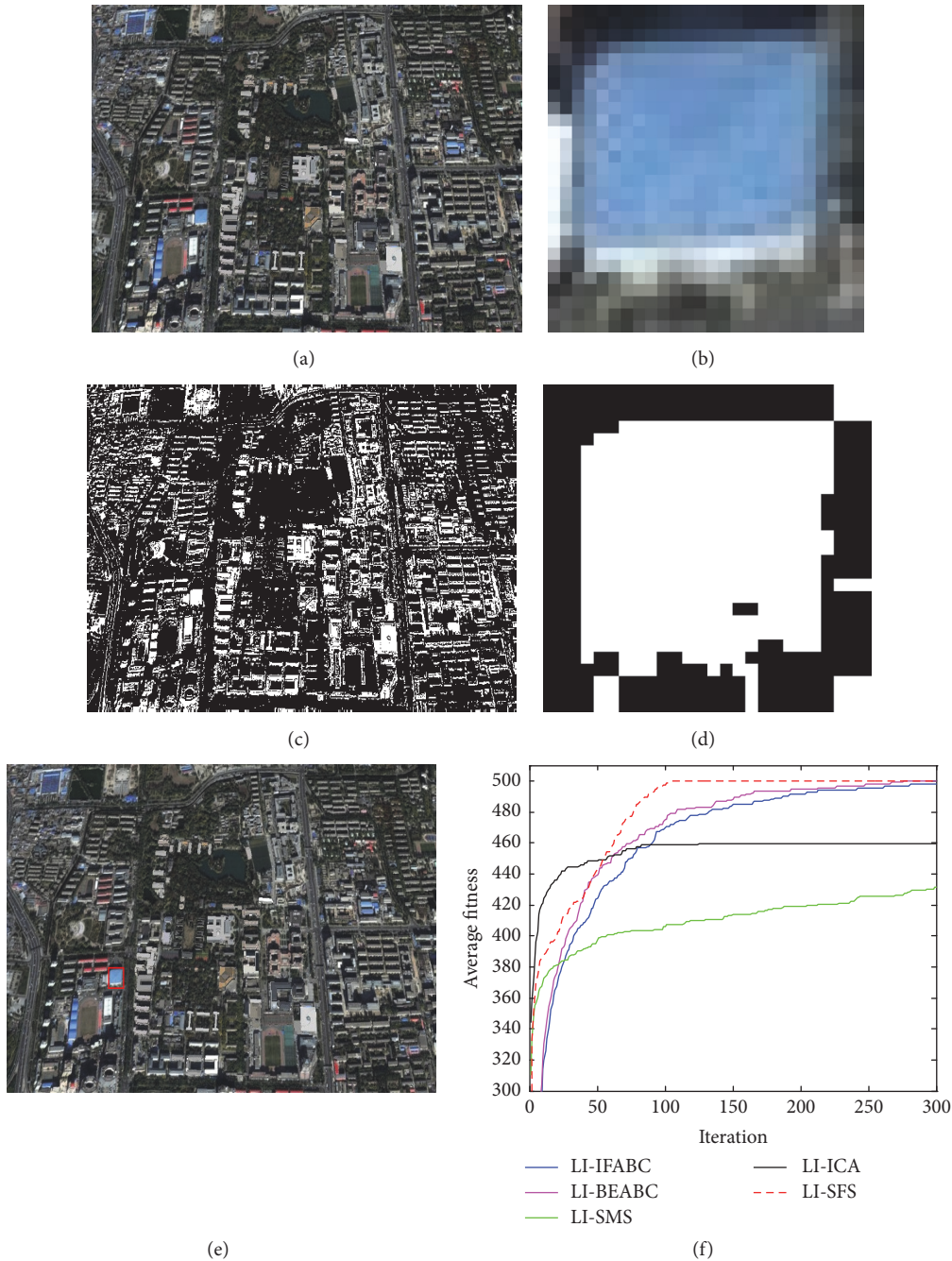


FIGURE 5: Experimental results for image 3. (a) Original image ( $687 \times 405$ ). (b) Template image ( $26 \times 27$ ). (c) Original image processed by the lateral inhibition. (d) Template image processed by the lateral inhibition. (e) The final template matching result. (f) Evolution curves of experimental results for image 3.

and the best results are denoted in bold type. The comparisons are analyzed considering four performance indexes: the average (Avg), the converged iteration, the correct rate, and the average CPU Time. The average indicates the mean fitness value. The converged iteration exhibits the times of iterations that have been visited during a single experiment. The correct rate represents the number of executions in percentage for which the algorithm successfully locates the

optimal detection point. The average CPU Time indicates the time in seconds which is employed during the execution of each single experiment. All these performance indexes are averaged considering a determined number of executions, aiming to assure statistical consistency.

