

Research Article

Binary Structuring Elements Decomposition Based on an Improved Recursive Dilation-Union Model and RSAPSO Method

Yudong Zhang,¹ Shuihua Wang,^{1,2} Yi Sun,³ Genlin Ji,¹
Preetha Phillips,⁴ and Zhengchao Dong⁵

¹ School of Computer Science and Technology, Nanjing Normal University, Nanjing, Jiangsu 210023, China

² School of Electronic Science and Engineering, Nanjing University, Nanjing, Jiangsu 210046, China

³ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

⁴ School of Natural Sciences and Mathematics, Shepherd University, Shepherdstown, WV 25443, USA

⁵ Translational Imaging Division and MRI Unit, Columbia University and New York State Psychiatric Institute, New York, NY 10032, USA

Correspondence should be addressed to Yudong Zhang; zhangyudongnuaa@gmail.com

Received 2 January 2014; Accepted 20 January 2014; Published 27 April 2014

Academic Editor: Baozhen Yao

Copyright © 2014 Yudong Zhang 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.

This paper proposed an improved approach to decompose structuring elements of an arbitrary shape. For the model of this method, we use an improved dilation-union model, adding a new termination criterion, as the sum of 3-by-3 matrix should be less than 5. Next for the algorithm of this method, we introduced in the restarted simulated annealing particle swarm optimization method. The experiments demonstrate that our method can find better results than Park's method, Anelli's method, Shih's SGA method, and Zhang's MFSGA method. Besides, our method gave the best decomposition tree of different SE shapes including "ship," "car," "heart," "umbrella," "vase," "tree," "cat," "V," "bomb," and "cup."

1. Introduction

Mathematical morphology (MM) is a theory and technique for the analysis and processing of geometrical structures based on set theory, lattice theory, topology, and random functions. MM not only is commonly applied to digital images, but also can be employed to graphs, surface meshes, solids, and many other spatial structures [1].

In these applications, the inherent strategy in MM is to explore the characteristics of an object by probing its microstructure with various forms, known as structuring element (SE). Most image processing architectures adapted to morphological operations use SEs of a limited size. However, implementation becomes difficult when a large-sized SE is employed. Hence, the techniques for decomposing a large-sized SE into combined small-sized SEs are of importance [2].

In the last decade, techniques have been proposed for the decomposition of SE, but they are intricate and

sometimes suffer from indecomposable cases [3]. Hashimoto and Barrera indicated that traditional algorithms have the disadvantage of being unable to decompose many simply connected decomposable SEs [4]. Shih and Wu developed a method for decomposing arbitrarily shaped binary SEs by standard genetic algorithm (SGA). The algorithm performed an iterative process to create new ones that minimize a predefined fitness function [2]. Zhang and Wu proposed a recursive model and used the migration fitness scaling genetic algorithm (MFSGA) for SE decomposition [5].

The abovementioned SGA and MFSGA algorithms are time-consuming, and they can be easily trapped into local optimal points, leading to a wrong solution. Particle swarm optimization (PSO) is an efficient algorithm compared to SGA [6–8]. Zhang and Wu proposed a restarted simulated annealing PSO (RSAPSO) algorithm [9] for further improvement. In their paper, they reported that (1) RSAPSO combined the global search ability of PSO and the local search

ability of restarted simulated annealing (RSA) algorithm, and (2) RSAPSO offset the weakness of both PSO and RSA. They also proved that the RSAPSO is superior to SGA, RSA, and PSO by six benchmark functions. In this paper, we proposed to introduce the recursive model and the RSAPSO algorithm for SE decomposition.

The rest of the paper is organized as follows. Section 2 described the methodology, including the concept of SE decomposition, the recursive dilation-union model, the encoding strategy, and the objective function. We also introduced the RSAPSO algorithm. Experiments in Section 3 compared the RSAPSO algorithm with Park's method, Shih's method, Anelli's SGA method, and Zhang's MFSGA method. Finally, Section 4 is devoted to discussions and conclusions.

2. Methodology

2.1. Concept of SE Decomposition. Suppose A denotes a binary image and S denotes a SE. If we decompose S into $S_1 \oplus S_2 \oplus \dots \oplus S_k$, the dilation and erosion will become as follows according to associative law:

$$\begin{aligned} A \oplus S &= A \oplus (S_1 \oplus S_2 \oplus \dots \oplus S_k) = (((A \oplus S_1) \oplus S_2) \dots \oplus S_k), \\ A \ominus S &= A \ominus (S_1 \oplus S_2 \oplus \dots \oplus S_k) = (((A \ominus S_1) \ominus S_2) \dots \ominus S_k). \end{aligned} \quad (1)$$

The computational cost (CC) for dilation/erosion operators is equal to the number of nonzero element of S . Usually the CC of S is extremely large; however, the sum of CC of S_1, S_2, \dots and S_k are relatively small. Therefore, the SE decomposition can dramatically reduce the CC.

2.1.1. Dilation Model. Dilation model is to decompose SE by only dilation operator. Suppose S is decomposed into $S_1 \oplus S_2 \oplus \dots \oplus S_k$; the serial computational cost (SCC) of dilation model is equal to the sum of CC of $S_1, S_2,$ and S_k .

Consider

$$\begin{aligned} \min \quad \text{SCC} &= \text{CC}(S_1) + \text{CC}(S_2) + \dots + \text{CC}(S_k) \\ \text{s.t.} \quad S &= S_1 \oplus S_2 \oplus \dots \oplus S_k. \end{aligned} \quad (2)$$

Figure 1 shows three examples. The first decomposes a square SE of size 3×3 into a row vector of size 1×3 and a column vector of size 3×1 . The SCC decreases from 9 to $3 + 3 = 6$. The second decomposes a square SE of size 5×5 into two row vectors of size 1×3 and two column vectors of size 3×1 , and the SCC decreases from 25 to $3 + 2 + 3 + 2 = 10$. The third example decomposes a diamond SE of diameter 7 into three small SEs of size 3×3 , and the SCC decrease from 25 to $5 + 4 + 4 = 13$.

