Soft Comput (2017) 21:5585–5599
DOI 10.1007/s00500-017-2752-y

FOCUS

# Swarmic approach for symmetry detection of cellular automata behaviour

**Mohammad Ali Javaheri Javid**[1] · **Wajdi Alghamdi**[1] · **Anna Ursyn**[2] · **Robert Zimmer**[1] · **Mohammad Majid al-Rifaie**[1]

Published online: 4 August 2017
© The Author(s) 2017. This article is an open access publication

**Abstract** Since the introduction of cellular automata in the late 1940s they have been used to address various types of problems in computer science and other multidisciplinary fields. Their generative capabilities have been used for simulating and modelling various natural, physical and chemical phenomena. Besides these applications, the lattice grid of cellular automata has been providing a by-product interface to generate graphical contents for digital art creation. One important aspect of cellular automata is symmetry, detecting of which is often a difficult task and computationally expensive. In this paper a swarm intelligence algorithm—Stochastic Diffusion Search—is proposed as a tool to identify points of symmetry in the cellular automata-generated patterns.

**Keywords** Cellular automata · Symmetry · Asthetics · Swarm intelligence

Communicated by C. M. Vide, A. H. Dediu.

✉ Mohammad Ali Javaheri Javid
m.javaheri@gold.ac.uk

Wajdi Alghamdi
walgh001@gold.ac.uk

Anna Ursyn
ursyn@unco.edu

Robert Zimmer
r.zimmer@gold.ac.uk

Mohammad Majid al-Rifaie
m.majid@gold.ac.uk

[1] Department of Computing, Goldsmiths, University of London, London SE14 6NW, UK

[2] School of Art and Design, University of Northern Colorado, Greeley, CO 80639, USA

## 1 Introduction

Creating aesthetically pleasing imagery has been dominated by the application of evolutionary computing. The "*Bimorphs*" of Dawkins (1986),"*Mutators*" of Latham Todd et al. (1991), and "*Virtual Creatures*" of Sims (1994) are the classical examples. Although some impressive results have been achieved, there still remain problems with the aesthetic selection. That is devising a fitness function to replace human adjustment in the process of generation and evaluation.

In this study we investigate the aesthetics of cellular automata (CA) behaviour. CA invented by von Neumann in the late 1940s as material-independent systems to investigate the possibility of self-reproduction. A cellular automaton consists of a lattice of uniformly arranged finite state automata each of which taking input from the neighbouring automata; they in turn compute their next states by utilising a state transition function. A synchronous or asynchronous interactive application of state transition function (or rule) over the states of automata (or cells) generates the global behaviour of a cellular automaton that sometimes can be very complex with high aesthetic quality.

The property of CA that makes them particularly interesting to digital artists is their ability to produce visually appealing and logically deep patterns on the basis of very simply stated preconditions. Traditional scientific intuition, and early computer art, might lead one to assume that simple programs would always produce pictures too simple and rigid to be of artistic interest. But extrapolating from Wolfram's work on CA, "it becomes clear that even a program that may have extremely simple rules will often be able to generate pictures that have striking aesthetic qualities sometimes reminiscent of nature, but often unlike anything ever seen before" (Wolfram 2002, p. 11). This is a new way of generating imagery which has no precedent in human culture (Roth and Deutsch 2011). CA have been used as artistic tool since the 1960s. The most notable examples are "*Pixillation*", one of the early computer-generated animations (Schwartz and Schwartz 1992), the digital art works of Struycken (Scha 2006; Struycken 1976) and Brown (Beddard and Dodds 2009; Brown 2001) [a review of the application of CA in digital art provided in Javaheri Javid et al. (2015)]. Images generation by CA has unique qualities compared to imagery generated by other approaches like evolutionary art. This is due to processing elements which are pixels in evolutionary approaches where in CA the processing elements are cells. In addition there is always an implicit relationship between cells and its surround because of the neighbourhood relation. Although classical one-dimensional CA with binary states can generate complex behaviours, experiments with multi-state two-dimensional (2D) CA have shown that adding more states significantly increases the complexity of behaviour, therefore, generating very complex symmetrical patterns (Javaheri Javid et al. 2014; Javaheri Javid and te Boekhorst 2006) which are extremely challenging to generate using conventional mathematical methods. A recent work by Adamatzky and Martínez (2016) offers insights in the production of art works, using simple computational models with morphological behaviour, at the edge of mathematics, computer science, physics and biology, where CA is explored in further details.

In this study we approach the problem in the framework of dynamical systems and define a criterion for aesthetic selection in terms of its association with symmetry. The association of aesthetics and symmetry has been investigated from different points of view. In this paper, a brief account on cellular automata is presented, followed by a section on symmetry and its significance in aesthetics. Then, a swarm intelligence algorithm—Stochastic Diffusion Search—is explained, highlighting its main features, including its unique partial function evaluation aspect. Afterwards, the application of the algorithm in detecting points of symmetry is detailed, illustrating the performance of the method proposed.

## 2 Cellular automata

For the purpose of this study we provide a formal definition for 2D CA as follows and for the rest of the paper all the notions will be referring to this definition.

**Definition 1** A deterministic finite automaton is formally defined (Linz 2001) as a quintuple of $\mathcal{M}$ such that:

$$\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle \tag{1}$$

1. $Q$ is a finite set of *states*,
2. $\Sigma$ is a finite set of symbols as *input alphabet*,
3. $\delta : Q \times \Sigma \mapsto Q$ is the *state transition function*,
4. $q_0 \in Q$ is the *start* or *initial state*,
5. $F \subseteq Q$ is a set of *accepting* or *final states*.

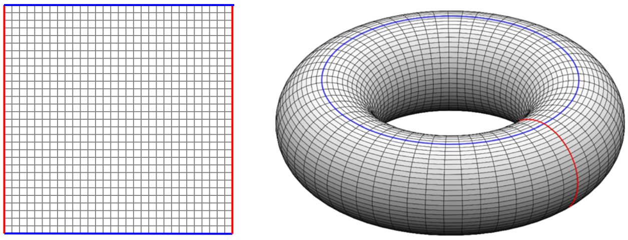


Fig. 1 Formation of virtual torus shape in a lattice with periodic boundary conditions

The state transaction function $\delta$ determines the transitions from one state to another state. It takes two arguments as $q \in Q$ and an input symbol $a \in \Sigma$ then maps them to a final state $q_1 \in Q$ (i.e $\delta(q, a) = q_1$).

**Definition 2** A lattice ($L$) is a regular tiling of a space by a unit cell.

The *Euclidean* plane is considered so the lattice $L$ is over $Z^2$. Lattices can have *square*, *hexagonal* or *triangle* for their unit cells. A lattice can be *infinite* with *open boundary conditions* or *finite* with *periodic boundary conditions*. A finite lattice with periodic boundary conditions where the opposite borders (up and down with left and right) are connected, forms a virtual torus shape (Fig. 1).

**Definition 3** A cellular automaton is a lattice of regularly tiled by deterministic finite state automaton as unit cell.

A cellular automaton is presented as a quadruple of $\mathcal{A}$ such that:

$$\mathcal{A} = \langle L, S, N, f \rangle \tag{2}$$

1. $L$ is a finite square lattice with *periodic boundary conditions*,
2. $S \subseteq \mathbb{N}^0$ is a finite set of non-negative integers as *states*,
3. $N \subseteq \mathbb{N}^+$ is a finite set of non-negative integers as *neighbourhood*,
4. $f : S^{|N|} \mapsto S$ is the *state transition function*.