To improve the performance evaluation of evolutionary algorithms, statistical tests should be conducted [24]. In order to determine whether the results of LI-SFS differ from



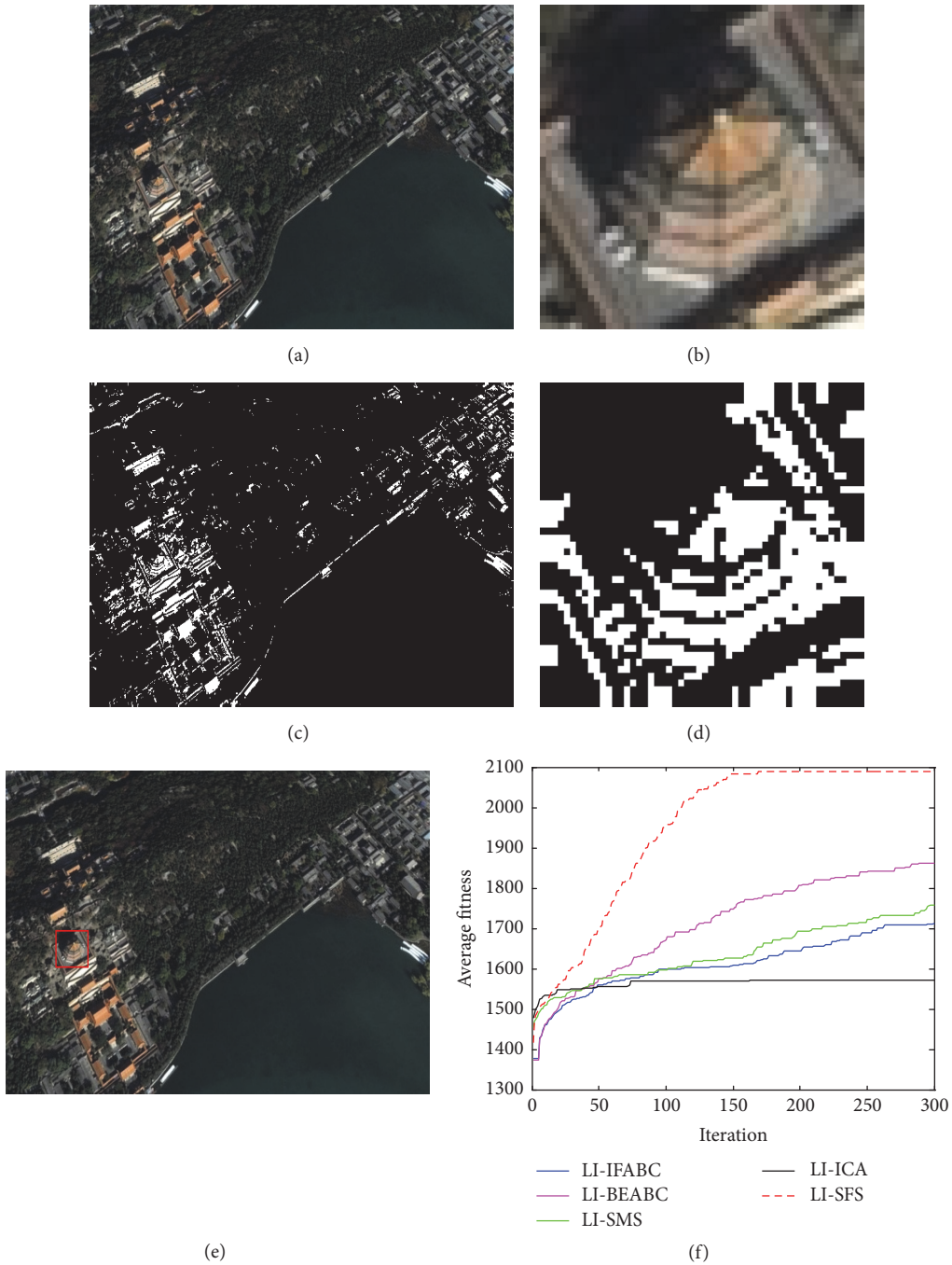


FIGURE 6: Experimental results for image 4. (a) Original image ( $690 \times 402$ ). (b) Template image ( $54 \times 47$ ). (c) Original image processed by the lateral inhibition. (d) Template image processed by the lateral inhibition. (e) The final template matching result. (f) Evolution curves of experimental results for image 4.