2.1.2. Dilation-Union Model. Dilation-union model is to decompose SE by both dilation and union operators. There are two types of computational cost for dilation-union model: the SCC and the parallel computational cost (PCC).

Consider

$$\begin{aligned} \text{PCC} &= \max(\text{CC}(S_1) + \text{CC}(S_2), \text{CC}(S_3)) \\ \text{s.t.} \quad S &= S_1 \oplus S_2 \cup S_3. \end{aligned} \quad (3)$$

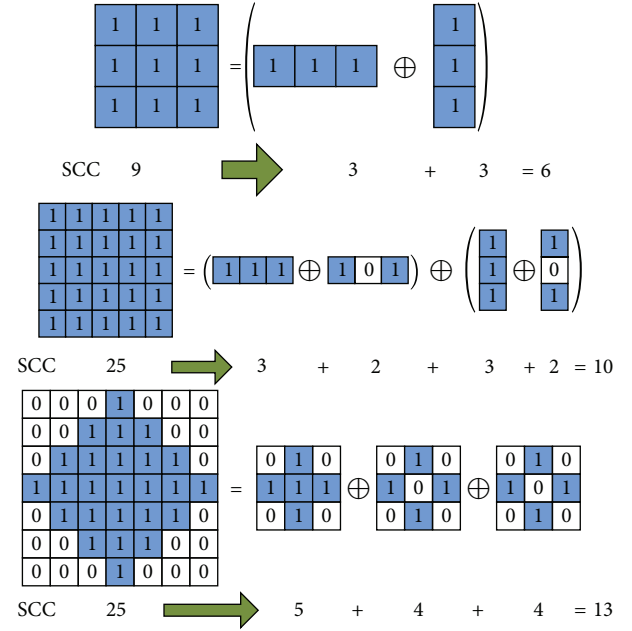


FIGURE 1: Examples of SE decomposition based on dilation model: (a) 3×3 square; (b) 5×5 square; (c) 7×7 diamond.

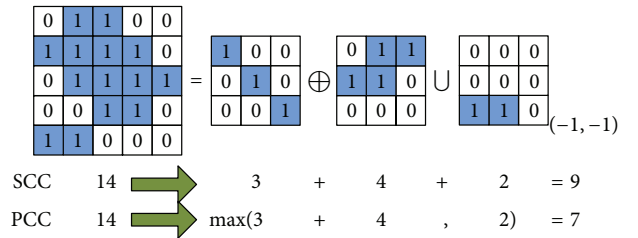


FIGURE 2: An example of SE decomposition based on dilation-union model.

Suppose S can be decomposed into the union of $S_1 \oplus S_2$ and S_3 ; then the SCC is equal to the sum of CC of $S_1, S_2,$ and S_3 ; the PCC is equal to the maximal value of $\text{CC}(S_1) + \text{CC}(S_2)$ and $\text{CC}(S_3)$. Figure 2 gives an example; here the subscript $(-1, -1)$ denotes that this square matrix should be translated -1 at both the x -axis and the y -axis. The SCC decreases from 14 to $3 + 4 + 2 = 9$. The PCC decreases from 14 to $\max(3 + 4, 2) = 7$.

2.1.3. Recursive Dilation-Union Model. Let S_{NN} denote a SE of size $N \times N$. The first level decomposition is written as

$$S_{NN} = V_{(N-2)(N-2)} \oplus F_{33} \cup R_{NN}. \quad (4)$$

Here we use the dilation-union model: V denotes the variable-size matrix, F denotes the fixed-size prime component, and R denotes the residue component. The R can be easily decomposed into union of factors of size 3×3 because the size 3×3 is the golden standard as the elementary structuring component for decomposition in the literature.

Consider

$$S_{NN} = V_{(N-2)(N-2)} \oplus F_{33}^{\text{PC}} \cup \left[\left(R_{33}^1 \right)_{(x_1, y_1)} \cup \left(R_{33}^2 \right)_{(x_2, y_2)} \cup \dots \cup \left(R_{33}^n \right)_{(x_n, y_n)} \right]. \quad (5)$$

Here n denotes the number of 3-by-3 residual matrix. The subscript (x_i, y_i) denotes that the R^i should be translated by (x_i, y_i) . Therefore, the iteration continues until the termination criteria are satisfied. In Zhang and Wu's paper [5], they defined the termination criteria (TC) as that the size of S is smaller than or equal to 3×3 :

$$\text{TC1: size}(S) \leq 3. \quad (6)$$

However, this is not perfect because the 3×3 binary SE can be decomposed further as shown in Figure 1(a). So our new criterion is set as

$$\text{TC2: size}(S) \leq 3, \quad \text{sum}(S) \leq 5. \quad (7)$$

We improved the TC by adding the rule $\text{sum}(S) \leq 5$. The reason is that any 3-by-3 SE with less than five 1s is indecomposable.

The flowchart of the recursive dilation-union model is depicted in Figure 3, which indicates that we record 4 variables ($V_k, F_k, R_k,$ and P_k) at each iteration level k , and the decomposition tree can be depicted via those variables. The following task is to develop an effective method to perform formulas (4) and (5).

2.2. Optimization Problem. The SE decomposition problem can be transformed into an optimization problem with the help of dilation-union model. In what follows we will briefly discuss the encoding strategy and the objective function of the optimization problem.

2.2.1. Encoding Strategy. Formula (4) or formula (5) can be regarded as an optimization problem. We transform the prime component F_k into a row vector string of chromosomes. For any SE at any iteration, the F_k is a 3-by-3 two-value image; therefore, the chromosome can be written as

$$\xi = \text{vector}(F) = f_1 f_2 \dots f_9. \quad (8)$$

Here ξ denotes the chromosome and f_i denotes the locus. Figure 4(a) illustrates the numbering scheme for ξ , namely, from left to right and then from top to bottom. Two examples are shown in Figures 4(b)-4(c), and their 1D strings of chromosomes are "001001101" and "101110101," respectively.