The state transition function $f$ maps from the set of neighbourhood states $S^{|N|}$ where $|N|$ is the cardinality of neighbourhood set, to the set of states $S = \{s_0, \ldots, s_{n-1}\}$ *synchronously* in *discrete time* intervals of $t = \{0, 1, 2, 3, \ldots, n\}$ where $t_0$ is the *initial time* of the cellular automaton. By convention every automaton on the lattice is in neighbourhood relation with itself. A mapping that satisfies $f(s_0, \ldots, s_0) = s_0$ where $s_0 \in S$ is called a *quiescent state*. In other words,

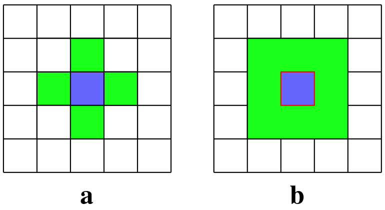


**Fig. 2** von Neumann (**a**) and Moore (**b**) neighbourhood templates

if a lattice is considered as a set, then its elements are deterministic finite automata such that $L = \{m_1, m_2, \ldots, m_n\}$. The state of each cell at time $(t+1)$ is determined by the states of immediate neighbouring cells (nearest neighbourhood) at time $(t)$ given a neighbourhood template. There are two commonly used neighbourhood templates considered for 2D CA. A five-cell mapping ($f : S^5 \mapsto S$) known as *von Neumann neighbourhood* (Eq. 3; Fig. 2a) and a nine-cell mapping ($f : S^9 \mapsto S$) known as *Moore neighbourhood* (Eq. 4; Fig. 2b).

$$s_{i,j}^{t+1} = f\begin{pmatrix} & s_{(i,j+1)}^t & \\ s_{(i-1,j)}^t & s_{(i,j)}^t & s_{(i+1,j)}^t \\ & s_{(i,j-1)}^t & \end{pmatrix} \tag{3}$$

$$s_{i,j}^{t+1} = f\begin{pmatrix} s_{(i-1,j+1)}^t & s_{(i,j+1)}^t & s_{(i+1,j+1)}^t \\ s_{(i-1,j)}^t & s_{(i,j)}^t & s_{(i+1,j)}^t \\ s_{(i-1,j-1)}^t & s_{(i,j-1)}^t & s_{(i+1,j-1)}^t \end{pmatrix} \tag{4}$$

A function $F : S^L \mapsto S^L$ which gives the *global state* as *configuration* $C$ over the lattice $L$ at time $t$ is the *global transition function*. $F$ maps the configuration at time $t$ to the configuration at time $t+1$ where $C \in S^L$.

$$C^{t+1} = F\left(C^t\right) \tag{5}$$

The sequence $(c^0, c^1, c^2, \ldots, c^{n-1})$ collectively is the *behaviour* of a cellular automata (also referred as the global behaviour) where $c^0$ is the *initial configuration* at $t = 0$. The graphical representation of the sequence of configurations as the behaviour of a cellular automaton by assigning a colour for cell states over the lattice called a *space–time diagram* (Fig. 3).

The total number of possible state transition functions as the size of rule space $\Phi$ can be obtained from

$$\Phi = |S|^{|S|^{|N|}} \tag{6}$$

where $|S|$ is the cardinality of $S$ and $|N|$ is the cardinality of $N$. For instance given a two state ($|S| = 2$) mapping with Moore neighbourhood template ($|N| = 9$), then $\Phi = 2^{2^9} = 2^{512} \approx 1.3 \times 10^{154}$. In order to put this number in perspective it can be noted that the number of atoms in visible universe is $\approx 10^{80}$. This excessively large number of state transition functions neither can be stored in any modern computer nor it can be algorithmically defined. A common approach to overcome this problem is to define a subset(s) of all possible state transition functions by a formula. Two commonly applied formulas to generate such subsets are *totalistic* rules and *outer totalistic* rules where the state of each cell is updated according to the sum of the states of the neighbouring cells in given template. Equations 7 and 9 show the generation of totalistic (tot) and outer totalistic (outer-tot) rules for Moore neighbourhood template. The new state by applying an outer totalistic rule depends on a tuple, namely the old state ($s_{i,j}^t$) and the sum of the neighbourhood without the current state. It does not only depend on the neighbourhood's sum as in the totalistic rule.

$$\text{Totalistic rules: } s_{i,j}^{t+1} = f_{tot}(\sigma_{tot}) \tag{7}$$

where

$$\sigma_{\text{tot}} = \sum \begin{Bmatrix} s_{(i-1,j+1)}^t & s_{(i,j+1)}^t & s_{(i+1,j+1)}^t \\ s_{(i-1,j)}^t & s_{(i,j)}^t & s_{(i+1,j)}^t \\ s_{(i-1,j-1)}^t & s_{(i,j-1)}^t & s_{(i+1,j-1)}^t \end{Bmatrix} \tag{8}$$

$$\text{Outer totalistic rules: } s_{i,j}^{t+1} = f_{\text{outer-tot}}(\sigma_{\text{outer-tot}}) \tag{9}$$

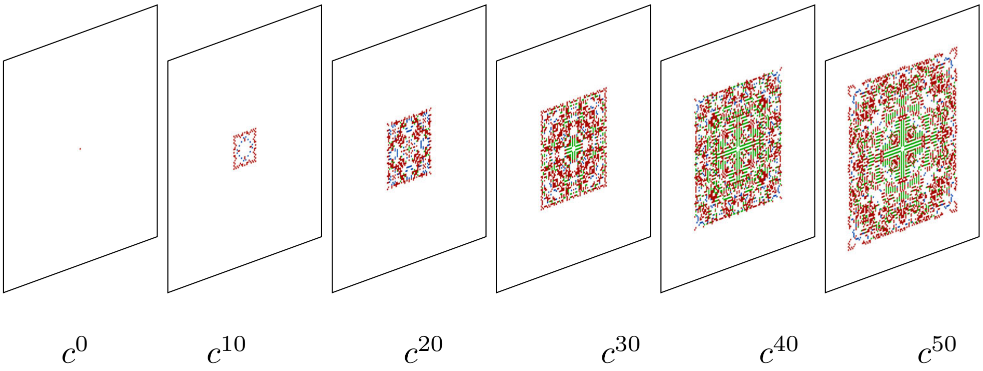


**Fig. 3** A space–time diagram depicts the behaviour of a cellular automaton as the sequence of configurations

**Table 1** A Comparison of the rule spaces for two state 2D CA using formulas

| Rule type | von Neumann | Moore |
|---|---|---|
| All | $2^{2^5} = 2^{32} \approx 4 \times 10^9$ | $2^{2^9} = 2^{512} \approx 1.3 \times 10^{154}$ |
| Outer totalistic | $2^{10} = 1024$ | $2^{18} = 262144$ |
| Totalistic | $2^5 = 32$ | $2^9 = 512$ |

**Table 2** Specifications of cellular automaton 1

| Cellular automaton 1 |
|---|
| $S = \{0, 1\}^*$ |
| $f_{\text{outer-tot}} = S^8 \mapsto S$ |
| $s^{t+1}_{i,j} = \left\{ \begin{array}{l} 1 \text{ if } s^t_{(i,j)} = 0 \text{ and } \sigma = 1 \\ 0 \text{ otherwise} \end{array} \right\}$ |

*Colour assignments are *white* for quiescent state (0) and *red* for (1)

**Table 3** Specifications of cellular automaton 2

| |
|---|
| $S = \{0, 1, 2\}^*$ |
| $f_{\text{outer-tot}} = S^8 \mapsto S$ |
| $s^{t+1}_{i,j} = \left\{ \begin{array}{l} 1 \text{ if } s^t_{(i,j)} = 1, 2 \text{ and } \sigma = 0 - 8 \\ 2 \text{ if } s^t_{(i,j)} = 0 \quad \text{ and } \sigma = 3 \\ 0 \text{ otherwise} \end{array} \right\}$ |

*Colour assignments are *white* for quiescent state (0), *red* for (1) and *green* for (2)

where

$$\sigma_{\text{outer-tot}} = \sum \left\{ \begin{array}{ccc} s^t_{(i-1,j+1)} & s^t_{(i,j+1)} & s^t_{(i+1,j+1)} \\ s^t_{(i-1,j)} & & s^t_{(i+1,j)} \\ s^t_{(i-1,j-1)} & s^t_{(i,j-1)} & s^t_{(i+1,j-1)} \end{array} \right\} \tag{10}$$

Table 1 demonstrates the reduction in the size of rule space by formulas to generate subset of state transaction functions.

The behaviour of a particular cellular automaton is constrained by initial configuration, transaction function, the number of states and the number of time intervals. In the lack of any viable model to evaluate the behaviour of 2D CA given state transition function, the only available method is to run simulations. Given the vast size of the rule space and the fact that rule space is an unstructured space such that by knowing the behaviour a particular cellular automaton or a set of CA, the behaviour of other CA cannot be induced. CA examples (Tables 2 and 3) demonstrate the formation of patterns from a single cell (Fig. 4) and glider as initial configurations (Fig. 5). And Fig. 6 shows experimental patterns generated by the authors to demonstrate the generative capabilities of CA in creating very complex symmetrical patterns.