the best results of other algorithms in a statistical method, a nonparametric test which is known as Wilcoxon’s rank-sum test [25, 26] is performed at 5% significance level. The calculated  $p$  values in Wilcoxon’s rank-sum test comparing LI-SFS and other algorithms over all the simulation cases are given in Table 2. In Table 2, N/A indicates “not applicable” which means the statistical test cannot be done because both algorithms found the optimum successfully in all the

runs. Usually,  $p$  values  $< 0.05$  can be considered as sufficient evidence against the null hypothesis.

Table 1 shows the results for all images. As can be seen in this table, the LI-SFS outperforms the other algorithms on images 1-2 and images 4-7 in terms of the Avg. converged iteration, and correct rate. For image 3, it turns out that the converged iteration of LI-SFS is 67, while LI-IFABC is 106, LI-BEABC is 94, LI-SMS is 291, and LI-ICA is

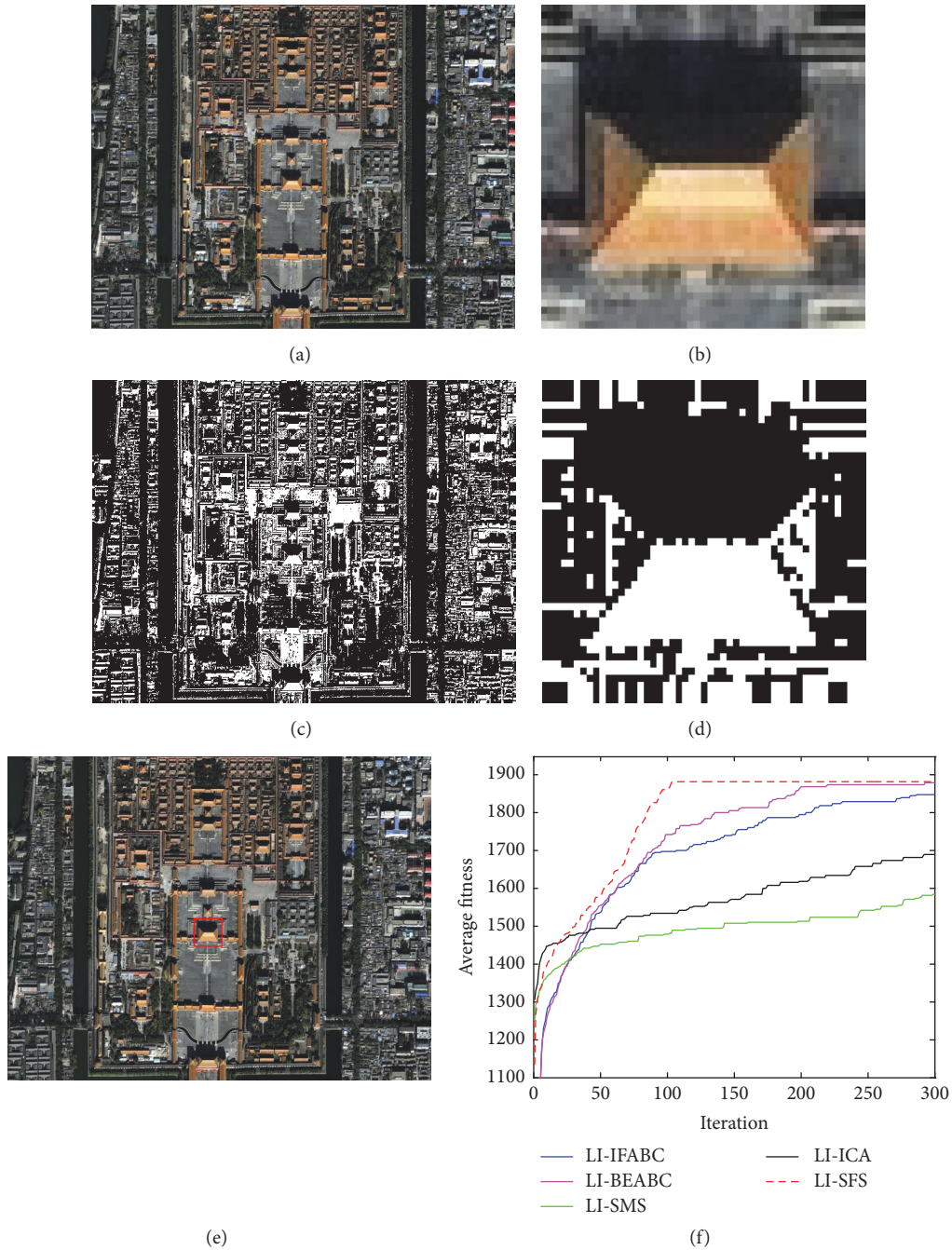


FIGURE 7: Experimental results for image 5. (a) Original image ( $749 \times 507$ ). (b) Template image ( $51 \times 45$ ). (c) Original image processed by the lateral inhibition. (d) Template image processed by the lateral inhibition. (e) The final template matching result. (f) Evolution curves of experimental results for image 5.

133, apparently showing that the proposed LI-SFS has the fastest convergence. From image 3, we see that, on Avg and correct rate, LI-BEABC and LI-SFS are well capable of providing the best values. From above analyses about images 1–7, we can conclude that the proposed LI-SFS demonstrates superiority in solving template matching optimization problem.

In addition, LI-SFS usually consumes less time than the other algorithms. This is because LI-SFS neither adds

complicated outside-world strategy nor changes the conventional algorithm framework.

According to the  $p$  values in Table 2, LI-SFS achieves significant improvement in images 1, 2, 4, and 6 compared to other algorithms. For images 3, 5, and 7, the  $p$  values show that the results of LI-SFS are significantly better in three out of four groups. Therefore, this is evidence that LI-SFS results are statistically significant and that it has not occurred by coincidence.