2.2.2. Objective Function. Since the prime component F is encoded, variable matrix of SEV and residual matrix R can be obtained through the following formula:

$$\begin{aligned} V &= S \odot F, \\ R &= S - V \oplus F. \end{aligned} \quad (9)$$

Then, we can extract two different types of costs, SCC and PCC, as follows:

$$\begin{aligned} \text{SCC} &= \sum V + \sum F + \sum R, \\ \text{PCC} &= \max \left(\sum V + \sum F, \sum R \right). \end{aligned} \quad (10)$$

Note that those variables satisfy the equality as $S = V \oplus F \cup R$. In this paper we choose SCC as the objective function, since the PCC needs pipelined architecture, which is difficult to implement in practice.

2.3. RSAPSO Algorithm. Now that the encoding strategy and objective function are established, we employed the RSAPSO algorithm [9] to search the optimal points. PSO is a method that optimizes a problem by iteratively improving a candidate solution with regard to a given measure of quality [10]. It is commonly known as metaheuristic method as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, PSO does not use the gradient of the problem being optimized, which means that PSO does not require the optimization problem to be differentiable as is required by classic optimization methods such as gradient descent and quasi-Newton methods [11]. PSO can therefore also be used in optimization problems that are partially irregular, noisy, adaptive, and so forth [12].

Simulated annealing (SA) was chosen as the local search method. SA comes from annealing in metallurgy [13], a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects [14]. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one [15]. We introduced the restarted simulated annealing (RSA) technique to improve the performance of SA.

The proposed algorithm RSAPSO combines both the exploitation ability from RSA and the exploration ability from PSO. It divides the population into two halves: one half runs PSO and the other half runs RSA. In what follows, we will describe those three algorithms in depth.

2.3.1. Particle Swarm Optimization. PSO is a population based stochastic optimization technique, which simulates the social behavior of a swarm of bees, a flock of birds, or a school of fish. By randomly initializing the algorithm with candidate solutions, the PSO successfully leads to a global optimum [16]. This is achieved by an iterative procedure based on the processes of movement and intelligence in an evolutionary system [17]. Figure 5(a) shows the flow chart of a PSO algorithm.

In PSO, each potential solution is represented as a particle. Two properties (position x and velocity v) are associated

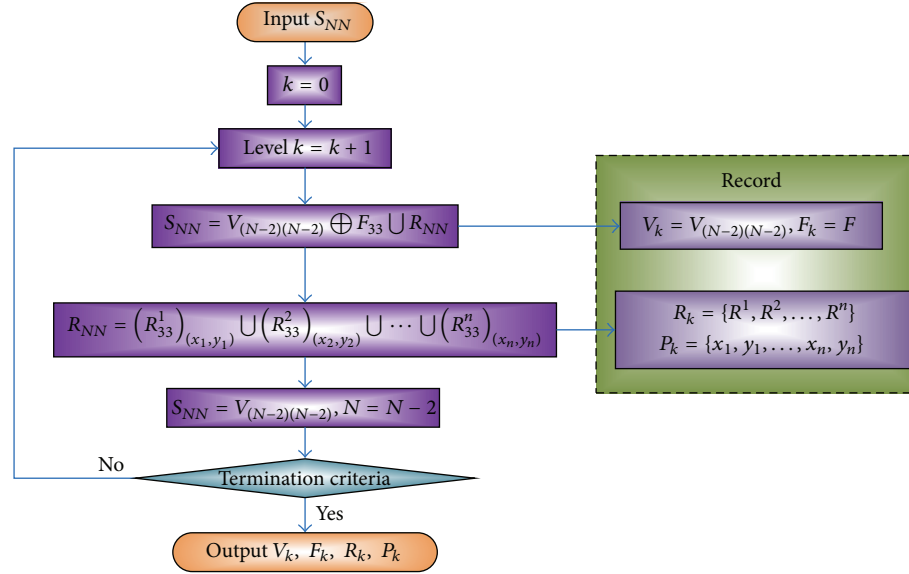


FIGURE 3: Flowchart of recursive dilation-union model.

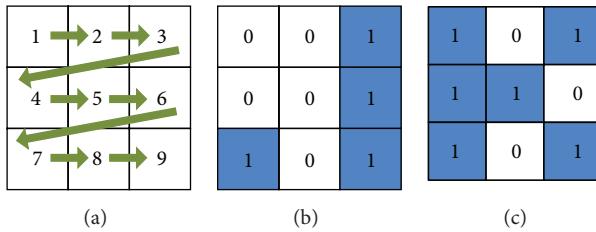


FIGURE 4: (a) Encoding order: left to right and then top to bottom; (b) an example of "001001101"; (c) an example of "101110101".

with each particle. Suppose that x and v of the i th particle are given as

$$\begin{aligned} x &= (x_{i1}, x_{i2}, \dots, x_{iN}), \\ v &= (v_{i1}, v_{i2}, \dots, v_{iN}), \end{aligned} \quad (11)$$

where N stands for the dimensions of the problem. In each iteration, a fitness function is evaluated for all the particles in the swarm [18]. The velocity of each particle is updated by keeping track of the two best positions: one is the best position a particle has traversed so far and is called " p Best" and the other is the best position that any neighbor of a particle has traversed so far. It is a neighborhood best called " n Best." When a particle takes the whole population as its neighborhood, the neighborhood best becomes the global best and is accordingly called " g Best." Hence, a particle's velocity and position are updated as follows:

$$\begin{aligned} v &= \omega \cdot v + c_1 r_1 (p\text{Best} - x) + c_2 r_2 (n\text{Best} - x), \\ x &= x + v \Delta t, \end{aligned} \quad (12)$$

where ω is called the "*inertia weight*" that controls the impact of the previous velocity of the particle on its current one.

The parameters c_1 and c_2 are positive constants, called "*acceleration coefficients*." The parameters r_1 and r_2 are random numbers that are uniformly distributed in the interval $[0, 1]$. These random numbers are updated every time when they occur. The parameter Δt stands for the given time step. The population of particles is then moved according to (12) and tends to cluster together from different directions. However, a maximum velocity v_{\max} should not be exceeded by any particle to keep the search within a meaningful solution space [19]. The PSO algorithm runs through these processes iteratively until the termination criterion is satisfied [20–22].