## 3 Symmetry and aesthetics

The association of aesthetic preferences and symmetry of a stimulus has been investigated extensively in the literature. Symmetry having proportionality and balance is considered to be an important element of aesthetics. The role of sym-

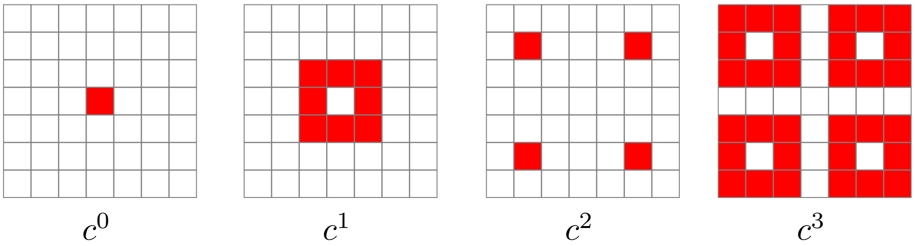


**Fig. 4** Formation of patterns from a single cell as initial configuration

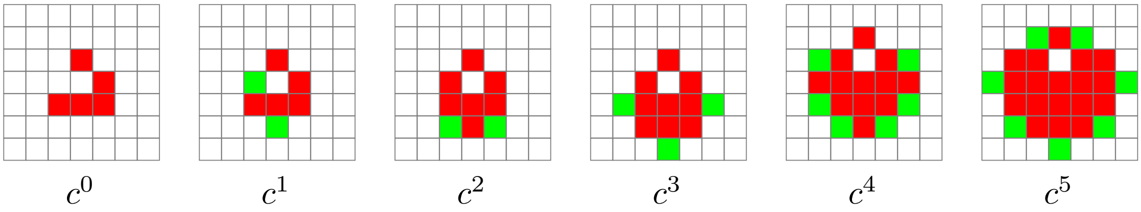


**Fig. 5** Formation of patterns from a glider as initial configuration

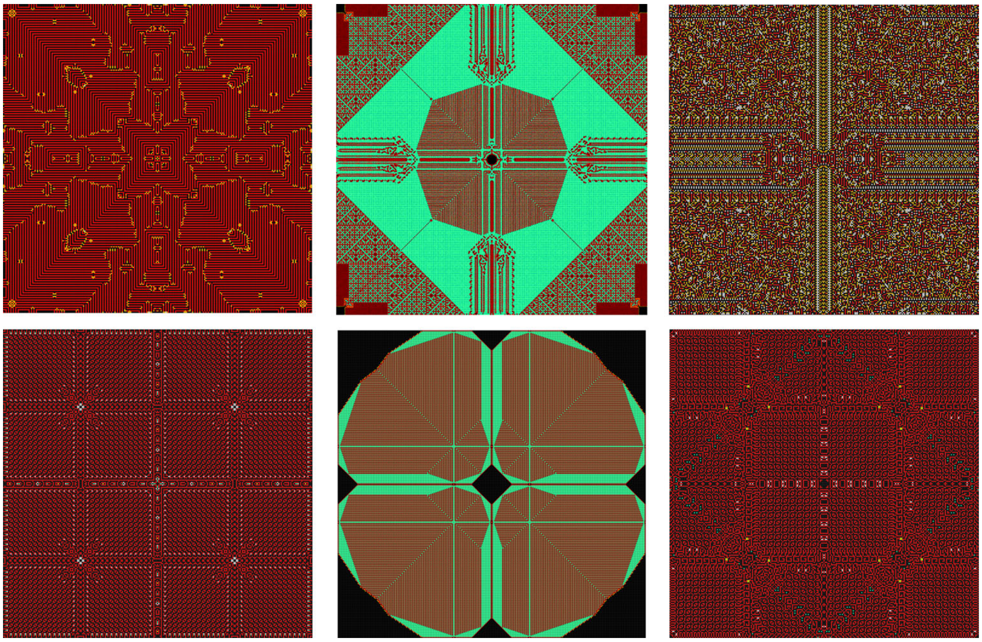


**Fig. 6** Sample complex symmetrical patters generated by CA

metry in art, architecture and its association with aesthetic preferences is a well-known concept (Møller 1998). Natural objects displaying symmetry evoke wonder and surprise because of their intricacy. For example, architecture and architectural details, such as stain windows, mosaics, and friezes, visual arts, pottery and ceramics, quilts, textiles, and carpets, make a varied use of symmetry as an important principle in their design. An examination of architecture and cell biology in terms of biosemiotics, with architectural structures, discussed as context-dependent semiotic objects with functional and/or aesthetic values. Both the natural and man-made environment can be perceived as locus, place, site, or a part of a mental map of a cultural framework. Maybe for that reason symmetry is so often seen not only beautiful but also conducive to visual communication (Ferreira 2012).

People find symmetrical patterns more beautiful than asymmetrical patterns (Jacobsen et al. 2006). A study investigating the effect of symmetry on interface judgements, and relationship between a higher symmetry value and aesthetic appeal for the basic imagery shows preference of symmetric over asymmetric images (Bauerly and Liu 2006). Further studies found that if symmetry is present in the face or the body, an individual is judged as being relatively more attractive and if the body is asymmetric the face is rated unattractive, even if the person doing the rating never sees the body (Randy and Steven 1993; Gangestad et al. 1994). Symmetry plays a crucial role in theories of perception and is even considered a fundamental structuring principle of cognition (Leyton 1992). From evolutionary perceptive physical appearances like as symmetry, and perceived level of aesthetics as an indirect measure in mate selection (Møller 1998; Møller and Cuervo 1999). It is not surprising that humans find sensory delight in symmetry, given the world in which we evolved. In our world the animals that have interested us and our ancestors (as prey, menace, or mate) are overwhelming symmetric along at least one axis (Railton 1998). Studies demonstrate the direct effects of symmetry on attractiveness (Grammer and Thornhill 1994). In other words symmetry is positively linked with both psychological and physiological health indicators (Shackelford 1997). The processing fluency theory states that a person has a certain mental state that facilitates the processing of specific information. This state may emerge from both stimulus attributes (e.g. it is easier to process symmetrical stimuli; Reber et al. 2004). In particular, symmetric objects are more readily perceived (Carroll 2003).

In geometry a shape is attributed as symmetrical if it is invariant to the application of one or more symmetry operations like translation, rotation, reflection, and glide reflection. The type of a symmetrical object is also specified with respect to a given symmetry operation(s) applied to the object when this operation preserves some property of the object. Such operations form a symmetry group of the object. In geometry symmetrical shapes are generated by applying symmetry operations like translations, rotations, reflections, and glide reflections. There are several types of symmetry, for example, the line or the radial mirror symmetry. Mirror symmetry, a symmetrical object is often defined as the correspondence in size, form, and arrangement of similar parts on the opposite sides of a point, line (axis), or plane. A figure that has line symmetry has two halves which coincide if folded along its line of symmetry, and these halves are congruent, it means, they are the same size and shape. Symmetrical objects show elements of symmetry, for example, a shape of a crystal may show rotation axes, a centre of symmetry, or mirror planes, imaginary planes that separate an object into halves. Radial symmetry in an object occurs when it can be rotated around the rotation axis and retain the same appearance as before rotating, repeating itself several times during a complete rotation. Symmetry exists not only in geometry but also in natural world and human works.

Scientists and artists see a purpose in symmetry investigations, for example, mathematicians, anthropologists, artists, designers, architects who conduct computer analysis of the facades, friezes, and some architectural details, as well as researchers in many fields of natural sciences, medicine, pharmacology, biology, geology, or chemistry. Many artists have created masterpieces this way. Artists used to transform patterns and repetitions to apply the unity or symmetry in their compositions (for example, by examining a Fibonacci sequence, prime numbers and magic squares, a golden section, or tessellation techniques). Mathematicians, computing scientists, and artists used to apply visual metaphors as a cognitive tool to visualise the world's structure and our knowledge. The theme of symmetry can certainly be considered inspirational to create biologically inspired art, because symmetrical forms and shapes possess an aesthetic beauty and an order reflected by their geometry. We can appreciate these forms finding the importance of adaptations that animals develop as an answer to the conditions of life, examining mathematical order in natural forms, and recreating it in our own artwork. With generative approach, artists draw from natural phenomena observed in biology and physics, and their creative process may evolve into a sequence of iterative solutions and modifications transforming the artwork.

However, developing computational methods which generate symmetrical patterns is still a challenge since it has to connect abstract mathematics with the noisy, imperfect, real-world; and few computational tools exist for dealing with real-world symmetries (Liu 2002). Applying evolutionary algorithms to produce symmetrical forms leaves the formulation of fitness functions, which generate and select symmetrical phenotypes, to be addressed. Lewis describes two strategies in evolutionary algorithms approach for generating and selecting symmetrical forms: "a common approach is to hope for properties like symmetry to gradually emerge by selecting for them. Another strategy is to build in symmetry functions which are sometimes activated and are appearing suddenly. However, this leads to a lack of control, as offspring resulting from slight mutations (i.e. small steps in the solution space) bear little resemblance to their ancestors" (Lewis 2008).

The next section explains the swarm intelligence algorithm which will be used in detecting symmetrical patterns.