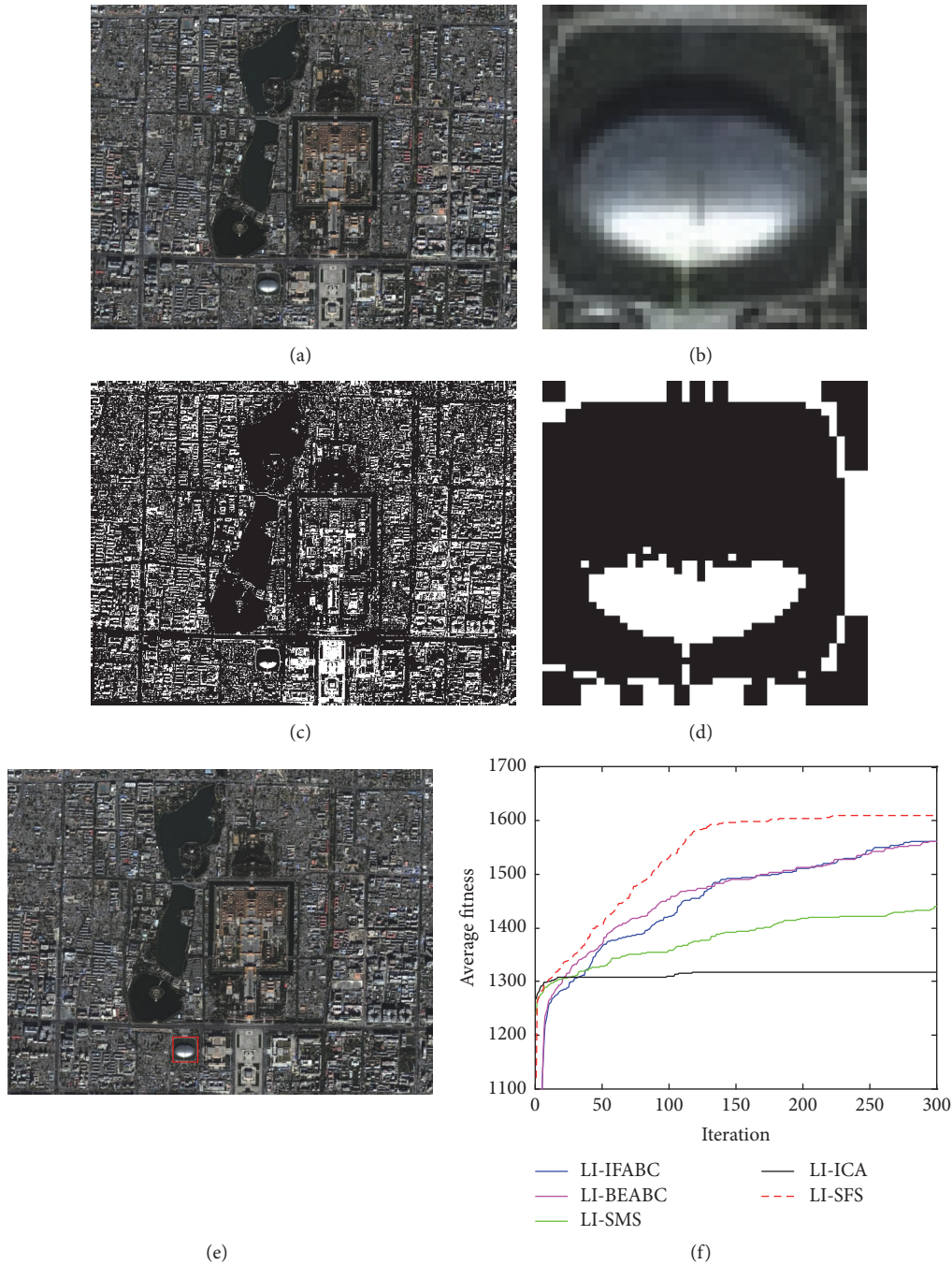


FIGURE 8: Experimental results for image 6. (a) Original image ( $684 \times 588$ ). (b) Template image ( $42 \times 47$ ). (c) Original image processed by the lateral inhibition. (d) Template image processed by the lateral inhibition. (e) The final template matching result. (f) Evolution curves of experimental results for image 6.

The results of the experiment with the original images shown in Figures 3(a)–9(a) and the template images shown in Figures 3(b)–9(b) are given. Original and template images processed by the lateral inhibition are shown in Figures 3(c)–9(c) and Figures 3(d)–9(d), respectively. It is obvious that the edges of these images are enhanced. So we can say that lateral inhibition has the ability to enhance the characters of image, which can improve the success rate of template

matching. Figures 3(e)–9(e) illustrate matching results for all images. From the matching results, it is clear that LI-SFS method can successfully find the precise location of the template fixed in the original image. Further experiments are given in Figures 3(f)–9(f) to investigate the advantage of LI-SFS.

Figures 3(f)–9(f) illustrate the averaged evolution curves of all algorithms dealing with all images over 50 independent