2.3.2. Restarted Simulated Annealing. SA algorithm is a probabilistic hill-climbing technique that is based on the annealing/cooling process of metals [23]. This annealing process occurs after the heat source is removed from a molten metal and its temperature starts to decrease. As the temperature decreases, the energy of the metal molecules reduces, and the metal becomes more rigid. The procedure continues until the metal temperature has reached the surrounding ambient temperature, at which stage the energy has reached its lowest value and the metal is perfectly solid [24].

The SA procedure begins by generating an initial solution at random. At initial stages, a small random change is made in the current solution X_c . The new solution is called X_n . The perturbation depends on a temperate parameter T and a scaling constant k .

Consider

$$\text{pert}(T) = k \times T \times r_3. \quad (13)$$

Here r_3 is a random value between 0 and 1 with uniform distribution. The temperature T decreases with each iteration of the algorithm, thus reducing the size of the perturbations as the search progresses. This mechanism produces large perturbation in the initial stages of the search and ensures that

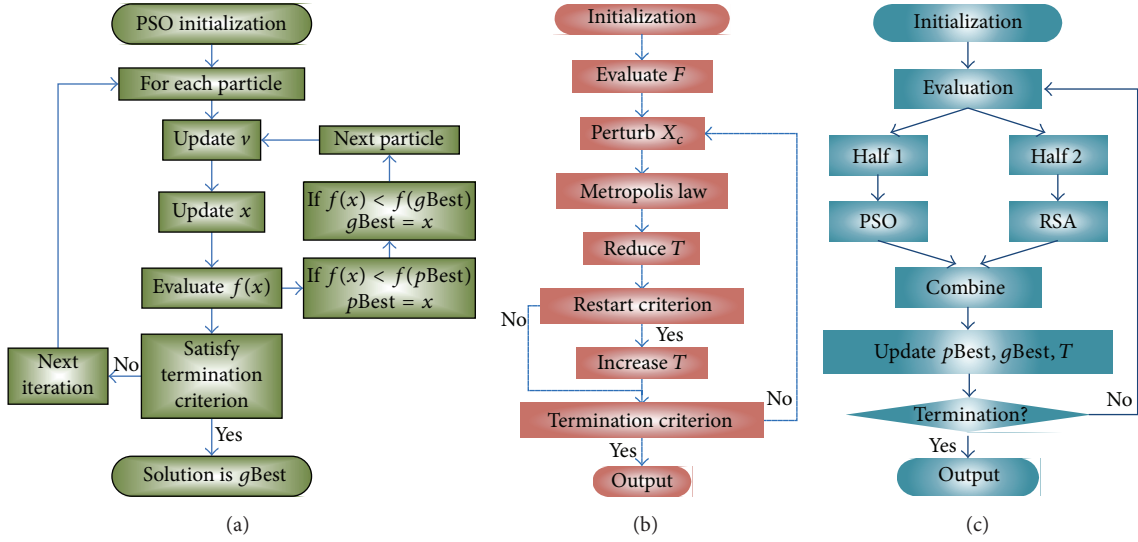


FIGURE 5: Flowchart of the (a) PSO; (b) RSA; and (c) RSAPSO.

the resulting parameters are fine-tuned towards the end of the optimization [25].

A move is made to the new solution X_n if it has smaller energy F or if the probability function has a higher value than a randomly generated number. Otherwise a new solution is generated, evaluated, and compared again. The probability p of accepting a new solution X_n which is called “Metropolis law” is given as follows:

$$p = \begin{cases} 1, & \text{if } F(X_n) < F(X_c), \\ \exp\left(\frac{F(X_c) - F(X_n)}{T}\right), & \text{otherwise.} \end{cases} \quad (14)$$

In order to avoid getting trapped at local extrema points, the reduction rate of T should be slow enough. In this study the following method to reduce the temperature has been used:

$$T_n = T_0 \times \beta^n. \quad (15)$$

Here T_0 is the initial temperature, β is the reduction constant, and n is the number of iterations. In general, most worsening moves may be accepted at initial stages, but at the final stage only improving ones are likely to be allowed. This can help the procedure jump out of a local minimum.

However, sometimes it is better to move back to a former solution that was significantly better rather than always moving from the current state. This process is called “restarting” of SA [26]. To do this we set the temperature to a former value and restart the annealing schedule. The decision to restart can be based on several criteria, including whether a fixed number of steps had passed, whether the current energy is too high compared with the best obtained so far, or whether the random number falls within prescribed range (randomly restart). In this paper, we restart the SA when the current energy is too high compared with the best energy because it performs best among all criterions [27]. The flowchart of RSA is shown in Figure 5(b).

2.3.3. Pseudocodes of RSAPSO. Traditional PSO algorithm suffers from getting trapped at the early stage [28]. On the other hand, RSA accepts a worse solution so it can escape from a local minimum, resist premature convergence, and increase the diversity [12]. Therefore, a new hybrid strategy was proposed and referred to as RSAPSO [9]. The proposed RSAPSO algorithm offsets the weaknesses of PSO and RSA. The main idea lies in the fact that it divides the population into two halves: one half runs PSO and the other half runs RSA. During each step, the results are combined and updated by the best result that is picked up from the whole population. The flowchart of RSAPSO is depicted in Figure 5(c), and its pseudocodes are given as follows.

Step 1 (initialization). Generate the population randomly.

Step 2 (evaluation). Evaluate each particle’s objective function f .

Step 3 (segmentation). Halve the population randomly: one half was updated by PSO according to formula (12) and the other half was updated by RSA according to formulas (13) and (14).

Step 4 (update). Update $pBest$ and $gBest$; update T according to formula (15).

Step 5 (repeat). Repeat Step 2 to Step 4 until the termination criterion is satisfied.

Step 6 (output). Output the final results.

3. Experiments and Discussions

The experiments were carried out on the platform of P4 IBM with 2.2 GHz Intel Core i3-2330 M CPU and 6 GB RAM, running under 64-bit Windows 7 operating system. The algorithm was in-house developed via the global optimization

TABLE 1: Parameters setting.