## 4 Swarm intelligence

The swarm intelligence algorithm used in this work is Stochastic Diffusion Search (SDS) (Bishop 1989; al-Rifaie and Bishop 2013) which is a probabilistic approach for solving best-fit pattern recognition and matching problems. SDS, as a multi-agent population-based global search and optimisation algorithm, is a distributed mode of computation utilising interaction between simple agents. Its computational roots stem from Geoff Hinton's interest 3D object classification and mapping. See Hinton (1981) for Hinton's work and Bishop and Torr (2004) for the connection between Hinton mapping and SDS. SDS algorithm has been used in various fields including optimisation, generative arts and medical imaging (e.g. al-Rifaie et al. 2012a, b). SDS has a strong mathematical framework

Unlike many natured inspired search algorithms, SDS has a strong mathematical framework, which describes the behaviour of the algorithm by investigating its resource allocation (Nasuto 1999), convergence to global optimum (Nasuto and Bishop 1999), robustness and minimal convergence criteria (Myatt et al. 2004) and linear time complexity (Nasuto et al. 1998). The full mathematical model and proof of SDS convergence are elaborated in al-Rifaie and Bishop (2013).

### 4.1 SDS architecture

Similar to other swarm intelligence algorithms, SDS commences a search or optimisation by initialising its population. In any SDS search, each agent maintains a hypothesis, $h$, defining a possible problem solution. After initialisation, the two phases of SDS are followed (see Algorithm 1 for a high-level description of SDS):

- Test phase
- Diffusion phase

In the test phase, SDS checks whether the agent hypothesis is successful or not by performing a partial hypothesis evaluation and returning a domain-independent boolean value. Later in the iteration, contingent on the strategy employed, successful hypotheses diffuse across the population and in this way information on potentially good solutions spreads throughout the entire population of agents.

In other words, in the Test phase, each agent performs *partial function evaluation,* pFE, which is some function of the agent's hypothesis, pFE $= f(h)$, where $f$ is the function and $h$ is the hypothesis; and in the diffusion phase, each agent recruits another agent for interaction and potential communication of hypothesis.

**Algorithm 1** SDS Algorithm

```
Initialising agents
While (stopping condition is not met)
   Testing hypotheses
      Determining agents status (active/inactive)
   Diffusing hypotheses
      Exchanging of information
End While
```

### 4.2 Standard SDS and passive recruitment

In standard SDS, *passive recruitment mode* is employed. In this mode, if the agent is inactive, a second agent is randomly selected for diffusion; if the second agent is active, its hypothesis is communicated (*diffused*) to the inactive one. Otherwise there is no flow of information between agents; instead a completely new hypothesis is generated for the first inactive agent at random (see Algorithm 2). Therefore, recruitment is not the responsibility of the active agents. Higher rate of inactivity boosts exploration, whereas a lower rate biases the performance towards exploitation. Details of the test phase and the fitness function is described later in this paper.

**Algorithm 2** Passive Recruitment Mode

```
For each agent ag
   If ( !ag.isActive )
      r_ag = pick a random agent
      If ( r_ag.isActive )
         ag.hypothesis = r_ag.hypothesis
      Else
         ag.hypothesis = generate random hypothesis
   End If
End For
```

### 4.3 Partial function evaluation

One of the concerns associated with many optimisation algorithms (e.g. Genetic Algorithm, Particle Swarm Optimisation and etc.) is the repetitive evaluation of a computationally expensive fitness function. In some applications, such as tracking a rapidly moving object or generation of CA patters, the repetitive function evaluation significantly increases the computational cost of the algorithm. Therefore, in addition to reducing the number of function evaluations, other measures can be used in an attempt to reduce the computations carried out during the evaluation of each possible solution, as part of the overall optimisation (or search) processes.

The commonly used benchmarks for evaluating the performance of swarm intelligence algorithms are typically small in terms of their objective functions computational costs (Digalakis and Margaritis 2002; Whitley et al. 1996), which is often not the case in real-world applications (examples of costly evaluation functions are seismic data interpretation, selection of sites for the transmission infrastructure of wireless communication networks and radio wave propagation calculations of one site, etc.).

Costly objective function evaluations have been investigated under different conditions (Jin 2005) and the following two broad approaches have been proposed to reduce the cost of function evaluations:

- The first is to estimate the fitness by taking into account the fitness of the neighbouring elements, the former generations or the fitness of the same element through statistical techniques introduced in Branke et al. (2001).
- In the second approach, the costly fitness function is substituted with a cheaper, approximate fitness function.

When agents are about to converge, the original fitness function can be used for evaluation to check the validity of the convergence (Jin 2005).

The approach that the standard SDS algorithm uses is similar to the second method. Many fitness functions are decomposable to components that can be evaluated separately. During the test phase of SDS, in partial function evaluation (pFE, which is some function of the agent's hypothesis, pFE $= f(h)$, where $f$ is the function and $h$ is the hypothesis), the evaluation of one or more of the components may provide partial information to guide the subsequent optimisation process.

In other words, instead of evaluating the hypothesis in its entirely, part of it, which is called *micro-feature*, is selected and evaluated accordingly. Therefore, during the test phase, only the randomly selected micro-features of the hypotheses are evaluated and the status of each agent is thus determined. Thus, if the micro-feature of each hypothesis consists of, say, $\frac{1}{10}$ of the entire hypothesis, the computational expense for the evaluation process of each hypothesis would be $\frac{9}{10}$ computationally cheaper.

Next, details of the process through which SDS performs its spatial-independent symmetry detection is presented.

## 5 Experiments and results

This section explains the design of the experiments conducted along with the results of applying SDS to identify partial or full symmetries on the cellular automata-generated patterns[1]. The inputs to the system are sample patterns used as proof of principle to show the functionality of the method; afterwards, some real-world cellular automata-generated patterns are fed in the system to evaluate the overall performance of the algorithm in detecting the aforementioned types of symmetries.

In order to adopt SDS to identify symmetries, the following important considerations are taken into account (see Algorithm 3 for more details):

- the search space comprises of the entire cells on the grid (see Fig. 7-left where the search space size is 5 × 5)
- SDS hypothesis is a cell index $(x, y)$. For instance the coordinate $(x, y) = (2, 2)$ could be the hypothesis and micro-features[2] can be selected by specifying the $x_d$ and $y_d$ distances from the hypothesis; for instance if the $(x_d, y_d)$ distance is $(2, 0)$, this micro-features can be found at the following coordinates: $(x - x_d, y - y_d)$ and $(x + x_d, y + y_d)$; therefore, the coordinates of the micro-features in this example will be $(2 - 2, 2 - 0)$ and $(2 + 2, 2 + 0)$ resulting in $(0, 2)$ and $(4, 2)$. See Fig. 7-right.
- the environment in cellular automata is torus, which means if moving downwards along the search space when we reach the last raw, the next row to be visited is the top row. The same is applicable when moving between columns (see Fig. 1 shows the 2D representation of the cellular automata and its real structure as torus).

**Algorithm 3** Detecting symmetry by SDS algorithm

```
Initialising agents
    Each agent picks a cell as its
     hypothesis *
Test Phase
    xd and yd distances are picked randomly
       for each agent
    If C(x-xd, y-yd) = C(x+xd, y+yd) **
        Agent becomes active
    Else
        Agent becomes inactive
Diffusion Phase
    Exchanging of information

*   Agent hypothesis refers to their (x,y)
     coordinate
**  C(i,j) returns the colour of the pixel at
     position (i,j)
```

The patterns in Fig. 8 show the hypothesis (3, 2) and the various possible micro-features, some of which resulting in the hypothesis' status to be *true* while some others lead to the hypothesis' status to be *false*. In other words, various micro-features are selected to test the symmetry of the pattern along various axes of symmetry. Figure 9 shows the four axes of symmetry in a fourfold symmetrical pattern on the centre of the pattern and Fig. 10 shows each of these axes separately, again on the centre of the pattern. The torus structure of cellular automata is demonstrated in the choice of some of the corresponding micro-features; see, for example, Fig. 8 top-right corner, where the micro-feature is chosen at $(-1, -1)$ distance. Thus the corresponding cell is chosen at $(1, 1)$ distance from the hypothesis, which means moving out of the 2D canvas from the right border and entering again from the left.

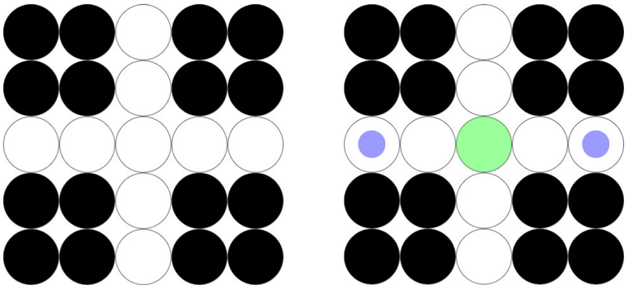


**Fig. 7** *Left*: search space (5 × 5); *right*: active hypothesis is shown in *green*; and the selected micro-features are highlighted in *blue* (colour figure online)

Having considered the details above, the process through which SDS commences with the initialisation phase and then cycle through the two phases and test and diffusion is explained next.