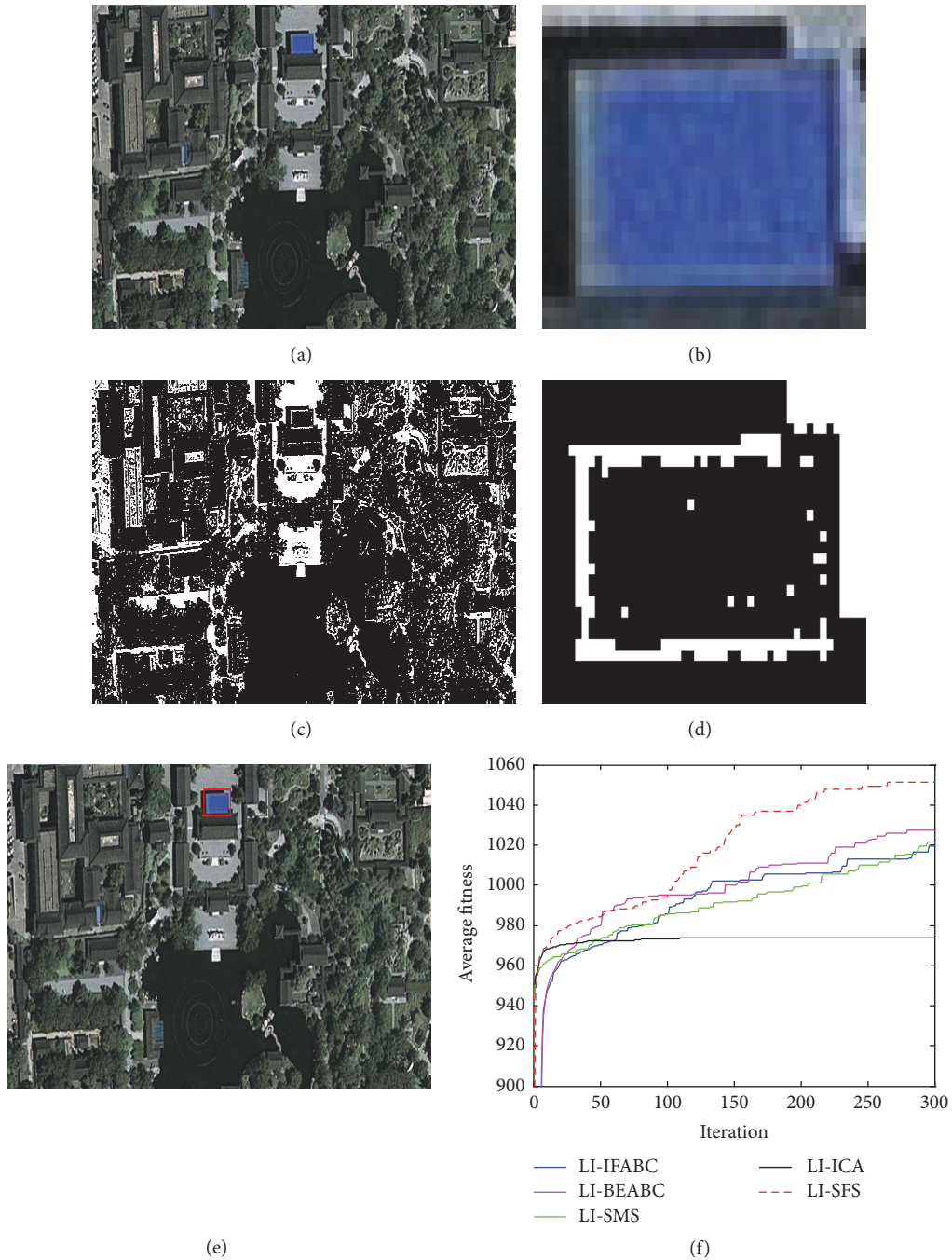


FIGURE 9: Experimental results for image 7. (a) Original image ( $749 \times 347$ ). (b) Template image ( $49 \times 30$ ). (c) Original image processed by the lateral inhibition. (d) Template image processed by the lateral inhibition. (e) The final template matching result. (f) Evolution curves of experimental results for image 7.

runs. As can be seen from Figures 4(f) and 6(f)–9(f), the LI-SFS has the fastest convergence rate. Through carefully looking at Figures 3(f) and 5(f), LI-ICA has a fast convergence initially towards the known maximum; however, it is outperformed by LI-SFS after 50 generations and 60 generations, respectively. As the procedure proceeds, LI-SFS

gets closer and closer to the maximum, when the Li-ICA becomes premature and trapped into local maximum.

According to this comprehensive study, the superior exploration and local maxima avoidance of SFS are able to solve complicated template matching problem efficiently and accurately under different environments.

TABLE 1: Comparative optimization results obtained using LI-SFS and other algorithms.

Image	Result	LI-IFABC [11]	LI-BEABC [12]	LI-SMS [13]	LI-ICA [7]	LI-SFS
Image 1	Avg	1069.88	1046.40	924.60	881.86	<b>1143.28</b>