Algorithm	Parameters
SGA	$N = 20, P_C = 0.8, P_M = 0.1$
MFSGA	$N = 20, P_C = 0.8, P_M = 0.1, M_I = 10, M_R = 2$
RSAPSO	$N = 20, \omega = 0.6, c_1 = 1, c_2 = 1, T_0 = 100, \beta = 0.95$

toolbox of MATLAB 2013a. Readers can repeat the results of the experiment on any desktop installing MATLAB.

3.1. Parameters Setting. We compared the proposed method RSAPSO with Park's method, Anelli's method, Shih's SGA method, and Zhang's MFSGA method. Some important parameters are obtained through trial-and-error method and listed in Table 1. Here, N denotes the number of population, P_C denotes the crossover probability, P_M denotes the mutation probability, M_I denotes the migration interval, and M_R denotes the migration rate.

3.2. Comparison with Park's Method. The SE in Figure 6 is indecomposable by Park's algorithm [4]. Its original SCC is 21. We ran our RSAPSO method 20 times, and all runs obtained the optimal decomposition shown in Figure 6(a) with SCC as 10. Besides, we ran SGA and MFSGA 20 times for comparison. Their results were shown in Table 2. The second row shows the result of all 20 runs. Take SGA as an example; "19(10) + 1(11)" represents that 19 runs obtained SCC as 10 and the left one run obtained SCC as 11. The third row "Averaged SCC" shows either the averaged SCC result of heuristic methods or the SCC result of deterministic methods. Table 2 shows that SGA obtained optimal result (10) 19 times and one suboptimal result (11) as shown in Figure 6(b). MFSGA obtained optimal result (10) 19 times and one suboptimal result (11) as shown in Figure 6(c). The SCC of the two suboptimal SE decomposition results was 11. The averaged SCC of SGA, MFSGA, and RSAPSO were 10.05, 10.05, and 10, respectively.

3.3. Comparison with Anelli's Method. We compared the proposed RSAPSO with Anelli's method [29], SGA, and MFSGA. We ran SGA, MFSGA, and RSAPSO 20 times, since the distribution of their initial population is random. Figures 7(a) and 7(b) show two successful decomposition trees of Anelli's SE, and both of their SCC are 18. Figures 7(c) and 7(d) show two failed decomposition trees, of which their SCC are 22 and 23, respectively.

The comparison results are shown in Table 3. The SCC of original SE was 41. Anelli's method reduced it to 22, and the corresponding decomposition tree can be found in [29]. Among 20 runs, SGA obtained 15 times of optimal results, MFSGA obtained 17 times of optimal results, while the proposed RSAPSO method obtained all 20 successful runs. The averaged SCC of SGA, MFSGA, and RSAPSO were 19.05, 18.6, and 18, respectively.

3.4. Comparison with SGA. Shih's paper proclaimed that the optimal SCC results of decomposed trees for "ship" and "car"

shapes by SGA were 47 and 46, respectively. After searching by RSAPSO, we obtained better results (see Table 5). For the "ship" shape, the original SCC was 125, the optimal SCC found by SGA was 47, and our method found better SCC result as 44 (see 2nd row in Table 4). For the "car" shape, the original SCC was 168, the optimal SCC by SGA was 46, and our method achieved better SCC as 43 (see the 3rd row in Table 4).

3.5. Comparison with MFSGA. In what follows, we compared the proposed RSAPSO with MFSGA method. Zhang et al. proclaimed that the optimal SCC for "heart" and "umbrella" by MFSGA were 32 and 44, respectively. Using RSAPSO, we obtained better results. For SE of "heart," the original SCC was 142. MFSGA minimized SCC to 32. Our method obtained better SCC result as 30. For SE of "umbrella," the original SCC was 94. MFSGA minimized SCC to 44. Our method obtained better result as 41.

3.6. Other Examples. We used the RSAPSO for six other benchmarks SEs of different shapes, such as vase, tree, cat, V, bomb, and cup. The results were shown in Table 6. For the "vase" shape, the SCC of the original and decomposed SE was 149 and 32, respectively. For the "tree" shape, the SCC of the original and decomposed SE was 73 and 26, respectively. For the "cat" shape, the SCC of the original and decomposed SE was 102 and 30, respectively. For letter "V," the SCC of the original and decomposed SE was 94 and 34, respectively. For the "bomb" shape, the SCC of the original and decomposed SE was 91 and 26, respectively. For the "cup" shape, the SCC of the original and decomposed SE was 113 and 25, respectively.

4. Conclusions

In this paper, a novel decomposition method for arbitrarily shaped SE was proposed. The SE decomposition problem was first transformed into an optimization problem with virtue of the improved recursive dilation-union model that contains a new termination criterion. Afterwards, the RSAPSO method was introduced as the searching algorithm. In the experiments, we compared our method with Park's method [4], Anelli's method [29], Shih's SGA method [2], and Zhang's MFSGA method [5]. The results based on several benchmark shapes showed that our method was more robust than the aforementioned algorithms and was able to find the optimal decomposition tree.

The contribution of the paper lies in the following 4 aspects: (1) we proposed improved termination criteria for the recursive dilation-union model, adding the rule "sum(S) \leq 5"; (2) we used the serial computational time as the objective function; (3) we introduced the RSAPSO algorithm and proved it is superior to Park's method, Anelli's method, SGA, and MFSGA for SE decomposition application; and (4) we gave the best decomposition results for the shapes "heart," "ship," and "car" among all the state-of-the-art SE decomposition methods.

The future tentative research will focus on other optimization algorithms, such as cuckoo search [30], harmony

TABLE 2: Decomposition of Park's SE (20 runs).

	Original	Park's	SGA	MFSGA	RSAPSO
SCC			19 (10) + 1 (11)	19 (10) + 1 (11)	20 (10)
Averaged SCC	21	21	10.05	10.05	10

TABLE 3: Decomposition of Anelli's SE (20 runs).