### 5.1 Initialisation phase

During the initialisation phase each one of the agents in the population is assigned a hypothesis which is a random $(x, y)$ coordinate from the search space. Additionally, the status of all agents are initially set to *false*.

### 5.2 Test phase

In the test phase, each agent, which is already allocated a hypothesis, picks a random $x_d$ and $y_d$ distances from the hypothesis cell as its micro-feature; the randomly selected micro-feature is then compared against its corresponding mirrored cells to check if the mirrored cell has the same value. If the values are the same, the status of the agent is set to *true*, otherwise *false*

### 5.3 Diffusion phase

The process in the diffusion phase is the same as the one detailed in the algorithm description: each inactive agent

[1] The technique presented in this paper is applicable to any CA-generated patterns. In this work a few sample CA-generated representations are used to illustrate the application of the method introduced.

[2] Micro-features are used in the test phase of SDS to determine the status of the agent (i.e. active or inactive). Micro-features, in this paper, refer to the pixels surrounding the hypothesis's $(x, y)$ coordinate; these pixels can be at the distance $(x_d, y_d)$ from the hypothesis.

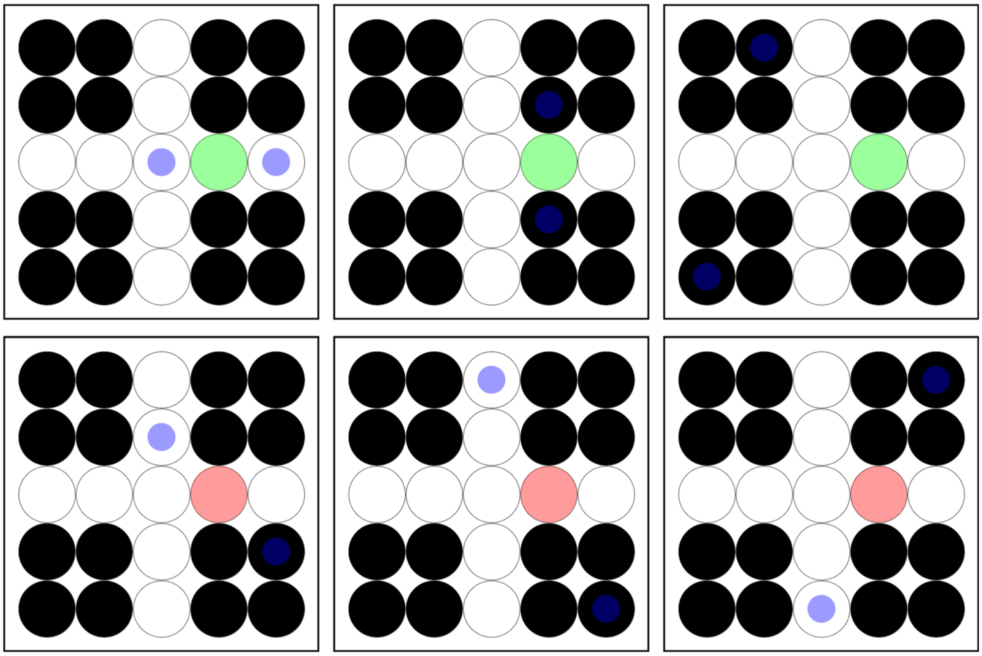


**Fig. 8** Sample hypothesis set to be (3, 2); active hypotheses are shown in *green* and the inactive ones are displayed in *red*; the selected micro-features are highlighted in *blue* (colour figure online)

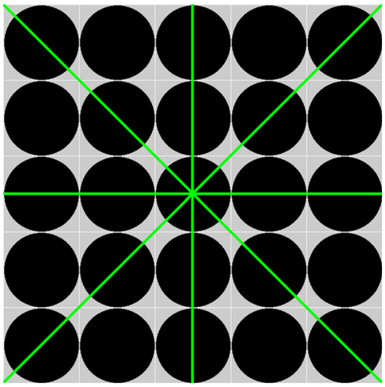


**Fig. 9** Four axes of symmetry in a fourfold symmetrical pattern

picks an agent randomly from the population; if the randomly selected agent is active, the inactive agent adopts the hypothesis of the active agent (i.e. the $(x, y)$ coordinate), otherwise the inactive agent picks a random coordinate from the search space.

After $n$ number of iterations, agents converge on the $(x, y)$ coordinates with the most symmetrical quality.

### 5.4 Experiments and discussion

One of the main features of SDS is partial function evaluation which here manifests itself in the following: each time comparing one cell on one side of the symmetrical point to its corresponding cell on the other side. Therefore, even when an agent is active, in the next iteration it picks another micro-feature and checks the point from "a different perspective" to ensure that the symmetry still holds. In other words, using this approach, the algorithm allocates its resources "wisely" and repeatedly tests the already maintained points of interest against any asymmetrical discovery. Therefore, the agents' focus is guided towards the more promising area of the search space, where the algorithm identifies partial symmetry. The area with the identified partial symmetry is

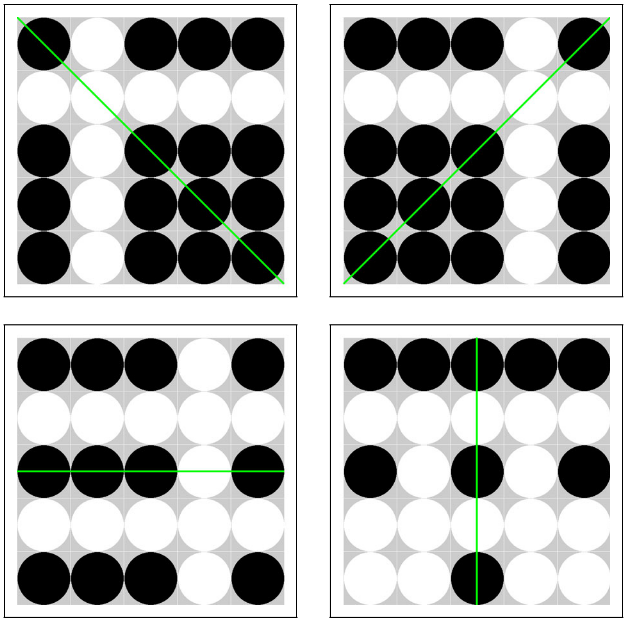


**Fig. 10** Individual axes of symmetry

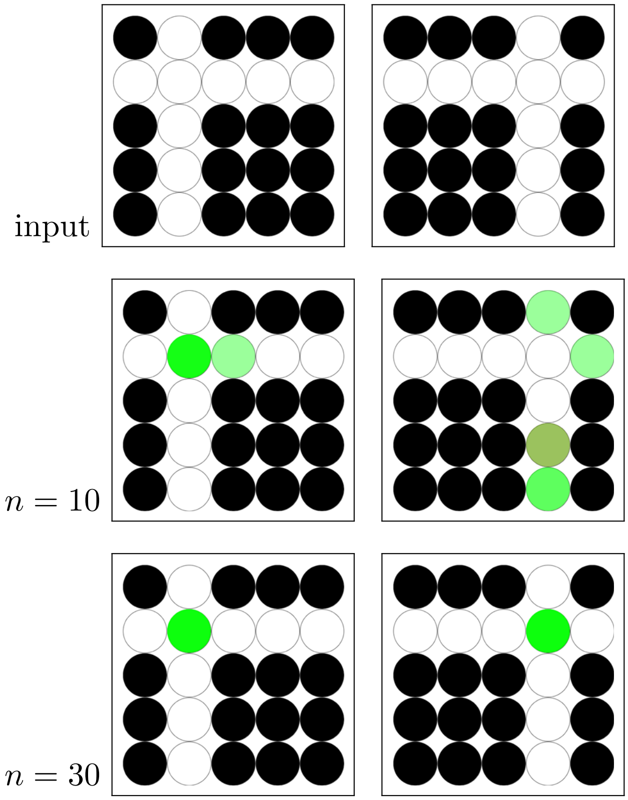


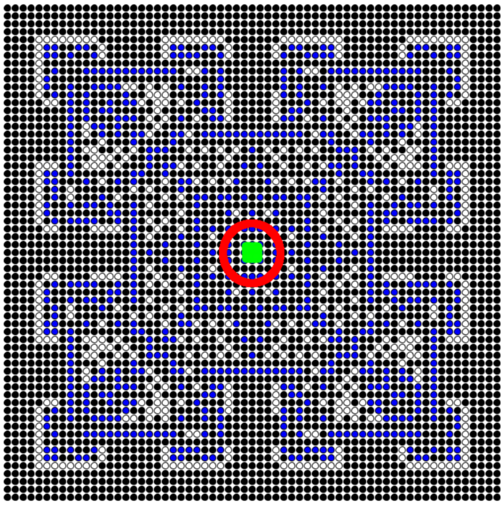
**Fig. 11** SDS agents convergence on the point of symmetry. *Top*: two input patterns; *middle*: SDS agents after $n = 10$ iterations; *bottom*: SDS agents after $n = 30$ iterations

further investigated; the repeated exploration of the promising areas guarantees the agents' convergence to the location of symmetry.