	Converged iter	186	199	287	246	<b>137</b>
	Correct rate (%)	66	60	26	20	<b>98</b>
	CPU Time (sec)	1.07	<b>1.04</b>	2.27	2.27	2.24
Image 2	Avg	1859.12	1786.40	1749.80	1570.30	<b>2064.66</b>
	Converged iter	212	247	263	272	<b>90</b>
	Correct rate (%)	52	36	30	10	<b>94</b>
	CPU Time (sec)	1.73	<b>1.72</b>	2.97	3.25	2.88
Image 3	Avg	497.74	<b>500</b>	431.32	459.48	<b>500</b>
	Converged iter	106	94	291	133	<b>67</b>
	Correct rate (%)	96	<b>100</b>	10	60	<b>100</b>
	CPU Time (sec)	0.66	1.32	1.74	1.75	<b>0.61</b>
Image 4	Avg	1712.16	1862.40	1757.58	1570.70	<b>2089.88</b>
	Converged iter	266	231	278	284	<b>90</b>
	Correct rate (%)	26	42	22	6	<b>98</b>
	CPU Time (sec)	1.69	2.88	3.39	2.88	<b>1.58</b>
Image 5	Avg	1847.82	1879.04	1584.42	1690.24	<b>1884</b>
	Converged iter	136	111	283	201	<b>72</b>
	Correct rate (%)	88	98	20	60	<b>100</b>
	CPU Time (sec)	1.66	<b>1.59</b>	3.08	2.73	2.88
Image 6	Avg	1561.54	1561.2	1440.82	1316.42	<b>1610</b>
	Converged iter	183	187	288	285	<b>92</b>
	Correct rate (%)	72	70	12	6	<b>100</b>
	CPU Time (sec)	2.64	1.36	2.69	2.72	<b>1.34</b>
Image 7	Avg	1019.60	1027.60	1021.64	973.74	<b>1051.16</b>
	Converged iter	272	262	284	300	<b>231</b>
	Correct rate (%)	22	24	18	0	<b>44</b>
	CPU Time (sec)	0.98	1.97	2.24	2.28	<b>0.96</b>