	Original	Anelli's	SGA	MFSGA	RSAPSO
SCC			15 (18) + 4 (22) + 1 (23)	17 (18) + 3 (22)	20 (18)
Averaged SCC	41	22	19.05	18.6	18

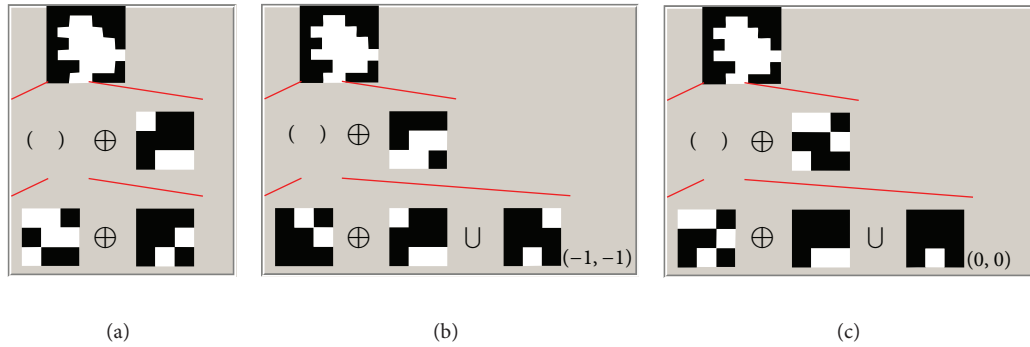


FIGURE 6: Decomposition tree of Park's SE (SCC = 21): (a) a successful run (SCC = 10); (b) a failed run by SGA (SCC = 11); and (c) a failed run by MFSGA (SCC = 11).