For the experiments reported in this work, the population size is empirically calculated using the following formula:

$$\text{pSize} = \left\lceil \frac{w^2}{4} \right\rceil \tag{11}$$

where $pSize$ is the population size and $w$ is the width of the search space. Using this setup, the agents land on fourth of the search space; therefore, for the search space introduced earlier, the population size will be set to $\text{pSize} = \left\lceil \frac{5^2}{4} \right\rceil = 6$. Figure 11 shows two patterns which are processed by SDS algorithm and the results are shown after $n = 10$ and $n = 30$ iterations[3]. The figures in the last row show better convergence to the point of symmetry as they are allowed more SDS iterations. As illustrated in the figures, some agents became active on $(x, y)$ coordinates which do not represent the full fourfold symmetry; these agents will eventually pick different micro-features in the next iterations and become inactive; consequently, when they are inactive, they need to choose random agents; given that the number of active agents on the centre of symmetry increases over time (thanks to the diffusion phase), it is likely that an active agent is chosen. This would lead to the inactive agents picking micro-features from the centre of symmetry in their next iterations and become/stay active. Note that in these experiments, alpha is used for the transparency of the agents' colour (where $\alpha = 200$); therefore, as shown in the figures, the cell with the largest number of active agents can be distinguished from others.



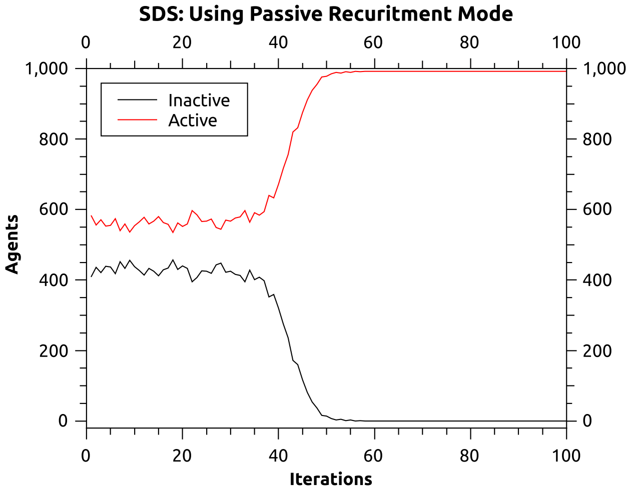


**Fig. 12** Passive recruitment mode

There are occasions when more than one centre of symmetry exists, or there exist some partial symmetries in the image along with full centre of symmetry; in this case another flavour of the recruitment strategy is deployed which is called *context-sensitive mechanism*. This strategy frees up some of the agents who are active and share the same hypothesis and therefore allows the algorithm to constantly check for traces of symmetry in the input pattern.

In other words, the use of context-sensitive mechanism biases the search towards global exploration. Therefore, if

[3] Further details about the number of iterations are given later in the paper.

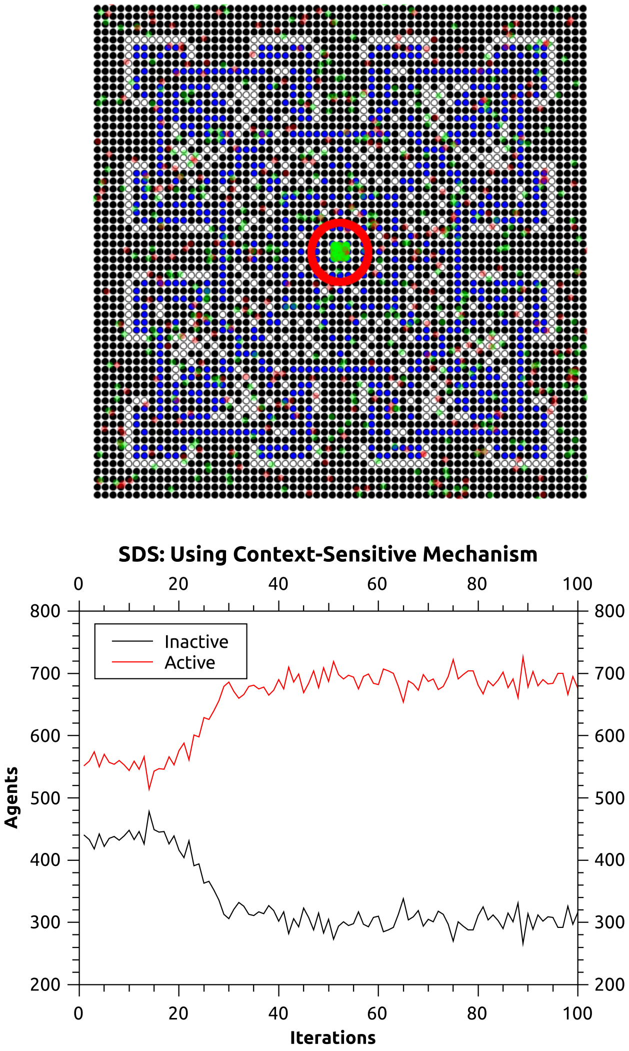


**Fig. 13** Context-sensitive mechanism

an active agent randomly chooses another active agent that maintains the same hypothesis, the selecting agent is set inactive and adopts a random hypothesis. This mechanism frees up some of the resources in order to have a wider exploration throughout the search space as well as preventing cluster size from overgrowing while ensuring the formation of large clusters in case there exists a perfect match or good sub-optimal solutions (see Algorithm 4).

**Algorithm 4** Context-Sensitive Mechanism

```
If ( ag.activity )
      r_ag = pick a random agent
      If ( r_ag.activity AND
           ag.getHypothsis == r_ag.
         getHypothsis )
           ag.setActivity ( false )
           ag.setHypotheis ( randomHypothsis )
      End If
End If
```

The next set of experiments use more complex patterns, generated by cellular automata techniques. Initially an experiment is run that utilises the *passive recruitment mode* without the introduced *context-sensitive mechanism* and later, the impact of *context-sensitivity* is discussed.

The graph in Fig. 12 illustrates the behaviour of the agents' activities; this graph demonstrates that after the initialisation phase, the number of active and inactive agents are balanced; however, over time, and due to the presence of a centre of symmetry in the pattern, the number of active agents increases and the number of inactive agents decreases. Therefore, ultimately, once the absolute centre of symmetry (where symmetry holds irrespective of the micro-features chosen) is identified, the entire agent population becomes active and the number of inactive agents drops to zero.

In the next experiment, the graph in Fig. 13 illustrates the behaviour of SDS algorithm using the context-sensitive mechanism, where the populations are biased towards global exploration. In this graph, while the increase in active agents and the decrease in inactive agents are visible, it is evident that there are always agents which are released back from the centre of symmetry to the search space to explore the possibility of further symmetrical points. This feature is particularly useful in dynamic environments, and where there are more than one absolute point of symmetry. The figure shows many active (green) and inactive (red) agents throughout the search space. On the other hand, there are no roaming agents in Fig. 12 and all agents are drawn to the centre of symmetry.

The next experiment, which uses a more symmetrically complex CA-generated pattern, demonstrates the crucial difference when using SDS with or without the context-sensitive mechanism. As stated before context-sensitive mechanism reduces the greediness of the agents and allows the agents to explore the search space for any undetected symmetry, while the passive mechanism is greedy and once it finds the absolute point of symmetry (where symmetry holds no matter which micro-feature is picked), it gradually pulls all the agents and stops them from locating partial symmetries in the canvas.

The new input to be used in this experiment has two identically CA grown patterns one on the top-left quarter and another on the bottom-left quarter. When running the SDS algorithm, it becomes clear that the passive recruitment strategy (see Fig. 14) initially locates two points of symmetry (when $n = 100$ iterations) and later (when $n = 200$ iterations) all agents are drawn to the absolute point of symmetry (note that the search spaces in cellular automata are torus).