TABLE 2:  $p$  values calculated for the Wilcoxon rank-sum test via comparing LI-SFS versus LI-IFABC, LI-BEABC, LI-SMS, and LI-ICA on seven test images ( $p \geq 0.05$ ).

LI-SFS versus	LI-IFABC	LI-BEABC	LI-SMS	LI-ICA
Image 1	5.5131E - 05	6.6687E - 06	1.5868E - 12	5.4321E - 14
Image 2	4.8973E - 06	8.9744E - 09	2.9742E - 10	1.9117E - 16
Image 3	1.5939E - 01	N/A	1.9534E - 17	7.9373E - 07
Image 4	1.2671E - 12	4.0174E - 09	2.7785E - 13	2.9243E - 18
Image 5	1.2283E - 02	3.2709E - 01	7.0990E - 15	7.8020E - 07
Image 6	6.4921E - 05	3.2246E - 05	6.7811E - 17	1.5605E - 18
Image 7	3.5032E - 02	4.6496E - 01	2.6490E - 02	7.7858E - 12

## 6. Conclusions and Future Works

Template matching is a fundamental and important problem in pattern recognition, image analysis, and computer vision. The central problem in template matching is to determine an efficient and effective approach to characterize image features and measure the degree of similarity between two images. In this work, the innovation of our approach rests with applying

SFS to dispose the LI-based template matching problem. Experimental results clearly demonstrate the efficiency of SFS in comparison with other intelligent algorithms involved in this work.

Moreover, theoretical analyses have proved that it is necessary for using LI model in template matching [12]. Such a preprocessing procedure is theoretically inefficient or misleading when the original reference image suffers from

two kinds of nonuniform illumination changes. In particular, when a reference image is distorted by Gaussian noise, LI model will make the situation even worse.

Our future work will focus on adopting SFS to deal with more complicated applications in the field of aeronautics and astronautics. Besides that, we will pay more attention to theoretical analyses in order to thoroughly reveal the advantages/disadvantages of LI model in template matching problem.

## Conflicts of Interest

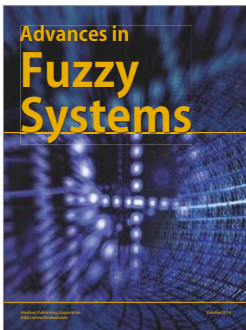
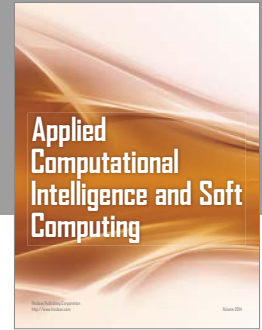
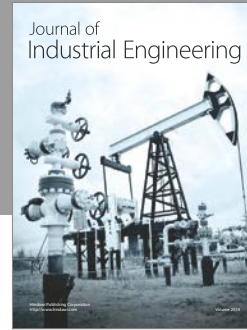
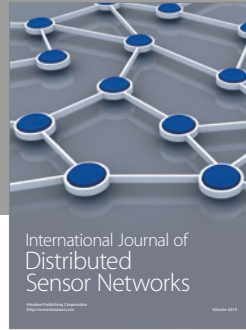
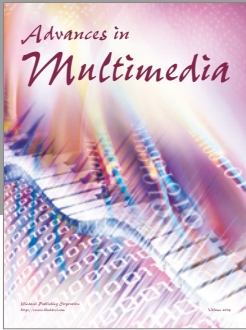
The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by Innovation Project of Guangxi Graduate Education under Grant no. gxun-chxz2016100, National Science Foundation of China under Grants nos. 61463007 and 61563008, and the Guangxi Natural Science Foundation under Grant no. 2016GXNSFAA380264.