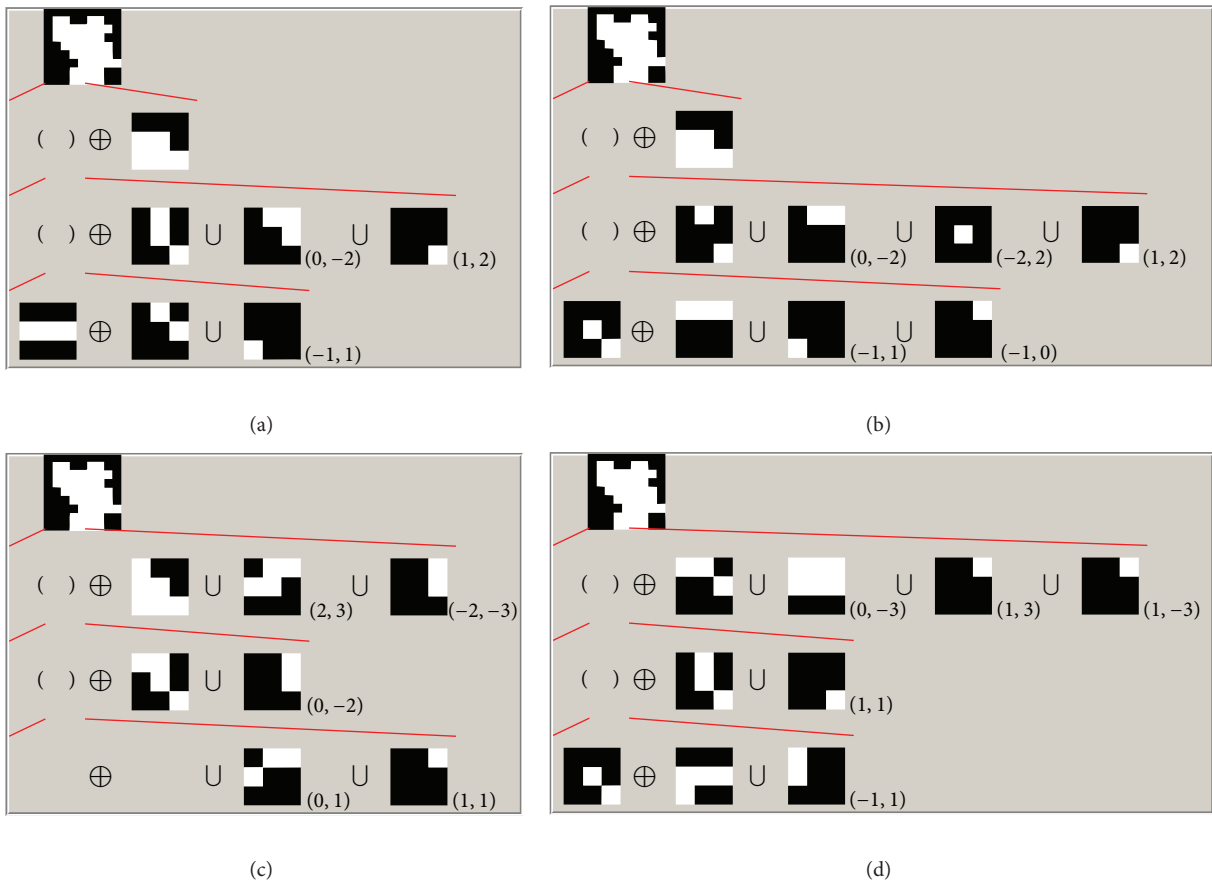

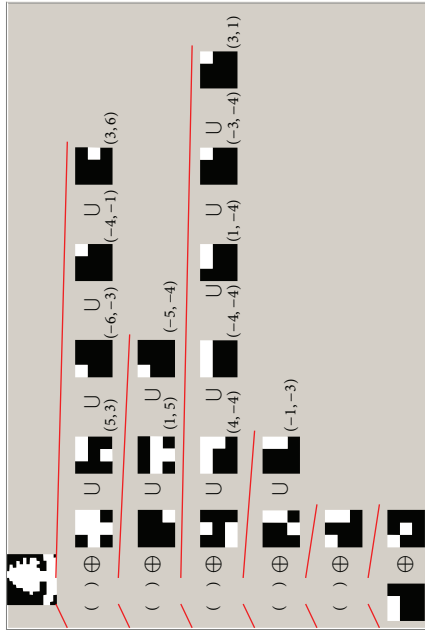
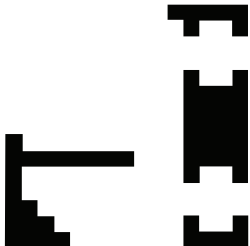
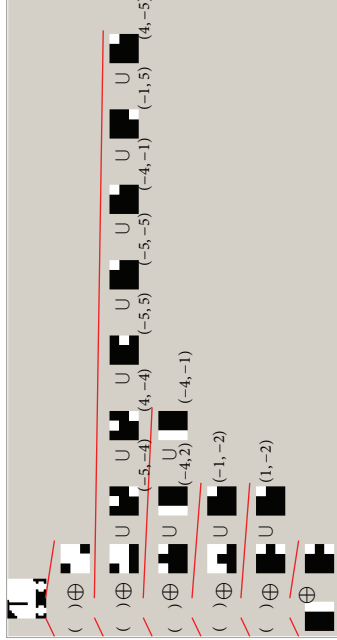


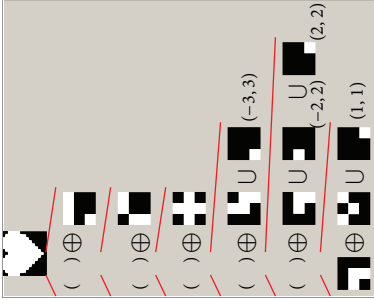
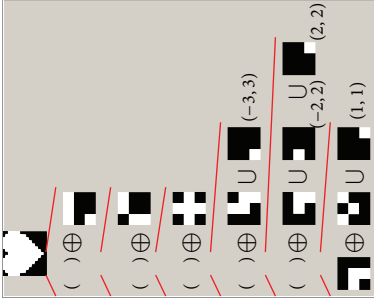

FIGURE 7: Decomposition tree of Anelli's SE (SCC = 41): (a) a successful run (SCC = 18); (b) another successful run (time = 18); (c) a failed run (SCC = 22); and (d) a failed run (SCC = 23).

TABLE 4: Two-SE decomposition by RSAPSO.

Name	SE	Decomposition by RSAPSO	Original	SCC SGA	RSAPSO
Ship			125	47*	44
Car			168	46*	43

*Data from Shih and Wu [2].

TABLE 5: Two-SE decomposition by RSAPSO.

Name	SE	Decomposition by RSAPSO		SCC	
		Original	MFSGA	RSAPSO	RSAPSO
Heart			142	32*	30
					
Umbrella			94	44*	41
					

*Data from Zhang and Wu [5].

TABLE 6: Six-SE decomposition examples by our RSAPSO method.

Name	SE	Original	SCC	Tree	Decomposition	SCC
Vase			149			32
Tree			73			26
Cat			102			30
V			94			34

TABLE 6: Continued.

Name	SE	Original	SCC	Tree	Decomposition	SCC
Bomb			91			26
Cup			113			25

search [31], genetic pattern research [32], Tabu search [33], firefly algorithm [34], honey bee mating [35], and artificial bee colony [36, 37]. We will also try other termination criteria [38] and check the effectiveness of the new SE decomposition methods.

Conflict of Interests

The authors declare that they do not have any commercial or associative interest that represents a conflict of interests in connection with the work submitted.

Acknowledgments

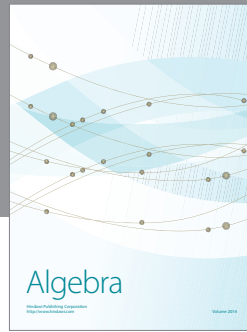
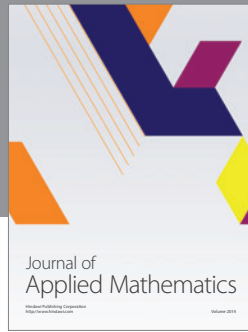
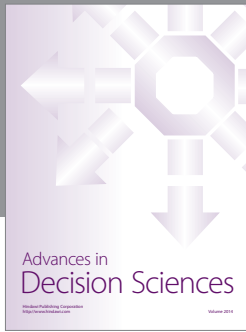
The authors would like to express their gratitude to the three anonymous reviewers. This paper is supported from the Nanjing Normal University Research Foundation for Talented Scholars (no. 2013119XGQ0061).

References

[1] O. Cuisenaire, "Locally adaptable mathematical morphology using distance transformations," *Pattern Recognition*, vol. 39, no. 3, pp. 405–416, 2006.
 [2] F. Y. Shih and Y.-T. Wu, "Decomposition of binary morphological structuring elements based on genetic algorithms,"

Computer Vision and Image Understanding, vol. 99, no. 2, pp. 291–302, 2005.
 [3] H. Park and J. Yoo, "Structuring element decomposition for efficient implementation of morphological filters," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 148, no. 1, pp. 31–35, 2001.
 [4] R. F. Hashimoto and J. Barrera, "A note on Park and Chin's algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 139–144, 2002.
 [5] Y. Zhang and L. Wu, "Recursive structure element decomposition using migration fitness scaling genetic algorithm," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6728, no. 1, pp. 514–521, 2011.
 [6] G. Fornarelli and A. Giaquinto, "Adaptive particle swarm optimization for CNN associative memories design," *Neurocomputing*, vol. 72, no. 16–18, pp. 3851–3862, 2009.
 [7] B. Alatas and E. Akin, "Chaotically encoded particle swarm optimization algorithm and its applications," *Chaos, Solitons and Fractals*, vol. 41, no. 2, pp. 939–950, 2009.
 [8] M. Rabbani, M. A. Bajestani, and G. Baharian Khoshkhou, "A multi-objective particle swarm optimization for project selection problem," *Expert Systems with Applications*, vol. 37, no. 1, pp. 315–321, 2010.
 [9] Y. Zhang and L. Wu, "A robust hybrid restarted simulated annealing particle swarm optimization technique," *Advances in Computer Science and Its Applications*, vol. 1, no. 1, pp. 5–8, 2012.