However, by using the context-sensitive approach, the largest partial symmetries are also identified and highlighted (see Fig. 15). The graphs at the bottom of Figs. 14 and 15 show clearly the behaviour of the agents in both modes.

As displayed in the graph of Fig. 15, while the number of active and inactive agents are distinguishably far from

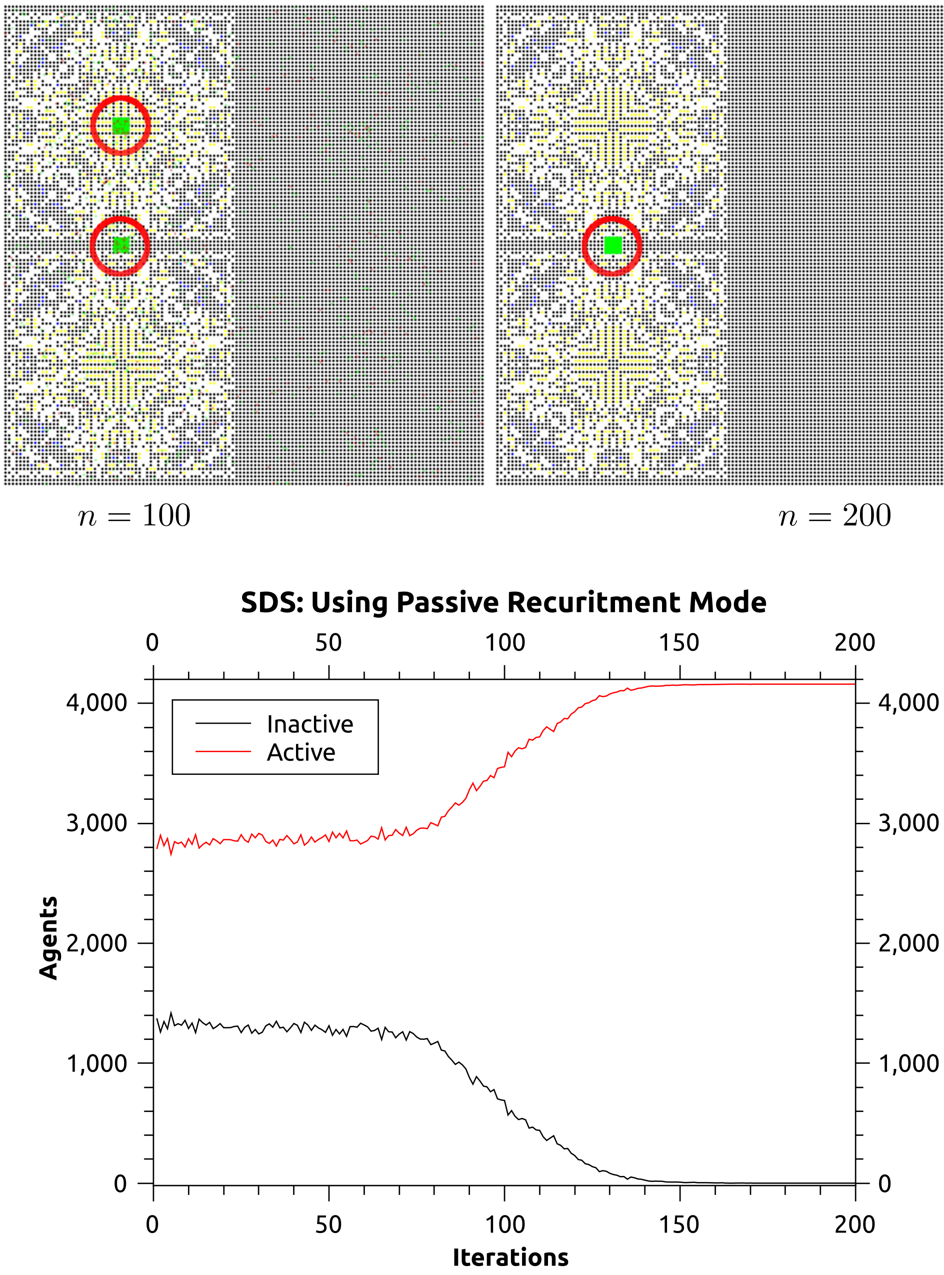


**Fig. 14** Passive recruitment mode: finding the absolute symmetry at iterations $n = 100$ and 200

one another, yet it is shown that the number of active agents does not reach the maximum possible[4] and the number of inactive agents does not drop to zero. This mechanism insures the identification of other symmetrical points in the input. Therefore, depending on the functionalities needed, either of these approaches could be used.

Whenever there is a full-match, SDS has been proven mathematically to converge (Nasuto and Bishop 1999) to the optimum solution. This proof is strongly based on the presence of the partial function evaluation aspect of SDS where a micro-feature is selected and evaluated individually of other micro-features. This makes the algorithm robust in dealing with false positive or false negative.

Another observation to be expanded in the future work is the direct proportionality of the agents' activity to the 'strength' of the symmetry. Therefore, while context-sensitive mechanism finds partial symmetries, it is able to *rank* the various clusters formed over the pattern.

[4] Given the size of the side of search space is $ssSize = 129$, the population size for this pattern is $pSize = \left\lceil \frac{129^2}{4} \right\rceil = 4,160$

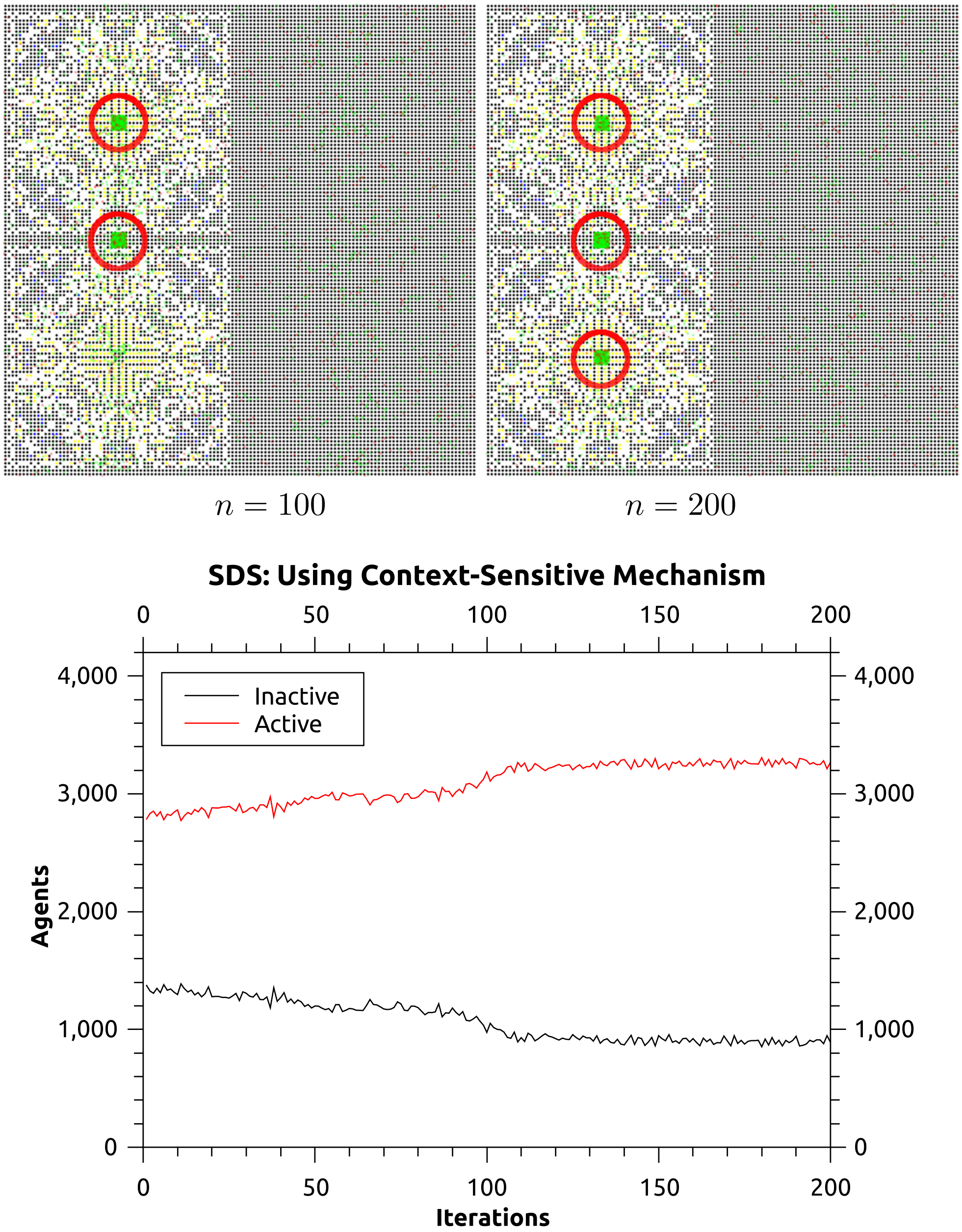


**Fig. 15** Context-sensitive mechanism: finding partial symmetry at iterations $n = 100$ and 200

## 6 Conclusion

CA provide perspective and powerful tools in generating graphical contents. The multi-state CA rule space is a vast set of possible rules which can generate interesting patterns with high aesthetic qualities. The interaction of CA rules at local level generates emergent global behaviour that can sometimes demonstrate attractive complexity. Some characteristics of CA, such as the regularity and complexity of the rules that are employed locally, suggest that they could be well suited to generate artificially generated aesthetic images.