## References

- [1] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, 1993.
- [2] Y. Lee, T. Hara, H. Fujita, S. Itoh, and T. Ishigaki, "Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique," *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 595–604, 2001.
- [3] H. Peng, F. Long, and Z. Chi, "Document image recognition based on template matching of component block projections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1188–1192, 2003.
- [4] G. Fu, H. Zhao, C. Li, and L. Shi, "Road Detection from Optical Remote Sensing Imagery Using Circular Projection Matching and Tracking Strategy," *Journal of the Indian Society of Remote Sensing*, vol. 41, no. 4, pp. 819–831, 2013.
- [5] H. Li, H.-B. Duan, and X.-Y. Zhang, "A novel image template matching based on particle filtering optimization," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1825–1832, 2010.
- [6] F. J. Veredas, H. Mesa, and L. Morente, "A hybrid learning approach to tissue recognition in wound images," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, no. 2, pp. 327–347, 2009.
- [7] L. Huang, H. Duan, and Y. Wang, "Hybrid bio-inspired lateral inhibition and Imperialist Competitive Algorithm for complicated image matching," *Optik*, vol. 125, no. 1, pp. 414–418, 2014.
- [8] H. Duan, C. Xu, S. Liu, and S. Shao, "Template matching using chaotic imperialist competitive algorithm," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1868–1875, 2010.
- [9] F. Liu, H. Duan, and Y. Deng, "A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching," *Optik*, vol. 123, no. 21, pp. 1955–1960, 2012.
- [10] H. Duan, Y. Deng, X. Wang, and C. Xu, "Small and Dim Target Detection via Lateral Inhibition Filtering and Artificial Bee Colony Based Selective Visual Attention," *PLoS ONE*, vol. 8, no. 8, Article ID e72035, 2013.
- [11] B. Li, L.-G. Gong, and Y. Li, "A novel artificial bee colony algorithm based on internal-feedback strategy for image template matching," *Scientific World Journal*, vol. 2014, Article ID 906861, 14 pages, 2014.
- [12] B. Li, "An evolutionary approach for image retrieval based on lateral inhibition," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 13, pp. 5430–5438, 2016.
- [13] E. Cuevas, A. Echavarría, D. Zaldivar, and M. Pérez-Cisneros, "A novel evolutionary algorithm inspired by the states of matter for template matching," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6359–6373, 2013.
- [14] H. Salimi, "Stochastic Fractal Search: a powerful metaheuristic algorithm," *Knowledge-Based Systems*, vol. 75, pp. 1–18, 2015.
- [15] G. G. Tejani, V. H. Bhensdadia, and S. Bureerat, "Examination of three meta-heuristic algorithms for optimal design of planar steel frames," *Advances in Computational Design*, vol. 1, no. 1, pp. 79–86, 2016.
- [16] V. H. Bhensdadia, *Investigation of size optimization of frame structure using grey wolf optimizer (GWO) and stochastic fractal search (SFS) algorithms dissertation, Department of mechanical engineering school of engineering [Doctoral, thesis]*, RK University, Department of mechanical engineering school of engineering, 2015.
- [17] B. Li, "Atomic potential matching: An evolutionary target recognition approach based on edge features," *Optik*, vol. 127, no. 5, pp. 3162–3168, 2016.
- [18] B. Li and Z. Shao, "Precise trajectory optimization for articulated wheeled vehicles in cluttered environments," *Advances in Engineering Software*, vol. 92, pp. 40–47, 2016.
- [19] M. Sohal, A. Singh, and R. S. Virk, "A framework for optimizing distributed database queries based on stochastic fractal search," *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 6, pp. 544–551, 2015.
- [20] H. Hartline K, "The response of single optic nerve fibers of the vertebrate eye to illumination of the retina," *Journal of Physiology—Legacy*, vol. 121, no. 2, pp. 400–415, 1938.
- [21] Z. Fang, Z. Dawei, and Z. Ke, "Image pre-processing algorithm based on lateral inhibition," in *Electronic Measurement and Instruments*, pp. 701–705, IEEE, 2007.
- [22] Z. Zhang and H. Duan, "A hybrid Particle Chemical Reaction Optimization for biological image matching based on lateral inhibition," *Optik*, vol. 125, no. 19, pp. 5757–5763, 2014.
- [23] X. H. Wang, H. B. Duan, and D. L. Luo, "Cauchy biogeography-based optimization based on lateral inhibition for image matching," *Optik*, vol. 124, no. 22, pp. 5447–5453, 2013.
- [24] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [25] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*, Springer, Berlin, Germany, 5th edition, 2011.
- [26] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric statistical methods*, John Wiley & Sons, Third edition, 2014.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