- [10] M. S. Arumugam, M. V. C. Rao, and A. W. C. Tan, "A novel and effective particle swarm optimization like algorithm with extrapolation technique," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 308–320, 2009.
- [11] A. Y. Abdelaziz, F. M. Mohammed, S. F. Mekhamer, and M. A. L. Badr, "Distribution Systems Reconfiguration using a modified particle swarm optimization algorithm," *Electric Power Systems Research*, vol. 79, no. 11, pp. 1521–1530, 2009.
- [12] Y. Zhang, Y. Jun, G. Wei, and L. Wu, "Find multi-objective paths in stochastic networks via chaotic immune PSO," *Expert Systems with Applications*, vol. 37, no. 3, pp. 1911–1919, 2010.
- [13] A. Nakib, H. Oulhadj, and P. Siarry, "Image histogram thresholding based on multiobjective optimization," *Signal Processing*, vol. 87, no. 11, pp. 2516–2534, 2007.
- [14] H. Mohamadi, J. Habibi, M. S. Abadeh, and H. Saadi, "Data mining with a simulated annealing based fuzzy classification system," *Pattern Recognition*, vol. 41, no. 5, pp. 1841–1850, 2008.
- [15] S. Rubenthaler, T. Rydén, and M. Wiktorsson, "Fast simulated annealing in Rd with an application to maximum likelihood estimation in state-space models," *Stochastic Processes and their Applications*, vol. 119, no. 6, pp. 1912–1931, 2009.
- [16] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization artificial ants as a computational intelligence technique," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [17] Y. Zhang, P. Agarwal, V. Bhatnagar et al., *Swarm Intelligence and Its Applications*, The Scientific World Journal, 2013.
- [18] Y. Zhang, L. Wu, and S. Wang, "UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 705238, 9 pages, 2013.
- [19] L. Lamberti, "An efficient simulated annealing algorithm for design optimization of truss structures," *Computers and Structures*, vol. 86, no. 19–20, pp. 1936–1953, 2008.
- [20] A. Liu, Y. Yang, Q. Xing, H. Yao, Y. Zhang, and Z. Zhou, "Improved collaborative particle swarm algorithm for job shop scheduling optimization," *Advanced Science Letters*, vol. 4, no. 6–7, pp. 2180–2183, 2011.
- [21] Y. Zhang and L. Wu, "Bacterial chemotaxis optimization for protein folding model," in *Proceedings of the 5th International Conference on Natural Computation (ICNC '09)*, pp. 159–162, Tianjian, China, August 2009.
- [22] Y. Zhang, S. Wang, G. Ji et al., "An MR brain images classifier system via particle swarm optimization and kernel support vector machine," *The Scientific World Journal*, vol. 2013, Article ID 130134, 9 pages, 2013.
- [23] E. Cosola, K. Genovese, L. Lamberti, and C. Pappalètere, "A general framework for identification of hyper-elastic membranes with moiré techniques and multi-point simulated annealing," *International Journal of Solids and Structures*, vol. 45, no. 24, pp. 6074–6099, 2008.
- [24] E. W. McGookin and D. J. Murray-Smith, "Submarine manoeuvring controller's optimisation using simulated annealing and genetic algorithms," *Control Engineering Practice*, vol. 14, no. 1, pp. 1–15, 2006.
- [25] Y. Zhang, L. Wu, N. Neggaz, S. Wang, and G. Wei, "Remote-sensing image classification based on an improved probabilistic neural network," *Sensors*, vol. 9, no. 9, pp. 7516–7539, 2009.
- [26] N. Cretu and M.-I. Pop, "Acoustic behavior design with simulated annealing," *Computational Materials Science*, vol. 44, no. 4, pp. 1312–1318, 2009.
- [27] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognition*, vol. 43, no. 1, pp. 5–13, 2010.
- [28] T. Niknam, H. D. Mojarrad, and M. Nayeripour, "A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch," *Energy*, vol. 35, no. 4, pp. 1764–1778, 2010.
- [29] G. Anelli, A. Broggi, and G. Destri, "Decomposition of arbitrarily shaped binary morphological structuring elements using genetic algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 217–224, 1998.
- [30] S. Walton, O. Hassan, K. Morgan, and M. R. Brown, "Modified cuckoo search: a new gradient free optimisation algorithm," *Chaos, Solitons and Fractals*, vol. 44, no. 9, pp. 710–718, 2011.
- [31] I. Ahmad, M. G. Mohammad, A. A. Salman, and S. A. Hamdan, "Broadcast scheduling in packet radio networks using Harmony Search algorithm," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1526–1535, 2012.
- [32] Y. Zhang, S. Wang, G. Ji et al., "Genetic pattern search and its application to brain image classification," *Mathematical Problems in Engineering*, vol. 2013, Article ID 580876, 8 pages, 2013.
- [33] B. Yao, P. Hu, X. Lu, J. Gao, and M. Zhang, "Transit network design based on travel time reliability," *Transportation Research Part C: Emerging Technologies*, 2014.
- [34] Y. Zhang, L. Wu, and S. Wang, "Solving two-dimensional HP model by firefly algorithm and simplified energy function," *Mathematical Problems in Engineering*, vol. 2013, Article ID 398141, 9 pages, 2013.
- [35] M.-H. Horng, "Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4580–4592, 2010.
- [36] Y. Zhang, L. Wu, and S. Wang, "Magnetic resonance brain image classification by an improved artificial bee colony algorithm," *Progress in Electromagnetics Research*, vol. 116, pp. 65–79, 2011.
- [37] B. Yao, B. Yao, B. Yao et al., "Artificial bee colony algorithm with scanning strategy for periodic vehicle routing problem," *SIMULATION: Transactions of the Society For Modeling and Simulation International*, vol. 89, no. 6, pp. 762–770, 2013.
- [38] J. Sun, B.-Z. Yao, C.-Y. Yang et al., "Tunnel surrounding rock displacement prediction using support vector machine," *International Journal of Computational Intelligence Systems*, vol. 3, no. 6, pp. 843–852, 2010.



Hindawi

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