This paper demonstrates the capability of a swarm intelligence algorithm—Stochastic Diffusion Search—in detecting absolute symmetries (when present) and the centre of partial symmetrical patterns within the input image. Evaluating the symmetry of cellular automata-generated patterns is often a difficult task partly due the large size of the search space, and partly due to the constantly changing, dynamic environment in which the cellular automata patterns are generated. These factors contribute to making the detection of symmetrical patterns computationally expensive. One of the main features of Stochastic Diffusion Search is partial function evaluation which is particularly useful when dealing with large problems with high dimensions and costly evaluation function (e.g. in this case, the expensive computational cost of detecting symmetry in cellular automata-generated patterns). The performance of this algorithm is explained in the paper and the results are accordingly demonstrated.

Following the introduction of this novel technique, among the future research topics are: conducting a comparison with other evolutionary and non-evolutionary techniques; computing the correlation between the size of the search space and the computational complexity of the process as well as ranking the quality of the symmetries detected.

### Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

Adamatzky A, Martínez GJ (2016) Designing beauty: the art of cellular automata, vol 20. Springer, Berlin

al-Rifaie MM, Bishop M (2013) Stochastic diffusion search review. Paladyn J Behav Robot 4(3):155–173

al-Rifaie MM, Bishop M, Blackwell T (2012) Information sharing impact of stochastic diffusion search on differential evolution algorithm. J Memet Comput 4(4):327–338

al-Rifaie MM, Bishop M, Caines S (2012) Creativity and autonomy in swarm intelligence systems. J Cognit Comput 4(3):320–331. doi:10.1007/s12559-012-9130-y

Bauerly M, Liu Y (2006) Computational modeling and experimental investigation of effects of compositional elements on interface and design aesthetics. Int J Man Mach Stud 64(8):670–682

Beddard H, Dodds D (2009) Digital pioneers. V&A pattern. V&A Publishing, London

Bishop J (1989) Stochastic searching networks. In: Proceedings of 1st IEE conference on artificial neural networks, pp 329–331. London, UK

Bishop JM, Torr P (2004) The stochastic search network. In: Linggard R, Myers DJ, Nightingale C (eds) Neural networks for images, speech and natural language. Chapman & Hall, New York, pp 370–387

Branke J, Schmidt C, Schmeck H (2001) Efficient fitness estimation in noisy environments. In: Proceedings of the 3rd annual conference on genetic and evolutionary computation. Morgan Kaufmann Publishers Inc, pp 243–250

Brown P (2001) Stepping stones in the mist. In: Bentley P, Corne D (eds) Creative evolutionary systems. Morgan Kaufmann Publishers Inc, pp 387–407

Carroll MJ (ed) (2003) HCI models, theories, and frameworks: toward a multidisciplinary science. Morgan Kaufmann Publishers, San Francisco

Dawkins R (1986) The blind watchmaker. W. W. Norton, New York

Digalakis J, Margaritis K (2002) An experimental study of benchmarking functions for evolutionary algorithms. Int J Comput Math 79:403–416

Ferreira MIA (2012) Interactive bodies: the semiosis of architectural forms. Biosemiotics 5(2):269–289

Gangestad SW, Thornhill R, Yeo RA (1994) Facial attractiveness, developmental stability, and fluctuating asymmetry. Ethol Sociobiol 15(2):73–85

Grammer K, Thornhill R (1994) Human (homo sapiens) facial attractiveness and sexual selection: the role of symmetry and averageness. J Comp Psychol 108(3):233

Hinton GF (1981) A parallel computation that assigns canonical object-based frames of reference. In: Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2, pp 683–685. Morgan Kaufmann Publishers Inc

Jacobsen T, Schubotz RI, Höfel L, Cramon DYV (2006) Brain correlates of aesthetic judgment of beauty. Neuroimage 29(1):276–285

Javaheri Javid MA, al Rifaie MM, Zimmer R (2014) Detecting symmetry in cellular automata generated patterns using swarm intelligence. In: Dediu AH, Lozano M, Martín-Vide C (eds) Theory and practice of natural computing, lecture notes in computer science, vol 8890. Springer, Berlin, pp 83–94

Javid MAJ, al-Rifaie MM, Zimmer R (2015) An informational model for cellular automata aesthetic measure. In: AISB 2015 Symposium on Computational Creativity. University of Kent, Canterbury, UK

Javid MAJ, te Boekhorst R (2006) Cell dormancy in cellular automata. In: Alexandrov VN, van Albada GD, Sloot PMA, Dongarra J (eds) International conference on computational science (3), Lecture notes in computer science, vol. 3993, pp. 367–374. Springer

Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput 9:3–12

Lewis M (2008) Evolutionary visual art and design. In: Romero J, Machado P (eds) The art of artificial evolution, natural computing series. Springer, Berlin, pp 3–37

Leyton M (1992) Symmetry, causality, mind. Bradford books. MIT Press, Cambridge

Linz P (2001) An introduction to formal languages and automata. Jones & Bartlett Publishers, Burlington

Liu Y (2002) Computational symmetry. In: Hargittai I, Laurent TC (eds) Symmetry 2000, Part 1. Wenner-Gren international series, vol 80. Portland Press, London, pp 231–246

Møller A,P, Cuervo JJ (1999) Asymmetry, size and sexual selection : meta-analysis, publication bias and factors affecting variation in relationships. Oxford University Press, Oxford

Møller AP, Thornhill R (1998) Bilateral symmetry and sexual selection: a meta-analysis. Am Nat 151(2):174–192

Myatt DR, Bishop JM, Nasuto SJ (2004) Minimum stable convergence criteria for stochastic diffusion search. Electron Lett 40(2):112–113

Nasuto SJ (1999) Resource allocation analysis of the stochastic diffusion search. Ph.D. thesis, University of Reading, Reading, UK

Nasuto SJ, Bishop JM (1999) Convergence analysis of stochastic diffusion search. Parallel Algorithms Appl 14(2):89–107

Nasuto SJ, Bishop JM, Lauria S (1998) Time complexity analysis of the stochastic diffusion search. In: Proceedings of the international ICSC/IFAC symposium on neural computation (NC 1998), September 23-15, 1998, Vienna, Austria, pp 260–266

Railton P (1998) Aesthetic value, moral value, and the ambitions of naturalism. In: Levinson J (ed) Aesthetics and ethics: essays at the intersection. Cambridge University Press, Cambridge, pp 59–105

Randy T, Steven G (1993) Human facial beauty. Human Nat 4:237–269

Reber R, Schwarz N, Winkielman P (2004) Processing fluency and aesthetic pleasure: is beauty in the perceiver's processing experience? Personal Soc Psychol Rev 8(4):364–382

Roth TO, Deutsch A (2011) Universal synthesizer and window: cellular automata as a new kind of cybernetic image. In: Oliver G, Thomas V (eds) Imagery in the 21st century. The MIT Press, pp 269–288

Scha IR (2006) Kunstmatige Kunst. De Commectie 2(1):4–7

Schwartz L, Schwartz L (1992) The computer artist's handbook: concepts, techniques, and applications. W W Norton & Company Incorporated, New York

Shackelford TKLRJ (1997) Facial symmetry as an indicator of psychological emotional and physiological distress. J Personal Soc Psychol 72:456

Sims K (1994) Evolving virtual creatures. In: Proceedings of the 21st annual conference on computer graphics and interactive techniques, pp 15–22. ACM

Struycken P (1976) Splash 1972/1974. In: Leavitt R (ed) Artist and computer. Harmony Books, pp 30–31

Todd S, Latham W, Hughes P (1991) Computer sculpture design and animation. J Vis Comput Anim 2(3):98–105

Whitley D, Rana S, Dzubera J, Mathias KE (1996) Evaluating evolutionary algorithms. Artif Intell 85(1–2):245–276

Wolfram S (2002) A new kind of science. Wolfram Media Inc., Champaign